

Modelling and Analysis of Communicating Systems

Jan Friso Groote and Mohammad Reza Mousavi

Voor papa en mem, en Paula
– Jan Friso

For my mother, my father, and Mona
– Mohammad

Preface

Robin Milner observed in 1973 that the primary task of computers is to interact with their environment. But the theory of programs and programming at that time ignored this [131, 132]. To remedy this, he set out working on a theory of interaction, leading to his seminal books [133, 135] in which he developed CCS, the Calculus of Communicating Systems. At the same time two other main process calculi were developed, namely ACP (Algebra of Communicating Processes, [25]) and CSP (Communicating Sequential Processes, [100, 101]).

Interesting as they were, these process algebras were too bare to be used for the description of actual systems, mainly because they lacked a proper integration of data. In order to solve this, process-algebraic specification languages have been designed (most notably LOTOS [106] and PSF [127]) which contained both data and processes. A problem with these languages was that they were too complex to act as a basic carrier for the development of behavioural analysis techniques.

We designed an intermediate language, namely mCRL2 (and its direct predecessor μ CRL [84, 75]) as a stripped-down process specification language or an extended process algebra. It contains exactly those ingredients needed to describe the behaviour of systems precisely in all its aspects, and its (relative) simplicity allows to concentrate on proof and analysis techniques for process behaviour.

Throughout the years many of these analysis techniques have been developed. These results include the Recursive Specification Principle, invariants, τ -confluence, cones and foci, the modal mu-calculus with time and data, abstract interpretation and coordinate transformations, parameterised boolean equation systems, and proof by patterns, to name a few. These results, when combined together, constitute a mathematical framework suitable to launch an ‘attack’ on several phenomena in the realm of process behaviour that are not properly understood. They also form an effective framework to formulate and prove the correctness of complex and intricate protocols.

Up till now, all these results were lingering around in the literature. We combined them in this book, added exercises and examples to make the developed material suitable for self study and for teaching. The book has been used in the past years as the basis for several graduate-level courses. These include the course on “System Validation” in the Embedded Systems masters program at Eindhoven University of Technology and Delft University of Technology in the Netherlands.

Acknowledgements

The first version of this book appeared as a handbook chapter [86]. This chapter formed the basis of a reader [60] used for courses at several universities (published as [59]). These earlier publications were based on the modelling language μ CRL (micro Common Representation Language, [84, 75]) essentially developed in 1991. In 2003 we decided that it was time for a successor to increase the usability of the μ CRL, and we decided to baptise its successor mCRL2. The essential difference is that mCRL2 has richer datatypes, including standard data types and functions, contrary to μ CRL which contained only a mechanism to define equational datatypes. This book is solely based

on mCRL2, and substantially extends [59].

The development of mCRL2 builds upon the development work on process algebras between 1970 and 1990. Especially the work on CCS (Calculus of Communicating Systems) by Robin Milner [133] and ACP (Algebra of Communicating Processes) by Jan Bergstra, Jan Willem Klop, Jos Baeten, Rob van Glabbeek and Frits Vaandrager [25, 18] forms an important basis. An essential step was the EC SPECS project, where a megalomane *Common Representation Language* had to be developed to encompass all behavioural description languages that existed at that time (LOTOS, CHILL, SDL, PSF) and that still had to be developed. As a reaction a micro Common Representation Language (μ CRL) had been developed in which Alban Ponse was instrumental. Bert Lissner was the main figure behind the maintenance and development of the tools to support μ CRL.

The following people have contributed to the development of mCRL2, its tools and its theory: Sjoerd Cranen, Tom Haenen, Frank van Ham, Jeroen Keiren, Aad Mathijssen, Bas Ploeger, Jaco van de Pol, Hannes Pretorius, Frank Stappers, Carst Tankink, Yaroslav Usenko, Muck van Weerdenburg, Wieger Wesselink, Tim Willemse, and Jeroen van der Wulp.

This book has been used as a reader for several courses among which are the courses ‘Requirements, Analysis, Design and Verification’ and ‘System Validation’ at Eindhoven University of Technology and Delft University of Technology. Many thanks go to Sjoerd Cranen, Veronica Gaspes, Jeroen Keiren, Michel Reniers and Erik de Vink for their careful proofreading. Valuable feedback also came from Michael Adriaansen, Muhammad Atif, Timur Bagautdinov, Ruud Bauhaus, Harsh Beohar, Debjyoti Bera, Dwight Berendse, Anton Bilos, Michiel Bosveld, Gert-Jan van den Braak, Christoph Brandt, Bram Cappers, Mehmet Çubuk, Edin Dudojević, Michiel Fortuin, Joe Ganett, Herman Geuvers, Sven Goossens, Christiaan Hartman, Albert Hofkamp, Hossein Hojjat, Albert Hofkamp, Tom Hubregtsen, Bas Kloet, Diana Koenraadt, Geert Kwintenberg, Koen van Langen, Tony Larsson, Mattias Lee, Josh Mengerink, Paul Mulders, Gerardo Ochoa, Chidi Okwudire, Mathijs Opdam, Mahboobeh Parsapoor, Eva Ploum, Sander de Putter, André van Renssen, Marcel Roeloffzen, Koos Rooda, Anson van Rooij, Vikram Saralaya, Frank Stappers, Carst Tankink, Sander Verdonschot, Twan Vermeulen, Maks Verver, Amrita Vikas Sinha, Migiel de Vos, Tim Willemse, Jia Yan, Umar Waqas and many others.

Contents

I	Modelling	11
1	Introduction	13
1.1	Motivation	13
1.2	The mCRL2 approach	14
1.3	An overview of the book	15
1.4	Audience and suggested method of reading	16
2	Actions, behaviour, equivalence and abstraction	17
2.1	Actions	17
2.2	Labelled transition systems	18
2.3	Equivalence of behaviours	21
2.3.1	Trace equivalence	21
2.3.2	★Language and completed trace equivalence	23
2.3.3	★Failures equivalence	24
2.3.4	Strong bisimulation equivalence	26
2.3.5	The Van Glabbeek linear time – branching time spectrum	29
2.4	Behavioural abstraction	32
2.4.1	The internal action τ	32
2.4.2	Weak trace equivalence	33
2.4.3	(Rooted) Branching bisimulation	33
2.4.4	★(Rooted) Weak bisimulation	38
2.5	Historical notes	40
3	Data types	43
3.1	Data type definition mechanism	43
3.2	Standard data types	49
3.2.1	Booleans	50
3.2.2	Numbers	51
3.3	Function data types	54
3.4	Structured data types	57
3.5	Lists	59
3.6	Sets and bags	60
3.7	Where expressions and priorities	61
3.8	Historical notes	61

4	Sequential processes	63
4.1	Actions	63
4.2	Multi-actions	64
4.3	Sequential and alternative composition	66
4.4	Deadlock	68
4.5	The conditional operator	69
4.6	The sum operator	70
4.7	Recursive processes	71
4.8	Axioms for the internal action	74
4.9	Historical notes	75
5	Parallel processes	77
5.1	The parallel operator	77
5.2	Communication among parallel processes	80
5.3	The allow operator	82
5.4	Blocking and renaming	85
5.5	Hiding internal behaviour	86
5.6	★Alphabet axioms	87
5.7	Historical notes	90
6	The modal μ-calculus	91
6.1	Hennessy-Milner logic	92
6.2	Regular formulas	93
6.3	Fixed point modalities	96
6.4	Modal formulas and labelled transition systems	100
6.5	Modal formulas with data	103
6.6	Equations	105
6.7	Historical notes	106
7	Modelling of system behaviour	111
7.1	Alternating bit protocol	111
7.2	Sliding window protocol	114
7.3	A patient support platform	119
7.4	Historical notes	127
8	Timed process behaviour	129
8.1	Timed actions and time deadlocks	129
8.2	Timed transition systems	131
8.3	Timed process equivalences	133
8.3.1	Timed (strong) bisimulation	133
8.3.2	Timed branching bisimulation	135
8.3.3	Timed trace and timed weak trace equivalence	138
8.4	Timed processes	139
8.5	Modal formulas with time	144
8.6	Historical notes	146

II	Analysis	147
9	Basic manipulation of processes	149
9.1	Derivation rules for equations	149
9.2	Derivation rules for formulas	154
9.3	The sum operator	156
9.4	The sum elimination lemma	157
9.5	Induction for constructor sorts	158
9.6	Recursive specification principle	160
9.7	Koomen’s fair abstraction rule	164
9.8	Parallel expansion	166
9.8.1	Basic parallel expansion	166
9.8.2	Parallel expansion with data: two one-place buffers	167
9.8.3	Parallel expansion with time	170
9.9	Historical notes	172
10	Linear process equations and linearisation	175
10.1	Linear process equations	175
10.1.1	General linear process equations	175
10.1.2	Clustered linear process equations	178
10.2	Linearisation	178
10.2.1	Linearisation of sequential processes	179
10.2.2	Parallelisation of linear processes	185
10.2.3	Linearisation of n parallel processes	187
10.3	Proof rules for linear processes	189
10.3.1	τ -convergence	189
10.3.2	The Convergent Linear Recursive Specification Principle (CL-RSP)	191
10.3.3	CL-RSP with invariants	192
10.4	Historical notes	195
11	Confluence and τ-prioritisation	197
11.1	τ -confluence on labelled transition systems	198
11.2	τ -prioritisation labelled transition systems	199
11.3	Confluence and linear processes	202
11.4	τ -prioritisation for linear processes	204
11.4.1	Using confluence for state space generation	205
11.5	Historical notes	206
12	Cones and foci	207
12.1	Cones and foci	207
12.2	Protocol verification using the cones and foci	212
12.2.1	Two unbounded queues form a queue	212
12.2.2	Milner’s scheduler	213
12.2.3	The alternating bit protocol	214
12.3	Historical notes	218

13 ★Verification of distributed systems	221
13.1 Tree identify protocol	221
13.1.1 The correctness of the tree identify protocol	223
13.2 Sliding window protocol	227
13.2.1 Some rules for modulo calculation	227
13.2.2 Linearisation	227
13.2.3 Getting rid of modulo arithmetic	228
13.2.4 Proving non-modulo SWP equal to a FIFO queue	234
13.2.5 Correctness of the sliding window protocol	237
13.3 Distributed summing protocol	237
13.3.1 A description in mCRL2	238
13.3.2 Linearisation and invariants	240
13.3.3 State mapping, focus points and final lemma	243
13.4 Historical notes	246
14 Verification of modal formulas using PBESs	249
14.1 Boolean equation systems	249
14.1.1 Boolean equation systems and model checking	251
14.1.2 Gaussian elimination	252
14.2 Parameterised boolean equation systems	254
14.3 Translating modal formulas to PBESs	255
14.4 Techniques for solving PBESs	259
14.4.1 Transforming a PBES to a BES	259
14.4.2 Global solving techniques for PBESs	260
14.4.3 Local solving techniques for PBESs	261
14.5 Historical notes	265
III Semantics	267
15 ★Semantics	269
15.1 Semantics of data types	269
15.1.1 Signatures	269
15.1.2 Well-typed data expressions	273
15.1.3 Free variables and substitutions	278
15.1.4 Data specifications	280
15.1.5 Semantics of data types	282
15.2 Semantics of processes	284
15.2.1 Processes, action declarations and process equations	284
15.2.2 Semantical multi-actions	288
15.2.3 Substitution on processes	289
15.2.4 Operational semantics	294
15.3 Validity of modal μ -calculus formulas	296
15.4 Semantics of a PBES	298
15.5 Soundness and completeness	300
15.6 Historical Notes	300

IV	Appendices	303
A	Equational definition of built in data types	305
A.1	Bool	306
A.2	Positive numbers	306
A.3	Natural numbers	308
A.4	Integers	311
A.5	Reals	313
A.6	Lists	315
A.7	Sets	316
A.8	Bags	320
A.9	Function update	323
A.10	Structured sorts	324
B	Plain-text notation	325
B.1	Data types	325
B.1.1	Sorts	325
B.1.2	Functions for any data type	325
B.1.3	Boolean expressions	326
B.1.4	Structured-data expressions	326
B.1.5	Numerical expressions	326
B.1.6	Function expressions	327
B.1.7	List expressions	327
B.1.8	Sets and bags	327
B.2	Processes	328
B.3	Modal logic	328
B.3.1	Action formulas	328
B.3.2	Regular expressions	329
B.3.3	State formulas	329
B.4	(Parameterised) boolean equation systems	329
C	Syntax of the formalisms	331
C.1	Comments	331
C.2	Keywords	331
C.3	Conventions to denote the context-free syntax	331
C.4	Identifiers and numbers	332
C.5	Sort expressions and sort declarations	332
C.6	Constructors and mappings	333
C.7	Equations	333
C.8	Data expressions	333
C.9	Communication and renaming sets	334
C.10	Process expressions	335
C.11	Actions	335
C.12	Process and initial state declaration	336
C.13	Data specification	336
C.14	mCRL2 specification	336

C.15 BES	336
C.16 PBES	337
C.17 Action formulas	337
C.18 Regular formulas	337
C.19 State formulas	338
C.20 Action Rename Specifications	338
D Axioms for processes	339
E Answers to exercises	345
References	361