

AUTOMATED DATABASE APPLICATIONS TESTING

Specification Representation for
Automated Reasoning

Rana Farid Mikhail
Donald Berndt
Abraham Kandel

S E R I E S I N
**MACHINE PERCEPTION
ARTIFICIAL INTELLIGENCE**
Volume 76

Contents

<i>Preface</i>	vii
<i>Acknowledgments</i>	ix
1. Introduction	1
1.1 The Need for Testing	1
1.2 Why Does Software have Errors?	2
1.3 Software Testing Definitions	3
1.3.1 Software Errors, Faults and Failures	3
1.3.2 Software Testing	3
1.4 When Should Testing Start in the Software Lifecycle?	5
1.5 Types of Testing Techniques	6
1.6 UML	7
1.7 Formal Specification Languages	8
1.8 Current Testing Technologies and Tools	8
1.8.1 Types of Automated Testing Tools	8
1.8.2 Design and Visual Modeling Tools	9
1.8.3 Automated Testing Tools	10
1.8.4 Disadvantage of Record and Playback Test Automation Tools	12
1.9 Related Literature	13
1.9.1 AGENDA: A Test Generator for Database Applications	14
1.9.2 Executable Software Specifications	14
1.10 Objectives of this Book	15
2. SpecDB: A Database Design for Software Specifications	21

2.1	Introduction	21
2.2	The Advantages of a Database Representation of Specifications	22
2.3	The Specification Database Design: SpecDB	23
2.4	Entities for Specific Software Requirements	24
2.4.1	Operator and Valid Operands	24
2.4.2	List_of_Tables and Table Descriptions	28
2.4.3	Types, Types_LOV and Type Restrictions	29
2.4.4	Variables and Restrictions on Variable Values	32
2.4.5	Procedure, Function, and Subroutine Descriptions	36
2.4.6	Subroutine Parameters and Post-Conditions	38
2.4.7	Dataflow, Triggers and Input/Output Definitions	42
2.4.8	Assignment and Calculation	45
2.4.9	Predicate	48
2.4.10	Database Operations	50
2.4.11	Classes, Objects and Class Relations	54
2.5	Conclusion	55
2.5.1	Other SpecDB Entities	57
3.	Representing Formal Specifications in SpecDB: A Trans- lation Algorithm	63
3.1	A Translation Algorithm from Formal Specifications to SpecDB	63
3.2	Assumptions and Restrictions	64
3.3	Preparing for Translation	65
3.4	Storing Formal Specifications	65
3.4.1	Defining Schema Names and Operations	65
3.4.2	Types of Schemas	66
3.4.3	User-Defined Types	67
3.4.4	Specifying Values for a User-Defined Type	67
3.4.5	Defining Sets or Tables	68
3.4.6	Function Declaration	69
3.4.7	Subroutine Input Restriction	69
3.4.8	Variable Definition and Subroutine Inputs	70
3.4.9	Output Definition	71
3.4.10	Setting Variable Restrictions	72
3.4.11	Assignment Operation	73
3.4.12	Assignment Operation and Function Call	74
3.4.13	Assignment Operation for Sets	75

3.4.14	Assignment Operation for a Set	76
3.4.15	Inserting Records in the Database	76
3.4.16	Output Table Definition	77
3.4.17	Assigning a Variable to a Calculation	78
3.4.18	Constraining the Values Populating a Table or Set	79
3.5	Conclusion	81
4.	An Automated Constraint Generator	83
4.1	The Design of Additional Tables in SpecDB	84
4.1.1	Using List_of_Tables and Table_Description	85
4.1.2	Adding the Constraints Table to SpecDB and Using the Predicate Table	87
4.1.3	Other Tables from SpecDB, Type_Excluded_Ranges and Types_LOV	90
4.2	Generating Database-Level Constraints	91
4.2.1	Generating Simple Static Constraints: The Algorithm	91
4.2.2	Enforcing Complex Dynamic Business Rules	95
4.3	Conclusion	97
5.	A Reverse Engineering Testing Tool	99
5.1	Technique	100
5.1.1	Input and Pre-Condition Classification and Categorization	101
5.1.2	Output and Post-Condition Classification and Categorization	104
5.2	Case Study	106
5.2.1	Problem Statement for the Vacation Salesman Commission Program	107
5.2.2	Implementation	107
5.2.3	Program Graph for the Salesman Commission Problem	110
5.2.4	DD-Path Graph for the Salesman Commission Problem	110
5.3	Examples	111
5.3.1	Test Case 1	113
5.3.2	Test Case 2	120
5.3.3	Test Case 3	122

5.4	Conclusion	124
6.	Enhancing Other Testing Tools Using SpecDB	125
6.1	Using the SpecDB Database to Expand the Testing Domain	125
6.2	Enhancing State Validation Tools	128
6.3	Expanding the Testing Scope Beyond Variables and Database States	129
6.4	Testing Different Operation Types	129
6.5	Conclusion	130
7.	Conclusion and Future Work	131
7.1	A Comparison Between Testing Tools	131
7.1.1	Test Cases and Scenarios	132
7.1.2	Comparing the Ability of Testing Tools to Identify Errors	134
7.2	Book Concepts at a Glance	135
7.3	Conclusion and Recommendations	136
7.4	Future Work at a Glance	138
Appendix A	SQL Scripts for SpecDB Constraints	143
Appendix B	PL/SQL Constraint Generator Code	173
	<i>Bibliography</i>	185
	<i>Index</i>	193