

DISS. ETH NO. 17643

USING PATTERNS TO DEVELOP CONSISTENT DESIGN CONSTRAINTS

A dissertation submitted to

SWISS FEDERAL INSTITUTE OF TECHNOLOGY ZURICH

for the degree of

Doctor of Sciences

presented by

MICHAEL WAHLER

Dipl.-Inf., TU München

born on March 10th, 1978

German citizen

accepted on the recommendation of

Prof. Dr. David Basin, examiner

Prof. Dr. Peter Müller, co-examiner

Dr. Jana Koehler, co-examiner

2008

Abstract

Developing constraint specifications for class models is a time-consuming and error-prone task because typical specifications contain numerous constraints, which in addition often state complex facts about the elements of the model. As the size and the complexity of constraint specifications grow, so does the probability of inadvertently specifying inconsistent models that cannot be instantiated because of contradictory constraints.

In this thesis, we introduce a novel approach to developing consistent constraint specifications based on constraint patterns. The input for this approach is an unconstrained class model and the output is a formal constraint specification. The approach comprises four phases. First, class models are automatically analyzed to elicit potentially missing constraints. Second, a library of composable constraint patterns allows developers to write concise constraint specifications based on the results from the elicitation phase. Third, this approach contains consistency assertions on the constraint pattern library that enable automatic consistency analysis of pattern-based constraint specifications. As basis for these consistency observations, we provide formal definitions of consistency properties of constrained class models. Fourth, pattern-based constraint specifications are transformed into logical expressions or code in a programming language.

The focus of the approach is on effectiveness and practicability. Therefore, we introduce a tool that allows model developers to follow the theoretic approach in a guided way and effectively apply it in modeling projects. We use this tool to conduct several case studies in which we validate that our approach improves the state-of-the-art constraint development in terms of detecting missing constraints, shortening development time, avoiding inconsistencies, and yielding more comprehensible constraints.

Zusammenfassung

Die Entwicklung von Constraintspezifikationen für Klassenmodelle ist zeit raubend und fehleranfällig, weil typische Spezifikationen zahlreiche Constraints enthalten, die darüber hinaus oft komplizierte Eigenschaften von Modellelementen beschreiben. Mit steigender Grösse und Komplexität von Constraintspezifikationen steigt auch die Wahrscheinlichkeit, versehentlich inkonsistente Modelle zu spezifizieren, die auf Grund widersprüchlicher Constraints nicht instantiiert werden können.

In dieser Dissertation führen wir einen neuartigen Ansatz zur Entwicklungen von konsistenten Constraintspezifikationen ein, der auf Constraintmustern basiert. Die Eingabe für diesen Ansatz ist ein constraintfreies Klassenmodell, und die Ausgabe ist eine formelle Constraintspezifikation. Dieser Ansatz beinhaltet vier Phasen: 1. Klassenmodelle werden automatisch analysiert, um potentiell fehlende Constraints zu eruieren. 2. Eine Bibliothek von zusammensetzbaren Constraintmustern erlaubt es Entwicklern, auf der Basis dieser Eruierung knappe und präzise Constraintspezifikationen zu entwerfen. 3. Dieser Ansatz beinhaltet Konsistenzaussagen über diese Bibliothek von Constraintmustern, die eine automatische Konsistenzanalyse von musterbasierten Constraintspezifikationen ermöglichen. Als Grundlage für diese Konsistenzbetrachtungen erstellen wir formelle Definitionen von Konsistenzigenschaften constraint-annotierter Klassenmodelle. 4. Musterbasierte Constraintspezifikationen werden in logische Ausdrücke oder Code in einer Programmiersprache übersetzt.

Der Schwerpunkt dieses Ansatzes liegt auf Anwendbarkeit. Deshalb stellen wir ein Werkzeug vor, das Modellentwickler bei der Benutzung des theoretischen Ansatzes führt, um ihn effektiv in Modellierungsprojekten einzusetzen. Wir benutzen dieses Werkzeug zur Durchführung mehrerer Fallstudien, in denen wir validieren, dass unser Ansatz den heutigen Stand der Constraintspezifizierung verbessert, indem er beim Finden von fehlenden Constraints hilft, die Entwicklungszeit verkürzt, Inkonsistenzen vermieden werden und leichter verständliche Constraints erstellt werden können.