

TEUBNER-TEXTE zur Informatik Band 2

M. Rupprecht

Implementierung und parallele
Verarbeitung von Kommunikationssoftware

TEUBNER-TEXTE zur Informatik

Herausgegeben von

Prof. Dr. Johannes Buchmann, Saarbrücken

Prof. Dr. Udo Lipeck, Hannover

Prof. Dr. Franz J. Rammig, Paderborn

Prof. Dr. Gerd Wechsung, Jena

Als relativ junge Wissenschaft lebt die Informatik ganz wesentlich von aktuellen Beiträgen. Viele Ideen und Konzepte werden in Originalarbeiten, Vorlesungsskripten und Konferenzberichten behandelt und sind damit nur einem eingeschränkten Leserkreis zugänglich. Lehrbücher stehen zwar zur Verfügung, können aber wegen der schnellen Entwicklung der Wissenschaft oft nicht den neuesten Stand wiedergeben.

Die Reihe „TEUBNER-TEXTE zur Informatik“ soll ein Forum für Einzel- und Sammelbeiträge zu aktuellen Themen aus dem gesamten Bereich der Informatik sein. Gedacht ist dabei insbesondere an herausragende Dissertationen und Habilitationsschriften, spezielle Vorlesungsskripten sowie wissenschaftlich aufbereitete Abschlußberichte bedeutender Forschungsprojekte. Auf eine verständliche Darstellung der theoretischen Fundierung und der Perspektiven für Anwendungen wird besonderer Wert gelegt. Das Programm der Reihe reicht von klassischen Themen aus neuen Blickwinkeln bis hin zur Beschreibung neuartiger, noch nicht etablierter Verfahrensansätze. Dabei werden bewußt eine gewisse Vorläufigkeit und Unvollständigkeit der Stoffauswahl und Darstellung in Kauf genommen, weil so die Lebendigkeit und Originalität von Vorlesungen und Forschungsseminaren beibehalten und weitergehende Studien angeregt und erleichtert werden können.

TEUBNER-TEXTE erscheinen in deutscher oder englischer Sprache.

Implementierung und parallele Verarbeitung von Kommunikationssoftware

Von Michael Rupprecht

Rheinisch-Westfälische Technische
Hochschule Aachen



B. G. Teubner Verlagsgesellschaft
Stuttgart · Leipzig 1993



Dipl.-Ing. Michael Rupprecht

Geboren 1960 in Kiel. Von 1979 bis 1987 Studium der Elektrotechnik mit dem Schwerpunkt Technische Informatik an der Rheinisch-Westfälischen Technischen Hochschule Aachen, 1987 Diplom. Von 1987 bis 1992 wiss. Mitarbeiter am Lehrstuhl für Informatik IV, bei Prof. Dr. Otto Spaniol, 1992 Promotion.

Arbeitsschwerpunkte: Datenkommunikation, insbesondere Hochgeschwindigkeitsnetze; Petrinetze; parallele Programmierung.

Dissertation an der Mathematisch-Naturwissenschaftlichen Fakultät der Rheinisch-Westfälischen Technischen Hochschule Aachen

D 82 (Diss. RWTH Aachen)

Die Deutsche Bibliothek – CIP-Einheitsaufnahme

Rupprecht, Michael :

Implementierung und parallele Verarbeitung von Kommunikationssoftware /

Michael Rupprecht. – Stuttgart ; Leipzig : Teubner, 1993

(Teubner-Texte zur Informatik ; Bd. 2)

Zugl. : Aachen, Techn. Hochsch., Diss., 1992

ISBN 978-3-8154-2050-8 ISBN 978-3-322-93436-9 (eBook)

DOI 10.1007/978-3-322-93436-9

NE : GT

Das Werk einschließlich aller seiner Teile ist urheberrechtlich geschützt. Jede Verwertung außerhalb der engen Grenzen des Urheberrechtsgesetzes ist ohne Zustimmung des Verlages unzulässig und strafbar. Das gilt besonders für Vervielfältigungen, Übersetzungen, Mikroverfilmungen und die Einspeicherung und Verarbeitung in elektronischen Systemen.

© B. G. Teubner Verlagsgesellschaft Leipzig 1993

Umschlaggestaltung: E. Kretschmer, Leipzig

Vorwort

Die vorliegende Arbeit entstand während meiner Tätigkeit als wissenschaftlicher Mitarbeiter am Lehrstuhl für Informatik IV (Prof. Dr. O. Spaniol) der Rheinisch-Westfälischen Technischen Hochschule Aachen.

Herrn Prof. Dr. Otto Spaniol, der durch seine Anregungen, seine konstruktive Kritik und durch die mir gewährten Freiheiten viel zum Gelingen dieser Arbeit beigetragen hat, gilt mein persönlicher Dank.

Herrn Prof. Dr. Wolfgang Effelsberg (Universität Mannheim) danke ich besonders herzlich für seine Bereitschaft zur Übernahme des Korreferats und für sein Interesse an meiner Arbeit.

Danken möchte ich auch allen meinen Kollegen am Lehrstuhl und den Studenten, die auf die eine oder andere Weise einen Beitrag zum Entstehen und Gelingen der Arbeit geleistet haben und hier nicht alle genannt werden können. Besonders hervorheben möchte ich jedoch Herrn Kai Jakobs und Herrn Prof. Dr. Peter Martini, die mir insbesondere in der Anfangsphase bei der Auffindung meines wissenschaftlichen und beruflichen Weges hilfreich waren und Herrn Bernd Heinrichs und Herrn Christian Engel, die mir in ähnlicher Weise in der Schlußphase dieser Arbeit sehr geholfen haben.

Ganz besonderes danken möchte ich den Menschen, die in anderer Weise beim Gelingen und beim Abschluß dieser Arbeit für mich sehr wichtig waren. Namentlich erwähnen möchte ich hier nur meine Frau Cordula Rupprecht, die wahrscheinlich mehr zu dieser Arbeit beigetragen hat, als ich bis heute begriffen habe und meinem Vater, der durch mich nicht nur Probleme mit dem Korrekturlesen dieser Arbeit hatte.

Aachen, September 1992

Michael Rupprecht

2.4.1.2.	Höhere Taktraten, neuere Technologien	36
2.4.1.3.	Direkte Umsetzung in VLSI-Schaltungen	36
2.4.1.4.	Multiprozessorarchitekturen für die Protokollverarbeitung	37
2.4.2.	Nicht standardkonforme Ansätze	37
2.4.3.	Parallele Verarbeitung von Kommunikationssoftware	37
2.4.3.1.	Pipeline-Controller	38
2.4.3.2.	Mehrere Prozessoren	39
2.4.3.3.	Multiport-Speicher	39
2.4.3.4.	Universeller Multiprozessor/Multiport-Speicher- Controller	40
2.4.3.5.	Universeller Multiprozessor Controller	41
2.4.4.	Die Verwendung von Transputern	41
2.4.5.	Zusammenfassung	42
3.	Spezifikation von Protokollen	44
3.1.	Methoden zur Spezifikation	44
3.1.1.	Nicht formale Protokollspezifikationstechniken	44
3.1.2.	Formale Protokollspezifikationstechniken	45
3.2.	Petrinetze	46
3.2.1.	Grundbegriffe	47
3.2.2.	Stellen/Transitionsnetze	50
3.3.	Erweiterungen	55
3.3.1.	Verbotskanten	55
3.3.2.	Abräumkanten	56
3.3.3.	Zeitbehaftete Petrinetze	57
3.3.4.	Individuelle Marken und Prädikate	58
3.4.	Produktnetze	59
3.4.1.	Individuelle Marken	60
3.4.2.	Markierung	60
3.4.3.	Kapazität	60
3.4.4.	Kantenanschrift	60
3.4.5.	Zulässige Interpretationen	62
3.4.6.	Schwellenmarkierung	63

3.4.7. Prädikate	63
3.4.8. Spezielle Kanten	65
3.4.9. Die Aktiviertheit einer Transition	66
3.4.10. Das Schalten einer Transition	66
3.4.11. Beispiel	66
3.4.12. Vereinfachung der graphischen Darstellung	68
4. Protokollimplementierung auf der Basis von Petrinetzen	70
4.1. Petrinetze als Programmiersprache	70
4.1.1. S/T-Netze	71
4.1.2. Produktnetze	72
4.1.3. Scheduler/Task-Konzept	73
4.1.4. Zusammenfassung	74
4.2. Funktionales Konzept der MDMA-Architektur	74
4.2.1. Aktionseinheiten	76
4.2.2. Globaler Datenspeicher	76
4.2.3. Entscheidungseinheit	76
4.2.4. Markenspeicher	78
4.2.5. Suchmodul	79
4.2.6. Sortierer	79
4.2.7. Suchspeicher	80
4.2.8. Zusammenfassung	80
4.3. Eingeschränktes Produktnetz	81
4.3.1. Problem : Testen aller gültigen Interpretationen	81
4.3.2. Problem : dynamische Speicherverwaltung	82
4.3.3. Lösung : Kapazität = 1	82
4.3.4. Lösung : Kapazität > 1	83
4.3.5. Formale Definition	86
4.3.5.1. Einfach-Marken-Stelle	86
4.3.5.2. Mehrfach-Marken-Stelle	87
4.3.6. Beispiele	88
4.3.7. Entflechtung	91
4.3.7.1. Entflechtung einer Eingangskante	92

4.3.7.2.	Entflechtung einer Ausgangskante	93
4.3.7.3.	Entflechtung der Verbots- und Abräumkanten	93
4.3.7.4.	Beispiel	94
4.3.7.5.	Schematisierte Kantenanschrift	94
4.3.7.6.	Verbotsstellen im eingeschränkten, entflochtenen Produktnetz	96
4.3.8.	Beispiel	98
4.3.9.	PENCIL/C-Produktnetz	100
4.3.10.	Zusammenfassung	102
5.	Die Programmiersprache PENCIL	103
5.1.	PENCIL/C als Erweiterung von C	104
5.1.1.	Syntax - Definition	104
5.1.1.1.	Transitionen	104
5.1.1.2.	Stellen	105
5.1.2.	Deklaration von Einfachstellen	107
5.1.3.	Deklaration von Mehrfachstellen	108
5.1.4.	Verwendung der Einfachstellen in der Aktivierungsfunktion	110
5.1.5.	Verwendung des Transitionsrumpfes	111
5.1.6.	Weitere Anmerkung zur Sprache PENCIL/C	113
5.1.6.1.	Verwendung von Zeigertypen	113
5.1.6.2.	Beschränkung der Markenanzahl	113
5.1.6.3.	Ausführungsprioritäten	113
5.1.7.	Beispiel	114
6.	Die Verwendung von PENCIL	116
6.1.	Umsetzung anderer Spezifikationssprachen in PENCIL	116
6.1.1.	Erweiterte endliche Automaten	116
6.1.2.	LOTOS	125
6.2.	Granularität	127
6.2.1.	Verschmelzen von Stellen	127
6.2.2.	Verschmelzen von Transitionen	129
6.2.3.	Vergrößern	131
6.2.4.	Prädikatsvereinfachung	131

Inhalt	11
6.2.5. Zusammenfassung	134
6.3. Parallelität	135
6.3.1. Das Verschmelzen von Stellen und Transitionen	136
6.3.2. Erweiterte endliche Automaten	137
6.3.3. Betriebssystem	138
6.3.4. Parallelität innerhalb einer Verbindung	139
6.3.5. Parallelität des Referenzmodells	141
6.3.6. Parallelität der Verbindungen untereinander	142
6.3.7. Kommunikation zwischen Prozessen	144
6.3.8. Pipeline-Verarbeitung und identische Teilnetze	146
6.3.9. Zusammenfassung	148
7. Die MDMA-Architektur	149
7.1. Funktionsweise der Entscheidungseinheit	149
7.1.1. Quittung	149
7.1.2. Zeigerliste	150
7.1.3. Stellenliste	150
7.1.4. Transitionsliste	151
7.1.5. Auftragsliste	152
7.1.6. Auftrag	152
7.1.7. Aktivierungsbedingungen	152
7.1.8. Zusammenfassung	154
7.2. Modifikation der MDMA-Architektur	154
7.2.1. Vorteil : Reduzierung der Kopiervorgänge	154
7.2.2. Vorteil : Parallele Ausführung der Sortierfunktionalität	155
7.2.3. Vorteil : Vereinfachte Auftragsvergabe	155
7.2.4. Vorteil : Reduzierter Synchronisationsaufwand	155
7.2.5. Zusammenfassung	156
7.3. Kommunikation	156
7.3.1. Synchronisation	157
7.3.2. Kommunikation zwischen Aktions- und Entscheidungseinheit	157

7.3.2.1.	Vergabe von Aufträgen	157
7.3.2.2.	Quittieren einer Schaltfunktion	158
7.3.2.3.	Status der Warteschlangen	158
7.3.3.	Externe Ereignisse	158
7.4.	Hardware-Realisierung der MDMA-Architektur	161
7.4.1.	Realisierung des MDMA-Konzepts mit der Basisarchitektur	161
7.4.2.	Alternative Implementierung von PENCIL/C	162
7.4.3.	Transputer	163
7.4.4.	Spezialisierte MDMA Hardware-Architektur	164
7.4.4.1.	Markenspeicher	164
7.4.4.2.	Warteschlangen	165
7.5.	Reaktionszeiten	166
7.5.1.	Lösung für eine feste Kundenanzahl	168
7.5.2.	Lösung mit veränderlicher Kundenanzahl	171
7.5.3.	Lösung für ein Multiprozessorsystem	172
7.5.4.	Auswertung	173
7.5.5.	Zusammenfassung	177
7.6.	Zusammenfassung	178
8.	Ausblick und Schlußwort	179
9.	Literatur	181
10.	Anhang	193
10.1.	Mathematische Symbole	193
10.1.1.	Relationen	193
10.1.2.	Quantoren	193
10.1.3.	Logische Operatoren	193
10.1.4.	Mengen	193
10.2.	Schreibweisen für Petrinetze	194
10.3.	Begriffe	196
10.4.	Syntaxerweiterungen	196