Structural- and Behavioral-based Design Pattern Recognition

Abstract

Requirements of existing software are constantly changing. Before changing the software, a software developer needs to understand a huge amount of source code which is frequently poorly documented. In software engineering, well-known solutions for recurring problems are described as design patterns. Identifying design patterns in the source code documents the implicit software design and supports the developer understanding the software.
Design patterns are defined by their structure as well as their behavior. Most of the existing design pattern recognition tools use static analyses of the source code. Static analyses are appropriate to identify the structure but insufficient to identify behavior. Dynamic analyses in contrast examine the behavior of the software during runtime but produce huge amount of data.
This thesis presents a design pattern recognition which combines an existing static analysis with a newly developed dynamic analysis. The static analysis identifies candidates of design patterns by their structure. The dynamic analysis subsequently examines the behavior of just these candidates to reduce the amount of data. The design patterns are therefore identified by their structure as well as their behavior and hence are a reliable basis for understanding the software.