

Contents

- 1 Introduction** 1
 - 1.1 Verification and Validation Problem Statement 2
 - 1.2 Systems Engineering 3
 - 1.3 Systems Engineering Standards 5
 - 1.4 Model-Driven Architecture 6
 - 1.5 Systems Engineering Modeling Languages 8
 - 1.5.1 UML 2.x: Unified Modeling Language 8
 - 1.5.2 SysML: Systems Modeling Language 9
 - 1.5.3 IDEF: Integration Definition Methods 10
 - 1.6 Outline 11

- 2 Architecture Frameworks, Model-Driven Architecture, and Simulation** 15
 - 2.1 Architecture Frameworks 16
 - 2.1.1 Zachman Framework 16
 - 2.1.2 Open Group Architecture Framework 17
 - 2.1.3 DoD Architecture Framework 18
 - 2.1.4 UK Ministry of Defence Architecture Framework 25
 - 2.1.5 UML Profile for DoDAF/MODAF 25
 - 2.2 AP233 Standard for Data Exchange 26
 - 2.3 Executable Architectures or from Design to Simulation 26
 - 2.3.1 Why Executable Architectures? 27
 - 2.3.2 Modeling and Simulation as an Enabler
for Executable Architectures 28
 - 2.4 DoDAF in Relation to SE and SysML 31
 - 2.5 Conclusion 35

- 3 Unified Modeling Language** 37
 - 3.1 UML History 37

3.2	UML Diagrams	38
3.2.1	Class Diagram	39
3.2.2	Component Diagram	40
3.2.3	Composite Structure Diagram	41
3.2.4	Deployment Diagram	42
3.2.5	Object Diagram	43
3.2.6	Package Diagram	43
3.2.7	Activity Diagram	44
3.2.8	Activity Diagram Execution	47
3.2.9	Use Case Diagram	48
3.2.10	State Machine Diagram	49
3.2.11	Sequence Diagram	53
3.2.12	Communication Diagram	55
3.2.13	Interaction Overview Diagram	56
3.2.14	Timing Diagram	57
3.3	UML Profiling Mechanisms	58
3.4	Conclusion	59
4	Systems Modeling Language	61
4.1	SysML History	61
4.2	UML and SysML Relationships	62
4.3	SysML Diagrams	63
4.3.1	Block Definition Diagram	64
4.3.2	Internal Block Diagram	65
4.3.3	Package Diagram	66
4.3.4	Parametric Diagram	66
4.3.5	Requirement Diagram	67
4.3.6	Activity Diagram	69
4.3.7	State Machine Diagram	71
4.3.8	Use Case Diagram	72
4.3.9	Sequence Diagram	72
4.4	Conclusion	73
5	Verification, Validation, and Accreditation	75
5.1	V&V Techniques Overview	76
5.1.1	Inspection	77
5.1.2	Testing	77
5.1.3	Simulation	78
5.1.4	Reference Model Equivalence Checking	79
5.1.5	Theorem Proving	79
5.2	Verification Techniques for Object-Oriented Design	79
5.2.1	Design Perspectives	80
5.2.2	Software Engineering Techniques	80
5.2.3	Formal Verification Techniques	81
5.2.4	Program Analysis Techniques	82

5.3	V&V of Systems Engineering Design Models	83
5.4	Tool Support	88
5.4.1	Formal Verification Environments	88
5.4.2	Static Analyzers	90
5.5	Conclusion	92
6	Automatic Approach for Synergistic Verification and Validation	95
6.1	Synergistic Verification and Validation Methodology	96
6.2	Dedicated V&V Approach for Systems Engineering	99
6.2.1	Automatic Formal Verification of System Design Models	99
6.2.2	Program Analysis of Behavioral Design Models	100
6.2.3	Software Engineering Quantitative Techniques	101
6.3	Probabilistic Behavior Assessment	101
6.4	Established Results	102
6.5	Verification and Validation Tool	103
6.6	Conclusion	105
7	Software Engineering Metrics in the Context of Systems Engineering .	107
7.1	Metrics Suites Overview	107
7.1.1	Chidamber and Kemerer Metrics	107
7.1.2	MOOD Metrics	108
7.1.3	Li and Henry's Metrics	109
7.1.4	Lorenz and Kidd's Metrics	109
7.1.5	Robert Martin Metrics	109
7.1.6	Bansiya and Davis Metrics	110
7.1.7	Briand et al. Metrics	110
7.2	Quality Attributes	111
7.3	Software Metrics Computation	111
7.3.1	Abstractness (A)	112
7.3.2	Instability (I)	112
7.3.3	Distance from the Main Sequence (DMS)	113
7.3.4	Class Responsibility (CR)	113
7.3.5	Class Category Relational Cohesion (CCRC)	114
7.3.6	Depth of Inheritance Tree (DIT)	114
7.3.7	Number of Children (NOC)	114
7.3.8	Coupling Between Object Classes (CBO)	115
7.3.9	Number of Methods (NOM)	116
7.3.10	Number of Attributes (NOA)	117
7.3.11	Number of Methods Added (NMA)	117
7.3.12	Number of Methods Overridden (NMO)	118
7.3.13	Number of Methods Inherited (NMI)	118
7.3.14	Specialization Index (SIX)	119
7.3.15	Public Methods Ratio (PMR)	119

7.4	Case Study	120
7.5	Conclusion	123
8	Verification and Validation of UML Behavioral Diagrams	125
8.1	Configuration Transition System	125
8.2	Model Checking of Configuration Transition Systems	127
8.3	Property Specification Using CTL	129
8.4	Program Analysis of Configuration Transition Systems	130
8.5	V&V of UML State Machine Diagram	131
8.5.1	Semantic Model Derivation	132
8.5.2	Case Study	134
8.5.3	Application of Program Analysis	138
8.6	V&V of UML Sequence Diagram	141
8.6.1	Semantic Model Derivation	141
8.6.2	Sequence Diagram Case Study	142
8.7	V&V of UML Activity Diagram	145
8.7.1	Semantic Model Derivation	145
8.7.2	Activity Diagram Case Study	145
8.8	Conclusion	152
9	Probabilistic Model Checking of SysML Activity Diagrams	153
9.1	Probabilistic Verification Approach	153
9.2	Translation into PRISM	155
9.3	PCTL* Property Specification	160
9.4	Case Study	161
9.5	Conclusion	166
10	Performance Analysis of Time-Constrained SysML Activity Diagrams	167
10.1	Time Annotation	167
10.2	Derivation of the Semantic Model	169
10.3	Model-Checking Time-Constrained Activity Diagrams	170
10.3.1	Discrete-Time Markov Chain	172
10.3.2	PRISM Input Language	172
10.3.3	Mapping SysML Activity Diagrams into DTMC	173
10.3.4	Threads Identification	173
10.4	Performance Analysis Case Study	176
10.5	Scalability	181
10.6	Conclusion	187
11	Semantic Foundations of SysML Activity Diagrams	189
11.1	Activity Calculus	189
11.1.1	Syntax	190
11.1.2	Operational Semantics	194

- 11.2 Case Study 200
- 11.3 Markov Decision Process 203
- 11.4 Conclusion 203

- 12 Soundness of the Translation Algorithm 205**
 - 12.1 Notation 205
 - 12.2 Methodology 206
 - 12.3 Formalization of the PRISM Input Language 206
 - 12.3.1 Syntax 207
 - 12.3.2 Operational Semantics 208
 - 12.4 Formal Translation 210
 - 12.5 Case Study 213
 - 12.6 Simulation Preorder for Markov Decision Processes 215
 - 12.7 Soundness of the Translation Algorithm 217
 - 12.8 Conclusion 222

- 13 Conclusion 223**

- References 227**

- Index 241**