

Chapter 2

Modeling Design Patterns for Semi-Automatic Reuse in System Design

Galia Shlezinger

Technion-Israel Institute of Technology, Israel

Iris Reinhartz-Berger

University of Haifa, Israel

Dov Dori

Technion-Israel Institute of Technology, Israel

ABSTRACT

Design patterns provide reusable solutions for recurring design problems. They constitute an important tool for improving software quality. However, correct usage of design patterns depends to a large extent on the designer. Design patterns often include models that describe the suggested solutions, while other aspects of the patterns are neglected or described informally only in text. Furthermore, design pattern solutions are usually described in an object-oriented fashion that is too close to the implementation, masking the essence of and motivation behind a particular design pattern. We suggest an approach to modeling the different aspects of design patterns and semi-automatically utilizing these models to improve software design. Evaluating our approach on commonly used design patterns and a case study of an automatic application for composing, taking, checking, and grading analysis and design exams, we found that the suggested approach successfully locates the main design problems modeled by the selected design patterns.

INTRODUCTION

Patterns are types of themes that specify recurring processes, events, or elements. Commonly used in different engineering fields, patterns can be classified into analysis patterns, design pat-

terns, organization patterns, process patterns, and domain-specific patterns. Batra (2005) claimed that pattern recognition can be considered as a conceptual modeling technique. He further stated that the main challenge with patterns is “to provide a limited number of patterns at a fairly high level of abstraction and an appropriate level of granularity...” (Batra, 2005, p. 87). Identifying locations in

DOI: 10.4018/978-1-61350-471-0.ch002

which patterns might be applied can be a tedious task (Eden, 2002). Purao, Storey and Han (2003) described a prototype for automatically generating a conceptual design based on analysis patterns. Their research, which concerns high level analysis and design, aims at producing models automatically from textual requirements. Blomqvist (2005) presented an approach for creating ontology patterns semi-automatically by utilizing knowledge organized as patterns from other areas, like data modeling, knowledge reuse, software analysis, and software design. While these approaches are appropriate for high-level conceptual modeling activities, they are less applicable at the detailed design stage. As the level of design becomes more specific, different parameters, such as code optimization, system limitations, and other non-functional requirements, have to be taken into consideration. These require careful and detailed descriptions of the patterns, their essence, and their exact usages.

In this work, we concentrate on design patterns, which describe reusable solutions for recurring design problems in given contexts (Gamma, Helm, Johnson & Vlissides, 1994). While design patterns may help produce better design and implementation of applications (Prechelt, Unger, Philipssen & Tichy, 2002; Schmidt, 1995), their appropriate use is often hindered due to the inherent ambiguity in the existing ways of description and representation (Taibi & Ngo, 2003). This may impede effective use of design patterns in particular applications, as designers may not be able to appreciate benefits or predict shortcomings associated with their proper use (Dong, Yang & Zhang, 2007). Misuse of design patterns typically results from failure to understand the rationale behind the patterns (Wendorff, 2001) or from difficulties in incorporating the patterns into a specific system design (Abdul Jalil & Azman Mohd Noah, 2007).

In view of these observations, we have developed an approach, accompanied by a tool, for clearly modeling design patterns and applying them semi-automatically to system models. In

doing so, we utilize knowledge we have gained from modeling and categorizing design patterns. Our approach supports modeling of the different aspects of design patterns, including their problem and solution specifications, their essence, and their correct usage in particular systems. The design pattern models are specified using Object-Process Methodology (OPM) (Dori, 2002), which enables describing structural and behavioral aspects of design patterns at different granularity and abstraction levels. These design pattern models provide the basis for an algorithm for searching design problems in a given system design and suggesting corrections for them in the form of design patterns. Evaluating our approach and the feasibility of its algorithm using a case study research methodology, we found that the suggested approach successfully locates the main design problems modeled by the selected design patterns.

The structure of the rest of the paper is as follows. First, the relevant literature is reviewed, followed by a section that introduces the suggested design patterns representation method. Then, the algorithm for improving a system's design via the use of design patterns is presented, followed by a report on the evaluation of the approach. Finally, we summarize and propose possible future research directions.

LITERATURE REVIEW: DESIGN PATTERNS REPRESENTATION

Design patterns are widely recognized as an important technique for software design and programming. They are usually described using a template, such as the well known one proposed by Gamma *et al.* (1994). These descriptions are often accompanied by graphical notations. Using UML (Object Management Group, 2002), for example, class diagrams are recruited for modeling the structural aspects of the design patterns, while sequence diagrams are used for specifying their

26 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the publisher's webpage:
www.igi-global.com/chapter/modeling-design-patterns-semi-automatic/63661

Related Content

Use of Qualitative Research to Generate a Function for Finding the Unit Cost of Software Test Cases

Mark L. Gillenson, Thomas F. Stafford, Xihui "Paul" Zhang and Yao Shi (2020). *Journal of Database Management* (pp. 42-63).
www.irma-international.org/article/use-of-qualitative-research-to-generate-a-function-for-finding-the-unit-cost-of-software-test-cases/249170

Database and the Web

Mark Gillenson (1998). *Journal of Database Management* (pp. 35-36).
www.irma-international.org/article/database-web/51203

Similarity Search in Time Series Databases

Maria Kontaki, Apostolos N. Papadopoulos and Yannis Manolopoulos (2005). *Encyclopedia of Database Technologies and Applications* (pp. 646-651).
www.irma-international.org/chapter/similarity-search-time-series-databases/11218

The Development of Ordered SQL Packages to Support Data Warehousing

Wilfred Ng and Mark Levene (2001). *Journal of Database Management* (pp. 26-39).
www.irma-international.org/article/development-ordered-sql-packages-support/3270

Similarity Retrieval and Cluster Analysis Using R* Trees

Jiaxiong Pi, Yong Shi and Zhengxin Chen (2009). *Handbook of Research on Innovations in Database Technologies and Applications: Current and Future Trends* (pp. 547-554).
www.irma-international.org/chapter/similarity-retrieval-cluster-analysis-using/20739