

Dines Bjørner

# SOFTWARE ENGINEERING

Volume 2

**Systems and Languages**

**Table of Contents**

November 13, 2005

Springer

Berlin Heidelberg New York  
Hong Kong London  
Milan Paris Tokyo

<b>PREFACE</b> .....	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

---

<b>1 Introduction</b> .....	<b>3</b>
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
Type Expressions .....	8
Type Definitions .....	10
1.3.2 The RSL Predicate Calculus .....	11
Propositional Expressions .....	11
Simple Predicate Expressions .....	12
Quantified Expressions .....	12
1.3.3 Concrete RSL Types .....	12
Set Enumerations .....	12
Cartesian Enumerations .....	13
List Enumerations .....	13
Map Enumerations .....	14
Set Operations .....	15
Cartesian Operations .....	17
List Operations .....	17
Map Operations .....	19
1.3.4 $\lambda$ -Calculus+Functions .....	21
The $\lambda$ -Calculus Syntax .....	21
Free and Bound Variables .....	21
Substitution .....	22
$\alpha$ -Renaming and $\beta$ -Reduction .....	22
Function Signatures .....	22
Function Definitions .....	23
1.3.5 Other Applicative Expressions .....	23
Let Expressions .....	23
Conditionals .....	25
Operator/Operand Expressions .....	25
1.3.6 Imperative Constructs .....	26
Variables and Assignment .....	26
Statement Sequences and <b>skip</b> .....	26
Imperative Conditionals .....	26
Iterative Conditionals .....	27
Iterative Sequencing .....	27
1.3.7 Process Constructs .....	27
Process Channels .....	27
Process Composition .....	27

	Input/Output Events .....	28
	Process Definitions .....	28
1.3.8	Simple RSL Specifications .....	29
1.4	Bibliographical Notes .....	29

---

## Part II SPECIFICATION FACETS

---

<b>2</b>	<b>Hierarchies and Compositions .....</b>	<b>35</b>
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
	“Top-down” .....	38
	“Bottom-up” .....	38
2.2.2	Principles and Techniques .....	38
2.3	The Main Example .....	39
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 .....	48
2.5	Bibliographical Notes .....	49
2.6	Exercises .....	50
<b>3</b>	<b>Denotations and Computations .....</b>	<b>55</b>
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
	An Extension .....	61
3.2.4	GOTO Continuations .....	62
	The Problem .....	64
	Syntactic Types .....	66
	Semantic Types .....	67
	The Main Semantic Functions .....	67
	A Profound Problem .....	70
	Further Remarks on Example 3.3 .....	70
	The Remaining Semantic Functions .....	70
	Discussion .....	72
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
	A Computational State .....	75
	Motivating the Control Stack .....	75
	The Elaboration Functions .....	76
	Discussion .....	78

3.3.4	Computational Semantics of GOTO Programs .....	78
	The Syntax .....	78
	Some Semantic Observations .....	79
	Semantic Types.....	80
3.3.5	Computational Semantics of Coroutine Programs .....	83
	"Coroutine" Program Text I.....	83
	Coroutine Program Text II.....	84
3.3.6	Discussion .....	85
	On the Mechanisation of Continuations .....	85
	On the "Power" of GOTOs via Label Variables .....	85
	On Computational Semantics.....	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
<b>4</b>	<b>Configurations: Contexts and States .....</b>	<b>93</b>
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
	The Context .....	100
	The State .....	101
	A Model .....	101
	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
	Model versus Specification States .....	103
	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
	Comments on the Semantics of Fig. 4.2[A] .....	106
	Comments on the Semantics of Fig. 4.2[B] .....	106
	Comments on the Semantics of Fig. 4.2[C] .....	106
	Comments on the Semantics of Fig. 4.2[D] .....	106
	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 .....	115
4.9	Final Summary: Contexts and States .....	115
4.10	Exercises .....	116

<b>5</b>	<b>Time, Space and Space/Time</b> .....	121
5.1	Time .....	122
5.1.1	Time — The Basics .....	122
	Time-Varying Entities = Dynamic Entities .....	122
	Time and Dynamicity .....	124
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
	Point, Curve, Surface and Volume .....	132
	Spatial “Events” .....	133
	Spatial “Behaviours” .....	133
	Representation of Spatial Bodies .....	133
	Operations on Space .....	134
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
	Discussion of the <i>Blizard</i> Model of Space-Time .....	137
5.4	Discussion .....	137
5.5	Bibliographical Notes .....	137
5.6	Exercises .....	137

---

## Part IV LINGUISTICS

---

<b>6</b>	<b>Pragmatics</b> .....	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
<b>7</b>	<b>Semantics</b> .....	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
	Static Inductive Semantics .....	159
	Dynamic Inductive Semantics .....	159
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
	A Symbolic Attributed Parse Tree Example .....	165
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
<b>8</b>	<b>Syntax .....</b>	<b>173</b>
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
	Syntax of Sentential Structures .....	175
	Syntax of Semantical Structures .....	176
8.2.2	Examples of Sentential Structures .....	176
	Modelling Simple Sentential Structures .....	177
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 .....	181
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
	Values and Value Types .....	197

	Locations and Location Types .....	198
	Storages .....	199
	Type Constraints .....	199
8.7.2	Abstract Syntaxes of other Storage Models .....	200
	Informal Exposition .....	200
	Formal Exposition .....	201
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
<b>9</b>	<b>Semiotics</b> .....	<b>213</b>
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
	A Physical System View .....	219
	Examples of Physical and Nonphysical Systems .....	220
	A Linguistic Systems View .....	222
	Example System Languages .....	223
	A Flowchart Language .....	226
	Informal Syntax of Simple Flowcharts .....	227
	Formal Syntax of Simple Flow Charts .....	228
	Structured Flowcharts .....	229
	System Diagrams Versus Formal Specifications .....	230
	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

---

<b>10</b>	<b>Modularisation</b> .....	<b>243</b>
10.1	Introduction .....	244
	10.1.1 Some Examples .....	244
	Review of Examples .....	248
	10.1.2 Preparatory Discussion .....	249
	Software Devices .....	250
	Abstract Data Types $\rightarrow$ Algebraic Semantics .....	250
	The Frames Approach .....	250

	The Entity-Relationship (ER) Approach . . . . .	251
	General Pragmatics of Modularisation . . . . .	252
	General Semantics of Modularisation . . . . .	252
	General Syntax of Modularisation . . . . .	252
	General Module Specification Method . . . . .	252
10.1.3	Structure of Chapter . . . . .	252
10.2	RSL Classes, Objects and Schemes . . . . .	253
10.2.1	Introducing the RSL “class” Concept . . . . .	253
	Meaning of RSL Declarations — A Review . . . . .	253
	First Motivation of the RSL “class” Concept: Focus . . . . .	253
	Second Motivation of the RSL “class” Concept: Semantic Algebras . . . . .	253
	Third Motivation of the RSL “class” Concept . . . . .	254
	Fourth Motivation of the RSL “class” Concept: Named schemes . . . . .	256
10.2.2	The RSL “class” Concept . . . . .	257
10.2.3	The RSL “object” Concept . . . . .	257
10.2.4	The RSL “scheme” Concept . . . . .	257
	Simple Schemes . . . . .	257
	Scheme Extensions . . . . .	259
	Hiding . . . . .	260
	Etcetera . . . . .	262
10.2.5	RSL “scheme” Parameterisation . . . . .	263
	Motivation: Why and How Scheme Parameters? . . . . .	263
	The Syntax and Semantics of Parameterised Schemes . . . . .	265
10.2.6	A “Large-Scale” Example . . . . .	265
	The Contrasting Background Example . . . . .	266
	Narrative . . . . .	266
	Formalisation . . . . .	266
	The Schematised Example . . . . .	268
10.2.7	Definitions: Class, Scheme and Object . . . . .	270
10.3	UML and RSL . . . . .	271
10.3.1	Overview of UML Diagrams . . . . .	271
	Use Case Diagrams . . . . .	271
	Sequence/Collaboration Diagrams . . . . .	272
	Statechart Diagrams . . . . .	272
10.3.2	Class Diagrams . . . . .	272
	UML “Standardisation” . . . . .	273
10.3.3	Class Diagrams . . . . .	273
	Classes . . . . .	273
	Association . . . . .	274
	Links . . . . .	275
	Generalisations . . . . .	276
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 . . . . .	279
10.4	Discussion . . . . .	280
10.4.1	Modularity Issues . . . . .	280
	Modular Specification and Programming . . . . .	280
	Stability of Modularity Concepts . . . . .	280
	Whither Object-Oriented (OO) “Programming” . . . . .	280
	Schema, Object and Module Calculi . . . . .	281
	Formalisations of UML’s Class Concept . . . . .	281
10.4.2	Principles, Techniques and Tools . . . . .	281
10.5	Bibliographical Notes . . . . .	282

10.6	Exercises	282
<b>11</b>	<b>Automata and Machines</b>	<b>283</b>
11.1	Discrete State Automata	284
11.1.1	Intuition	286
11.1.2	Motivation	286
11.1.3	Pragmatics	286
11.2	Discrete State Machines	288
11.3	Finite State Automata	289
11.3.1	Regular Expression Language Recognisers	290
11.3.2	Regular Expressions	291
11.3.3	Formal Languages and Automata	292
11.3.4	Automaton Completion	293
11.3.5	Nondeterministic Automata	293
11.3.6	Minimal State Finite Automata	294
11.3.7	Finite State Automata Formalisation, I	295
11.3.8	Finite State Automata Realisation, I	295
11.3.9	Finite State Automaton Formalisation, II	296
11.3.10	Finite State Automata Realisation, II	297
11.3.11	Finite State Automata — A Summary	297
11.4	Finite State Machines	298
11.4.1	Finite State Machine Controllers	299
11.4.2	Finite State Machine Parsers	301
11.4.3	Finite State Machine Formalisation	303
11.4.4	Finite State Machine Realisation	303
11.4.5	Finite State Machines — A Summary	305
11.5	Pushdown Stack Devices	305
11.5.1	Pushdown Stack Automata and Machines	305
11.5.2	Formalisation of Pushdown Stack Machines	307
11.5.3	Pushdown Stack Device Summary	309
11.6	Bibliographical Notes: Automata and Machines	309
11.7	Exercises	310

---

## Part VI CONCURRENCY AND TEMPORALITY

---

<b>12</b>	<b>Petri Nets</b>	<b>313</b>
	Christian Krog Madsen is chief author of this chapter	
12.1	The Issues	313
12.2	Condition Event Nets (CENs)	314
12.2.1	Description	314
12.2.2	Small CEN Examples	315
12.2.3	An RSL Model of Condition Event Nets	319
	Syntax of CENs and A Static Semantics	319
	A Dynamic Semantics	320
12.3	Place Transition Nets (PTNs)	321
12.3.1	Description	321
12.3.2	Small PTN Examples	321
12.3.3	An RSL Model of Place Transition Nets	322
	Syntax of PTPNs and A Static Semantics	322
	A Dynamic Semantics	324
12.3.4	Railway Domain Petri Net Examples	326
	Route Descriptions	326
	Interlocking Tables	327

	Petri Net for Units .....	328
	Petri Net for Switches .....	328
	Petri Net for Signals.....	329
	Petri Net for Routes .....	329
	Construction of Petri Net for Interlocking Tables.....	329
	Summary .....	330
12.4	Coloured Petri Nets (CPNs) .....	331
12.4.1	Description.....	331
12.4.2	A CPN Example.....	332
12.4.3	An RSL Model of Coloured Petri Nets.....	334
	Syntax of CPNs and A Static Semantics .....	334
	Dynamic Semantics of Coloured Petri Nets .....	337
12.4.4	Timed Coloured Petri Nets.....	339
12.5	CEN Example: Work Flow System .....	340
12.5.1	Project Planning.....	340
	Project Plans.....	340
	Project Plan Construction .....	343
12.5.2	Project Activities .....	344
	Project Flow of Control: “Waves” and Traces.....	344
12.5.3	Project Generation .....	351
	Process Generation .....	351
	Channel Allocation .....	352
12.6	CPN and RSL Examples: Superscalar Processor .....	354
12.6.1	Description.....	354
12.6.2	Coloured Petri Net Model .....	355
12.6.3	RSL Model: Superscalar Processor .....	360
12.7	Discussion.....	368
12.8	Bibliographical Notes .....	369
12.9	Exercises .....	370
<b>13</b>	<b>MESSAGE AND LIVE SEQUENCE CHARTS .....</b>	<b>371</b>
	Christian Krog Madsen is chief author of this chapter	
13.1	Message Sequence Charts .....	372
13.1.1	The Issues .....	372
13.1.2	Basic MSCs (BMSCs) .....	372
	Informal Presentation .....	372
	An Example BMSC .....	376
	An RSL Model of BMSC Syntax .....	377
13.1.3	High Level MSCs (HMSCs) .....	379
	An Informal Presentation .....	379
	An Example HMSC .....	380
13.1.4	An RSL Model of HMSC Syntax .....	380
13.1.5	MSCs are HMSCs .....	381
13.1.6	Syntactic Well-formedness of MSCs .....	381
13.1.7	An Example: IEEE 802.11 Wireless Network .....	387
	Description.....	387
	An RSL Model of the IEEE 802.11 Example .....	389
13.1.8	Semantics of Basic Message Sequence Charts .....	395
13.1.9	Semantics of High Level Message Sequence Charts .....	397
13.2	Live Sequence Charts: Informal Presentation.....	397
13.2.1	Live Sequence Chart Syntax.....	397
	Graphical Syntax of Live Sequence Charts .....	397
13.2.2	A Live Sequence Chart Example, I .....	403
13.3	Process Algebra .....	404

13.3.1	The Process Algebra $PA_\epsilon$ . . . . .	405
	Signature . . . . .	405
	Equations . . . . .	406
	Derivability . . . . .	407
	Reduction to Basic Terms . . . . .	408
13.3.2	Semantics of $PA_\epsilon$ . . . . .	412
13.3.3	The Process Algebra $PAC.\epsilon$ . . . . .	415
13.3.4	Semantics for $PAC_\epsilon$ . . . . .	418
13.4	Algebraic Semantics of Live Sequence Charts . . . . .	422
13.4.1	Textual Syntax of Live Sequence Charts . . . . .	422
13.4.2	Semantics of Live Sequence Charts . . . . .	422
13.4.3	The Live Sequence Chart Example, II . . . . .	426
13.5	Relating Message Charts to RSL . . . . .	427
13.5.1	Types of Integration . . . . .	427
13.5.2	An RSL Subset . . . . .	428
	Syntax . . . . .	428
	Operational Semantics with Communication Behaviour . . . . .	429
13.5.3	Relating Live Sequence Charts to RSL . . . . .	430
	Syntactical Restrictions . . . . .	430
	Satisfaction Relation . . . . .	431
13.5.4	Checking Satisfaction . . . . .	437
13.5.5	Tool Support . . . . .	438
13.6	Communicating Transaction Processes (CTP) . . . . .	438
13.6.1	Intuition . . . . .	438
13.6.2	Narration of CTPs . . . . .	439
	CTP Diagrams: . . . . .	439
	CTP Processes . . . . .	440
	CTP Transaction Schemas . . . . .	441
	CTP Transaction Charts . . . . .	441
	Simple CTP Message Sequence Charts . . . . .	441
	Enabled CTP Transaction Charts . . . . .	443
	CTP Transitions . . . . .	443
	Enabled Versus Invoked Schemas and Charts . . . . .	444
	Details of Invocation and Execution . . . . .	444
13.6.3	Formalisation of CTPs . . . . .	445
	Syntax of CTP Programs . . . . .	445
	Types of CTP Interpretation States . . . . .	446
	Interpretation Functions . . . . .	447
13.7	Discussion . . . . .	447
13.7.1	General . . . . .	447
13.7.2	Principles, Techniques and Tools . . . . .	448
13.8	Bibliographical Notes . . . . .	449
13.9	Exercises . . . . .	450
<b>14</b>	<b>Statecharts</b> . . . . .	<b>455</b>
	Christian Krog Madsen is chief author of this chapter	
14.1	Introduction . . . . .	455
14.2	A Narrative Description of Statecharts . . . . .	456
14.3	An RSL Model of the Syntax of Statecharts . . . . .	461
14.4	Examples . . . . .	464
14.4.1	Railway Line Automatic Blocking . . . . .	464
	Narrative . . . . .	465
	Statecharts . . . . .	466
14.4.2	Railway Line Direction Agreement System . . . . .	468

	Narrative .....	468
	Internal Behaviour of LDAS (Statechart).....	469
	Internal Behaviour of Station A (Statechart) .....	470
	Internal behaviour of Station B (Statechart).....	470
	External Behaviour (Live Sequence Chart) .....	471
14.4.3	Wireless Rain Gauge .....	471
	Description.....	472
	Statechart Model .....	473
	RSL Model .....	473
14.5	A Process Algebra for Statecharts .....	477
14.5.1	SPL: The Statechart Process Language .....	478
14.5.2	Semantics of SPL .....	479
14.5.3	Equivalence for SPL Terms .....	479
14.6	Semantics of Statecharts .....	481
14.6.1	An SPL Semantics for Statecharts.....	481
14.6.2	Statechart Example .....	483
14.7	Relating Statecharts to RSL .....	485
14.7.1	Syntactical Restrictions .....	485
14.7.2	Satisfaction Relation .....	485
14.7.3	Checking Satisfaction.....	486
14.7.4	Tool Support .....	486
14.8	Discussion.....	487
14.8.1	General.....	487
14.8.2	Principles, Techniques and Tools .....	487
14.9	Bibliographical Notes .....	488
14.10	Exercises .....	488
<b>15</b>	<b>QUANTITATIVE MODELS OF TIME</b> .....	<b>495</b>
15.1	The Issues .....	495
15.1.1	Soft Temporalities .....	495
15.1.2	Hard Temporalities .....	496
15.1.3	Soft and Hard Real-Time .....	496
15.1.4	Examples — "Ye Olde Way"!.....	496
15.1.5	Structure of This Chapter .....	498
15.2	Temporal Logic .....	498
15.2.1	The Issues .....	499
15.2.2	A Philosophical Linguistics Background .....	499
15.2.3	Interval Temporal Logic, ITL .....	500
	Intervals and Subintervals: $[c, d] \sqsubseteq [b, e]$ .....	500
	Length of an Interval: $[\cdot]$ .....	501
	The "Sometime" Modality: $\diamond$ .....	501
	The "Always" Modality: $\square$ .....	502
	The Right Neighbourhood Expanding Modalities .....	502
	The Left Neighbourhood Expanding Modalities .....	503
	The "Chop" Modality .....	504
	Defining $\diamond$ in Terms of Chop .....	504
15.2.4	The Classic Temporal Operators: $\diamond, \square$ .....	505
15.3	The Duration Calculus .....	506
15.3.1	Examples, Part I .....	506
15.3.2	Some Basic Notions .....	507
	Boolean States, State Assertions and Characteristic Functions .....	507
	State Durations.....	508
15.3.3	Examples, Part II .....	510
	Problem Description .....	510

	Requirements: <b>L_Req</b> .....	512
15.3.4	The Syntax .....	514
	Simple Expressions .....	514
	State Expressions and Assertions .....	514
	Durations and Duration Terms .....	515
	Duration Formulas .....	515
	Common Duration Formula Abbreviations .....	515
15.3.5	The Informal Semantics .....	516
15.3.6	Examples, Part III .....	517
	The Elevator cum Lift Example: Design .....	517
	The Railway Road-Rail Level Crossing Examples .....	519
	Problem Description .....	520
	Formalisation .....	521
	Problem Description .....	524
	Some Observations .....	527
15.3.7	Transitions and Events .....	528
15.3.8	Discussion: From Domains to Designs .....	533
15.4	TRSL: RSL with Timing .....	533
15.4.1	TRSL Design Criteria .....	533
	The First TRSL Design Decision .....	534
	The Second TRSL Design Decision .....	534
15.4.2	The TRSL Language .....	536
	Syntax .....	536
	Semantics .....	536
15.4.3	Another Gas Burner Example .....	537
15.4.4	Discussion .....	540
15.5	RSL with Timing and Durations .....	541
15.5.1	Review of TRSL .....	541
15.5.2	TRSL and Duration Calculus .....	542
	Problem Description .....	543
	DC Requirements .....	543
	TRSL Specification .....	544
	Satisfaction Relation .....	544
15.6	Discussion .....	545
15.6.1	General .....	545
15.6.2	Principles, Techniques and Tools .....	545
15.7	Bibliographical Notes .....	546
15.8	Exercises .....	546

---

## Part VII LANGUAGE DEFINITIONS

---

<b>16</b>	<b>SAL: Simple Applicative Language</b> .....	<b>551</b>
16.1	A Caveat .....	552
16.2	The SAL Syntax .....	552
	16.2.1 Informal Exposition of SAL Syntax .....	552
	16.2.2 Formal Exposition of SAL Syntax .....	553
	16.2.3 Comments .....	554
16.3	A Denotational Semantics .....	554
	16.3.1 An Informal Semantics .....	554
	16.3.2 A Formal Semantics .....	555
	Semantic Types .....	555
	Operator Meanings .....	555
	Semantic Functions .....	556

16.3.3	Review of SAL Semantics, 1	557
16.3.4	Two Asides	558
	Of Things to Come!	558
	The Most Recent Error	558
16.4	A First-Order Applicative Semantics	560
16.4.1	Syntactic Types	560
16.4.2	Semantic Types	560
16.4.3	Abstraction Functions	561
16.4.4	Auxiliary Functions	562
16.4.5	Semantic Functions	563
16.4.6	Review	566
16.4.7	Review of SAL Semantics, 2	566
16.5	An Abstract, Imperative Stack Semantics	567
16.5.1	Design Decisions — Informal Motivation	567
16.5.2	Semantics Style Observations	568
16.5.3	Syntactic Types	568
16.5.4	Semantic Types	569
16.5.5	Abstraction Functions	569
16.5.6	Run-Time Functions	569
16.5.7	Semantic Functions	570
	Two Invariants	570
	[0] Interpret Programs	571
	[1] Interpret Constant Expressions	571
	[2] Interpret Variable Expressions	572
	[3] Interpret Prefix Expressions	572
	[4] Interpret Infix Expressions	573
	[5] Interpret Conditional Expressions	573
	[6] Interpret Lambda Expressions	574
	[7] Interpret Simple Let Expressions	574
	[8] Interpret Recursive Let Expressions	575
	[9] Interpret Function Application Expressions	575
16.5.8	Review of SAL Semantics, 3	576
16.6	A Macro-Expansion Semantics	576
16.6.1	Analysis of Stack Semantics	577
	Informal Design Description	580
	[1] The Run-Time State	581
	[2] Macro-expansion	581
	[3] Realisation of CLOSures	582
	[4] The Compile State — Compile-Time Specification	582
	[5] The Compile State — The Dictionary	583
	Execution	583
16.6.2	Syntactic Types	584
16.6.3	Compile-Time Types	584
16.6.4	Run-Time Semantic Types	584
16.6.5	Run-Time State	584
16.6.6	Run-Time Stack Operations	585
16.6.7	Run-Time Stack Search for Variable Values	585
16.6.8	Macro-Expansion Functions	586
	[0] Program Macro-Expansion	586
	[1] Constant Expression Macro-Expansion	587
	[2] Variable Expression Macro-Expansion	587
	[3] Prefix Expression Macro-Expansion	588
	[4] Infix Expression Macro-Expansion	588
	[5] Conditional Expression Macro-Expansion	588

	[6] Lambda-Expression Macro-Expansion . . . . .	589
	[7] Simple Let Expression Macro-Expansion . . . . .	590
	Block Macro-Expansion . . . . .	590
	[8] Recursive Function/Let Expression Macro-Expansion . . . . .	592
	[9] Function Application Expression Macro-Expansion . . . . .	593
16.6.9	Review of SAL Semantics, 4. . . . .	593
16.7	ASM: An Assembler Language . . . . .	594
16.7.1	Semantic Types . . . . .	594
16.7.2	The Computer State . . . . .	595
16.7.3	The Address Concept . . . . .	595
16.7.4	Machine Instructions . . . . .	596
16.7.5	Machine Semantics . . . . .	598
	Interpreting Code . . . . .	598
	Find Label Index . . . . .	598
	The Store Immediate and Store Instructions . . . . .	599
	The Load Immediate and Load Instructions . . . . .	599
	The Apply Function Instructions . . . . .	599
	The Unconditional Jump Instruction . . . . .	601
	The Conditional Jump Instruction . . . . .	601
	The Register Move and Adjust Instructions . . . . .	602
	The Pack and Unpack Instructions . . . . .	602
	The Output Instruction . . . . .	602
	The Finish Instruction . . . . .	603
16.7.6	Review of ASM . . . . .	603
16.8	A Compiling Algorithm . . . . .	603
16.8.1	Syntactic Types . . . . .	604
16.8.2	Compile Time Types and State . . . . .	604
16.8.3	Compile-Time Dynamic Function . . . . .	604
16.8.4	Compile-Time Static Function . . . . .	604
16.8.5	Run-Time Constant Values . . . . .	605
16.8.6	Compilation Functions . . . . .	605
	[0] Program Compilation . . . . .	605
	[1] Constant Expression Compilation . . . . .	606
	[2] Variable Expression Compilation . . . . .	606
	[3] Prefix Expression Compilation . . . . .	608
	[4] Infix Expression Compilation . . . . .	608
	[5] Conditional Expression Compilation . . . . .	608
	[6] Lambda-Expression Compilation . . . . .	609
	[7] Simple Let Expression Compilation . . . . .	610
	[*] Block Expression . . . . .	610
	[8] Recursive Function/Let Expression Compilation . . . . .	611
	[9] Function Application Expression Compilation . . . . .	612
16.8.7	Review of Compiling Algorithm . . . . .	613
16.9	An Attribute Grammar Semantics . . . . .	613
16.9.1	Abstract Syntactic Types . . . . .	614
16.9.2	SAL BNF Grammar, 1 . . . . .	614
16.9.3	Node Attributes . . . . .	615
16.9.4	Constants . . . . .	615
16.9.5	Some Typographical Distinctions . . . . .	615
16.9.6	Compilation Functions . . . . .	616
	[0] Program Compilation . . . . .	616
	[1] Constant Expression Compilation . . . . .	616
	[2] Variable Expression Compilation . . . . .	617
	[3] Prefix Expression Compilation . . . . .	617

	[4] Infix Expression Compilation .....	618
	[5] Conditional Expression Compilation .....	618
	[6] Lambda-Expression Compilation .....	619
	[7] Simple Let Expression Compilation .....	619
	[*] Block Expression Compilation .....	620
	[8] Recursive Function/Let Expression Compilation.....	620
	[9] Function Application Expression Compilation.....	621
16.9.7	Review of Attribute Semantics, 1 .....	622
16.10	Another Attribute Grammar Semantics .....	623
16.10.1	Abstract Syntactic Types .....	623
16.10.2	SAL BNF Grammar, 2 .....	623
16.10.3	Global Variables .....	624
16.10.4	Constants .....	624
16.10.5	Node Attributes .....	624
16.10.6	Compilation Functions.....	625
	[0] Program Compilation .....	625
	[1] Constant Expression Compilation .....	625
	[2] Variable Expression Compilation .....	625
	[3] Prefix Expression Compilation .....	626
	[4] Infix Expression Compilation .....	626
	[5] Conditional Expression Compilation .....	627
	[6] Lambda-Expression Compilation .....	627
	[7] Simple Let Expression Compilation .....	628
	[●] Block Compilation .....	628
	[8] Recursive Function / Let Expression Compilation .....	628
	[9] Function Application Expression Compilation.....	629
16.10.7	Review of Attribute Semantics, 2 .....	630
16.11	Discussion.....	630
16.11.1	General.....	630
16.11.2	Principles, Techniques and Tools .....	631
16.12	Review and Bibliographical Notes .....	631
16.13	Exercises .....	633
<b>17</b>	<b>SIL: Simple Imperative Language.....</b>	<b>635</b>
17.1	The Background .....	635
17.2	Syntactic Types .....	636
17.2.1	Concrete, Schematic Syntax .....	636
17.2.2	Abstract Syntax .....	636
17.3	Imperative Denotational Semantics .....	637
17.3.1	Semantic Types.....	637
17.3.2	Auxiliary Semantic Functions.....	638
17.3.3	Semantic Functions .....	638
	Procedure Denotations .....	638
	Statement and Expression Function .....	639
17.4	Macro-Expansion Semantics .....	639
17.4.1	Syntactic Types .....	640
17.4.2	Compile-Time Semantic Types .....	640
17.4.3	Run-Time Semantic Types .....	640
	“Snapshot” of a Run-time State .....	640
	Semantic Types.....	641
17.4.4	Run-Time State Declaration and Initialisation .....	641
17.4.5	Abstraction Functions .....	642
17.4.6	Macros .....	643
17.5	Discussion.....	644

17.5.1	General	644
17.5.2	Principles, Techniques and Tools	644
17.6	Bibliographical Notes	645
17.7	Exercises	645
<b>18</b>	<b>SMIL: Simple Modular, Imperative Language</b>	<b>647</b>
18.1	Syntactic Types	647
18.2	A Denotational Semantics	648
18.2.1	Semantic Types	648
18.2.2	Auxiliary Functions	649
	Static Functions	649
	Temporal Functions	649
18.2.3	Semantic Functions	649
	Semantic Function Types	650
	Semantic Function Definitions	650
18.3	A Macro-Expansion Semantics	651
18.3.1	Run-Time Semantic Types	651
18.3.2	Compile/Run-time Semantic Types	652
18.3.3	Compile-Time Semantic Types	653
18.3.4	Semantic Functions	653
18.4	Discussion	655
18.4.1	General	655
18.4.2	Principles, Techniques and Tools	655
18.5	Bibliographical Notes	656
18.6	Exercises	656
<b>19</b>	<b>SPIL: Simple Parallel, Imperative Language</b>	<b>657</b>
19.1	The Problem	657
19.2	Syntax	658
19.2.1	Informal Syntax	658
	Process Expressions	658
	Expressions and Statements	659
	System Processes	659
19.2.2	Formal Syntax	660
19.3	Process Concepts and Semantic Types	660
19.3.1	Syntactic Notions	661
	Textual	661
19.3.2	Machines and Interpreters	662
19.3.3	Semantic Notions and Types	662
	Actions	662
19.4	Process-Oriented Semantic Types	664
19.4.1	Unique Process Identifiers $\pi : \Pi$	664
19.4.2	The Heap $\xi : \Xi$	665
19.4.3	Input/Output Channel Bindings	666
19.4.4	Environments $\rho : \text{ENV}$	666
19.4.5	State Composition $\Psi, \Gamma, \Xi, \Sigma, \Omega$	667
	The Global State	668
19.5	Initial and Auxiliary Semantic Functions	668
19.5.1	Start Function	668
19.5.2	System Function	669
19.5.3	Bind and Allocate Functions	669
19.5.4	Free and Bound Functions	670
19.5.5	Distribute Function	670
19.5.6	Transition Loop	671

19.6	Semantic Functions .....	671
19.6.1	The <i>Next-State</i> Transition Function .....	671
19.6.2	The Assignment Statement .....	672
19.6.3	The <i>case</i> Statement .....	672
19.6.4	The <i>while</i> Loop .....	673
19.6.5	The <i>repeat until</i> Loop .....	673
19.6.6	Simple Input/Output Processes .....	674
19.6.7	The Parallel Process Command,    .....	675
19.6.8	The <i>stop</i> Process Technicality .....	675
19.6.9	The Process <i>call</i> Command .....	676
19.6.10	Internal Nondeterministic Processes .....	676
19.6.11	External Nondeterministic Processes .....	676
19.6.12	Nondeterministic Input/Output Processes .....	677
19.6.13	The Embedded System Process Command .....	678
19.6.14	A <i>finish</i> Process Technicality .....	678
19.7	Discussion .....	678
19.7.1	General .....	678
19.7.2	Principles, Techniques and Tools .....	679
19.8	Bibliographical Notes .....	679
19.9	Exercises .....	680

---

## Part VIII CLOSING

---

<b>20</b>	<b>Closing</b> .....	685
20.1	A Summary .....	685
20.2	Conclusion: Volumes 1 and 2 .....	686
20.3	Preview of Volume 3 .....	686
20.4	“UML”-ising Formal Techniques .....	688

---

## Part IX APPENDIXES

---

<b>A</b>	<b>Naming Convention</b> .....	693
<b>B</b>	<b>Indexes</b> .....	697
B.1	Symbols Index .....	698
B.1.1	General .....	698
B.1.2	Duration Calculus .....	698
B.1.3	Other Temporal Logics .....	698
B.1.4	Live Sequence Charts .....	699
B.1.5	Message Sequence Charts .....	699
B.1.6	Petri Nets .....	699
B.1.7	RSL .....	699
B.1.8	Statecharts .....	699
B.1.9	Time/Space .....	699
B.1.10	Abbreviations .....	700
B.2	Concepts Index .....	701
B.3	Characteriations and Definitions Index .....	720
B.4	Authors Index .....	722
	<b>References</b> .....	727