DISS. ETH NO. 17610

Making classes provable through contracts, models and frames

A dissertation submitted to the
SWISS FEDERAL INSTITUTE OF TECHNOLOGY ZURICH
(ETH Zürich)

for the degree of
Doctor of Sciences

presented by
Bernd Schoeller
Diplom-Informatiker, TU Berlin

born
April 5th, 1974

citizen of
Federal Republic of Germany

accepted on the recommendation of
Prof. Dr. Bertrand Meyer, examiner
Prof. Dr. Martin Odersky, co-examiner
Prof. Dr. Jonathan S. Ostroff, co-examiner

2007

ABSTRACT

Software correctness is a relation between code and a specification of the expected behavior of the software component. Without proper specifications, correct software cannot be defined.

The *Design by Contract* methodology is a way to tightly integrate specifications into software development. It has proved to be a light-weight and at the same time powerful description technique that is accepted by software developers. In its more than 20 years of existence, it has demonstrated many uses: documentation, understanding object-oriented inheritance, runtime assertion checking, or fully automated testing.

This thesis approaches the formal verification of contracted code. It conducts an analysis of Eiffel and how contracts are expressed in the language as it is now. It formalizes the programming language providing an operational semantics and a formal list of correctness conditions in terms of this operational semantics.

It introduces the concept of *axiomatic classes* and provides a full library of axiomatic classes, called the *mathematical model library* to overcome problems of contracts on *unbounded data structures*.

This thesis argues that modular verification is essential for the reuse of trusted object-oriented code. Modular verification introduces problems with hidden interference of components, known as the *frame problem*. This thesis introduces the concept of *dynamic frame contracts* and shows how such contracts can overcome the frame problem, at the same time retaining full information hiding and being faithful to the inheritance relation.

The thesis includes an experimental implementation of a fully automated verifier called *Ballet*. This verifier transforms Eiffel into proof obligations that are handed over to a fully automated theorem prover.

ZUSAMMENFASSUNG

Korrekte Software beschreibt eine Relation zwischen Programmtext und einer Spezifikation des zu erwartenden Verhaltens einer Software-Komponente. Ohne eine geeignete Spezifikation ist der Begriff *korrekter Software* bedeutungslos.

Design by Contract ist eine Methode die Spezifikationen in den Softwareentwicklungsprozess einbindet. Sie hat sich bewährt als eine einfache und gleichermaßen mächtige Technologie und wird von Entwicklern angenommen. In den 20 Jahren seit ihrer Entstehung haben sich viele Anwendungsfelder für Design by Contract ergeben: Dokumentation, Verständnis der objektorientierten Vererbung, Überprüfung zur Laufzeit, oder vollautomatisches Testen.

Diese Dissertation beschäftigt sich mit der formalen Verifikation von *contracted Code*. Sie analysiert die Programmiersprache Eiffel und wie in dieser Programmiersprache Verträge (Contracts) benutzt werden können. Eine operationelle Semantik formalisiert die Programmiersprache. Die Beweisverpflichtungen zur Überprüfung der Korrektheit werde definiert.

Sie entwickelt den Begriff der *axiomatischen Klasse* und entwickelt eine Bibliothek solcher Klassen, die *Mathematical Model Library*, um Probleme mit Verträgen über unbeschränkten Datenstrukturen zu beheben.

Diese Dissertation argumentiert, dass modulares Beweisen essentiell für die Wiederverwendungen von vertrauenswürdigen, objektorientierten Komponenten ist. Eine modulare Beweisführung scheitert an versteckten Interferenzen verschiedener Komponenten. Dieses Problem ist unter dem Namen *Frame Problem* bekannt. Diese Dissertation erweitert Design by Contracts um *Dynamic Frame Contracts*, um diese Probleme zu beheben. Dynamic Frame Contracts bewahren die Kapselung von Komponenten und sind verträglich mit der Vererbungsbeziehung der objektorientierten Entwicklung.

Die Dissertation enthält die experimentelle Implementierung eines vollautomatischen Beweiswerkzeugs mit dem Namen *Ballet*. Dieses Werkzeug wandelt Eiffel Programmtext in Beweisverpflichtungen für einen automatischen Beweiser um.