

Re-Engineering Approach for PLC Programs based on Formal Methods

*Re-Engineering-Ansatz für SPS-Programme auf
Basis formaler Beschreibungen*

vom

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

Doktor der Ingenieurwissenschaften (Dr.-Ing.)

genehmigte Dissertation

von

M.Sc. Mohammed Bani Younis

geb. in Dair Abi Said (Jordanien)

D386

Eingereicht am: 29. Mai 2006
Tag der mündlichen Prüfung: 01. September 2006
Dekan des Fachbereichs: Prof. Dr.-Ing. Wolfgang Kunz

Promotionskommission

Vorsitzender: Prof. Dr.-Ing. habil. Lothar Litz
Berichterstattende: J. Prof. Dr.-Ing. Georg Frey
Prof. Dr.-Ing. Luca Ferrari

Berichte aus der Automatisierungstechnik

Mohammed Bani Younis

**Re-Engineering Approach for PLC Programs
based on Formal Methods**

Re-Engineering-Ansatz für SPS-Programme
auf Basis formaler Beschreibungen

D 386 (Diss. Technische Universität Kaiserslautern)

Shaker Verlag
Aachen 2006

Bibliografische Information der Deutschen Nationalbibliothek

Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über <http://dnb.d-nb.de> abrufbar.

Zugl.: Kaiserslautern, TU, Diss., 2006

Copyright Shaker Verlag 2006

Alle Rechte, auch das des auszugsweisen Nachdruckes, der auszugsweisen oder vollständigen Wiedergabe, der Speicherung in Datenverarbeitungsanlagen und der Übersetzung, vorbehalten.

Printed in Germany.

ISBN-10: 3-8322-5674-1

ISBN-13: 978-3-8322-5674-6

ISSN 0945-4659

Shaker Verlag GmbH • Postfach 101818 • 52018 Aachen

Telefon: 02407 / 95 96 - 0 • Telefax: 02407 / 95 96 - 9

Internet: www.shaker.de • E-Mail: info@shaker.de

Acknowledgements

I wish to acknowledge and thank those people who contributed to this thesis:

Much heartiest to my supervisor, Prof. Georg Frey who has provided me with a lot of advice and help throughout the four years of my PhD studies. He has always been there to help and support not only in my work. His support was from the first time I met him with a vague knowledge about the programmable logic controllers up to the submitting of my thesis.

I am grateful to the reviewers of this thesis, Prof. Georg Frey and Prof. Luca Ferrarini for their interest and the time and effort they invested in reviewing this thesis. My overwhelming thanks to the evaluation committee chair Prof. Dr.-Ing. habil Lothar Litz.

I would like to thank my colleagues who have created a very pleasant atmosphere in the Juniorprofessorship Agentbased Automation and stood beside to me till the end of the work. Working at Juniorprofessorship Agentbased Automation in Kaiserslautern is as being part of a team, as well as part of a family. During this period I have gained fruitful memories I will never forget.

A special thank goes to Dr. Bruno Denis from the LURPA in Cachan for the interest in my work and for the long term interesting discussions which have aid me a lot in my work.

Thank you very much my master thesis supervisor Dr. Roland Hecker whom through his support and heedfulness I came to the decision to start my promotion.

Thanks to Tanvir Hussain and Florian Wagner for the support and proof reading of the manuscript. Without their help there could be considerably more errors and flaws in this thesis. Any errors or shortcomings are my responsibility alone.

I am forever indebted to my wife Ghadeer for her understanding, endless patience and encouragement when it most needed. I must thank to my family. I would not be where I am today without them. Thank you Mum, Dad.

Last, but not least, I would like to dedicate this thesis to my Parents, my wife, my son Bassem and my daughter Tuqa, for their love, patience, and understanding—they allowed me to spend most of the time on this thesis. For the past five months, my son kept on reminding me: “Papa, go to Uni. and write your thesis”!

Kaiserslautern, September 2006

Mohammed Bani Younis

Abstract

Today there is a standard for the implementation of programs on Programmable Logic Controllers (PLCs). Furthermore there are methods for the formal development of these programs. The standard allows the interchange of algorithms (e.g. if a new hardware should be used) and the formal methods allow the rigid proof of functionality especially needed in safety critical applications (e.g. using model checking). However, there are a lot of existing PLC programs that have been implemented in proprietary languages before a standard existed and even today formal methods are scarcely used during design. This work outlines a re-engineering approach based on the formalization of PLC programs. The PLC program modules are modeled as Finite State Machines (FSMs). These FSMs are able to communicate with each other to describe the complete dynamic of the PLC system. The resulting formal model can serve as a basis for editing and analyzing the system. The transformation of PLC programs into a vendor independent format and the visualization of its structure is identified as an important intermediate step in this process. It is shown how XML and corresponding technologies can be used for the formalization, visualization, re-implementation, and software measurement of existing PLC programs.

Für die Implementierung von Speicherprogrammierbaren Steuerungen (SPS) existiert heute ein Standard. Darüberhinaus gibt es eine ganze Reihe formaler Entwicklungsmethodiken für SPS-Programme. Während der SPS-Standard den Austausch von Algorithmen ermöglicht (z.B., wenn eine neue Hardware zum Einsatz kommen soll), stellt die Anwendung formaler Methoden die Möglichkeit dar, die Funktionalität der Programme nachzuweisen. Letzteres ist insbesondere für sicherheitskritische Anwendungen unerlässlich. Leider gibt es jedoch eine ganze Reihe bestehender SPS-Programme die implementiert wurden, noch bevor der Standard existierte und selbst heute werden viele Programme noch ohne die Unterstützung formaler Methodiken entworfen. Im Rahmen dieser Arbeit wird ein Re-Engineering Ansatz vorgestellt, welcher auf der Formalisierung von SPS-Programmen beruht. Hierfür werden die SPS-Programmbausteine zunächst auf Basis von Automaten (Finite State Machines, FSM) modelliert. Diese FSMs kommunizieren miteinander um die vollständige Dynamik des SPS-Systems abzubilden. Das so entstehende formale Modell kann als Basis für den weiteren Systembearbeitungs- und -analyseprozess verwendet werden. Die Transformation des SPS-Programms in ein vom Hersteller unabhängiges Format stellt ebenso wie die Visualisierung der Programmstruktur einen wichtigen Schritt des Re-Engineering-Prozesses dar. In der Arbeit wird gezeigt, wie XML und die zugehörigen Technologien für die Formalisierung, Visualisierung, Re-Implementierung und Softwarequalitätsmessung vorhandener SPS-Programme verwendet werden können.

Contents

1. Introduction	1
1.1 Motivation	1
1.2 Structure of the Thesis.....	1
1.3 New and Important Approaches.....	2
2. PLC Programs and Re-Engineering	5
2.1 PLC Programs	5
2.2 Terminology of Re-Engineering.....	5
2.3 Need for the Re-Engineering.....	9
2.4 Commercial Re-Engineering Solutions for STEP5.....	10
2.5 Re-Engineering based on Formal Methods.....	10
2.5.1 Sources used for the Formalization.....	12
2.5.2 Scope of Formalization.....	12
2.5.3 Aim of the Formalization.....	13
2.5.4 Model used for the Formal Description	14
2.5.5 Application of Re-Engineering to PLC Programs	14
2.5.6 Summary	17
2.6 Re- and Software-Engineering Methods	17
2.6.1 Reverse Engineering or Re-Implementation.....	18
2.6.2 Re-Engineering based on Internet and Object-Oriented Approaches.....	19
2.7 Compound Re-Engineering.....	21
3. PLC Programs' Formalization	23
3.1 PLCs and STEP5.....	23
3.2 Mathematical Description of PLC Programs	24
3.2.1 PLC as Discrete Event System (DES)	24
3.2.2 PLC Program as Communicating Automata.....	25
3.2.3 PLC Program and PLC Cycle.....	26
3.3 Formalization of PLC Programs.....	28
3.3.1 General Considerations.....	28
3.3.2 Concluding Remarks.....	36
3.3.3 Formalization of Binary Programs.....	37
3.3.3.1 Classification of Binary Operations	37
3.3.3.2 Conversion Algorithm for Binary Programs	39
3.3.4 Formalization of Digital Programs	43
3.3.4.1 Classification of Digital Operations	43
3.3.4.2 Transformation of Digital Programs	43
3.3.4.3 Abstraction of Digital Programs	45
3.3.4.4 Example.....	49
3.3.4.5 Conversion Algorithm for Digital Programs.....	49
3.3.5 Formal Description of Counters.....	50
3.3.5.1 Structure and Operating Mode of Counters	50
3.3.5.2 Formal Description and Integration of Counters in the Automaton Model	51
3.3.5.3 An Example of a Program with Counter	52

3.3.6	Formal Description of Timers.....	53
3.3.6.1	Structure and Operating Mode of Timers.....	53
3.3.6.2	Formal Description and Integration of Timers in the Automaton Model	54
3.3.6.3	An Example of a Program with Timer	56
3.4	Discussion.....	57
4.	Visualization of the Formalized PLC Programs	59
4.1	Re-Engineering Approach	59
4.2	Software Engineering Methods and Internet Technologies	59
4.2.1	Introductory Overview.....	59
4.2.2	UML.....	60
4.2.3	XML as a Tool for Visualization	61
4.2.4	XMI.....	64
4.2.5	SVG.....	64
4.3	Visualization Concept	65
4.3.1	Overview	65
4.3.2	Conversion of a PLC Program into a well-formed XML	66
4.3.3	XML Validation against the XML Schema	67
4.3.4	Identification of Instructions.....	68
4.3.5	Visualization of XML	69
4.3.6	Separation of the Modules using XMI.....	70
4.4	Implementation of the Formalization using Java	71
4.5	Implementation Example.....	72
4.6	Summary.....	78
5.	Re-Implementation of PLC Programs	79
5.1	Overview	79
5.2	Conversion of STEP5 to IEC 61131-3	80
5.3	Conversion Concept of the Dynamic Semantic.....	82
5.3.1	Conversion of Binary Algorithms.....	85
5.3.2	Conversion of Timers and Counters	86
5.3.3	Conversion of Non-Binary Instructions and Data Blocks	86
5.4	Discussion.....	88
6.	SW Quality and Measurements	89
6.1	SW Quality Definition.....	89
6.2	PLC Programs' Software Quality.....	90
6.3	Structure and Complexity Metrics.....	92
6.3.1	General Overview	92
6.3.2	LOC.....	93
6.3.3	Halstead Measure.....	93
6.3.4	McCabe's Cyclomatic Complexity.....	94
6.3.5	Information Flow	95
6.3.6	Tree Impurity	96
6.3.7	Coupling.....	97
6.3.8	Summary Assessment of the Structural Metrics	98
6.4	Metrics Implementation.....	98

6.4.1	Overview	98
6.4.2	Size Metric	99
6.4.3	Halstead Measure	99
6.4.4	McCabe Cyclomatic Complexity	102
6.4.5	Tree Impurity	104
6.4.6	Coupling	106
6.5	Conclusions	106
7.	Case Studies	109
7.1	Didactic Case Study	109
7.1.1	Plant Description	109
7.1.2	Conversion Process	111
7.1.3	SW Quality	114
7.2	Industrial Case Study	118
7.2.1	Introduction and Description of the System	118
7.2.2	Visualization and Conversion Process	120
7.2.3	Plant SW Quality	123
7.3	Conclusion	126
8.	Summary	127
8.1	Summary in English	127
8.2	Kurzfassung in deutscher Sprache	128
9.	Bibliography and Indices	135
9.1	Bibliography	135
9.2	Index of Tables	145
9.3	Index of Figures	146
9.4	List of often used Symbols and Abbreviations	147