
Contents

PREFACE	VII
Overview	VII
“UML”-ising Formal Techniques	VIII
The RAISE Specification Language: RSL	VIII
Acknowledgments	VIII
Brief Guide to Volume 2	IX

Part I OPENING

1 Introduction	3
1.1 Introduction	3
1.1.1 Why This Volume?	3
1.1.2 Why Master These Principles, Techniques and Tools?	4
1.1.3 What Does This Volume “Contain”?	4
1.1.4 How Does This Volume “Deliver”?	5
1.2 Formal Techniques “Lite”	6
1.3 An RSL Primer	8
1.3.1 Types	8
1.3.2 The RSL Predicate Calculus	11
1.3.3 Concrete RSL Types	12
1.3.4 λ -Calculus+Functions	21
1.3.5 Other Applicative Expressions	24
1.3.6 Imperative Constructs	26
1.3.7 Process Constructs	27
1.3.8 Simple RSL Specifications	29
1.4 Bibliographical Notes	29

Part II SPECIFICATION FACETS

2	Hierarchies and Compositions	35
2.1	The Issues	35
2.1.1	Informal Illustrations	36
2.1.2	Formal Illustrations	36
2.2	Initial Methodological Consequences	37
2.2.1	Some Definitions	37
2.2.2	Principles and Techniques	38
2.3	The Main Example	40
2.3.1	A Hierarchical, Narrative Presentation	40
2.3.2	A Hierarchical, Formal Presentation	42
2.3.3	A Compositional, Narrative Presentation	45
2.3.4	A Compositional, Formal Presentation	47
2.4	Discussion	49
2.5	Bibliographical Notes: Stanisław Leshniewski	49
2.6	Exercises	50
3	Denotations and Computations	55
3.1	Introduction	55
3.1.1	Computations and Denotations	56
3.1.2	Syntax and Semantics	56
3.1.3	Characterisations	56
3.2	Denotational Semantics	57
3.2.1	A Simple Example: Numerals	57
3.2.2	The Denotational Principle	58
3.2.3	Expression Denotations	58
3.2.4	GOTO Continuations	62
3.2.5	Discussion of Denotational Semantics	72
3.3	Computational Semantics	74
3.3.1	The Issues	74
3.3.2	Two Examples	74
3.3.3	Expression Computations	74
3.3.4	Computational Semantics of GOTO Programs	78
3.3.5	Computational Semantics of Coroutine Programs	83
3.3.6	Discussion	85
3.4	Review: Denotations and Computations	86
3.5	Some Pioneers of Semantics	86
3.5.1	John McCarthy	86
3.5.2	Peter Landin	88
3.6	Exercises	90

4	Configurations: Contexts and States	93
4.1	Introduction	94
4.2	The Issues	97
4.3	“Real-World” Contexts and States	98
4.3.1	A Physical System: Context and State	99
4.3.2	Configurations of Contexts and States	99
4.3.3	Nonphysical System: Context and State	100
4.3.4	Discussion, I	101
4.3.5	Discussion, II	102
4.4	First Summary: Contexts and States	102
4.4.1	General	102
4.4.2	Development Principles and Techniques	103
4.5	Programming Language Configurations	104
4.6	Concurrent Process Configurations	104
4.6.1	The Example	104
4.6.2	Summary	110
4.7	Second Summary: Contexts and States	111
4.8	Information States and Behaviour States	112
4.8.1	Program Flowcharts as State Machine Data	112
4.8.2	Flowcharts \equiv Machines	113
4.8.3	Flowchart Machines	113
4.8.4	Observations	114
4.8.5	Conclusion	114
4.9	Final Summary: Contexts and States	115
4.10	Exercises	116

Part III A CRUCIAL DOMAIN AND COMPUTING FACET

5	Time, Space and Space/Time	121
5.1	Time	122
5.1.1	Time — The Basics	122
5.1.2	Time — General Issues	124
5.1.3	“A-Series” and “B-Series” Models of Time	125
5.1.4	A Continuum Theory of Time	125
5.1.5	Temporal Events	126
5.1.6	Temporal Behaviour	127
5.1.7	Representation of Time	127
5.1.8	Operations “on” Time	128
5.2	Space	129
5.2.1	Space — The Basics	129
5.2.2	Location-Varying Entities	129
5.2.3	Locations and Dynamicity	131
5.2.4	Space — General Issues	132
5.3	Space/Time	135

5.3.1	A Guiding Example	135
5.3.2	Representation of Space/Time	135
5.3.3	Blizard's Theory of Time-Space	136
5.4	Discussion	137
5.5	Bibliographical Notes	137
5.6	Exercises	137

Part IV LINGUISTICS

6	Pragmatics	145
6.1	Introduction	145
6.2	Everyday Pragmatics	146
6.3	"Formal" Pragmatics	146
6.4	Discussion	147
6.4.1	General	147
6.4.2	Principles and Techniques	148
6.5	Bibliographical Note	148
6.6	Exercises	149
7	Semantics	151
7.1	Introduction	151
7.2	Concrete Semantics	152
7.3	"Abstract" Semantics	152
7.4	Preliminary Semantics Concepts	152
7.4.1	Syntactic and Semantic Types	153
7.4.2	Contexts	153
7.4.3	States	154
7.4.4	Configurations	154
7.4.5	Interpretation, Evaluation and Elaboration	154
7.5	Denotational Semantics	155
7.5.1	Simple Case	156
7.5.2	Composite Case	156
7.6	Macro-expansion Semantics	157
7.6.1	Rewriting	157
7.6.2	Macro-expansion	158
7.6.3	Inductive Rewritings	158
7.6.4	Fix Point Evaluation	161
7.7	Operational and Computational Semantics	161
7.7.1	Stack Semantics	162
7.7.2	Attribute Grammar Semantics	162
7.8	Proof Rule Semantics	166
7.9	Discussion	169
7.9.1	General	169
7.9.2	Principles, Techniques and Tools	169

7.10	Bibliographical Notes	170
7.11	Exercises	170
8	Syntax	173
8.1	The Issues	174
8.1.1	Form and Content: Syntax and Semantics	174
8.1.2	Structure and Contents of This Chapter	175
8.2	Sentential Versus Semantical Structures	175
8.2.1	General	175
8.2.2	Examples of Sentential Structures	176
8.2.3	Examples of Semantical Structures	178
8.3	The First Abstract Syntax, John McCarthy	181
8.3.1	Analytic Grammars: Observers and Selectors	182
8.3.2	Synthetic Grammars: Generators	182
8.4	BNF Grammars \approx Concrete Syntax	183
8.4.1	BNF Grammars	183
8.4.2	BNF \leftrightarrow RSL Parse Trees Relations	184
8.5	Structure Generators and Recognisers	186
8.5.1	Context-Free Grammars and Languages	186
8.5.2	Parse Trees	188
8.5.3	Regular Expressions and Languages	189
8.5.4	Language Recognisers	190
8.6	XML: Extensible Markup Language	190
8.6.1	An Example	191
8.6.2	Discussion	192
8.6.3	Historical Background	192
8.6.4	The Current XML “Craze”	193
8.6.5	XML Expressions	193
8.6.6	XML Schemas	195
8.6.7	References	197
8.7	Abstract Syntaxes	197
8.7.1	Abstract Syntax of a Storage Model	197
8.7.2	Abstract Syntaxes of Other Storage Models	200
8.8	Converting RSL Types to BNF	202
8.8.1	The Problem	202
8.8.2	A Possible Solution	202
8.9	Discussion of Informal and Formal Syntax	203
8.9.1	General	203
8.9.2	Principles, Techniques and Tools	204
8.10	Bibliographical Notes	204
8.11	Exercises	205

9	Semiotics	213
9.1	Semiotics = Syntax \oplus Semantics \oplus Pragmatics	213
9.2	Semiotics	214
9.3	Language Components	215
9.4	Linguistics	216
9.5	Languages and Systems	217
9.5.1	Professional Languages	218
9.5.2	Metalanguages	219
9.5.3	Systems	219
9.5.4	System Diagram Languages	232
9.5.5	Discussion of System Concepts	232
9.5.6	Systems as Languages	233
9.6	Discussion	233
9.6.1	General	233
9.6.2	Principles, Techniques and Tools	234
9.7	Charles Sanders Peirce	234
9.8	Bibliographical Notes	234
9.9	Exercises	235

Part V FURTHER SPECIFICATION TECHNIQUES

10	Modularisation	243
10.1	Introduction	244
10.1.1	Some Examples	244
10.1.2	Preparatory Discussion	249
10.1.3	Structure of Chapter	252
10.2	RSL Classes, Objects and Schemes	253
10.2.1	Introducing the RSL “class” Concept	253
10.2.2	The RSL “class” Concept	257
10.2.3	The RSL “object” Concept	257
10.2.4	The RSL “scheme” Concept	257
10.2.5	RSL “scheme” Parameterisation	263
10.2.6	A “Large-Scale” Example	265
10.2.7	Definitions: Class, Scheme and Object	270
10.3	UML and RSL	271
10.3.1	Overview of UML Diagrams	271
10.3.2	Class Diagrams	272
10.3.3	Class Diagrams	273
10.3.4	Example: Railway Nets	276
10.3.5	Comparison of UML and RSL OO Constructs	278
10.3.6	References	279
10.3.7	Class Diagram Limitations	280
10.4	Discussion	280
10.4.1	Modularity Issues	280

10.4.2	Principles, Techniques and Tools	281
10.5	Bibliographical Notes	282
10.6	Exercises	282
11	Automata and Machines	285
11.1	Discrete State Automata	286
11.1.1	Intuition	287
11.1.2	Motivation	288
11.1.3	Pragmatics	288
11.2	Discrete State Machines	290
11.3	Finite State Automata	291
11.3.1	Regular Expression Language Recognisers	292
11.3.2	Regular Expressions	293
11.3.3	Formal Languages and Automata	294
11.3.4	Automaton Completion	295
11.3.5	Nondeterministic Automata	295
11.3.6	Minimal State Finite Automata	296
11.3.7	Finite State Automata Formalisation, I	297
11.3.8	Finite State Automata Realisation, I	297
11.3.9	Finite State Automaton Formalisation, II	298
11.3.10	Finite State Automata Realisation, II	299
11.3.11	Finite State Automata — A Summary	299
11.4	Finite State Machines	300
11.4.1	Finite State Machine Controllers	300
11.4.2	Finite State Machine Parsers	303
11.4.3	Finite State Machine Formalisation	304
11.4.4	Finite State Machine Realisation	305
11.4.5	Finite State Machines — A Summary	306
11.5	Pushdown Stack Devices	307
11.5.1	Pushdown Stack Automata and Machines	307
11.5.2	Formalisation of Pushdown Stack Machines	309
11.5.3	Pushdown Stack Device Summary	310
11.6	Bibliographical Notes: Automata and Machines	311
11.7	Exercises	311

Part VI CONCURRENCY AND TEMPORALITY

12	Petri Nets	315
	Christian Krog Madsen is chief author of this chapter	
12.1	The Issues	315
12.2	Condition Event Nets (CENs)	316
12.2.1	Description	316
12.2.2	Small CEN Examples	317
12.2.3	An RSL Model of Condition Event Nets	320

12.3	Place Transition Nets (PTNs)	323
12.3.1	Description	323
12.3.2	Small PTN Examples	324
12.3.3	An RSL Model of Place Transition Nets	324
12.3.4	Railway Domain Petri Net Examples	328
12.4	Coloured Petri Nets (CPNs)	333
12.4.1	Description	333
12.4.2	A CPN Example	336
12.4.3	An RSL Model of Coloured Petri Nets	336
12.4.4	Timed Coloured Petri Nets	341
12.5	CEN Example: Work Flow System	342
12.5.1	Project Planning	342
12.5.2	Project Activities	346
12.5.3	Project Generation	353
12.6	CPN and RSL Examples: Superscalar Processor	356
12.6.1	Description	356
12.6.2	Coloured Petri Net Model	357
12.6.3	RSL Model: Superscalar Processor	362
12.7	Discussion	371
12.8	Bibliographical Notes	372
12.9	Exercises	372
13	Message and Live Sequence Charts	375
Christian Krog Madsen is chief author of this chapter		
13.1	Message Sequence Charts	376
13.1.1	The Issues	376
13.1.2	Basic MSCs (BMSCs)	376
13.1.3	High-Level MSCs (HMSCs)	383
13.1.4	An RSL Model of HMSC Syntax	385
13.1.5	MSCs Are HMSCs	385
13.1.6	Syntactic Well-formedness of MSCs	386
13.1.7	An Example: IEEE 802.11 Wireless Network	391
13.1.8	Semantics of Basic Message Sequence Charts	400
13.1.9	Semantics of High-Level Message Sequence Charts	401
13.2	Live Sequence Charts: Informal Presentation	402
13.2.1	Live Sequence Chart Syntax	402
13.2.2	A Live Sequence Chart Example, I	408
13.3	Process Algebra	409
13.3.1	The Process Algebra PA_ϵ	410
13.3.2	Semantics of PA_ϵ	416
13.3.3	The Process Algebra PAC_ϵ	420
13.3.4	Semantics for PAC_ϵ	423
13.4	Algebraic Semantics of Live Sequence Charts	427
13.4.1	Textual Syntax of Live Sequence Charts	427
13.4.2	Semantics of Live Sequence Charts	428

13.4.3	The Live Sequence Chart Example, II	431
13.5	Relating Message Charts to RSL	431
13.5.1	Types of Integration	432
13.5.2	An RSL Subset	433
13.5.3	Relating Live Sequence Charts to RSL	436
13.5.4	Checking Satisfaction	442
13.5.5	Tool Support	443
13.6	Communicating Transaction Processes (CTP)	443
13.6.1	Intuition	443
13.6.2	Narration of CTPs	444
13.6.3	A Dining Philosophers Example	450
13.6.4	Formalisation of CTPs	453
13.7	Discussion	467
13.7.1	General	467
13.7.2	Principles, Techniques and Tools	468
13.8	Bibliographical Notes	469
13.9	Exercises	470
14	Statecharts	475
	Christian Krog Madsen is chief author of this chapter	
14.1	Introduction	475
14.2	A Narrative Description of Statecharts	476
14.3	An RSL Model of the Syntax of Statecharts	481
14.4	Examples	484
14.4.1	Railway Line Automatic Blocking	484
14.4.2	Railway Line Direction Agreement System	488
14.4.3	Wireless Rain Gauge	493
14.5	A Process Algebra for Statecharts	498
14.5.1	SPL: The Statechart Process Language	499
14.5.2	Semantics of SPL	500
14.5.3	Equivalence for SPL Terms	500
14.6	Semantics of Statecharts	503
14.6.1	An SPL Semantics for Statecharts	503
14.6.2	Statechart Example	504
14.7	Relating Statecharts to RSL	505
14.7.1	Syntactical Restrictions	506
14.7.2	Satisfaction Relation	506
14.7.3	Checking Satisfaction	507
14.7.4	Tool Support	508
14.8	Discussion	508
14.8.1	General	508
14.8.2	Principles, Techniques and Tools	508
14.9	Bibliographical Notes	509
14.10	Exercises	509

15	Quantitative Models of Time	517
15.1	The Issues	517
15.1.1	Soft Temporalities	517
15.1.2	Hard Temporalities	518
15.1.3	Soft and Hard Real-Time	518
15.1.4	Examples — “Ye Olde Way”!	518
15.1.5	Structure of This Chapter	520
15.2	Temporal Logic	520
15.2.1	The Issues	521
15.2.2	A Philosophical Linguistics Background	521
15.2.3	Interval Temporal Logic, ITL	522
15.2.4	The Classic Temporal Operators: \Diamond, \Box	527
15.3	The Duration Calculus	528
15.3.1	Examples, Part I	528
15.3.2	Some Basic Notions	529
15.3.3	Examples, Part II	532
15.3.4	The Syntax	536
15.3.5	The Informal Semantics	538
15.3.6	Examples, Part III	539
15.3.7	Transitions and Events	550
15.3.8	Discussion: From Domains to Designs	554
15.4	TRSL: RSL with Timing	555
15.4.1	TRSL Design Criteria	555
15.4.2	The TRSL Language	558
15.4.3	Another Gas Burner Example	559
15.4.4	Discussion	562
15.5	RSL with Timing and Durations	563
15.5.1	Review of TRSL	563
15.5.2	TRSL and Duration Calculus	564
15.6	Discussion	567
15.6.1	General	567
15.6.2	Principles, Techniques and Tools	567
15.7	Bibliographical Notes	568
15.8	Exercises	568

Part VII INTERPRETER AND COMPILER DEFINITIONS

16	SAL: Simple Applicative Language	573
16.1	A Caveat	574
16.2	The SAL Syntax	574
16.2.1	Informal Exposition of SAL Syntax	574
16.2.2	Formal Exposition of SAL Syntax	575
16.2.3	Comments	576
16.3	A Denotational Semantics	576

16.3.1	An Informal Semantics	576
16.3.2	A Formal Semantics	577
16.3.3	Review of SAL Semantics, 1	579
16.3.4	Two Asides	580
16.4	A First-Order Applicative Semantics	582
16.4.1	Syntactic Types	582
16.4.2	Semantic Types	582
16.4.3	Abstraction Functions	583
16.4.4	Auxiliary Functions	584
16.4.5	Semantic Functions	585
16.4.6	Review	588
16.4.7	Review of SAL Semantics, 2	588
16.5	An Abstract, Imperative Stack Semantics	589
16.5.1	Design Decisions — Informal Motivation	589
16.5.2	Semantics Style Observations	590
16.5.3	Syntactic Types	590
16.5.4	Semantic Types	591
16.5.5	Abstraction Functions	591
16.5.6	Run-Time Functions	591
16.5.7	Semantic Functions	592
16.5.8	Review of SAL Semantics, 3	598
16.6	A Macro-expansion Semantics	598
16.6.1	Analysis of Stack Semantics	599
16.6.2	Syntactic Types	606
16.6.3	Compile-Time Types	606
16.6.4	Run-Time Semantic Types	606
16.6.5	Run-Time State	606
16.6.6	Run-Time Stack Operations	607
16.6.7	Run-Time Stack Search for Variable Values	607
16.6.8	Macro-expansion Functions	608
16.6.9	Review of SAL Semantics, 4	616
16.7	ASM: An Assembler Language	616
16.7.1	Semantic Types	616
16.7.2	The Computer State	617
16.7.3	The Address Concept	617
16.7.4	Machine Instructions	618
16.7.5	Machine Semantics	620
16.7.6	Review of ASM	625
16.8	A Compiling Algorithm	625
16.8.1	Syntactic Types	626
16.8.2	Compile-Time Types and State	626
16.8.3	Compile-Time Dynamic Function	626
16.8.4	Compile-Time Static Function	627
16.8.5	Run-Time Constant Values	627
16.8.6	Compilation Functions	628

16.8.7	Review of Compiling Algorithm	635
16.9	An Attribute Grammar Semantics	636
16.9.1	Abstract Syntactic Types	637
16.9.2	SAL BNF Grammar, 1	637
16.9.3	Node Attributes	637
16.9.4	Constants	638
16.9.5	Some Typographical Distinctions	638
16.9.6	Compilation Functions	638
16.9.7	Review of Attribute Semantics, 1	641
16.10	Another Attribute Grammar Semantics	643
16.10.1	Abstract Syntactic Types	645
16.10.2	SAL BNF Grammar, 2	645
16.10.3	Global Variables	646
16.10.4	Constants	648
16.10.5	Node Attributes	648
16.10.6	Compilation Functions	648
16.10.7	Review of Attribute Semantics, 2	651
16.11	Discussion	651
16.11.1	General	651
16.11.2	Principles, Techniques and Tools	653
16.12	Review and Bibliographical Notes	655
16.13	Exercises	658
17	SIL: Simple Imperative Language	659
17.1	The Background	659
17.2	Syntactic Types	660
17.2.1	Concrete, Schematic Syntax	660
17.2.2	Abstract Syntax	660
17.3	Imperative Denotational Semantics	661
17.3.1	Semantic Types	661
17.3.2	Auxiliary Semantic Functions	662
17.3.3	Semantic Functions	662
17.4	Macro-expansion Semantics	663
17.4.1	Syntactic Types	664
17.4.2	Compile-Time Semantic Types	664
17.4.3	Run-Time Semantic Types	664
17.4.4	Run-Time State Declaration and Initialisation	665
17.4.5	Abstraction Functions	666
17.4.6	Macros	666
17.5	Discussion	668
17.5.1	General	668
17.5.2	Principles, Techniques and Tools	668
17.6	Bibliographical Notes	669
17.7	Exercises	669

18	SMIL: Simple Modular, Imperative Language	671
18.1	Syntactic Types	671
18.2	A Denotational Semantics	672
18.2.1	Semantic Types	672
18.2.2	Auxiliary Functions	673
18.2.3	Semantic Functions	673
18.3	A Macro-expansion Semantics	675
18.3.1	Run-Time Semantic Types	675
18.3.2	Compile/Run-Time Semantic Types	676
18.3.3	Compile-Time Semantic Types	677
18.3.4	Semantic Functions	677
18.4	Discussion	679
18.4.1	General	679
18.4.2	Principles, Techniques and Tools	679
18.5	Bibliographical Notes	680
18.6	Exercises	680
19	SPIL: Simple Parallel, Imperative Language	681
19.1	The Problem	681
19.2	Syntax	682
19.2.1	Informal Syntax	682
19.2.2	Formal Syntax	684
19.3	Process Concepts and Semantic Types	684
19.3.1	Syntactic Notions	685
19.3.2	Machines and Interpreters	686
19.3.3	Semantic Notions and Types	686
19.4	Process-Oriented Semantic Types	688
19.4.1	Unique Process Identifiers $\pi : \Pi$	688
19.4.2	The Heap $\xi : \Xi$	689
19.4.3	Input/Output Channel Bindings	690
19.4.4	Environments $\rho : \text{ENV}$	691
19.4.5	State Composition $\Psi, \Gamma, \Xi, \Sigma, \Omega$	691
19.5	Initial and Auxiliary Semantic Functions	693
19.5.1	Start Function	693
19.5.2	System Function	693
19.5.3	Bind and Allocate Functions	694
19.5.4	Free and Bound Functions	694
19.5.5	Distribute Function	694
19.5.6	Transition Loop	695
19.6	Semantic Functions	695
19.6.1	The Next-State Transition Function	695
19.6.2	The Assignment Statement	696
19.6.3	The case Statement	696
19.6.4	The while Loop	697
19.6.5	The repeat until Loop	697

19.6.6	Simple Input/Output Processes	698
19.6.7	The Parallel Process Command, 	699
19.6.8	The <i>stop</i> Process Technicality	699
19.6.9	The Process <i>call</i> Command	700
19.6.10	Internal Nondeterministic Processes, []	700
19.6.11	External Nondeterministic Processes, []	700
19.6.12	Nondeterministic Input/Output Processes	701
19.6.13	The Embedded System Process Command	702
19.6.14	A <i>finish</i> Process Technicality	702
19.7	Discussion	702
19.7.1	General	702
19.7.2	Principles, Techniques and Tools	703
19.8	Bibliographical Notes	703
19.9	Exercises	704

Part VIII CLOSING

20	Closing	709
20.1	A Summary	709
20.2	Conclusion: Volumes 1 and 2	710
20.3	Preview of Volume 3	710
20.4	“UML”-ising Formal Techniques	712

Part IX APPENDIXES

A	Naming Convention	717
B	Indexes	721
B.1	Symbols Index	722
B.1.1	Time/Space	722
B.1.2	Modular RSL	722
B.1.3	Petri Nets	722
B.1.4	Message Sequence Charts	723
B.1.5	Live Sequence Charts	723
B.1.6	Statecharts	723
B.1.7	Temporal Logics	723
B.1.8	Duration Calculus	723
B.1.9	Timed RSL: TRSL	723
B.1.10	Abbreviations	724
B.2	Concepts Index	725
B.3	Characteriations and Definitions Index	744
B.4	Authors Index	746

References	751
-------------------------	-----

Software Engineering 2
Specification of Systems and Languages
Bjørner, D.
2006, XXIV, 780 p. 151 illus., Hardcover
ISBN: 978-3-540-21150-1