

Contents

<i>Preface</i>	vii
PART 1 Theoretical Aspects of Real-Time Systems	1
Chapter 1 A Discrete Model for Real-Time Environments	3
1.1 Introduction	3
1.2 Time and Events	4
1.3 Discrete Real-Time Systems	8
1.4 Composition and Decomposition of iDRTS Structures	9
1.5 DRTS as Algebras	14
1.6 A Model for Sequential Processes	18
1.7 Conclusions	24
Chapter 2 Distributed Synchronous Processes	25
2.1 Introduction: Real-Time and Reactive Systems	25
2.2 Reactive Programming: Asynchronism versus Strong Synchronism	28
2.3 The Weak Synchronous Paradigm	31
2.4 CoREA: A Weak Synchronous Process Algebra	32
2.4.1 Abstract Syntax	32
2.4.2 Weak Synchronous Operational Semantics	34
(a) Transition	36
(b) Inaction	36

(c) Case	36
(d) Concurrency	37
(e) Recursion	37
2.4.3 Congruence	39
2.4.4 Equational Laws	41
2.4.5 Brief Comparison with CCS	43
2.5 Application to Distributed Reactive Programming	43
2.5.1 ESTEREL: Brief Overview	44
2.5.2 $\mathcal{EC} = \text{ESTEREL} + \text{CoREA}$	45
2.6 Towards Weak Synchronous Distributed Implementations	47
(f) Protocol SB (Single Bus)	49
2.7 Application to Embedded Systems	52
2.8 Conclusion	54
Chapter 3 A Model of Probabilistic Processes	57
3.1 Introduction	58
3.2 Syntax of PCSP	59
3.3 Domain of Probabilistic Processes	60
3.4 Operator Semantics	62
3.4.1 Simple Operators	63
3.4.2 External Choice	63
3.4.3 Parallel Composition	64
3.4.4 Recursion	65
3.5 Example	66
3.6 Testing Semantics	67
3.6.1 Probabilistic Tests	67
3.6.2 Simple Operators	69
3.6.3 External Choice	69
3.6.4 Continuation of a Process after an Action in a State	73
3.7 Conclusion	75
Chapter 4 Modeling and Proving Grafquets with Transition Systems	77
4.1 Introduction	77
4.2 Grafquet	77
4.2.1 Graphical Elements	78
4.2.2 Temporal Aspect	78

4.2.3	Evolution Rules and Interpretation	79
4.3	First Modeling	80
4.3.1	Transition System	80
4.3.2	Synchronization Constraints	81
4.3.3	Temporization	83
4.3.4	Limits	85
4.4	Second Modeling	86
4.4.1	Construction of Basic Transition Systems	86
4.4.2	Building of the Global Transition System	88
4.5	Proof	89
4.5.1	Second Modeling	89
4.5.2	First Modeling	92
4.6	Example and Results	93
4.6.1	Example	93
4.6.2	Results	94
4.7	Conclusion	96

Chapter 5 Focus Points and Convergent Process

	Operators	97
5.1	Introduction	97
5.2	Preliminaries	99
5.2.1	A Short Description of μ CRL	99
5.2.2	Linear Process Operators	100
5.2.3	Internal Actions	104
5.3	Sufficient Conditions for the Equality of LPOs	104
5.4	Abstraction and Idle Loops	111
5.5	Examples	117
5.5.1	The Concurrent Alternating Bit Protocol	119
5.5.1.1	Specification	119
5.5.1.2	Expansion	122
5.5.1.3	Invariant	124
5.5.1.4	Abstraction and focus points	125

PART 2	Verification Methods for Real-Time Systems	135
Chapter 6	The Automatic Verification Using Symbolic Model-Checking	137
6.1	Introduction	137
6.2	Specification Method for Real-Time Systems	139
6.2.1	Specification by Timed Buchi Automaton	139
6.2.2	Generation of Timed Kripke Structure	141
6.3	Real-Time Temporal Logic	143
6.4	Verification Algorithm for Real-Time Symbolic Model Checking	144
6.4.1	Inverse Image Computation	144
6.4.2	DBMs (Differences Bounds Matrices)	145
6.4.2.1	Reachability Analysis (Test Timing Constraints)	145
6.4.3	Real-Time Symbolic Model Checking	147
6.5	The Verification System	149
6.5.1	Configuration of the Verification System	149
6.5.2	Verification Example	149
6.5.2.1	Specification	149
6.5.2.2	Verification	150
6.6	Conclusion	152
Chapter 7	Property Verification within a Process Algebra Framework	153
7.1	Introduction	153
7.2	The Circal Process Algebra	154
7.2.1	Informal Semantics	155
7.2.2	Formal Semantics	157
7.3	The Methodology	158
7.4	Constraint-Based Modeling	159
7.5	A Temporal Logic for Simultaneous Actions	161
7.6	The Representation of Properties	164
7.6.1	Formula-Based Characterization	165
7.6.2	Model-Based Characterization	169
7.7	Discussion and Future Work	173

PART 3 Synthesis Methods for Real-Time Systems 175

Chapter 8 Verifying Real-Time Systems with Standard Tools 177

8.1 Introduction 177

8.2 Timed Transition Models 180

 8.2.1 TTM Semantics 182

 8.2.2 Real-Time Temporal Logic 184

 8.2.3 An Example of a TTM 185

8.3 Translating Timed into Fair Systems 186

 8.3.1 The Conversion Procedure 189

8.4 Verifying Clocked Properties 190

8.5 A Real-Time Mutual Exclusion Protocol Example 191

8.6 Conclusion 195

Chapter 9 Beyond the Verification Approach: The Synthesis Approach 197

9.1 Introduction 197

9.2 Supervisory Control Theory 199

 9.2.1 Preliminaries 200

 9.2.2 Synthesis Procedures 203

9.3 Synthesis Algorithms for Totally Observed DES 204

 9.3.1 Wonham and Ramadge Synthesis Algorithm 205

 9.3.2 Barbeau, Kabanza, and St-Denis Synthesis Algorithm . 208

 9.3.3 Barbeau, Kabanza, and St-Denis Synthesis Algorithm (Safety Properties) 210

9.4 Description of the Experiment 211

9.5 Performance Study 215

9.6 Conclusion 217

PART 4 Extensions to Formal Languages 219

Chapter 10 Testing Semantics for Urgent Timed Process Algebras 221

10.1 Introduction 221

12.1.1	Motivation	265
12.1.2	Didactic and Industrial Objectives	266
12.1.3	Scope and Limitations of this Work	267
12.1.4	Overview of the Environment	267
12.1.4.1	Overview of the PADD notation	267
(g)	Introduction	267
(h)	Mathematical semantics	268
(i)	Explicit parallelism and communication	268
(j)	Parametric abstract types and monitors	268
(k)	DD schema embedded documentation	269
(l)	An example	269
12.1.5	Ramon Llull	271
12.1.6	Prior Usage	271
12.2	Some System Forms and Transformations	271
12.2.1	Introduction	271
12.2.2	Sequential (<i>SQ</i>) Form	272
12.2.3	Communicating Sequential (<i>CS</i>) Form	272
12.2.4	To Connections Interface Transformation (T_{ci})	272
12.2.5	Simple Cyclic (<i>SC</i>) Form	273
12.2.6	Structural (<i>ST</i>) Form	273
12.2.7	To Communicating Process Transformation (T_{cp})	273
12.3	Communication-Extended Abstract Types	274
12.4	Algebraic Framework	275
12.4.1	Introduction	275
12.4.2	Equivalences for Parallel Communicating Processes	275
12.4.3	Algebraic Semantics of Communications	276
12.4.4	Time Interval Algebra	277
12.5	Methods and Tools	278
12.5.1	Documentation	278
12.5.2	Purely Communicating System Modeling and Specification	279
12.5.3	Simulation	279
12.5.4	Allocation-Mapping Transformation	279
12.5.5	Communications Simplification	280
12.5.6	System Refinement Based on CATs	281
12.5.7	Proper Monitor Elimination Transformation (T_{me})	281
12.6	Conclusion and Future Work	283

Chapter 13	Analysis of Real-Time Systems Using OSA	285
13.1	Introduction	285
13.2	Object-Interaction Models	287
13.3	Object-Behavior Models	292
13.4	Object-Relationship Model	301
13.5	Tunable Formalism	305
13.6	State of OSA	306
13.7	Conclusion	307
Chapter 14	Algebraic Implementation of Model Checking Algorithms	309
14.1	Introduction	309
14.2	Algebraic Specification of CTL	312
14.3	Algebraic Implementation of a Model Checker	318
14.3.1	Structure of an Algebraic Compiler	318
14.3.2	The Macro Processor Generating Satisfiability Sets	321
14.3.3	Generating a Model Checker Program	323
14.3.4	Implementing the Macro Processor $\mathcal{M}_{A_{sets}}$	325
14.4	Conclusions	326
PART 6	Industrial Applications of Real-Time Systems	329
Chapter 15	An Automaton Based Algebra for Specifying Robotic Agents	331
15.1	Introduction	331
15.2	Related Work	332
15.3	Example	333
15.4	Our Framework	334
15.4.1	Elementary Processes	336
15.4.2	Composition Operators	337
15.5	Synthesis Example	342
15.6	Conclusion	344

Chapter 16	A Three-Level Analysis of a Simple Acceleration Maneuver, with Uncertainties	345
16.1	Introduction	345
16.2	Hybrid Input/Output Automata	346
16.3	Mathematical Preliminaries	348
16.3.1	Assumptions about the Constants	348
16.3.2	Some Useful Functions	348
16.3.2.1	Function f	348
16.3.2.2	Function g	349
16.3.2.3	Function f_1	349
16.3.2.4	Function h	350
16.4	High Level Specification V	351
16.4.1	Overview	351
16.4.2	Formal Description	352
16.5	Derivative Automaton D	353
16.5.1	Formal Description	353
16.5.2	Some Properties of D	355
16.5.3	D Implements V	357
16.5.4	An Approximate Result	359
16.6	Modifications to V and D to Incorporate Periodic Feedback	360
16.6.1	Modified High Level Specification V_1	360
16.6.2	Modified Derivative Automaton D_1	361
16.6.3	Modified Correctness Proof	361
16.7	The Implementation $Impl$	363
16.7.1	<i>Vehicle</i>	363
16.7.2	<i>Controller</i>	365
16.7.3	$Impl$	366
16.7.4	$Impl$ Implements D_1	368
16.8	Discussion	371
Chapter 17	Interface Specifications with Conjunctive Timing Constraints	375
17.1	Introduction	375
17.2	Timing Diagram Specifications	376
17.3	Causal Partitions over TD Specifications	381
17.4	Compatibility of Realizable Timing Diagrams	385

17.5 Extension to Cyclic Behaviors	388
17.6 Conclusions	392

Chapter 18 Experiments on a Fault Tolerant

Distributed System 393

18.1 Introduction	393
18.2 The Experiment Context	394
18.2.1 The Modular Project	394
18.2.2 Analyzed Mechanisms and Their Properties	395
18.2.3 Targeted Experiments	400
18.3 Synchronization Validation with Spin	401
18.3.1 Promela	401
18.3.2 Specification of the Particular Case	401
18.3.3 Generalization	402
18.3.4 Properties	404
18.3.5 Verification	405
18.4 The Analysis of the Detection Mechanisms	405
18.4.1 Validation of the Communication Strategy with Spin	406
18.4.1.1 Specification	406
18.4.1.2 Properties and Verification	407
18.4.2 Validation of the Local Detection with Kronos	408
18.4.2.1 Timed Automata	408
18.4.2.2 Specification	409
18.4.2.3 Properties	410
18.4.2.4 Verification	411
18.5 Discussion	412

Chapter 19 Specifying Multi-Level Security for an

Embedded Real-Time Control System 415

19.1 Introduction	415
19.2 An Avionics Real-Time Embedded Computer System	418
19.3 The TCSEC Guidelines	420
19.3.1 Security Policy	421
19.3.2 Accountability	423
19.3.3 Assurance and Documentation	423
19.4 Multi-Level Security	424
19.5 LOTOS	425

19.5.1	Using LOTOS in the Assurance Cycle	426
19.5.2	LOTOS Specifications	426
19.6	The Formal Security Model	427
19.6.1	Separability	428
19.6.2	Restrictiveness	431
19.7	Specifying Security of Networked Processors	431
19.7.1	TNIU Mechanism	432
19.7.2	Resettable Processors	436
19.7.3	A Security Policy Approach	438
19.8	Conclusion	439
 Chapter 20 An Algebraic Framework for the Feature Interaction Problem		441
20.1	Motivation and Background	441
20.2	Basic Concepts and Notation	442
20.3	A Method for Analyzing and Detecting Feature Interactions . .	444
20.3.1	Specification of Features in the Context of a System (Step 1)	444
20.3.2	Integration versus Composition of Features (Steps 2 and 3)	448
20.3.3	Derivation of Test Cases to Detect Interactions (Step 4)	450
20.3.4	Executing the System and Analysing the Results (Steps 5 and 6)	455
20.4	Conclusions and Research Directions	455
 Appendix A Algebraic Specification		457
	<i>Bibliography</i>	459
	<i>Index</i>	479