Chapter III

Using Linguistic Patterns to Model Interactions

Isabel Díaz, Central University of Venezuela, Venezuela
Oscar Pastor, Technical University of Valencia, Spain
Lidia Moreno, Technical University of Valencia, Spain
Alfredo Matteo, Central University of Venezuela, Venezuela

Abstract

The dynamic behavior of a system is elicited, specified, and analyzed by means of interaction modelling. This activity is important for object-oriented software development because it provides the information that is required to construct the system conceptual model. Several interaction modelling techniques have been proposed. However, this activity continues to be a complex and tedious task for the modellers. These problems arise from the intrinsic difficulties of interaction modelling and the lack of effective techniques and tools to support to the modellers. In this chapter, heuristic-oriented techniques and linguistics-oriented techniques proposed by several authors to model interactions are analyzed. In addition, a
framework to facilitate and to improve the interaction modelling is described. This framework was conceived to be integrated into automatic software production environments. It uses linguistic patterns to recognize interactions from use case models. The patterns were validated with expert modellers. The validation process carried out and the main results are also presented.

Introduction

Dynamic models fulfil an important role in the development of object-oriented software systems. These describe the behavior of a system in terms of: (1) the state change of an object, which is due to an event or the execution of an operation (intraobject dynamic perspective); and (2) how the objects should interact to provide the system with a determined behavior (inter-object dynamic perspective). This chapter is based on the study of the interobject dynamic perspective and, in particular, on the construction process of the object interaction models in order to describe the behavior of a system.

The modelling of interactions is one of the most frequently overlooked practices in software development. While the structural model is considered to be fundamental for the analysis and design of the systems, the dynamic model is considered to be optional (Larman, 2004; Rosenberg & Scott, 1999). Nevertheless, both models contribute two complementary views of the system design that, taken separately, would be insufficient. Our experience, which coincides with that reported by other authors, has led us to believe that this problem may have originated in the high level of difficulty of interaction modelling, especially for inexperienced modellers (Larman, 2004; Rosenberg & Scott, 1999; Song, 2001).

On the one hand, modelling is an inherently complex activity that, in any case, depends on the experience and the domain knowledge of the modeller. On the other hand, the difficulty of constructing interaction models is also determined by other circumstances that are explained below.

The result of a thorough review of the literature has indicated that software development approaches that describe a technique for interaction modelling are scarce. The aids offered to the modeller to facilitate the task of identifying and specifying interactions are very few when compared to the extensive descriptions that are made of the modelling language. The nature of the diagrams that are used to graphically represent the interaction models, generally sequence diagrams (SDs) or message sequence charts (MSCs), also constitutes an obstacle for modelling (ITU, 2000; OMG, 2003). The amount of time that must be spent on elaborating and troubleshooting these diagrams makes them tedious activities, which many modellers attempt to avoid. Model editing tools available
Related Content

Planning and Managing the Human Factors for the Adoption and Diffusion of Object-Oriented Software Development Processes
www.igi-global.com/chapter/planning-managing-human-factors-adoption/25639?camid=4v1a

Agile RUP: Taming the Rational Unified Process
www.igi-global.com/chapter/agile-rup-taming-rational-unified/25638?camid=4v1a

Detailed Design of the Transactions and Class Methods
www.igi-global.com/chapter/detailed-design-transactions-class-methods/18742?camid=4v1a

Mapping Class Diagrams to Relational Schemas
www.igi-global.com/chapter/mapping-class-diagrams-relationalschemas/18735?camid=4v1a