Diss. ETH No. 21939

Specified and Verified Reusable Components

A thesis submitted to attain the degree of
DOCTOR OF SCIENCES of ETH ZURICH
(Dr. sc. ETH Zurich)

presented by
NADIA POLIKARPOVA
Master of Applied Mathematics and Informatics, SPb SU ITMO, Russia

born on
May 20th, 1985

citizen of
Russia

accepted on the recommendation of

Prof. Dr. Bertrand Meyer, examiner
Prof. Dr. K. Rustan M. Leino, co-examiner
Prof. Dr. Peter Müller, co-examiner
Dr. Natarajan Shankar, co-examiner

2014

Abstract

For *reusable software components*—program modules designed for black-box usage in arbitrary, *a priori* unknown contexts—quality assurance is particularly important and easily justified. Although this is widely agreed upon, the industry standard is still far removed from a perfect world, in which components are unambiguously documented and their correctness is established with certainty. This thesis aims at providing programmers with practical tools and techniques for assessing and improving the quality of reusable components at various stages of the software development process.

Formal specifications play a central role in quality assurance, documenting the interface between a component and its clients and acting as oracles for verification. Writing good interface specifications—those that include all relevant details and none of the irrelevant—is challenging in the absence of precise guidelines and formal assessment criteria. The present work addresses this challenge with *model-based contracts*: a specification methodology that enhances Design by Contract with mathematical models, and supports strong yet abstract specifications. The thesis assesses feasibility and costs of deploying such strong specifications, and demonstrates their benefits, which include boosting automated testing, preventing inconsistent library designs, and decreasing the density of implementation defects.

Achieving high confidence in the quality of component implementations requires formal correctness proofs. Although static program verification has made significant progress in recent years, existing methods and tools provide insufficient support for model-based contracts, and for the design patterns often found in object-oriented component libraries. This thesis advances the state of the art in program verification with two contributions. First, it proposes *semantic collaboration*: a new methodology to reason about class invariants in the presence of inter-object dependencies; the methodology is flexible enough to accommodate advanced design patters, but comprises useful default annotations, which reduce the specification overhead in common scenarios. Second, the thesis details a practical verification methodology for model-based contracts, featuring advanced support for model classes, and

an approach to frame specifications that works well for complex inheritance hierarchies. Both proposed methodologies have been implemented in the AutoProof program verifier for the Eiffel programming language.

Another contribution of this thesis facilitates understanding and debugging failed verification attempts—one of the biggest remaining obstacles to usable program verification.

Practical solutions targeting reusable components must be evaluated on real software libraries. Two Eiffel data structure libraries, EiffelBase and its successor EiffelBase2, serve as case studies throughout the thesis. In particular, EiffelBase2—the first example of a data structure library developed from the start with strong specifications and verified (to a significant extent) for full functional correctness—embodies the vision of high-quality reusable components promoted in this work.

Zusammenfassung

Für *wiederverwendbare Software-Komponenten*—Programmmodule, die zur Black-Box-Nutzung in verschiedenen, im Voraus unbekannten Kontexten entwickelt werden—ist Qualitätssicherung besonders wichtig und der damit verbundene Aufwand gerechtfertigt. Obwohl diese Ansicht weitläufig akzeptiert wird, bleibt der Industriestandard weit von dem Idealfall entfernt, in dem alle Komponenten eine eindeutige Dokumentation haben und ihre Richtigkeit zweifellos sichergestellt ist. Ziel dieser Dissertation ist es, Programmierern anwendbare Werkzeuge und Techniken zur Verfügung zu stellen, die eine Bewertung und Verbesserung der Qualität von wiederverwendbaren Komponenten erlauben.

Formale Spezifikationen spielen bei der Qualitätssicherung eine zentrale Rolle, da sie die Schnittstelle zwischen Komponenten und ihre Kunden dokumentieren, und zur Verifikation von Implementierungen verwendet werden. Gute Schnittstellenspezifikationen zu schreiben—solche, die alle relevanten aber keine irrelevanten Details beinhalten—ist ohne genaue Richtlinien und formalen Bewertungskriterien schwierig. Die vorliegende Arbeit beschreibt einen Ansatz zur Überwindung dieser Schwierigkeiten mit Hilfe von *Modell-basierten Verträgen*: eine Spezifikationsmethode, die Design by Contract um mathematische Modelle ergänzt, und damit starke aber zugleich abstrakte Spezifikationen erlaubt. Die Dissertation evaluiert Durfürbarkeit und Kosten solch starker Spezifikationen, und zeigt ihren Nutzen auf, einschliesslich der Vorteile für automatische Testverfahren, der Vermeidung von inkonsistenten Bibliothek-Entwürfen, und der Abnahme der Häufigkeit von Implementierungsfehlern.

Um eine hohe Implementierungsqualität von Komponenten zu garantieren, benötigt man Richtigkeitsbeweise. Trotz bedeutenden Fortschritten, die statische Programmverifikation in den letzten Jahren gemacht hat, bieten existierende Methoden und Werkzeuge ungenügende Unterstützung für Modell-basierte Verträge, sowie für einige Entwurfsmuster, die häufig in den objektorientierten Komponenten-Bibliotheken zu finden sind. Diese Dissertation trägt zum Fortschritt der Programmverifikation in zweierlei Hinsicht

bei. Erstens führt sie eine neuartige Methode, *Semantische Kollaboration* genannt, ein, um über Klasseninvarianten in Gegenwart von Abhängigkeiten zwischen Objekten schlussfolgern zu können; die Methode ist flexibel genug sich komplexen Entwurfsmustern anzupassen, enthält aber zugleich nützliche Standardwerte, die die Kosten einer Spezifikation in Normalfall gering halten. Zweitens stellt die Dissertation eine praktische Verifikationsmethode für Modell-basierte Verträge vor, die eine fortgeschrittene Unterstützung von Modell-Klassen bietet und einen Ansatz für Frame-Spezifikationen umfasst, der gut mit komplexen Vererbung-Hierarchien funktioniert. Beide Methoden wurden in AutoProof, einem Programmverifizierer für die Eiffel Programmiersprache, implementiert.

Ein weiterer Beitrag der vorliegenden Arbeit erleichtert das Verstehen und die Fehlerbehebung im Fall von misslungenen Beweisversuchen—eines der grössten verbleibenden Hindernisse für nutzbare Programmverifikation.

Praktische Lösungen, die auf wiederverwendbare Software-Komponenten zielen, müssen anhand echter Software-Bibliotheken evaluiert werden. Zwei Eiffel Datenstruktur-Bibliotheken, EiffelBase und ihr Nachfolger EiffelBase2, dienen in der gesamten Dissertation als Fallstudien. Insbesondere, EiffelBase2—das erste Beispiel einer Datenstruktur-Bibliothek, die von Anfang an mit starken Spezifikationen entwickelt wurde und deren volle funktionelle Richtigkeit grösstenteils bewiesen wurde—verkörpert die in dieser Arbeit verfolgte Vision von hochwertigen wiederverwendbaren Komponenten.