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 μCRL [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 μCRL (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 μCRL, 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 μCRL 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 (μCRL) had been developed in which Alban Ponse was instrumental. Bert Lisser was the main figure behind the maintenance and development of the tools to support μCRL.

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

II Analysis 147