Chapter I

An Integrated Approach to Design Patterns Formalization

Toufik Taibi, United Arab Emirates University, UAE

Abstract

A Design pattern describes a set of proven solutions for a set of recurring design problems that occur within a context. As such, reusing patterns improves both quality and time-to-market of software projects. Currently, most patterns are specified in an informal fashion, which gives rise to ambiguity, and limits tool support and correct usage. This chapter describes balanced pattern specification language (BPSL), a language intended to accurately describe patterns in order to allow rigorous reasoning about them. BPSL incorporates the formal specification of both structural and behavioral aspects of patterns. Moreover, it can formalize pattern composition and instances of patterns (possible implementations of a given pattern).
Introduction

A Design pattern describes a set of proven solutions for a set of recurring design problems that occurs within a certain context. Hence, reusing patterns yields better quality software within reduced time frames.

Currently, most patterns are described using a combination of textual descriptions, object-oriented (OO) graphical notations such as unified modeling language (UML) (Rumbaugh, Jacobson, & Booch, 1998), and sample code fragments. The intention was to make them easy to read and use, and to build a pattern vocabulary. However, informal descriptions give rise to ambiguity, and limit tool support and correct usage. Tool support can play a great role in automated pattern mining, detection of pattern variants, and code generation from pattern specification.

The pattern community mostly focuses on the solution element of a pattern and not on its other elements, such as the problem solved, the context, the important forces (Alexander, Ishikawa, & Silverstein, 1977) acting within the problem, or the way the pattern resolves these forces. Indeed, the verbal description of the solution element is the most coherent and the easiest to formalize. As such, this work also focuses on specifying the solution element of patterns.

Most formal approaches for pattern specification lack in the area of approachability due to the assumption that complex mathematical notations are necessary to achieve precision, thus favoring mathematically mature modelers rather than normal modelers (Taibi & Ngo, 2003b). Another problem of formal approaches is that they are not comprehensive enough to describe both aspects (structural and behavioral) of patterns. Additionally, only a few of the formal approaches attempted formalizing pattern composition (Taibi & Ngo, 2003b).

Balanced pattern specification language (BPSL) (Taibi & Ngo, 2003a) was developed in order to formally specify the structural as well as behavioral aspects of patterns at three levels of abstraction: pattern composition, patterns, and pattern instances.

First order logic (FOL) (Smullyan, 1995) is used as the formal basis for specifying the structural aspect of patterns, because relations between pattern participants can be easily expressed as predicates. Temporal logic of actions (TLA) (Lamport, 1994) is used as the formal basis for specifying the behavioral aspect of patterns, because it is best suited to describe the collective behavior of objects. BPSL has been successfully used to specify patterns for stand-alone systems (Taibi & Ngo, 2003a) and also for distributed object computing systems (Schmidt, Stal, Rohnert, & Buschmann, 2000; Taibi & Ngo, 2004).

The design of component-based software involves the composition of different components. Patterns are special types of components offering a flexible means of reuse. Since each pattern represents a well-tested abstraction that has many instances, patterns can be considered building blocks from which reusable and flexible software designs can be built. Checking the correctness of pattern composition allows detecting problems early in the lifecycle, which saves time and the cost of fixing errors at later stages. Thus, if formalized, pattern composition can lead to ready-made architectures from which only instantiation is required to build robust implementations. Since the specification of the structural and behavioral aspects of patterns uses two different formalisms (FOL and TLA), pattern composition is formalized independently for each aspect.
17 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the product's webpage:
www.igi-global.com/chapter/integrated-approach-design-patterns-formalization/8149?camid=4v1

Recommend this product to your librarian:
www.igi-global.com/e-resources/library-recommendation/?id=1

Related Content

3D Modeling for Environmental Informatics: Parametric Manifold of an Object under Different Viewing Directions
www.igi-global.com/chapter/3d-modeling-for-environmental-informatics/139594?camid=4v1a

An Evaluation Framework and a Benchmark for Multi/Hyperspectral Image Compression
www.igi-global.com/chapter/evaluation-framework-benchmark-multi-hyperspectral/77549?camid=4v1a
A Novel Approach of Restoration of Digital Images Degraded by Impulse Noise

A Brain-Inspired Visual Pattern Recognition Architecture and Its Applications
Fok Hing Chi Tivive and Abdesselam Bouzerdoum (2008). *Pattern Recognition Technologies and Applications: Recent Advances* (pp. 244-264).
www.igi-global.com/chapter/brain-inspired-visual-pattern-recognition/28033?camid=4v1a