

Abstract

For functioning correctly, complex object-oriented software relies on application-specific *constraints* regarding the definition and use of program elements. This dissertation presents *CoffeeStrainer*, a system for checking programmer-defined constraints on Java source code.

CoffeeStrainer constraints are unique in that they are modular, extensible and composable, and supports both constraints regarding the definition of program elements and constraints regarding the use of program elements. Additionally, CoffeeStrainer constraints can consist of static (compile-time) and dynamic (run-time) parts. Static constraints are checked using a program representation which consists of the complete abstract syntax tree together with information obtained by name and type analysis. The system checks compilation units separately; the time taken to check one compilation unit is comparable to the time of compiling the compilation unit. The system has been fully implemented.

Unlike other approaches, CoffeeStrainer does not define a special-purpose language for specifying constraints — instead, it uses Java for constraints as well, making it easy for proficient Java programmers to specify constraints. Constraint code is embedded in Javadoc comments. Thus, constraint code and base-level code share the same structure, making it easy to find the rules that apply to a given part of the program, and allowing arbitrary compilers and tools to be applied to the source code that contains constraints.

Zusammenfassung in deutscher Sprache

Das korrekte Funktionieren komplexer objektorientierter Software beruht auf der Einhaltung anwendungsspezifischer Programmierrichtlinien, die sich sowohl auf die richtige Definition von Programmbestandteilen als auch auf deren richtige Benutzung beziehen können. Die Dissertation beschreibt ein System *CoffeeStrainer*, das Java-Programmierern einen Rahmen für die Spezifikation solcher Programmierrichtlinien bereitstellt und sie anhand von Java-Quelltext automatisch überprüft.

Das System zeichnet sich dadurch aus, daß die Programmierrichtlinien modular, erweiterbar und komponierbar sind, und daß sowohl Definitions- als auch Benutzungsrichtlinien gleichermaßen unterstützt werden. Programmierrichtlinien können sich auf die statische Struktur eines Programms oder auf die Ausführung eines Programms beziehen; außerdem ist es möglich, hybride Richtlinien zu formulieren, die aus statischen und dynamischen Anteilen bestehen. Das System ist vollständig implementiert und stellt den Programmierrichtlinien das vollständige zu überprüfende Programm als abstrakten Syntaxbaum mit zusätzlichen semantischen Informationen zur Verfügung. *CoffeeStrainer* überprüft Übersetzungseinheiten jeweils separat und benötigt zur Überprüfung einer Übersetzungseinheit vergleichbar viel Zeit wie ein Übersetzer für ihre Übersetzung.

Im Gegensatz zu anderen Arbeiten auf dem Gebiet wird keine spezielle Sprache zur Formulierung von Programmierrichtlinien definiert — stattdessen wird die Sprache Java verwendet, um erfahrenen Java-Programmierern die Benutzung des Systems zu erleichtern. Programmierrichtlinien sind in das Programm in Form von sogenannten "formalen Kommentaren" der Sprache Java integriert und beziehen sich logisch jeweils auf den kommentierten Programmteil, wodurch ihre Struktur sich an die Struktur des Programms anlehnt. Außerdem ermöglicht diese Technik, daß vorhandene Übersetzer und andere Werkzeuge, die auf den Quelltext zugreifen, weiterhin benutzt werden können.

Lebenslauf

Boris Bokowski, geboren am 18.10.1969 in Berlin

- 1988 Abitur, Justus-Liebig-Schule Darmstadt
- 1988-1994 Studium der Informatik mit Nebenfach Philosophie an der Technischen Universität Darmstadt; Abschluß: Diplom-Informatiker
- 1994-1997 Stipendiat des Graduiertenkollegs "Kommunikationsbasierte Systeme" der Technischen Universität Berlin, der Freien Universität Berlin und der Humboldt-Universität zu Berlin
- 1998-1999 Wissenschaftlicher Mitarbeiter der GMD-FIRST, Berlin
- 2000- Senior Software Developer, Object Technology International, Ottawa

Erklärung

Hiermit erkläre ich, daß ich die vorliegende Arbeit auf der Basis der angegebenen Literatur selbständig verfaßt habe.