Design and formal Analysis of Petri Net based Logic Control Algorithms

Entwurf und formale Analyse Petrinetz-basierter Steuerungsalgorithmen

vom

Fachbereich Elektro- und Informationstechnik
der Universität Kaiserslautern
zur Erlangung des akademischen Grades eines

Doktor der Ingenieurwissenschaften (Dr.-Ing.)

genehmigte Dissertation

von

Dipl.-Ing. Georg Frey
geb. in Mosbach

D386

Eingereicht am:	23. Januar 2002
Tag der mündlichen Prüfung:	15. Februar 2002
Dekan des Fachbereichs:	Prof. Dr.-Ing. Ralph Urbansky

Promotionskommission	
Vorsitzender:	Prof. Dr.-Ing. habil. Norbert Wehn
Berichterstattende:	Prof. Dr.-Ing. habil. Lothar Litz
	Prof. Dr. Habil. Jean-Jacques Lesage

Berichte aus der Automatisierungstechnik

Georg Frey

Design and formal Analysis of Petri Net based Logic Control Algorithms

Entwurf und formale Analyse
Petrinetz-basierter Steuerungsalgorithmen

D 386 (Diss. Universität Kaiserslautern)

Shaker Verlag
Aachen 2002

Die Deutsche Bibliothek - CIP-Einheitsaufnahme

Frey, Georg:
Design and formal Analysis of Petri Net based Logic Control Algorithms : Entwurf und formale Analyse Petrinetz-basierter Steuerungsalgorithmen / Georg Frey. Aachen : Shaker, 2002
(Berichte aus der Automatisierungstechnik)
Zugl.: Kaiserslautern, Univ., Diss., 2002
ISBN 3-8322-0043-6

Printed in Germany.

ISBN 3-8322-0043-6
ISSN 0945-4659

Shaker Verlag GmbH • P.O. BOX 1290 • D-52013 Aachen
Phone: 0049/2407/9596-0 • Telefax: 0049/2407/9596-9
Internet: www.shaker.de • eMail: info@shaker.de

Acknowledgements

The work that has resulted in this thesis could not be performed without the support from several people.

First of all, I would like to thank, Prof. Dr.-Ing. habil. Lothar Litz for the opportunity to be a member of his research group at the Institute of Automatic Control at the University of Kaiserslautern, and for all encouragement and guidance he has given to me. Prof. Litz not only knows Logic Control, but also how to provide a working atmosphere of intellectual freedom and fruitful co-operation and to work in his team is certainly a privilege.

Being a member of a group like the Institute of Automatic Control in Kaiserslautern has given me memories I will never forget: Joint excursions and festivities, interesting discussions on a variety of topics (not necessarily related to our research), and much more. Working in an inspiring and warm atmosphere like this makes this group special: Many thanks to all of you.

I would like to thank the advisors of this thesis, Prof. Dr.-Ing. habil. Lothar Litz and Prof. Dr. Habil. Jean-Jacques Lesage from the Ecole Normale Supérieure de Cachan (France) for the interest they took in my work and the time and effort they invested in reviewing this thesis. I would also like to thank Prof. Dr.-Ing. habil. Norbert Wehn for acting as Chair of the Evaluation Committee.

This work was supported financially by the „Deutsche Forschungsgemeinschaft“ (German Research Council, DFG) and the „Stiftung Rheinland-Pfalz für Innovation“ (Innovation Foundation of Rhineland-Palatinate).

I would like to thank all the people involved in the successful implementation of our SIPN-Editor at infoteam GmbH. Especially Karl-Heinz John for giving me the opportunity to see a part of the concepts presented in this thesis entering the area of practical applications.

To Dr. Mark Minas at the University of Erlangen-Nürnberg: Thank you Mark for the discussions on JAVA and the implementation of the SIPN-Editor.

Thanks to Prof. Dr.-Ing. habil. Jörg Raisch at the MPI in Magdeburg for countless inspiring discussions about this work and other even more important topics.

To Stéphane Klein for the help with the French references and for proof reading of this thesis and to Sandra Zilles and Anja Wiesen for proof reading essential parts of the manuscript: Without you there would be considerably more errors in this work. The remaining flaws are all my fault.

To my parents: Thank you for your love and support despite the fact that I am often not capable of explaining to you what I am actually doing. I hope to improve on this matter.

Finally, I give my greatest gratitude to my wife, Nora, for her love, support, patience, and encouragement: This thesis is to you and because of you. I love you so much.

Kaiserslautern, January 2002 Georg Frey

Abstract

The development of logic control algorithms lies outside the realm of classical continuous Control Theory with its strong mathematical foundations and purely formal design approaches. Logic controller development is closer to software development, in the sense that a special algorithm has to be developed for every new problem. However, Computer Science offers a variety of formal methods that help to avoid errors in the software development process and allow checking and evaluating the resulting algorithms. In this thesis, concepts from Discrete Event Control Theory and Software Engineering are combined to a formal development approach for logic controllers. Based on Petri Nets the complete controller development process from an informal specification to the final implementation on a programmable logic controller is discussed. This process includes the steps of design, verification, validation, evaluation (measurement of quality), and implementation. Special emphasis is put on the evaluation step that is new to logic controller development.

Der Entwurf von Steuerungsalgorithmen liegt außerhalb der klassischen, kontinuierlichen Regelungstheorie mit ihrem gesicherten mathematischen Fundament und ihren strikt formalen Entwurfsmethoden. Der Steuerungsentwurf ist eher mit der Softwareentwicklung verwandt, da für jedes neue Steuerungsproblem ein spezifischer Algorithmus entwickelt werden muss. Die Informatik bietet jedoch eine Reihe formaler Methoden, die helfen, Fehler in diesem Softwareentwicklungsprozess zu vermeiden und die resultierenden Algorithmen zu prüfen und zu bewerten. Im vorliegenden Band werden Konzepte aus den Bereichen der ereignisdiskreten Systemtheorie und des Software-Engineering kombiniert. Basierend auf Petrinetzen wird der vollständige Steuerungsentwicklungsprozess ausgehend von der informellen Spezifikation über Entwurf, Verifikation, Validierung und Bewertung (Messung der Qualität) bis zur abschließenden Implementierung auf einer speicherprogrammierbaren Steuerung betrachtet. Besonderes Gewicht wird dabei dem im Steuerungsbereich noch neuen Bereich der Bewertung beigemessen.

Contents

1. Introduction **1**

1.1 Motivation 1

1.2 Organization of the Thesis 1

1.3 New and Important Approaches 2

2. Development of Logic Controllers **5**

2.1 Introductory Overview 5

2.2 The Control Engineering Point of View 5

2.2.1 The Standard Approach 5

2.2.2 Application of Formal Methods 8

2.2.3 Verification and Validation 10

2.2.4 Summary 13

2.3 The Software Engineering Point of View 14

2.3.1 Software Quality in General 14

2.3.2 Software Quality in Logic Controller Development 16

2.3.3 Summary 17

2.4 The Combined Point of View 18

3. Theory of Signal Interpreted Petri Nets (SIPN) **19**

3.1 SIPN as a new PLC Programming Language 19

3.2 Definition and Meaning of SIPN 22

3.2.1 Basic Definitions and Notations 22

3.2.2 SIPN Definition 25

1.1.3 Graphical Representation 29

1.1.4 Logic Control Semantics 30

1.1.5 Principles of SIPN 30

1.3 SIPN Analysis 31

1.3.1 Reachability in SIPN 31

1.1.2 Dynamic Synchronization (DS) 36

1.1.3 Basic Petri Net Properties and their Adaptation to SIPN 37

1.1.4 Analysis based on the Reachability Graph of the underlying Petri Net 40

1.1.5 Additional SIPN Properties 42

1.1.6 Analysis based on the Reachability Graph of the SIPN 42

1.1.7 A Note on Algebraic Analysis 43

1.4 Hierarchical SIPN 43

1.4.1 The Concept of Hierarchy 43

1.4.2 Formal Definition of Hierarchical SIPN 45

1.1.3 Analysis of Hierarchical SIPN by Unfolding 47

1.1.4 Analysis of Hierarchical SIPN by Extension 48

3.5 Relation to other Formalisms 51

3.5.1 Relation to other Petri Net Models 51

3.5.2 Relation to Grafcet according to IEC 60848 53

3.5.3 Relation to SFC according to IEC 61131 55

3.6 Discussion 57

4. Verification **59**
4.1 The Verification Process ... 59
4.2 Informal Specification of Standard Functional Properties ... 60
4.3 Formalization: Criteria for Formal Correctness ... 61
4.4 Summary ... 63
5. Evaluation **65**
5.1 The Evaluation Process ... 65
5.2 Informal Specification of non-functional Properties ... 66
5.3 Formalization: Transparency Metrics ... 66
5.3.1 Overview and Introductory Example ... 66
5.3.2 Documentation (P1) ... 68
5.3.3 Graphical Representation (P2) ... 68
5.3.4 Redundant Information (P3) ... 69
5.3.5 Visible Dynamics (P4) ... 71
5.3.6 Complexity (P5) ... 72
5.3.7 Combination to one Transparency Metric ... 76
5.4 Conclusions ... 79
6. Experimental Validation of the Transparency Metrics **81**
6.1 Design of Software Engineering Experiments ... 81
6.2 Definition Phase ... 82
6.3 Planning Phase ... 83
6.3.1 Formulation of the Aim of the Experiment ... 83
6.3.2 Assignment of Test Candidates to Groups ... 85
6.3.3 Assignment of Projects to Groups ... 87
6.4 Operation Phase ... 88
6.4.1 Aspects of an Experiment's Operation ... 88
6.4.2 Identification and Treatment of Outliers ... 89
6.4.3 Preliminary Analysis: Correlation Analysis ... 90
6.4.4 Primary Analysis: Hypotheses Testing ... 93
6.4.4.1 Selection of Tests ... 93
6.4.4.2 t-Test ... 95
6.4.4.3 Sign-Test ... 95
6.4.5 Results of the Analysis ... 97
6.5 Interpretation ... 97
6.5.1 The Interpretation Phase ... 97
6.5.2 Scalability ... 98
6.5.3 Reproducibility ... 98
6.5.4 Validity of Experimental Results ... 99
6.5.4.1 Types of Validity ... 99
6.5.4.2 Construct Validity ... 99
6.5.4.3 Internal Validity ... 100
6.5.4.4 External Validity ... 101
6.6 Summary ... 102

7. Implementation **103**
7.1 Introductory Overview ... 103
7.2 Transformation of Net Elements ... 103
7.3 Concurrency ... 104
7.4 Iteration ... 105
7.4.1 Implementation of Iteration in the PLC Code ... 105
7.4.2 Simulative Determination of a Transition Sequence ... 105
7.4.3 Analytical Determination of a Transition Sequence ... 106
7.4.4 Comparison of the presented Iteration Methods ... 108
7.5 Implementation of Hierarchical SIPN using the IEC 61131 POU-Concept ... 109
7.6 Summary ... 110
8. Validation **111**
8.1 Introductory Overview ... 111
8.2 Compressed Air Accumulator Example ... 112
8.3 Validation based on SIPN ... 113
8.4 Validation using the Generated Code ... 114
8.5 Comparison and Concluding Remarks on Validation ... 118
9. Summary **119**
9.1 Summary in English ... 119
9.2 Kurzfassung in deutscher Sprache ... 120
10. Bibliography and Indices **125**
10.1 Bibliography ... 125
10.2 Index of Definitions ... 135
10.3 Index of Tables ... 136
10.4 Index of Figures ... 136
10.5 List of often used Symbols and Abbreviations ... 138