Contents

1 Introduction **1**
1.1 Motivation and Goal 1
1.2 Challenges and Contributions 2
1.3 Terminology 5
1.4 Outline 6

2 The EiffelBase2 Library **7**
2.1 Overview 7
2.2 Design Goals 8
2.3 The EiffelBase2 Architecture 8
2.3.1 Specification and Verification Challenges 10
2.3.2 The Mathematical Model Library 12

3 Specifying Reusable Components **15**
3.1 Introduction 15
3.2 Motivating Examples 17
3.2.1 Some limitations of Design by Contract 17
3.2.2 Enhancing Design by Contract with models 19
3.3 Foundations of Model-Based Contracts 21
3.3.1 Abstract Data Types 22
3.3.2 Abstract equality and abstract state space 23
3.3.3 Model-Based Specifications 26
3.4 Model-Based Contracts in Practice 31
3.4.1 From Abstract Data Types to Classes 31
3.4.2 Assessing Quality in Practice 38
3.5 Experimental Evaluation 43
3.5.1 Case studies 43
3.5.2 Results and discussion 43
3.6 Related Work 46
3.7 Summary and Future Work 47

Polikarpova, Nadia
Specified and verified reusable components
2014

digitalisiert durch
IDS Basel/Bern, im Auftrag der
Schweizerischen Nationalbibliothek

4 Testing against Strong Specifications 49
4.1 Introduction 49
4.2 A Motivating Example 51
4.3 Strong Specifications for Testing 52
4.3.1 Representation Constraints 53
4.3.2 Avoiding False Positives 54
4.3.3 Tool support 56
4.4 Experiments 57
4.4.1 Research Questions 57
4.4.2 Eiffel Experiments 58
4.4.3 C# Experiment 61
4.4.4 EiffelBase2 Experiment 62
4.5 Results 63
4.5.1 Faults Found 63
4.5.2 Fault Complexity 65
4.5.3 Usage of Testing Time 67
4.5.4 Runtime Performance Overhead 68
4.5.5 Specification Writing Overhead 68
4.5.6 C# Experiments 69
4.5.7 EiffelBase2 Experiments 70
4.5.8 Threats to Validity 70
4.6 Related Work 70
4.6.1 Formal specifications for testing. 71
4.6.2 Inferred specifications for testing. 71
4.6.3 Model-based specifications at runtime. 72
4.7 Summary 72

5 Flexible Invariants for Complex Object Structures 73
5.1 Introduction 73
5.2 Motivating Examples 75
5.2.1 Observer pattern 75
5.2.2 Iterator Pattern 77
5.3 Existing Approaches 79
5.4 Semantic Collaboration 82
5.4.1 Preliminaries and Definitions 82
5.4.2 Semantic Collaboration: Goals and Proof Obligations . 84
5.4.3 Soundness of the Methodology 86
5.4.4 Examples 89
5.4.5 Default Annotations 93
5.4.6 Update guards 94
5.4.7 Extensions 94

5.5 Experimental Evaluation 96
5.5.1 Challenge Problems 96
5.5.2 Results and Discussion 98
5.5.3 Comparison with Existing Approaches 99
5.6 Summary 101

6 Verifying Reusable Components 103
6.1 Introduction 103
6.2 A Motivating Example 104
6.3 Verification Methodology for Model-Based Contracts 106
6.3.1 Encoding of Model Classes 106
6.3.2 Encoding of Model Queries 110
6.3.3 Abstract Framing 111
6.4 Experimental Evaluation 114
6.5 Related Work 116
6.6 Summary and Future Work 118

7 Debugging Failed Verification Attempts 119
7.1 Introduction 119
7.2 A Motivating Example 121
7.2.1 Comparison with other approaches 123
7.3 A Runtime Semantics of Boogie Programs 124
7.3.1 Input Language 125
7.3.2 Runtime Operational Semantics 127
7.4 Boogaloo: Implementation Details 133
7.5 Experimental Evaluation 136
7.6 Related Work 138
7.7 Summary and Future Work 140

8 Conclusions 141