Chapter 2
Applying Learner-Centered Design Principles to UML Sequence Diagrams

Debra VanderMeer
Florida International University, USA

Kaushik Dutta
Florida International University, USA

ABSTRACT
The Unified Modeling Language has been shown to be complex and difficult to learn. The difficulty of learning to build the individual diagrams in the UML, however, has received scant attention. In this paper, we consider the case of the UML sequence diagram. Despite the fact that these diagrams are among the most frequently used in practice, they are difficult to learn to build. In this paper, we consider the question of why these diagrams remain so difficult to learn to build. Specifically, we analyze the process of learning to build sequence diagrams in the context of cognitive complexity theory. Based on this analysis, and drawing on the theory of learner-centered design, we develop a set of recommendations for presenting the sequence diagram building task to the student analyst to reduce the complexity of learning how to build them.

INTRODUCTION
Object-oriented analysis and design (OOAD) is the dominant software design method among practitioners. The Unified Modeling Language (UML) is an ISO standard graphical modeling language used in OOAD (International Organization for Standardization, 2005). The UML consists of a set of diagrams and associated notations, where each diagram represents a different view of a software analysis and design model–structure, interaction, or state (Blaha & Rumbaugh, 2005; Dori, 2002). In the OOAD process, analysts develop a set of objects, where each object has a set of attributes.
Applying Learner-Centered Design Principles to UML Sequence Diagrams

(data that the object knows about) and behaviors (operations that the object can perform). Objects interact by invoking one another’s operations to fulfill the high-level behaviors required of the system, representing the business logic of the to-be software system. The analyst captures this logic in a set of UML sequence diagrams (SDs). Figure 1 shows an example of such a diagram (all diagrams in this paper were developed using Visual Paradigm, a software tool from Visual Paradigm International).

SDs are a crucial building block in system design. Analyst errors in building them lead to significant rework efforts or serious software defects at implementation time, which in turn leads to increased costs. Thus, it is critical that analysts have the skills to produce high-quality SDs. While sequence diagrams are among the most widely used diagrams in the UML in practice (Dobing & Parsons, 2006; Dobing & Parsons, 2008; Fowler, 1997) and are critical to the design process, they are difficult to learn to build (Bołożu & Leung, 2006; Siau & Loo, 2006).

The OOAD process consists of five basic steps: (1) developing high-level requirements, (2) use case analysis, (3) domain data modeling, and (4) building sequence diagrams (George, Batra, Valacich & Hoffer, 2006), and (5) building a class diagram. In steps (1) and (2), the analyst develops a description of what the system should do. In step (3), the analyst develops the structure of the system from a data perspective and, in step (4),

Figure 1. Example UML sequence diagram