Verifying Correctness of Pattern-based Composition in Coq

Qiang Liu[1.1], Zongyuan Yang[1] and Jinkui Xie[1]

[1]Department of Computer Science and Technology, East China Normal University
Dongchuan Rd. 500, 200241 Shanghai, China
{lqiang, yzyuan, jkxie}@cs.ecnu.edu.cn

Abstract. Design patterns capture elegant design solutions and facilitate reuse in design level. In the Component Based Software Engineering (CBSE), design patterns were treated as design components, which serve as elemental components and can be composed to construct a large software system. In the process of composition, the key problem is how to ensure the correctness of composition. To address this problem, a criterion for correct composition is needed. The well known "faithfulness" principle is chosen as the correctness criterion and our discussions are based on this principle. In this paper, we at first use First Order Logic to model some elemental entities and relations in Object Oriented Design, which serve as an ontology in the domain. Then using the vocabulary in the ontology, we formally specify design patterns and formalize the "faithful" principle as theorems in Coq. Finally, we prove the theorems and thus show the correctness of composition. As a case study, we described and verified the composition of Composite pattern and Decorator pattern. Once a composition is proven to be correct, one can use the composition repeatedly. This would facilitate reuse of design in a larger scale and reduce errors in design phase, which justifies all the efforts of verifying their correctness.

Keywords: Design patterns, Composition, Verification, Coq, Faithfulness

1 Introduction

Design patterns [8] are widely used in modern software systems. In the Component-Based approach, design patterns were treated as design components, which serve as elemental components and can be composed to construct a large software system. In the process of composition, the key problem is how to ensure the correctness of composition. This is a difficult and larger top, and far from being resolved. In this paper, we set our discussion to the domain of the object oriented design, and restrict our discussion in the composition of design patterns. This paper is organized in the following way:
Section 2 is related works.

In section 3, we illustrate the problem by a case study, the composition of Composite and Decorator pattern. In order to model design patterns in Coq, we use First Order Logic to capture the basic entities (Class and Method) and relations in the domain of OOD (Inherit etc). They are both modeled by Inductive types in Coq. These entities and relations are the basic vocabularies based on which we can faithfully