

EATCS
Monographs on Theoretical Computer Science

Editors: W. Brauer G. Rozenberg A. Salomaa

**Advisory Board: G. Ausiello M. Broy S. Even
J. Hartmanis N. Jones T. Leighton M. Nivat
C. Papadimitriou D. Scott**



Ryszard Janicki Peter E. Lauer

Specification and Analysis of Concurrent Systems

The COSY Approach

With 128 Figures

Springer-Verlag

Berlin Heidelberg New York

London Paris Tokyo

Hong Kong Barcelona

Budapest

Authors

Prof. Dr. Ryszard Janicki

Prof. Dr. Peter E. Lauer

Department of Computer Science and Systems

McMaster University

1280 Main Street West

Hamilton, Ontario L8S 4K1, Canada

Editors

Prof. Dr. Wilfried Brauer

Institut für Informatik, Technische Universität München

Arcisstrasse 21, W-8000 München 2

Prof. Dr. Grzegorz Rozenberg

Institute of Applied Mathematics and Computer Science

University of Leiden, Niels-Bohr-Weg 1, P. O. Box 9512

2300 RA Leiden, The Netherlands

Prof. Dr. Arto Salomaa

The Academy of Finland

Department of Mathematics, University of Turku

SF-20 500 Turku, Finland

This book is the result of the combined efforts of the authors. Their names have been listed in alphabetical order with no implication that one is senior and the other junior.

ISBN-13: 978-3-642-77339-6 e-ISBN-13: 978-3-642-77337-2

DOI: 10.1007/978-3-642-77337-2

Library of Congress Cataloging-in-Publication Data

Janicki, Ryszard. Specification and analysis of concurrent systems: the COSY approach / Ryszard Janicki, Peter E. Lauer. p. cm. - (EATCS monographs on theoretical computer science; v. 26) Includes bibliographical references and index. (U.S.) 1.

Parallel processing (Electronic computers) I. Lauer, Peter E., 1934-. II. Title. III. Series. QA76.58.J36 1992 005.1'2-dc20 92-7430

This work is subject to copyright. All rights are reserved, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilm or in any other way, and storage in data banks. Duplication of this publication or parts thereof is permitted only under the provisions of the German Copyright Law of September 9, 1965, in its current version, and a permission for use must always be obtained from Springer-Verlag. Violations are liable for prosecution under the German Copyright Law.

© Springer-Verlag Berlin Heidelberg 1992

Softcover reprint of the hardcover 1st edition 1992

The use of general descriptive names, registered names, trademarks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

UNIX is a trademark of Bell Laboratories

OCCAM is a trademark of the INMOS Group of Companies

Typesetting: Camera ready by authors

45/3140-5 4 3 2 1 0 - Printed on acid-free paper

Preface

Complex systems of processes which can progress in parallel (or concurrently) and which involve critical periods of activity during which they may interfere with each other in some detrimental manner are becoming increasingly common in practice. Examples of such systems are air, train, or car traffic control, process control (such as a chemical or nuclear plant), communication systems and operating systems, etc. The complexity of the systems and the limitations of human capabilities for making complex decisions have led to attempts to reduce the complexity of the decisions required from the human operator by delegating large amounts of detailed sifting of intermediate information required for making decisions to automatic computers.

Most of the types of system listed above usually require a very high degree of validation before implementation because of the extremely critical nature of the processes being controlled, for example, the various traffic control systems whose malfunctioning could entail not only loss of property but loss of human life as well. The complexity of such validation can be reduced by initially validating the soundness of the intended control strategy independently from implementation details, and only subsequently validating that the implementation of the strategy by a specific computer system preserves the soundness of the strategy.

Concurrent systems are more difficult to specify and analyse than sequential ones, because they require the conceptualization not only of their sequential subsystems, but also of the complex interactions between them. It follows that the programmer's intuition is not enough, being unreliable in cases of high complexity. Here solution of the problem of verification of correct behaviour of the design becomes crucial, and a satisfactory conceptual apparatus for rigorous verification becomes essential.

In such an apparatus, reduction of complexity, by abstracting away from all irrelevant detail specific to some implementation of a concurrent and distributed system strategy, is desirable. But it must be easy to obtain implementations from the abstract specification of a strategy, given enough information about

the synchronization mechanisms of the concrete system on which it is to be implemented. We therefore need a notion of “system” sufficiently abstract to allow analysis only of those aspects of systems arising from their concurrency and yet capable of being readily translated into practical terms.

The COSY (COncurrent SYstem) notation and theory presented in this book constitutes such a conceptual apparatus, and was developed by the authors and others in the last decade as one of a number of mathematical approaches for conceptualizing concurrent and reactive systems.

It is based on the recognition that formal language and automata theory have been very successful in the specification and analysis of sequential and centralized systems, including both software and hardware. Hence, the COSY approach extends the conventional notions of grammar and automaton to collections of “synchronized” grammars and automata, permitting system specification and analysis of “true” concurrency without reduction to non-determinism.

In the book the COSY theory is developed to a great level of detail and this constitutes the first uniform and self-contained presentation of all results about COSY published since 1974, as well as including many new results not published elsewhere.

The COSY theory is used to analyze a sufficient number of typical problems involving concurrency, synchronization and scheduling, to allow the reader to apply the techniques presented to similar problems of interest to him or her. Furthermore, a high-level notation supporting modular development and verification of specifications is presented. A number of computer based analysis systems are also presented briefly and their possible use indicated.

The COSY model is also related to a number of major alternative models of concurrency, particularly Petri Nets, which have since the beginning served as one standard truly concurrent semantics of COSY. Indeed, much of the central part of the chapter on COSY theory is concerned with a detailed study of the relationship between COSY and Petri Nets. Other major alternative models related to COSY are the Communicating Sequential Processes (CSP) of C.A.R. Hoare, and the Calculus of Communicating Systems (CCS) of R. Milner, both of which were developed after COSY had already been introduced.

Finally, although COSY incorporates Path notation, which is due to R. Campbell and A.N. Habermann, it should be understood that COSY Paths are a generalization of the original Paths first introduced by R. Campbell and P.E. Lauer in 1974. Most references to our work in the literature have missed this important point and hence published remarks about COSY paths tend to be wrong since they concern only the original Path notation.

Concerning the structure of the book, it consists of six chapters and eight appendices. The appendices present basic mathematical concepts used throughout the book as well as long, complex and tedious proofs. Chapter 1 provides

detailed motivation, intentions of the book and acknowledgements. Chapter 2 is devoted to the formal theory of (basic) COSY. It contains the majority of formal definitions and results. Its major contents are: two kinds of semantics for COSY, one direct in terms of vector firing sequences; the other indirect in terms of Petri Nets; analysis of adequacy properties, execution strategies, and priorities. Chapter 3 deals with high-level COSY specifications including Process Notation, the Macro Notation and expansion theory. Chapter 4 analyzes 6 different applications of COSY theory. Chapter 5 compares COSY with other approaches including CCS and CSP. The last chapter gives a historical perspective and information about what has been excluded from the book.

R. Janicki
P. E. Lauer

Hamilton, Ontario, Canada, January 1992

Contents

Preface	v
1 What COSY Is and What It Is For	1
1.1 Introduction	1
1.2 Concepts, Objectives and Design Decisions	6
1.3 Structure of Book	10
1.4 Acknowledgements	16
2 Formal Theory of Basic COSY	19
2.1 Basic COSY Syntax and Semantics	19
2.2 VFS Semantics of COSY	23
2.2.1 Simple Properties of Vector Sequences	23
2.2.2 Vector Sequences and Partial Orders	29
2.2.3 Vector Sequences of Path Programs	32
2.2.4 Concurrent Product of Regular Expressions	38
2.3 Petri Net Semantics of COSY	39
2.3.1 Elements of Net Theory	41
2.3.2 VFS semantics of SMD nets	51
2.3.3 Semantics of Labelled Petri Nets	57
2.3.4 Labelled Petri Nets defined by Path Programs	65
2.3.5 Uniquely Named and Labelled Uniquely Named Regular Expressions	71
2.3.6 Labelled Nets and Products of Regular Expressions	75
2.3.7 Labelled Nets and COSY Path Programs	83
2.3.8 Other Constructions.	89
2.4 Adequacy Properties of Path Programs	97
2.4.1 Syntactically Conflict-Free Path Programs	97
2.4.2 Checking Adequacy of SCF-Path Programs	102
2.4.3 Adequacy-Preserving Transformations	114

	Insertions Preserving Adequacy	116
	Deletions Preserving Adequacy	128
	Applications of Adequacy-Preserving Transformations	132
2.4.4	Free-Choice Path Programs	137
2.5	Execution Semantics for COSY	140
2.5.1	Execution Strategies	143
2.5.2	Maximally Concurrent Strategies	156
2.6	Semantics of COSY with Priorities	161
2.6.1	Observations and Multiple Sequences	167
2.6.2	MFS Semantics for Priority COSY	170
2.6.3	Starvation Problem	175
2.6.4	Infinite MFS and Formal Definition of Starvation	180
2.6.5	Simulation of Dynamic Priorities	184
2.6.6	VFS Semantics for Priority COSY	186
2.7	Bibliographic Notes	194
3	High-level COSY Programs and System Design	197
3.1	High-level COSY Syntax and Semantics	197
3.2	The Process Notation	226
3.3	Macro Generators for COSY Notation	231
3.3.1	The Macro Program	235
3.3.2	The Collectivisors	236
3.3.3	The Body-Replicators	241
3.3.4	Macro Paths and Macro Processes	241
3.3.5	Sequence Replicators	242
3.3.6	Left and Right Sequence Replicators	246
3.3.7	The Distributors	247
3.3.8	Expansion of Macro COSY Programs	256
	The Expansion of Replicators	260
	The Expansion of Distributors	262
	Expansion Theorem	268
3.4	The Semantics of Macro COSY Programs	273
3.5	The COSY Environment	274
3.5.1	The Architecture of BCS	275
	The BCS Analysis Mechanism	275
	The BCS Semantic Transformation Mechanism	277
	The BCS Interface	278
3.5.2	The Macro COSY Environment	278
3.6	The COSY System Dossier	280
3.7	Bibliographical Notes	283

4	COSY Applications	285
4.1	Two-Way Channel with Disconnect	286
4.1.1	Bibliographic Notes	292
4.2	The Hyperfast Banker	292
4.2.1	The Bank Director	292
4.2.2	The Vault	294
4.2.3	The Doors	296
4.2.4	The Fair Doormen	297
4.2.5	The Counters	297
4.2.6	The Clerks	299
4.2.7	The Fair Reimbursers	300
4.2.8	The Customers	300
4.2.9	Priority Constraints	301
4.2.10	Final Comment and Bibliographical Notes	303
4.3	Cigarette Smokers	303
	Bibliographic Notes	319
4.4	Merlin-Randell Problem of Train Journeys	320
4.4.1	Movement Graphs and Minimal Critical Patterns	320
4.4.2	COSY Specification of the Synchronization \rightsquigarrow	325
4.4.3	Final Comment and Bibliographic Notes	329
4.5	Transforming Sequential Systems into Concurrent Systems	331
4.5.1	e-Paths and e-Path Programs	335
4.5.2	Concurrent e-Paths and Proper Concurrent e-Paths	337
4.5.3	Synchronized e-Path Programs	340
4.5.4	Final Comment and Bibliographic Notes	344
4.6	Modelling N-Modular Redundancy	345
4.6.1	Replicated Computation in Networks of Computing Nodes	346
4.6.2	Formal Specification	349
4.6.3	Bibliographical Notes	354
5	Comparison of COSY with Other Models	355
5.1	COSY and CCS	355
5.2	COSY and CSP	363
5.3	Full COSY and Petri Nets	371
5.3.1	Uniquely Named Comma&Star-free Programs and Asymmetric Choice Nets	371
5.3.2	Priority Path Programs, Priority Nets and Inhibitor Nets	381
5.4	Vector Sequences and Mazurkiewicz Traces	390
5.5	COSY and Synchronized Behaviours	392

6 Historical Perspective	395
6.1 Introduction	395
6.2 Conceptual and Methodological Framework of COSY Approach .	396
Bibliography	407
Appendices	
A Algebra of Relations	423
B Automata and Formal Language Theory	427
B.1 Strings	427
B.2 Languages	427
B.3 Regular Expressions and Languages	428
B.4 Finite State Automata and Grammars	429
C Elements of Graph Theory	433
D Proofs of Theorems 2.25, 2.26 and 2.28	435
E Proofs of Theorems 2.37 and 2.38	441
F Proof of Theorem 2.29	447
G Proof of Theorem 4.3	451
H Basic COSY Notations and Macro COSY Notation	457
List of Figures	463
List of Algorithms	467
List of Definitions	467
List of Theorems	468
List of Corollaries	470
List of Lemmas	471
Index	473