Chapter 6
The Sequence Diagram

While the purpose of the state machine diagram presented in the last chapter is to model the *intra*-object behavior—that is, the life cycle of an object—in this chapter we look at the modeling of the *inter*-object behavior—that is, the interactions between the objects in a system.

An interaction specifies how messages and data are exchanged between interaction partners. The interaction partners are either human, such as lecturers or students, or non-human, such as a server, a printer, or executable software. An interaction can be a conversation between multiple persons—for example, an oral exam. Alternatively, an interaction can model communication protocols such as HTTP or represent the message exchange between humans and a software system—for example, between a lecturer and the student administration system when the lecturer publishes exam results. An interaction can also be a sequence of method calls in a program or signals such as a fire alarm and the resulting communication processes.

An interaction describes the interplay between multiple interaction partners and comprises a sequence of *messages*. The sending or receipt of a message can be triggered by the occurrence of certain events, for example, the receipt of another message, and can take place at specified times, for example, at 05:00. Predefined constraints specify any necessary preconditions that must be met for successful interactions. For example, continuing the communication process outlined above, the lecturer must be logged into the system before entering the students’ grades.

In UML, you use *interaction diagrams* to specify interactions. In an interaction diagram, you always model a concrete scenario, meaning that the message exchange takes place within a specific context to fulfill a specific task. Interactions usually only describe a specific part of a situation. There are often other valid execution paths that the interaction
diagram does not cover. Although data exchanged through the messages and processed or stored by the interaction partners can be represented in interaction diagrams, the purpose of modeling interactions is not to specify exactly how this data is to be manipulated. If required, you can add this type of information to interaction diagrams, but other diagrams such as the activity diagram (see Chapter 7) would take preference to model this information.

Interactions offer a mechanism for describing communication sequences at different levels of detail, both for computer experts as well as for end users and decision-makers. Interaction diagrams are therefore used in various situations. For example, they are used to represent the interaction of a complete system with its environment. In this case, the system can be interpreted as a black box of which only the interfaces visible to the outside are known. You can also use interaction diagrams to model the interaction between system parts in order to show how a specific use case (see Chapter 3) can be implemented. In late design phases, you can use interaction diagrams to precisely model interprocess communication in which the partners involved must observe certain protocols. Interaction diagrams can also zoom in much further into the system to be realized and can model communication at class level, meaning that you can use them to model operation calls and inter-object behavior.

Of the four interaction diagrams offered by UML, the sequence diagram is the one most frequently used—often in an informal way to quickly present interaction sequences. However, in this chapter, we describe the elements of the sequence diagram in detail and examine how to apply them according to the UML standard. In Section 6.7 we briefly introduce the other three interaction diagrams and compare them to the sequence diagram.

### 6.1 Interaction Partners

In a sequence diagram, the interaction partners are depicted as lifelines. A lifeline is shown as a vertical, usually dashed line that represents the lifetime of the object associated with it (see Fig. 6.1). At the top end of the line is the head of the lifeline, a rectangle which contains an expression in the form $\text{roleName:Class}$ (Fig. 6.1(c)). This expression indicates the name of the role and the class of the object associated with the lifeline. In the same way as for the object diagram (see Chapter 4.1 on page 50), one of the two names may be omitted. If you omit the class, you can omit the colon (Fig. 6.1(a)); however, if you specify only the class, the