IDEA GROUP PUBLISHING
701 E. Chocolate Avenue, Hershey PA 17033-1117, USA
Tel: 717/533-8845; Fax 717/533-8661; URL-http://www.idea-group.com
ITB7421

Chapter XIX

The Role of Use Cases in the UML: A Review and Research Agenda

Brian Dobing
University of Lethbridge, Canada

Jeffrey Parsons
Memorial University of Newfoundland, Canada

A use case is a description of a sequence of actions constituting a complete task or transaction in an application. Use cases were first proposed by Jacobson (1987) and have since been incorporated as one of the key modeling constructs in the UML (Booch, Jacobson, & Rumbaugh, 1999) and the Unified Software Development Process (Jacobson, Booch, & Rumbaugh, 1999). This chapter traces the development of use cases, and identifies a number of problems with both their application and theoretical underpinnings. From an application perspective, the use-case concept is marked by a high degree of variety in the level of abstraction versus implementation detail advocated by various authors. In addition, use cases are promoted as a primary mechanism for identifying objects in an application, even though they focus on processes rather than objects. Moreover, there is an apparent inconsistency between the so-called naturalness of object models and the commonly held view that use cases should be the primary means of communicating and verifying requirements with users. From a theoretical standpoint, the introduction of implementation issues in use cases can be seen as prematurely anchoring the analysis to particular implementation decisions. In addition, the fragmentation of objects across use cases creates conceptual difficulties in developing a comprehensive class diagram from a set of use cases. Moreover, the role of categorization in human thinking suggests that class diagrams may serve directly as a good mechanism for communicating and verifying application requirements with users. We conclude by outlining a framework for further empirical research to resolve issues raised in our analysis.

INTRODUCTION

The Unified Modeling Language, or the UML (Booch, Jacobson, & Rumbaugh, 1999), has rapidly emerged as a standard language and notation for object-oriented modeling in systems development, while the accompanying Unified Software Development Process (Jacobson, Booch, & Rumbaugh, 1999) has recently been developed to provide methodological support for the application of the UML in software development. The adoption of the UML brings focus to object-oriented developers faced with the task of choosing among dozens of proposed approaches to object-oriented analysis and design. In light of this activity, driven primarily by practitioners, it is important from an academic perspective to independently evaluate the capabilities and limitations of the UML and the Unified Process. Such evaluations can contribute to the development of theoretical underpinnings of the UML, to an improvement in its modeling power and usability, and to its appropriate application in systems development projects.

This chapter focuses on two components of the UML: use cases and class diagrams. In particular, we consider whether use cases add value as a component of an object-oriented modeling language by looking at two of their key roles, gathering requirements and developing class diagrams. We examine the variability in the amount of detail use cases should contain, according to various proponents, and introduce a theoretical rationale for including fewer task details than many proponents advocate. We discuss the lack of "object"-orientation in use cases, and present a theoretical argument that use cases may, in fact, not be necessary or valuable in the UML. Finally, we develop a framework for empirical research to evaluate the value of use cases and their relationship to class diagrams in the UML.

USE CASE FUNDAMENTALS

The term "use case" was introduced by Jacobson (1987) to refer to "a complete course of events in the system, seen from a user's perspective" (Jacobson, Christerson, Jonsson, & Overgaard, 1992, p. 157). The concept resembles others being introduced around the same time. Rumbaugh, Blaha, Premerlani, Eddy, and Lorensen (1991), Wirfs-Brock, Wilkerson, and Wiener (1990), and Rubin and Goldberg (1992) use the terms "scenario" or "script" in a similar way. But, despite concerns about the awkwardness of the name, the use case has become an important part of most object-oriented analysis and design methodologies. Use cases were incorporated into the UML in late 1995, after Ivar Jacobson joined forces with Grady Booch and James Rumbaugh.

The use case differs from typical structured requirements analysis tools that preceded it in two important ways. First, the use case is largely text-based. Structured analysis emphasized the importance of graphical tools, such as Work Flow and Data Flow Diagrams. The rationale for preferring diagrams to text was the oft-cited "a picture is worth a thousand words." In addition, before structured methodologies became available, analysts often generated extensive and unstructured text descriptions of existing and proposed systems that were very difficult to use. The UML has not abandoned diagrams; there are currently eight in the UML Version 1.3 (OMG, 1999), and the class, activity, sequence, statechart and use-case diagrams all play important roles during analysis. But use cases are written in the customer's language, so that "users and customers no longer have to learn complex notation" (Jacobson et al., 1999, p. 38).