

Selbstadaptierende NC-Steuerung für rekonfigurierbare Werkzeugmaschinen

Von der Fakultät Konstruktions-, Produktions- und Fahrzeugtechnik
der Universität Stuttgart
zur Erlangung der Würde eines
Doktors der Ingenieurwissenschaften (Dr.-Ing.)
genehmigte Abhandlung

vorgelegt von
Dipl.-Ing. Christian Kircher
geboren in Stuttgart-Bad Cannstatt

Hauptberichter:
Prof. Dr.-Ing. Dr. h.c. A. Verl
Mitberichter:
Prof. Dr.-Ing. C. Brecher

Tag der mündlichen Prüfung: 20. Mai 2011

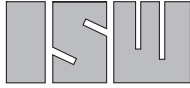
Institut für Steuerungstechnik der Werkzeugmaschinen
und Fertigungseinrichtungen
der Universität Stuttgart

2011

ISW/IPA Forschung und Praxis

Berichte aus dem Institut für Steuerungstechnik
der Werkzeugmaschinen und Fertigungseinrichtungen
der Universität Stuttgart und dem Fraunhofer-Institut für
Produktionstechnik und Automatisierung IPA

Herausgeber: Prof. Dr.-Ing. Dr. h.c. A. Verl




Institut für Steuerungstechnik
der Werkzeugmaschinen und Fertigungseinrichtungen



Fraunhofer
IPA

Christian Kircher



Selbstadaptierende NC-Steuerung für rekonfigurierbare Werkzeugmaschinen

Nr. 185

JOST-JETTER VERLAG
Fachverlag · 71296 Heimsheim

D 93

ISBN 978-3-939890-79-9
Jost Jetter Verlag, Heimsheim

Dieses Werk ist urheberrechtlich geschützt. Die dadurch begründeten Rechte, insbesondere die der Übersetzung, des Nachdrucks, des Vortrags, der Entnahme von Abbildungen und Tabellen, der Funksendung, der Mikroverfilmung oder der Vervielfältigung auf anderen Wegen und der Speicherung in Datenverarbeitungsanlagen, bleiben, auch bei nur auszugsweiser Verwertung, vorbehalten. Eine Vervielfältigung dieses Werkes oder von Teilen dieses Werkes ist auch im Einzelfall nur in den Grenzen der gesetzlichen Bestimmungen des Urheberrechtsgesetzes der Bundesrepublik Deutschland vom 9. September 1965 in der jeweils gültigen Fassung zulässig. Sie ist grundsätzlich vergütungspflichtig. Zuwiderhandlungen unterliegen den Strafbestimmungen des Urheberrechtsgesetzes.

© Jost Jetter Verlag, Heimsheim 2011.

Printed in Germany.

Die Wiedergabe von Gebrauchsnamen, Handelsnamen, Warenbezeichnungen usw. in diesem Werk berechtigt auch ohne besondere Kennzeichnung nicht zu der Annahme, daß solche Namen im Sinne der Warenzeichen- und Markenschutz-Gesetzgebung als frei zu betrachten wären und daher von jedermann benutzt werden dürften.

Sollte in diesem Werk direkt oder indirekt auf Gesetze, Vorschriften oder Richtlinien (z. B. DIN, VDI, VDE) Bezug genommen oder aus ihnen zitiert worden sein, so kann der Verlag keine Gewähr für die Richtigkeit, Vollständigkeit oder Aktualität übernehmen. Es empfiehlt sich, gegebenenfalls für die eigenen Arbeiten die vollständigen Vorschriften oder Richtlinien in der jeweils gültigen Fassung hinzuzuziehen.

Druck: printsystem GmbH, Heimsheim

Geleitwort des Herausgebers

In der Reihe „ISW Forschung und Praxis“ wird fortlaufend über Forschungsergebnisse des Instituts für Steuerungstechnik der Werkzeugmaschinen und Fertigungseinrichtungen der Universität Stuttgart (ISW) berichtet, das sich in vielfältiger Form mit der mechatronischen, steuerungstechnischen und antriebstechnischen Weiterentwicklung von Werkzeugmaschinen und Fertigungseinrichtungen beschäftigt. Die Arbeiten dieses Instituts konzentrieren sich im Besonderen auf die Bereiche Numerische Steuerungstechnik, Planungs- und Leitsysteme, Softwaretechnik, Maschinen- und Industrierobotertechnik sowie Mess-, Regel- und Antriebssysteme, also auf die aktuellsten Bereiche der Fertigungstechnik. Dabei stehen Grundlagenforschung und anwenderorientierte Entwicklung in einem stetigen Austausch, wodurch ein ständiger Technologietransfer zur Praxis sichergestellt wird.

Diese Buchreihe erscheint in zwangloser Folge und stützt sich auf Berichte über abgeschlossene Forschungsarbeiten und Dissertationen. Sie soll dem Ingenieur bei der Weiterbildung dienen und ihm Hilfestellungen zur Lösung spezifischer Probleme geben. Für den Studierenden bietet sie eine Möglichkeit zur Wissensvertiefung. Sie bleibt damit unter erweitertem Namen und in der inzwischen dritten Generation unverändert in der bewährten Konzeption, die ihr der Gründer des ISW, der leider allzu früh verstorbene Prof. Dr.-Ing. G. Stute, im Jahre 1972 gegeben hat und die Prof. Dr.-Ing. h.c. mult. Dr.-Ing. E.h. G. Pritschow von 1984 bis 2005 fortgesetzt hat.

Seit 2007 ist das ISW durch die gemeinsame Institutsleitung mit dem Fraunhofer IPA verbunden. Folgerichtig erscheinen nun auch die Berichte des IPA im Bereich Automatisierungstechnik in dieser Reihe.

Der vorliegende Band beschäftigt sich mit dem Entwurf einer selbstadaptierenden NC-Steuerung für rekonfigurierbare Werkzeugmaschinen durch Übertragung von Plug-and-Play-Prinzipien auf die NC-Konfigurierung. Zunächst werden in der Arbeit Merkmale von Plug-and-Play-Systemen analysiert, die anschließend auf mechatronische Werkzeugmaschinenmodule übertragen werden. Zur Abbildung von Maschinen- und Steuerungskonfigurationen, sowie ihrer funktionalen Zusammenhänge wird ein Informationsmodell geschaffen, aus dem heraus automatisiert Konfigurierungslisten für NC-Steuerungen generiert werden können. Die Ergebnisse der Arbeit wurden im Rahmen des SFB 467 und dem TFB 59 exemplarisch umgesetzt.

Der Herausgeber dankt der Druckerei für die drucktechnische Betreuung und dem Jost Jetter Verlag für die Aufnahme der Reihe in sein Lieferprogramm.

A. Verl

Vorwort

Die vorliegende Arbeit entstand während meiner Tätigkeit als wissenschaftlicher Mitarbeiter am Institut für Steuerungstechnik der Werkzeugmaschinen und Fertigungseinrichtungen (ISW) der Universität Stuttgart.

Mein sehr herzlicher Dank gilt meinem Doktorvater, Herrn Professor Dr.-Ing. Dr. h. c. Alexander Verl für die Betreuung der Arbeit und die Übernahme des Hauptberichts, sowie für die vertrauensvolle Zusammenarbeit und die interessante Zeit am Institut.

Herrn Professor Dr.-Ing. Christian Brecher, Institutsdirektor am Werkzeugmaschinenlabor WZL der RWTH Aachen, danke ich herzlich für das Interesse und die Übernahme des Mitberichts.

Herrn Professor Dr.-Ing. Wolfgang Schinköthe danke ich für die kurzfristige Übernahme des Prüfungsvorsitzes.

Allen Mitarbeitern des ISW, besonders den Kollegen im Fachbereich Steuerungstechnik, danke ich für die konstruktive und stets freundschaftliche Zusammenarbeit. Die vielen Diskussionen im Kollegenkreis haben mich fachlich und persönlich sehr bereichert und geprägt. Besonders meine ersten Jahre am ISW mit meinem Abteilungsleiter Dr.-Ing. Thomas Bürger, meinem Gruppenleiter Dipl.-Ing. Michael Seyfarth und den ersten Zimmerkollegen Dr.-Ing. Stefan Staudt und Dr.-Ing. Andreas Selig habe ich sehr positiv in Erinnerung.

Ganz besonders herzlich bedanke ich mich bei Thomas und Jörg, die während meiner Zeit am ISW nicht nur Mentor und Kollege waren, sondern auch wertvolle Freunde geworden sind.

Mein besonderer Dank gilt den Herren Dr.-Ing. Thomas Bürger, Dipl.-Ing. Michael Seyfarth und meinem Vater für die gewissenhafte und kritische Durchsicht meiner Arbeit und die stets sehr hilfreichen fachlichen Diskussionen.

Meinen Eltern danke ich sehr herzlich für die ständige Unterstützung in jeglicher Hinsicht und die Ermutigungen während meines bisherigen Werdegangs. Meiner Carolin, sowie allen meinen Freunden, danke ich für den moralischen Rückhalt, die Aufmunterungen und Ablenkungen während des Erstellens dieser Arbeit. Ihnen allen verdanke ich auch die Einsicht, dass es nicht nur die Dissertation gibt.

Christian Kircher

Inhaltsverzeichnis

Abkürzungen und Symbole	12
Abstract	14
1 Einleitung	15
2 Begriffe	17
2.1 Komponenten eines Steuerungssystems	17
2.2 Grundmechanismen für die Adaption von Steuerungssystemen	19
2.2.1 Rekonfigurierbare Systeme	21
2.3 Objektorientierung und baukastenbasierte Informationsmodellierung	23
2.4 Software- und Referenzarchitektur	24
3 Anforderungen an die Adaptierbarkeit von Steuerungssystemen	25
3.1 Interner Aufbau und Informationsfluss numerischer Steuerungen	25
3.1.1 Steuerungssoftware und Steuerungsplattform	25
3.1.2 Interner Informationsfluss	26
3.1.3 NC-SPS-Schnittstelle	27
3.1.4 NC-/SPS-Schnittstellen zur Feldebene	28
3.2 Anpassbarkeit von Steuerungssystemkomponenten	29
3.2.1 Anpassbarkeit der NC-Steuerung	29
3.2.2 Anpassbarkeit der Speicherprogrammierbare Steuerung	30
3.2.3 Anpassbarkeit des Interaktionssystems	30
3.2.4 Anpassbarkeit des Antriebssystems	31
3.3 Zusammenfassung der Anforderungen an die Adaption von Steuerungssystemen	31
4 Stand der Technik	34
4.1 Konfigurierbare und rekonfigurierbare Systeme	34
4.1.1 Baukastensysteme für konfigurierbare Werkzeugmaschinen	35
4.1.2 Rekonfigurierbare Werkzeugmaschinen	36
4.1.3 Werkzeugunterstützte Konfigurierung von rekonfigurierbaren Maschinen	39
4.1.4 Zusammenfassung der Arbeiten im Bereich rekonfigurierbarer Maschinen	41
4.2 Modularisierung von Steuerungssystemen	41
4.3 Adaption von Steuerungssystemen	43
4.3.1 Adaption von NC-Steuerungen	43
4.3.2 Adaption von speicherprogrammierbaren Steuerungen	44
4.3.3 Adaption des Interaktionssystems	45
4.3.4 Werkzeugunterstütztes Konfigurieren von Steuerungssystemen	46
4.3.5 Zusammenfassung der Arbeiten im Bereich Adaption von Steuerungssystemen	49




4.4	Informationsmodellierung	50
4.5	Bewertung bestehender Arbeiten und Konzepte	53
4.6	Zusammenfassung und Zielsetzung der Arbeit	56
5	Analyse und Systematisierung des NC-Konfigurierungsprozesses	59
5.1	Analyse des Vorgehens bei der Adaption von NC-Steuerungen	59
5.1.1	Abhängigkeiten zwischen Maschinen- und Steuerungskonfiguration	59
5.1.2	Anpassung der NC-Steuerung auf Basis mentaler Modelle	61
5.2	Systematisierung des Vorgehens bei der NC-Konfigurierung	63
5.3	Analyse der herstellerübergreifenden Anpassbarkeit von NC-Steuerungen	67
5.3.1	Analyse der Konfigurierungsmöglichkeiten	68
5.3.2	Strukturmodell einer adaptierbaren NC-Steuerung	72
5.4	Zusammenfassung	75
6	Methode zur Selbstadaption von NC-Steuerungen	76
6.1	Analyse von Plug-and-Play-Systemen aus der Computertechnik	77
6.1.1	Selbstkonfigurierung von Plug-and-Play-Systemen	77
6.1.2	Eigenschaften von Plug-and-Play-Systemen	81
6.2	Methoden zur Automatisierung der Inbetriebnahmetätigkeiten	82
6.2.1	Automatische Erkennung der Maschinenmodule	84
6.2.2	Automatisches Ableiten von NC-Konfigurierungsobjekten und einer Steuerungsstruktur	85
6.2.3	Auslesen von Maschinenmodulparametern	86
6.3	Anforderungen an Plug-and-Play-fähige Maschinenmodule	86
6.4	Voraussetzungen für die Übertragbarkeit von Plug-and-Play-Prinzipien	87
6.5	Methode zur Selbstadaption von NC-Steuerungen	89
6.6	Zusammenfassung	91
7	Konzept für eine selbstadaptierende NC-Steuerung	94
7.1	Modulkonzept für Werkzeugmaschinen auf Basis mechatronischer Module	94
7.1.1	Mechatronische Module	94
7.1.2	Steuerungstechnischer Aufbau von mechatronischen Modulen	96
7.1.3	Baukastensystem auf Basis mechatronischer Module	97
7.2	Definition eines durchgängigen mechatronischen Informationsmodells	99
7.2.1	Definition von NC-Konfigurierungsobjekten	101
7.2.2	Definition von Referenzarchitekturen	104
7.2.3	Definition von Maschinenmodulobjekten	105
7.2.4	Definition von Kinematikkonfigurationen	106
7.2.5	Abbildung der Zusammenhänge zwischen den Teilmodellen	110
7.2.6	Überblick über das Informationsmodell	111
7.3	Methode zur modellbasierten Konfigurierung der NC-Steuerung	112

7.4	Mechanismus zur Koordination der Modul- und Systemkonfigurierung	115
7.4.1	Selbstkonfigurierendes Kommunikationssystem	116
7.4.2	Zustandsautomat eines PnP-fähigen Moduls	117
7.4.3	Zustandsautomat des Konfigurierungskordinators	118
7.5	Konzept der selbstadaptierenden NC-Steuerung	121
7.6	Zusammenfassung	124
8	Ergebnisse der Arbeit	125
8.1	Komponentenweise Realisierung einer selbstadaptierenden NC-Steuerung	126
8.1.1	Modulkonzept für Werkzeugmaschinen	127
8.1.2	Informationsmodell und modellbasierte Konfigurierungsmethode	128
8.2	Bewertung der Realisierung	129
9	Zusammenfassung	131
10	Literaturverzeichnis	133

Abkürzungen und Symbole

AT	Antwort-Telegramm
CSR	Control and Status Register
FE	Funktionale Einheit
FÖDERAL	Föderale Software-Architektur für die Erstellung von Maschinen und Anlagen
HIPARMS	Highly Productive and Reconfigurable Manufacturing Systems
HÜMNOS	Herstellerübergreifende Module für den nutzerorientierten Einsatz der offenen Steuerungsarchitektur
IfW	Institut für Werkzeugmaschinen der Universität Stuttgart
ISW	Institut für Steuerungstechnik der Werkzeugmaschinen und Fertigungseinrichtungen der Universität Stuttgart
JOP	Japan Open System Promotion Group
MOF	Meta Object Facility
MPST	Mehrprozessor-Steuerungssystem
MAREA	Study and Definition of Machining Workstation Reference Architecture
MDT	Master-Daten-Telegramm
MOSYN	Modular Synthesis of advanced Machine Tools
MoWiMa	Modellierung und Wiederverwendung objektorientierter Maschinensoftware
NC	Numerical Control (Numerische Steuerung)
NC-KO	NC-Konfigurierungsobjekt
NRT	Non-Real-Time
OCEAN	Open Controller Enabled by an Advanced real-time Network
OMAC	Organization for Machine Automation and Control
OMG	Object Management Group
OSACA	Open System Architecture for Controls within Automation Systems
OSEC	Open System Environment for Manufacturing
PnP	Plug-and-Play
ROM	Read Only Memory
SFB 467	Sonderforschungsbereich 467: „Wandlungsfähige Unternehmensstrukturen für die variantenreiche Serienproduktion“

SPS	Speicherprogrammierbare Steuerung
TCP/IP	Transmission Control Protocol/Internet Protocol
TFB 059	Transferbereich 059: „Wandlungsfähigkeit in der variantenreichen Serienfertigung“
UML	Unified Modelling Language
USB	Universal Serial Bus
XML	Extensible Markup Language

	„Ist-ein“-Beziehung (Vererbung)
	„besteht-aus“-Beziehung (Aggregationsbeziehung)
	Instanziierungsbeziehung
<i>kursiv</i>	Klasse oder Objekt des Informationsmodells

Abstract

In the future, manufacturing companies will have to deal with market turbulences triggered by globalization and mass customization more than ever before. In addition, due to rapidly changing market demands the life cycles of consumer goods are shorter than the payback period of the machinery and equipment on which they are manufactured. Therefore machine tools must be rapidly adaptable to alternating and unpredictable production orders. They must be reconfigurable. This affects both the mechanical design and the control system.

If a machine tool is reconfigured by exchanging machine modules, the Numerical Control (NC) must be adapted to the new machine structure also. In doing so dependencies between the machine configuration and the NC must be considered. Specialists normally have these dependencies in mind during the initial design and commissioning phase of the NC configuration. That's why the NC is closely related to a special machine configuration, and the NC has to be configured anew after each mechanical reconfiguration. So, the configuration of the NC must be as automated as far as possible.

In this doctoral thesis plug-and-play-methods are transferred to the configuration of NC and a self-adaptable NC is designed which adapts itself automatically to a machine configuration. To sum up, the following innovations were developed:

- A manufacturer independent concept of an adaptable NC was created by systematisation the NC configuration process. The concept is based on general NC configuration objects which are used to configure the NC.
- Plug-and-play-systems were analysed in order to identify properties of plug-and-play-devices and to define preconditions to apply plug-and-play-methods to NC technology.
- A self-adaptable NC and a modular machine concept based on mechatronic modules were defined by transferring properties of plug-and-play-systems to machine tools.
- An information model was defined which represents the functional relations between machine and NC configuration. It includes the NC configuration objects of an adaptable NC.
- For the self-adaptation of the NC a model based configuration method was developed. This method uses the functional relations of the information model to identify the NC configuration objects related to the machine modules of the machine configuration. The combination of the NC configuration objects is defined by patterns for NC configurations. A generator is used to create a configuration file out of the information model. This configuration file is afterwards interpreted by the adaptable NC.

1 Einleitung

Aufgrund zunehmender Globalisierung und steigender Individualisierung von Massenprodukten agieren produzierende Unternehmen in einem **turbulenten wirtschaftlichen Umfeld** /1, 2/. Es ist gekennzeichnet durch Verkürzung der Produktlebenszyklen, schwankende Nachfragemengen, sinkende Losgrößen und steigende Variantenvielfalt /3, 4, 5, 6/. Dieses turbulente Umfeld erfordert von produzierenden Unternehmen eine schnelle **Reaktions- und Anpassungsfähigkeit** aller Fabrikobjekte auf allen Ebenen des Fabrikbetriebs (Bild 1.1). Dies umfasst sowohl die technische Infrastruktur, die Produktionstechnik als auch die organisatorischen Prozesse sowie Führungs- und Controllinginstrumente /7/. Für den Erfolg am Markt ist daher entscheidend, inwieweit es den Unternehmen gelingt, die Turbulenzursachen zu beherrschen oder durch geeignete Maßnahmen deren Auswirkungen bereits im Ansatz zu vermeiden. Die Eigenschaft, frühzeitige und vorausschauende Anpassungen der Unternehmensstrukturen und -prozesse durchzuführen, wird als **Wandlungsfähigkeit** bezeichnet /8, 9, 10/.

Eine Studie zur Wandlungsfähigkeit zeigt, dass ein Großteil der produzierenden Unternehmen als wandlungsträge bezeichnet werden kann /11, 12/. Dies liegt unter anderem auch an den **wandlungsträgen Produktionsmitteln**. In einer durch den Industriekreis des Rahmenkonzepts „Forschung für die Produktion von morgen“ durchgeführten Voruntersuchung zum Thema „Werkzeugmaschine 2010“ bestätigt sich dieses Bild /13, 14/. Die Gestaltung **rekonfigurierbarer Maschinen** und Anlagen wird dort als eines der fünf wichtigsten Handlungsfelder identifiziert, um den Widerspruch zwischen hoher Lebensdauer und Kostenintensität von Investitionsgütern auf der einen und Verkürzung des Produktlebenszyklus sowie sinkenden Losgrößen und steigender Variantenvielfalt auf der anderen Seite aufzulösen. Auch auf internationaler Ebene wird die Rekonfigurierbarkeit der Produktionstechnik als entscheidender Wettbewerbsfaktor für die Zukunft gesehen /15, 16, 17, 18, 19, 20/.

Aufgrund der hohen Investitionskosten für Produktionsmittel, die erst nach mehreren Produktlebenszyklen abgeschrieben sind, stellt die **Wandlungsfähigkeit der Produktionsmittel** eine große Herausforderung dar /21, 22, 23/. Maschinen müssen sowohl mechanisch als auch steuerungstechnisch im Laufe ihres Lebens mehrere Male an geänderte Randbedingungen angepasst werden. Ziel der Produktion im turbulenten Umfeld ist es daher, neben der Durchlaufzeitverkürzung, insbesondere bei häufiger auftretenden Umbauvorgängen, die Anlaufzeiten zu verkürzen, um die Produktivität der Maschinen- und Anlagen auf einem hohen Niveau zu halten. Produktionsmittel müssen daher **rekonfigurierbar**, das heißt anpassbar hinsichtlich Struktur, Technologie und Kapazität sein /24, 25/. Rekonfigurierbarkeit von Maschinen und Anlagen lässt sich durch **Modularisierung** und **Standardisierung von Modulschnittstellen**

erreichen /23, 24, 26/. Damit lassen sich bestehende modulare Systeme zerlegen und entsprechend einer neuen bzw. veränderten Produktionsaufgabe optimiert neu strukturieren.

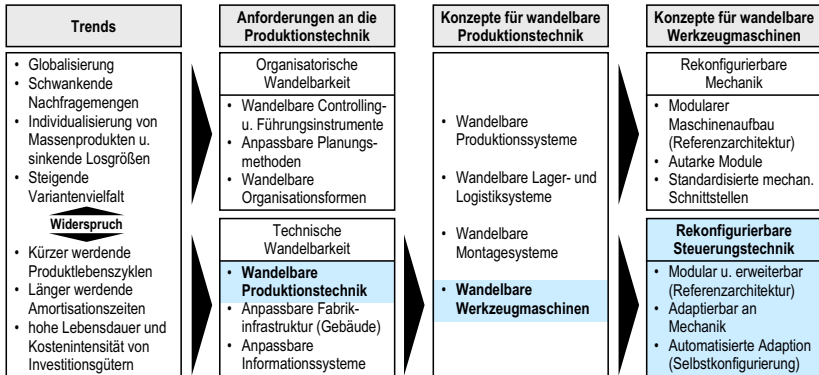


Bild 1.1: Einordnung und Schwerpunkt der Arbeit

Am Markt verfügbare Maschinenkonzepte für numerisch gesteuerte Werkzeugmaschinen haben bereits einen hohen Grad an Modularität. Die Module sind zumindest herstellerepezifisch standardisiert und ermöglichen somit eine **kundenindividuelle Konfigurierung** der Maschine /27, 28, 29, 30, 31/. Die Modularität bezieht sich allerdings bisher nur auf die mechanischen Baugruppen. Die **Rekonfigurierung der Steuerungstechnik** ist hingegen noch nicht möglich, da die Software der eingesetzten Systeme statisch konfiguriert ist. Dies geschieht einmalig und zwar während der Inbetriebnahme. Die NC-Steuerungen lassen sich zwar über so genannte Konfigurationslisten an die Maschinenkonfiguration anpassen. Diese sind allerdings jedes Mal **herstellerspezifisch** zu erstellen und erfordern ein **hohes Maß an Spezialwissen**, da funktionale Zusammenhänge zwischen NC-Softwareparametern und dem mechanischem Aufbau der Maschine zu berücksichtigen sind. Nach jeder mechanischen Rekonfigurierung muss bisher die NC-Steuerung neu konfiguriert werden. Wegen mangelnder Modularität der Steuerungssysteme und fehlenden **Adaptionsmethoden** ist dieser Vorgang komplex und dauert relativ zur mechanischen Rekonfigurierung zu lang.

Aufgabenstellung der vorliegenden Arbeit ist es deshalb, ein **selbstadaptierendes NC-Steuerungssystem** für rekonfigurierbare Werkzeugmaschinen zu konzipieren. Es soll selbstorganisierend sein. Dazu soll es mechanische Rekonfigurierungen erkennen, **automatisch** die notwendigen **Anpassungen der NC-Steuerung** ermitteln und durchführen. Dabei sind die Abhängigkeiten zwischen der Steuerungskonfiguration und dem mechanischen Aufbau der Maschine zu berücksichtigen und formal abzubilden.

2 Begriffe

Im folgenden Kapitel werden die der Arbeit zugrundeliegenden Begriffe definiert.

2.1 Komponenten eines Steuerungssystems

Zur Steuerung von Werkzeugmaschinen wird ein **Steuerungssystem** eingesetzt. Es besteht im Wesentlichen aus den folgenden **Steuerungssystemkomponenten** (Bild 2.1): Numerische Steuerung (NC) für die Bewegungserzeugung koordinierter Achsen, Speicherprogrammierbare Steuerung (SPS) für die Steuerung von Maschinenfunktionen, Interaktionssystem bestehend aus Benutzungsoberfläche und Maschinensteuerung, Stell- und Messglieder (Antriebssystem, Aktoren und Sensoren).

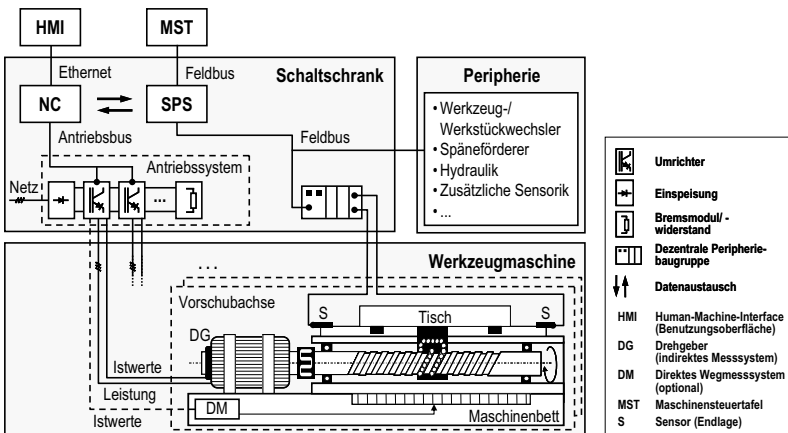


Bild 2.1: Beispielhafte Darstellung von Steuerungssystemkomponenten einer numerisch gesteuerten Werkzeugmaschine

Für die Steuerung von Bewegungen werden numerische Steuerungen (**NC-Steuerungen**) eingesetzt /32/. Ihre **Programmierung** wurde in der DIN 66025 /33/ genormt. Diese legt einen Programmaufbau aus NC-Sätzen fest, die Geometrie- und Technologieinformationen und Aufrufe von Maschinenfunktionen (M- und T-Befehle) beinhalten. Die Programmabarbeitung erfolgt satzweise. Eine NC-Steuerung erzeugt aus dem NC-Programm eine definierte Relativbewegung zwischen Werkzeug und Werkstück. Dazu wird das Programm sequentiell in einem steuerung-internen Informationsfluss, dem **NC-Kanal**, von NC-Funktionen verarbeitet. Die Steuerung setzt dadurch die programmierte Bahn in separate Verfahrbefehle für die Maschinenachsen der Maschinenkinematik um (Transformation), indem sie die Achspositionier-

systeme synchronisiert (zyklisch typischerweise im 1 ms-Takt) mit Sollwerten versorgt (Interpolation). Durch diese synchronisierten Einzelachsbewegungen erzeugt die NC-Steuerung eine Bewegung auf der Bahn. Dabei berücksichtigt sie die physikalischen Gesetzmäßigkeiten der Achsen, so dass die physikalischen Grenzen der Mechanik (z.B. Beschleunigungsgrenzen) nicht überschritten werden.

Die NC-Steuerung hat **Schnittstellen** zur Benutzungsoberfläche, zu den Antriebssystemen, zur SPS. Die Schnittstelle zur SPS ist eine interne Schnittstelle. Sie ist meist als Koppelspeicher ausgeführt. Die Antriebsschnittstelle ist inzwischen typischerweise ein digitales Bussystem /34, 35, 36/. Die Benutzungsoberfläche ist über einen Ethernetbus angebunden.

Für die Steuerung von logischen Abläufen für Maschinen- und Sicherheitsfunktionen werden **Speicherprogrammierbare Steuerungen** (SPS) eingesetzt. Die SPS ist freiprogrammierbar. Ihre Programmierung ist in der DIN EN 61131-3 /37/ genormt. Hauptaufgabe der SPS ist die Steuerung der maschinenspezifischen Funktionen der Peripherieaggregate (z.B. Kühlschmiermittelpumpe, Werkzeug- und Werkstückwechselsysteme). Sie werden von der SPS über Aktoren und Sensoren angesteuert, indem Funktionsaufrufe aus dem NC-Programm und Quittierungen zwischen der NC und SPS ausgetauscht werden. Die SPS wird daher auch als Anpassteuerung bezeichnet, weil sie die Maschinenfunktionen maschinenneutral anpasst und über Befehle im NC-Programm ansteuerbar macht.

Die SPS hat eine **Schnittstelle** zur Maschinensteuertafel und zur Feld-Ebene (Sensor/Aktor-Ebene). Beide Schnittstellen sind heute als Feldbusschnittstellen ausgeführt. Hier hat sich Profibus, CAN und DeviceNet als Standard durchgesetzt /38/. Die Peripherie ist meist durch sogenannte dezentrale Peripheriebaugruppen mit einer Feldbusschnittstelle an die SPS angeschlossen.

Für die Visualisierung und Beeinflussung des Maschinenzustands wird an die NC-Steuerung ein Interaktionssystem bestehend aus einer **Benutzungsoberfläche** und einer **Maschinensteuertafel** angekoppelt. Über die Maschinensteuertafel nimmt der Benutzer Einfluss auf die Maschinenfunktionen (z.B. Ein-/Ausschalten von Kühlung und Spindel). Er kann damit die Betriebsart (Hand-, Einzelsatz-, Automatikbetrieb) der Maschine auswählen und somit die Maschine von Hand bewegen oder ein NC-Programm starten. Die Bussysteme zur Benutzerebene sind meist ethernetbasierte Busse. Das Protokoll ist herstellerspezifisch und nutzt häufig die OPC-Technologie (OLE for Process Control) zur Datenübertragung /39/. Die Maschinensteuertafel ist über einen Feldbus an die SPS angeschlossen.

Stell- und Messglieder haben die Aufgabe, den Fertigungsprozess zu beeinflussen (**Aktoren**) und Zustandsinformationen zu erfassen (**Sensoren**) und in die übergeordnete Steuereinrich-

tung (NC, SPS) zurückzuführen. Dazu werden unterschiedliche Wirkprinzipien (hydraulisch, pneumatisch, elektrisch, mechanisch, ...) benutzt /40/. Aktoren und Sensoren werden in **Peripherieaggregaten** für die Erzeugung der Maschinenfunktionen eingesetzt. Die Peripherieaggregate bilden meist geschlossene Einheiten. Ihr Anschluss an die SPS erfolgt über einen Feldbus und dezentralen Feldbuskomponenten (dezentrale Peripheriebaugruppe).

Antriebssysteme sind nach /41/ geregelte Systeme und werden für die Positionierung von Maschinenachsen eingesetzt. Eine Achse wird üblicherweise in Lageregelung betrieben. Achsen werden unterschieden in Bearbeitungs- und Hilfsachsen.

Die **Bussysteme** zum Antrieb sind spezielle Feldbusse und werden auch als Motion- oder Antriebsbus bezeichnet. Im Gegensatz zu den Feldbussen, die an die SPS angeschlossen sind (Zykluszeit: 5-20 ms), haben sie schnellere Zykluszeiten (1 ms und kleiner) und arbeiten mit Protokollen, welche spezielle Antriebsprofile (definierter Funktionsumfang) unterstützen /42, 43/. **Antriebsbusse** sind inzwischen typischerweise digitale Bussysteme. Übliche offene Standards für Antriebsbusse sind SERCOS und Profibus /36/. Mit der Einführung von ethernetbasierten Echtzeitbussystemen haben sich neben SERCOS III und Profinet, EtherCat und Powerlink durchgesetzt /38, 44, 45/. Diese einheitlichen Bussysteme heben die Trennung zwischen Feld- und Antriebsbussen auf /46, 47/.

2.2 Grundmechanismen für die Adaption von Steuerungssystemen

Unter Adaption wird die Anpassung der Software von Steuerungssystemen an die Anforderungen der Maschine verstanden. Dafür gibt es die Mechanismen **Parametrierung, Konfiguration und Funktionsentwicklung** /48/ (Bild 2.2).

Parametrieren heißt nach /49/, den Parametern von Funktionseinheiten oder Programmbausteinen Werte zuzuweisen. Damit wird der intern festgelegte Programmablauf einer Funktionseinheit durch Parameter von außen beeinflusst. Im Bereich der Steuerungssoftware werden aus Parametern programminterne Variablen abgeleitet. Parameter im Bereich der Steuerungstechnik können z.B. Koeffizienten des Lageregelkreises sein, die aus physikalischen Größen der Maschine abgeleitet werden. Wichtig für diese Art der Adaption ist die Beschreibung der Steuerungsaufgabe in allgemeingültiger, parametrisierbarer Form.

Konfigurieren ist nach /49/ das Zusammensetzen aus vorgegebenen einzelnen Funktions- oder Baueinheiten zu einem Ganzen (System). Das Ergebnis ist eine Konfiguration. Konfigurierbare Systeme müssen daher aus Komponenten mit einheitlichen Schnittstellen bestehen. Übertragen auf die Steuerungssoftware wird damit das Bilden der Struktur der Steuerungssoftware aus mehreren funktionalen Einheiten bezeichnet. Diese Art der Adaption

wird gewählt, wenn das Aufgabenspektrum durch unterschiedliche Kombinationen von einzelnen Funktionseinheiten beschrieben werden kann.

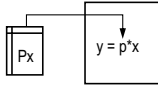
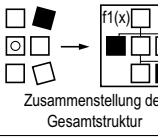
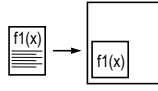

Basismechanismen für die Adaption der Steuerungssoftware			
	Parametrierung	Konfigurierung	Funktionsentwicklung
Prinzip	 Steuerung des Programmablaufs	 Zusammenstellung der Gesamtstruktur	 Entwicklung und Integration neuer Funktionen
Vorausgesetzte Kenntnisse	Prozess (Funktion)	●	●
	System (Struktur)	○	●
	Software (Realisierung)	○	●
Voraussetzung für den Einsatz	Aufgabe allgemeingültig beschreibbar	Aufgabenspektrum in Einzelaufgaben zerlegbar	Funktion einfach integrierbar
Relativer Aufwand (bei optimaler Auslegung)			
	● benötigt ○ Realisierungsunabhängig (bei optimaler Auslegung nicht benötigt)		

Bild 2.2: Adaptionsmechanismen für Steuerungssoftware (nach /48/)

Während Konfigurieren das erstmalige Erzeugen einer Konfiguration ist (z.B. bei der Erstinbetriebnahme), ist das **Rekonfigurieren** ein Umgestalten einer bestehenden Konfiguration. Dabei werden die Einstellungen der vorherigen Konfiguration soweit wie möglich übernommen, um die Zeit beim Rekonfigurieren zu reduzieren. Das Ergebnis einer Rekonfigurierung ist eine Rekonfiguration.

Wenn eine gewünschte Funktion nicht als Software vorhanden ist, muss sie im Rahmen einer **Funktionsentwicklung** neu entwickelt, das heißt, als Quellcode geschrieben werden. Eine wichtige Voraussetzung dafür ist die Möglichkeit, neu entwickelte Funktionen in die Steuerungssoftware integrieren zu können. Offene Steuerungssysteme basierend auf Plattformen bieten diese Möglichkeit für den Maschinenhersteller. Bei geschlossenen Systemen ist dazu die Unterstützung des Steuerungsherstellers notwendig.

Am Markt verfügbare NC-Steuerungen stellen sich bei der Konfigurierung als geschlossenes, **monolithisches System** dar (Bild 2.3, rechts). Die strukturelle Anpassung erfolgt von außen über Vorgabe von Werten (Parameter), die über eine Konfigurierungsschnittstelle eingelesen werden. Ein steuerungsinternes **Konfigurationslaufzeitsystem** sorgt durch Interpretation dieser Daten für die Konfigurierung (strukturelle Anpassung) der NC beim Steuerungshochlauf /48/. Eine Unterscheidung zwischen Parametrierung und Konfigurierung fällt daher schwer und findet nur in der Bezeichnung der Parameter statt. Sie werden eingeteilt in

Parametrierungs- und Konfigurierungsdaten (Bild 3.1).

In dieser Arbeit wird daher unter **Adaption** sowohl die Konfigurierung (strukturelle Anpassung) als auch die Parametrierung (funktionale Anpassung) als Anpassung der NC-Steuerung über Vorgabe von Parameterwerten verstanden.

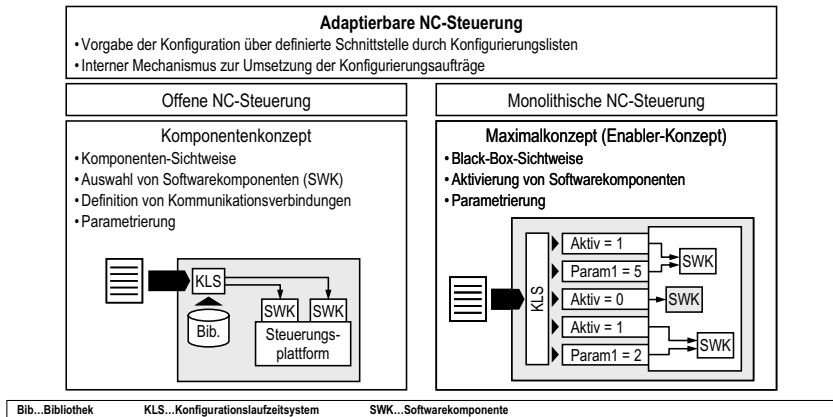


Bild 2.3: Adaptierbare NC-Steuerung

Adaptiere NC-Steuerungen sind in dieser Arbeit NC-Steuerungen mit einer Schnittstelle und einem internen Adaptionmechanismus für von außen vorgegebene Konfigurationenaufträge. Sie lassen sich unterscheiden in offene und monolithische NC-Steuerungen (Bild 2.3). **Offene NC-Steuerungen** basieren auf Komponenten, die austauschbar sind und definierte Schnittstellen besitzen (Komponentenkonzept). Eine offene NC-Steuerung wird durch Auswahl und Kombination von Softwarekomponenten konfiguriert. Anschließend werden die Kommunikationsverbindungen definiert und die Softwarekomponenten parametrierung. Monolithische NC-Steuerungen stellen sich als Blackbox dar. Die Konfiguration erfolgt von außen. Über Parameter werden Softwarekomponenten aktiviert (Enabler-Konzept).

2.2.1 Rekonfigurierbare Systeme

Die Rekonfigurierbarkeit eines Systems leitet sich aus dessen **Konfigurierbarkeit** ab. Die Konfigurierung entspricht dem Auslegen, Auswählen und Zusammensetzen von Modulen aus Baukastensystemen entsprechend den Anforderungen und dem Anwenderbedarf /50/. Rekonfigurationen stellen die **späteren Umbaumaßnahmen** und **Veränderungen von Struktur, Funktionalität, Kapazität und Technologie** durch Ersatz, Ergänzung oder Entfernung von einzelnen, in sich funktionsfähigen Modulen dar.

Es gibt zwei Arten von rekonfigurierbaren Systemen (Bild 2.4) /51/. Systeme vom Typ 1 bestehen aus Modulen, die in einer **Referenzarchitektur** vordefiniert sind. Systeme vom Typ 2 bestehen aus Modulen, die nicht vordefiniert und nicht in einer Referenzarchitektur vorgesehen sind. Sie müssen nachträglich in der Referenzarchitektur vordefiniert werden. Diese Arbeit beschränkt sich auf die Betrachtung der Maschinen des Typs 1, da automatisierte Konfigurations- und Rekonfigurierungsmethoden nur bei diesem Typ angewandt werden können.

Eine Rekonfigurierung kann auf unterschiedlichen Ebenen stattfinden. Auf der Ebene der Produktionssysteme findet eine Rekonfigurierung durch Austauschen, Entfernen oder Ersetzen von ganzen Maschinen statt. Rekonfigurierbare Produktionssysteme bestehen aus rekonfigurierbaren Bearbeitungs- und Montagesystemen, einem rekonfigurierbaren Steuerungs- und Informationssystem und einem rekonfigurierbaren Materialhandling- und Logistiksystem /52/. Die Module des rekonfigurierbaren Produktionssystems sind in diesem Fall ganze Maschinen. In rekonfigurierbaren Produktionssystemen spielt die rekonfigurierbare Verkettung dieser Module eine wichtige Rolle. Auf der Ebene von rekonfigurierbaren Werkzeugmaschinen sind die austauschbaren Module Baugruppen und Komponenten wie z.B. Vorschubeinheiten, Werkzeugwechselsysteme, angetriebene Werkzeuge, zusätzliche Hilfsachsen.

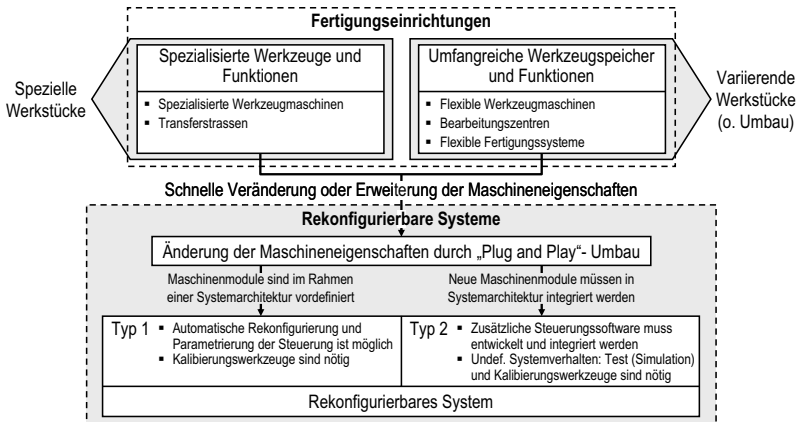


Bild 2.4: Typen konfigurierbarer und rekonfigurierbarer Systeme (nach /51/)

Kennzeichnend für rekonfigurierbare Werkzeugmaschinen sind der **modulare Aufbau** und die **Anpassbarkeit** durch Ersetzen, Ergänzen oder Entfernen von Modulen. Module rekonfigurierbarer Maschinen besitzen **standardisierte** mechanische, informations- und energetische **Schnittstellen**. Um die Schnittstellenvielfalt zu reduzieren, werden Module so

weit es geht als **mechatronische, autarke Einheiten** ausgelegt /24, 53/.

2.3 Objektorientierung und baukastenbasierte Informationsmodellierung

Objektorientierung bedeutet allgemein die Vereinigung von Datenstruktur und Verhalten. Reale Objekte werden dabei als Softwareobjekte abgebildet, die Daten und Methoden kapseln. Daten beschreiben den Zustand und Methoden das Verhalten von Objekten.

Objekte werden in Objektinstanzen (**Instanzen**) und Objektklassen (**Klassen**) unterschieden /54/. Eine Klasse, deren Instanzen Klassen sind, wird als Metaklasse bezeichnet. Oberklassen stellen die Verallgemeinerung einer Menge von Objekten mit ähnlicher Datenstruktur und ähnlichem Verhalten dar. Die Klassifizierung hat das Ziel, gleichartige Objekte zu Klassen zu abstrahieren und zusammenzufassen. Durch Aufbau einer **Hierarchie** entsteht eine Ordnung (Ober- und Unterklassen), die zur Strukturierung der Objekte und Klassen dient. In dieser Arbeit werden mechanische und steuerungstechnische Objekte zum Aufbau und zur Strukturierung von Informationsmodellen klassifiziert.

Die **objektorientierte Analyse** (OO-Analyse) wird beim Entwurf von Softwaresystem während der Anforderungsphase angewendet. Sie gliedert sich nach /55/ in fünf Phasen: das Auffinden von Objekten, das Organisieren und Strukturieren der Objekte, das Beschreiben der Objektinteraktionen, die Definition der Objektoperationen und die Definition des internen Objektaufbaus. In dieser Arbeit wird die OO-Analyse eingesetzt, um durch Abstraktion ein Strukturmodell einer adaptierbaren NC-Steuerung und ein Informationsmodell zur Abbildung der funktionalen Zusammenhänge zwischen Mechanik und NC-Steuerungstechnik zu entwerfen.

Ein **Baukastensystem** besteht aus einer Bausteinbibliothek mit kombinierbaren **Baukastenelementen** (Bausteinen) und einer **Baukastensystematik** (Baumusterplan), die angibt wie die Elemente der Baukastenbibliothek unter Beachtung von Unverträglichkeiten miteinander kombiniert werden können /56/. Dadurch ermöglicht die Baukastensystematik nur sinnvolle Kombinationen. Baukastensysteme sind nach bestimmten Sichtweisen meist fachdisziplinspezifisch strukturiert (Klassifikation).

Zur Abbildung von realen Objekten in Baukastensystemen werden strukturierte **Informationsmodelle** eingesetzt. Ein Informationsmodell ist eine realitätsbezogene, semantisch und datenverarbeitungsinvariante, formale Abbildung eines Anwendungsbereichs, die eine Rechnerverarbeitung ermöglicht /57/. Das Informationsmodell entsteht durch **Formalisierung und Abbildung** von nutzerspezifischem mentalen Wissen in einem rechnerinternen Modell (Repräsentationsmodell). Dazu werden die realen Betrachtungseinheiten im Informationsmodell

auf Strukturobjekte abgebildet. Strukturobjekte und die Beziehungen zwischen Strukturobjekten bilden die Modellierungselemente eines Informationsmodells.

Das methodische Vorgehen zur Erfassung der für das Informationsmodell relevanten Informationen wird unter dem Begriff **Erstellungssystematik** zusammenfasst /57, 58/. Dabei wird der abzubildende Gegenstand oder Sachverhalt nach bestimmten Kriterien schrittweise hierarchisch zerlegt. Die dabei gefundenen Strukturelemente werden zueinander in Beziehung gesetzt. Die Prinzipien, die bei der Erstellung von Baukastensystemen angewendet werden, sind **Abstraktion, Modularisierung und Hierarchisierung** /59/.

Zur Beschreibung von Objekten und Informationsmodellen hat sich die **Unified Modeling Language** (UML) als **Standard-Modellierungssprache** durchgesetzt /60, 61/. Sie ist fach- und realisierungsunabhängig und von der Object Management Group (OMG) /62/ standardisiert. Sie definiert sowohl textuelle als auch grafische Beschreibungsmittel. Mit der UML können reale Systeme als Objekte und Klassen modelliert werden. Zur Beschreibung von Zusammenhängen werden in der UML Assoziationen zwischen Objekten und Klassen benutzt.

2.4 Software- und Referenzarchitektur

Als **Softwarearchitektur** wird die Beschreibung der **Struktur eines Softwaresystems** auf einer geeigneten Abstraktionsebene in Form von Komponenten (Subsysteme, Module, Pakete, Klassen, ...) und ihrer Beziehungen zueinander bezeichnet /59/. Die Softwarearchitektur beschreibt die Komponenten, deren grundlegende Organisation und deren Zusammenspiel innerhalb eines Softwaresystems. Der Definition der Schnittstellen zwischen den Komponenten kommt daher eine wichtige Bedeutung zu.

Eine **Referenzarchitektur** ist in dieser Arbeit ein **Modellmuster** (Referenzmodell) für eine Klasse von Softwarearchitekturen. Eine Referenzarchitektur ist also ein abstraktes, idealtypisches Muster, welches als Vorlage für eine konkrete Ausprägung genutzt wird, indem vorgegebene Softwarestrukturen mit konkreten Softwarekomponenten ausgefüllt werden.

3 Anforderungen an die Adaptierbarkeit von Steuerungssystemen

Im folgenden Kapitel wird ausgehend vom internen Aufbau und dem Informationsfluss numerischer Steuerungen die Anpassbarkeit der Steuerungssystemkomponenten dargestellt, um daraus die Anforderungen an die Adaptierbarkeit von Steuerungssystemen abzuleiten.

3.1 Interner Aufbau und Informationsfluss numerischer Steuerungen

3.1.1 Steuerungssoftware und Steuerungsplattform

Der interne Aufbau der am Markt verfügbaren numerischen Steuerungssysteme ist zwar nicht offengelegt, aber aufgrund der äußeren, sichtbaren Schnittstellen und aus Gründen der Wart- und Handhabbarkeit der Software /64/ ist davon auszugehen, dass der **innere Aufbau** (Struktur) plattformbasiert ist und sich im Allgemeinen an Strukturierungsprinzipien der MPST-Initiative /64, 65, 66, 67, 68/ und des OSACA-Projektes orientiert /69, 70/.

Bild 3.1 zeigt den **plattformbasierten Aufbau** einer numerischen Steuerung nach /71/ bestehend aus Systemhardware, Systemsoftware und Anwendungssoftware. Die Systemhardware stellt die notwendigen Ressourcen wie Speicher, Prozessoren, Ein- und Ausgabebaugruppen bereit. Durch die Systemsoftware wird eine Abstraktionsschicht über die Hardware gelegt, so dass die Anwendungssoftware prozessor- und herstellerunabhängig entwickelt werden kann. Zur **Systemsoftware** gehören Betriebs-, Kommunikations-, Datenbank- und Grafiksysteme. Die Systemhardware und die Systemsoftware bilden zusammen die Systemplattform. Ihre Dienste und Funktion werden in einer Anwendungsprogrammierschnittstelle (API) bereitgestellt und dienen als Grundlage für die Implementierung und den Betrieb der Anwendungssoftware.

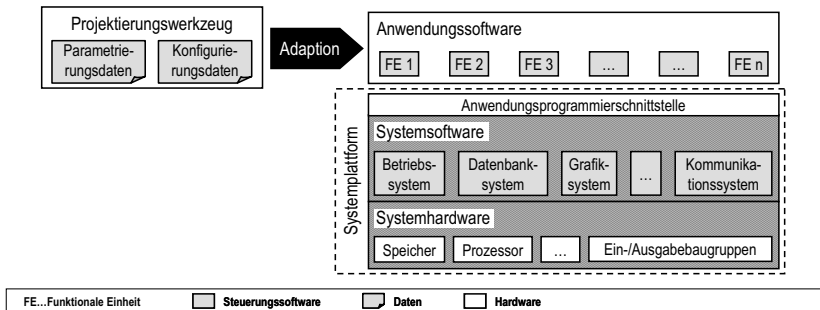


Bild 3.1: Strukturierung der Steuerungssoftware (in Anlehnung an /71/ und /72/)

Die **Anwendungssoftware** ist für die eigentliche Umsetzung der Funktionalität einer NC verantwortlich. Sie setzt sich aus Anwendungsmodulen zusammen, welche die spezifischen Funktionen für die Umsetzung der steuerungstechnischen Aufgabe erbringen. Sie werden in der Literatur auch als **Funktionale Einheiten** (FE) bezeichnet /72/. Dadurch lässt sich NC-Steuerungssoftware modular aus FE aufbauen. Typische FE sind Decoder, Werkzeuggeometriekorrektur, Bahnvorbereitung, Bahnausführung und Lageeinstellung. Die Menge der FE kann durch Konfigurierung angepasst werden /40, 73/.

Neben der Unterteilung von Software in System- und Anwendungssoftware werden im Bereich der numerischen Steuerungssoftware auch **Parametrierungs- und Konfigurierungsdaten** unterschieden /48, 72/. Die Parametrierungs- und Konfigurierungsdaten werden durch Softwarewerkzeuge im Rahmen der Inbetriebnahme erzeugt. Mit den **Konfigurierungsdaten** wird die Struktur (Topologie) der Anwendungsprogramme definiert. Damit wird die Anwendungssoftware an die konkreten maschinenspezifischen (z.B. Anzahl Achsen) und technologiespezifischen (z.B. verfügbare Werkzeuge) Anforderungen adaptiert. **Parametrierungsdaten** ermöglichen die Anpassung der Anwendungssoftware an die physikalischen Randbedingungen (z.B. maximal zulässige Beschleunigung der Antriebe) der zu steuernden Maschine.

3.1.2 Interner Informationsfluss

Zur Erzeugung von **Sollwertvorgaben** für die Achsen wird das NC-Programm von der NC-Steuerung decodiert und in einen internen NC-Steuerdatensatz umgewandelt. Die Steuerdaten durchlaufen sequentiell die Softwarekomponenten Decoder, Werkzeuggeometriekorrektur, Bahnvorbereitung, Bahnausführung und Lageeinstellung (**Bild 3.2**). Diesen festgelegten Informationsfluss nennt man **NC-Kanal**. Der Informationsfluss in einem NC-Kanal ist unidirektional. Ein NC-Kanal erzeugt Sollwerte für eine koordinierte Achsgruppe (gemeinsam interpolierte Achsen). Dazu sind jedem NC-Kanal ein NC-Programm und mehrere Achsen zugeordnet. Eine Steuerung kann je nach Leistungsfähigkeit mehrere NC-Kanäle haben. Damit können mehrere Achsverbände von einer NC-Steuerung gleichzeitig koordiniert werden. Die Kanalstruktur der NC-Steuerung muss bei der Inbetriebnahme konfiguriert werden.

Da die Versorgung der **Antriebe** mit Sollwerten hohen Echtzeitanforderungen im Bereich von 1 ms genügen muss, werden viele NC-Funktionen mit denselben Rechenzeitanforderungen in einer Softwarekomponente zusammengefasst. **Bild 3.2** zeigt den Funktionsumfang einer NC und eine allgemein gängige Strukturierung der NC-Software in fünf **Funktionale Einheiten** (FE) /40, 73, 74/.

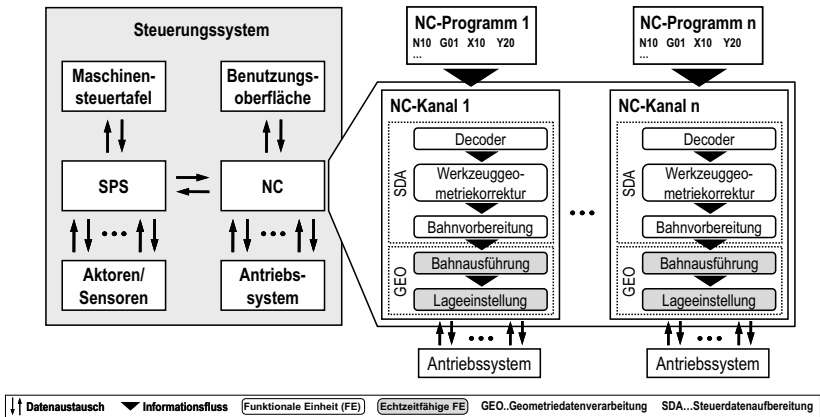


Bild 3.2: Aufbau eines NC-Kanals und steuerungsisernen Informationsfluss

Die Implementierung der verschiedenen NC-Funktionen als FE ist allerdings herstellerspezifisch und wenig öffentlich dokumentiert. Wegen denselben Rechenzeitanforderungen ist eine Zusammenfassung von Decoder, Werkzeuggeometriekorrektureinheit und Bahnvorbereitung zur Softwarekomponente **Steuerdatenaufbereitung** (SDA) und eine Zusammenfassung von Bahnausführung und Lageeinstellung zur Softwarekomponente **Geometriedatenverarbeitung** (GEO) sinnvoll und üblich /40, 73/. Im Rahmen der Inbetriebnahme erfolgt die Anpassung der NC, indem die Struktur (NC-Kanal mit Achsen) der Steuerung konfiguriert wird und den Parametern der Algorithmen der FE Werte zugewiesen werden.

3.1.3 NC-SPS-Schnittstelle

Die Schnittstelle zwischen NC und SPS ist herstellerspezifisch (Bild 3.3). Sie dient dem Daten- und Signalaustausch und der Synchronisation von Signalen zwischen der NC und der SPS. Aus der NC heraus werden **Maschinenfunktionen** (M- und T-Befehle) angestoßen und Statusmeldungen an die SPS (z.B. Programm läuft/beendet) gesendet. Die SPS sendet Statusmeldungen (z.B. Fehlermeldungen) über den Maschinenzustand und Steuersignale (z.B. Betriebsartenwahl, Spindelansteuerung, Kanalauswahl) sowie Quittierungen an die NC.

Die Kopplung der beiden Steuerungen erfolgt über einen gemeinsamen **Speicherbereich** und definierte Bitlisten (Variablenliste). Der Speicherbereich kann gemeinsam genutzt werden (**Koppelspeicher**, Dual-Ported RAM), wenn beide Systeme auf einer Hardwareplattform untergebracht sind (z.B. PC-basiertes System) oder wenn sie über einen Rückwandbus miteinander gekoppelt sind. Die Syntax und Semantik der Schnittstelle ist nicht offengelegt son-

dem herstellerspezifisch. Der Zugriff auf den Koppelspeicher erfolgt meist über interne Schnittstellen und vom Steuerungshersteller bereitgestellte Funktionsbausteine.

Funktionen der SPS können aus dem NC-Programm über Maschinenfunktionen angesprochen werden. Sie sind teilweise in der DIN 66025 /33/ genormt und werden auch als M-Funktionen bezeichnet. Die Norm definiert allerdings nur einen minimalen Befehlssatz. Jeder Hersteller bietet über die Norm hinaus eigene M-Funktionen an. Sie lassen sich in kanal- und achsspezifische Funktionen unterteilen. Kanalspezifische M-Funktionen sind einem NC-Kanal zugeordnet während achsspezifische M-Funktionen zu einer Achse gehören, unabhängig davon welchem NC-Kanal die Achse zugeordnet ist. Die M-Funktionen werden nach dem Zeitpunkt ihrer Auswirkung, der Dauer ihrer Auswirkung und der Art ihrer Quittierung unterschieden /33/.

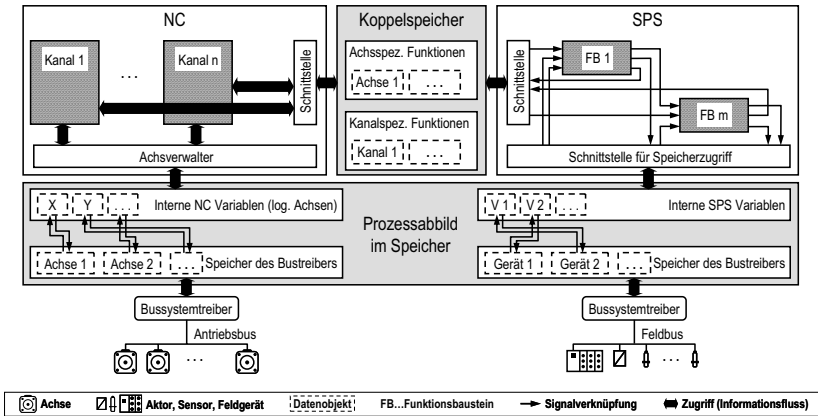


Bild 3.3: Steuerunginterne Kopplung über Speicher

3.1.4 NC-/SPS-Schnittstellen zur Feldebene

Die Ankopplung der NC zur **Feldebene** erfolgt ebenso wie bei der SPS über einen gemeinsamen Speicherbereich, den sich die NC bzw. die SPS mit einem Bustreiber teilt. Dort legt der Bustreiber Datenobjekte der Feldebene an. Die NC- und SPS haben dort einen Speicherbereich für ihre internen Variablen (**Prozessabbild**) auf die sie über entsprechende interne Schnittstellen zugreifen können. Für die Datenkommunikation werden die internen Variablen der NC und SPS bei der Konfigurierung der Steuerung mit den Datenobjekten der Feldebene verknüpft (Feldbus-Linking). Über diese vorkonfigurierte Signalverknüpfung findet im Speicher eine Zuordnung zwischen internen Variablen der NC und der SPS und dem Prozessabbild.

bild der Feldebene (Aktor/Sensor-Ebene) statt.

Bild 3.3 zeigt einen gemeinsam genutzten Speicher einer Einprozessorlösung bei der beide Systeme auf einer Hardware untergebracht sind. Bei geringen Anforderungen an die Zykluszeiten kann auf einen separaten Feldbus (Antriebsbus) für die Achsen verzichtet werden. Die Kommunikation wird dann von einem Feldbus für die NC und die SPS übernommen.

3.2 Anpassbarkeit von Steuerungssystemkomponenten

Leistungsfähige NC-Steuerungssysteme treten in aller Regel als **Gesamtpaket** aus NC- und speicherprogrammierbarer Steuerung auf. Ein Grund dafür ist auch die Tatsache, dass es keine genormte Schnittstelle zum Datenaustausch zwischen NC und SPS gibt und Lösungen hierfür grundsätzlich herstellerspezifisch sind. SPS und NC sind daher auf einer Hardware untergebracht, werden jedoch separat adaptiert. HMI und Maschinensteuertafel sind bei PC-basierten Architekturen meist zusammen untergebracht und werden als **Interaktionssystem** bezeichnet.

3.2.1 Anpassbarkeit der NC-Steuerung

Der Großteil der am Markt üblichen NC-Steuerungen wird als geschlossenes, monolithisches System geliefert, das in einem gewissen, vorgegebenen Rahmen an die Bearbeitungstechnologie und die Maschinenkonfiguration durch Konfigurierung und Parametrierung angepasst werden kann. Die **Anpassung** erfolgt über maschinenspezifische Parameter (Maschinendaten), die Merkmale der Maschinenkonfiguration aus Sicht der Steuerung beschreiben und festlegen. Maschinenparameter werden in ASCII-basierten Textdateien auf der Steuerung abgelegt. Die Strukturierung der Textdatei ist herstellerspezifisch. Bei einem System /75/ wird XML /76/ zur Strukturierung der Textdatei eingesetzt. Beim Hochlauf der Steuerung werden diese Dateien interpretiert und die Software der Steuerung durch ein **Konfigurationslaufzeitsystem** angepasst (Bild 3.4).

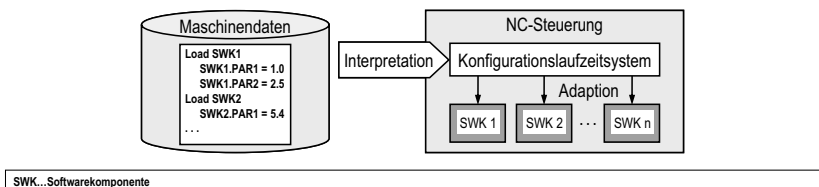


Bild 3.4: Adaption einer NC-Steuerung über Maschinendaten

Der Speicherbereich der NC-SPS-Schnittstelle wird ebenfalls über die allgemeinen Maschinendaten konfiguriert. Dazu werden Speicherbereiche für globale, kanal- und achsspezifische

Variablen definiert. Außerdem wird im Speicher Platz für die Werkzeugverwaltung angelegt.

Können funktionale Anforderungen nicht erfüllt werden, so müssen neue Funktionen im Quellcode hinzugefügt, das System neu gelinkt und kompiliert werden. Diese Art der Funktionsentwicklung kann nur durch den Steuerungshersteller erfolgen, da der Quellcode der NC-Steuerung dem Maschinenhersteller gegenüber nicht offengelegt ist und zur Verfügung gestellt wird. Bietet die Funktionserweiterung kein entsprechendes Marktpotenzial, haben insbesondere größere Steuerungshersteller kein kommerzielles Interesse an dieser Entwicklung. Lediglich kleinere Steuerungshersteller bieten Spezialfunktion und entsprechend modulare und erweiterbare NC-Steuerungen an. Der Quellcode steht dem Maschinenhersteller in der Regel zwar auch dann nicht zur Verfügung aber durch entsprechende Kooperationen lassen sich schnell und einfach Funktionen durch den Steuerungshersteller hinzufügen.

Maschinenherstellern mit großem eigenem Wissen im Bereich der Steuerungstechnik bieten die Firmen Siemens mit dem Konzept der **Compilezyklen** /77/ und Bosch Rexroth mit dem Konzept der **Joblisten** /78/ spezielle Mechanismen zur Erweiterung der Steuerungsfunktionalität durch Eigenentwicklungen an. Für den Großteil der Maschinenhersteller ist dies allerdings zu aufwändig und wird daher nicht genutzt.

3.2.2 Anpassbarkeit der Speicherprogrammierbare Steuerung

Da die SPS frei programmierbar ist, erfolgt die **Anpassung** der SPS an die Steuerungsaufgabe über das SPS-Programm. Die SPS-Programmierung ist zwar in der DIN EN 61131-3 /37/ genormt, da aber viele Hersteller eigene Programmiersprachen und erweiterte Funktionsbibliotheken anbieten, sind Programme nicht immer auf andere Steuerungen übertragbar. Die Programmierung erfolgt in der Regel auf Basis **vordefinierter Bausteine**, die die Funktionen der anzusteuernenden Peripherieaggregate kapseln. Das SPS-Programm wird zur Laufzeit zyklisch abgearbeitet, in dem die Eingänge gelesen werden, das Programm mit diesem Eingangsabbild abgearbeitet wird und anschließend die Ausgänge geschrieben werden.

Der Aufwand für die Anpassung kann sich durch **baukastenbasierte Programmierung** und eine Programmstruktur, die über Funktionsbausteine mit definierten Schnittstellen eine Hardwareabstraktion schafft, erheblich reduzieren /79. 80/.

3.2.3 Anpassbarkeit des Interaktionssystems

Benutzungsoberflächen müssen für eine Maschine konfiguriert werden. Steuerungshersteller bieten dafür Projektierungswerkzeuge mit vorgefertigten Masken für die gängigen Maschinenkonfigurationen an. Bei der Inbetriebnahme müssen lediglich die Anzahl und die Art der Achsen für die Statusanzeigen (Istwerte, Schleppabstand, Zustände) konfiguriert werden. Die

Sprachauswahl und die Einstellung der Anzeigefeinheit und der Einheiten erfolgt durch Parametrierung.

Da sich für die Hardware der Oberfläche PC-basierte Systeme mit Windows-Betriebssystem durchgesetzt haben, lassen sich speziell angepasste Oberflächen mit Entwicklungswerkzeugen und **bereitgestellten Bibliotheken** leicht erstellen. Der Zugriff auf NC-Daten erfolgt über herstellerspezifisch definierte Funktionsbibliotheken, die meist auf der OPC-Technologie basieren /39/.

Die Maschinensteuertafel ist über einen Feldbus an die SPS angeschlossen und stellt sich wie ein Feldgerät über Ein- und Ausgänge dar. Diese müssen bei der Projektierung (Feldbus-Linking) mit entsprechenden Variablen und Funktionsbausteinen im SPS-Programm verknüpft werden. Die Steuerungshersteller bieten für die Ansteuerung von Maschinensteuertafeln **fertige Funktionsbausteine** an. Sie sind entweder in Bibliotheken verfügbar oder bereits in ein SPS-Grundprogramm eingebettet.

3.2.4 Anpassbarkeit des Antriebssystems

Das **Antriebssystem** besteht aus den Teilmodulen Einspeisung, Umrichter und ggf. Bremsmodul. Das Umrichtermodul beinhaltet eine Regeleinrichtung für die Strom- und Geschwindigkeitsregelung. Je nach Anforderung und Rechenleistung des Reglers kann auf dem Umrichter auch ein Lageregler implementiert sein. Die für die Regelung benötigten Istgrößen werden entweder über indirekte Messsysteme oder über direkte Messsysteme erfasst.

Die **Parametrierung** der Antriebe erfolgt durch die achsspezifischen Maschinendaten über die NC-Steuerung. Sie lädt während ihres Hochlaufs über die bestehende Kommunikationsverbindung die Antriebsparameter auf den Antrieb. Dazu müssen auf den Antrieben lediglich die physikalischen Kommunikationsadressen festgelegt sein. Die Festlegung des Antriebstyps, die Parametrierung des Lage-, Geschwindigkeit- und Stromregelkreises, sowie die Anpassung der Gebersysteme erfolgt ebenfalls über die achsspezifischen Maschinendaten.

3.3 Zusammenfassung der Anforderungen an die Adaption von Steuerungssystemen

Durch die zunehmende **Modularisierung** der mechanischen Baugruppen und der Standardisierung der Modulschnittstellen wird die **mechanische Rekonfiguration** zukünftig schneller und einfacher durchführbar sein /12, 13/. Dies erfordert jedes Mal eine Neukonfigurierung und Parametrierung des **Steuerungssystems**, da die Konfigurierung und Parametrierung des Steuerungssystems abhängig von dem mechanischen Aufbau der Maschine und den physikalischen Eigenschaften der verwendeten Baugruppen ist.

Die **Anpassung** heutiger **Steuerungssysteme** an individuelle Maschinenkonfigurationen ist zeitaufwändig, da sie aufgrund dieser Abhängigkeit detailliertes **Spezialwissen** und langjährige Erfahrung bei der Inbetriebnahme und dem Umgang mit herstellerspezifischen Konfigurierungswerkzeugen erfordert. Das Wissen über die Erstellung einer korrekten Steuerungskonfiguration bei der die **Abhängigkeiten** von der Maschinenkonfiguration berücksichtigt sind liegt nur als **mentales Modell** in den Köpfen von Spezialisten vor.

Da Steuerungssysteme für rekonfigurierbare Maschinen nicht nur einmalig während der Inbetriebnahme konfiguriert und parametrisiert, sondern ständig rekonfiguriert werden, werden zum einen **einfach anpassbare Steuerungssysteme** benötigt, die sich in einem durch die Kombinierbarkeit der mechanische Module vorgegebenen Rahmen an die Maschine adaptieren lassen. Zum anderen werden neue **herstellerunabhängige Konfigurierungsmethoden** benötigt, die die Neu- und Umkonfigurierung von Steuerungssystemen systematisch unterstützen und dadurch die Steuerungsinbetriebnahme verkürzen. Diese Methode muss die Zusammenhänge zwischen mechanischem Aufbau der Maschine und notwendiger Konfigurierung der Steuerung berücksichtigen, so dass nicht jedes Mal ein entsprechender Experte für die Adaption von Steuerungssystemen vor Ort sein muss.

Eine **Vereinfachung** der Inbetriebnahme und eine **Verkürzung** der Inbetriebnahmezeiten bei gleichzeitiger Reduzierung der Fehler während der Inbetriebnahme lässt sich nur durch **Automatisierung** der Routineaufgaben und selbstständige Adaption des Steuerungssystems erreichen. Für eine kurzfristige Rekonfigurierung von Werkzeugmaschinen werden deshalb Mechanismen in der Steuerung benötigt, die die Steuerungssoftware **selbstständig** der geänderten maschinenbaulichen Konfiguration anpassen. Zusammenfassend lassen sich folgende Anforderungen an adaptierbare Steuerungssysteme für rekonfigurierbare Werkzeugmaschinen formulieren (Bild 3.5):

- Die Steuerung muss anpassbar sein in Bezug auf Struktur und Funktion der Maschine. Dies wird durch einen **modularen, strukturellen Aufbau der Steuerungssoftware** und **Konfigurierungsmechanismen in der Steuerung** erreicht.
- Die Adaption muss einfach sein, das heißt die Konfigurierung soll soweit wie möglich ohne Expertenwissen möglich sein. Dazu sind **formale Abbildungen** des mental vorhandenen Expertenwissens notwendig.
- Die Adaption des Steuerungssystems soll weitestgehend selbstständig sein. Dazu ist eine Methode für eine **automatisierte, selbstständige Adaption** notwendig. Die Methode zur Anpassung soll herstellerunabhängig sein und die **Abhängigkeiten zwischen Maschinen- und Steuerungskonfiguration** berücksichtigen.

- Die Adaption des Steuerungssystems soll mit einer **herstellerübergreifenden Methode** möglich sein. Die Methode soll für **marktübliche Systeme** funktionieren und einfach mit bisheriger Technik, die an der Maschine schon vorhanden ist, umsetzbar sein.

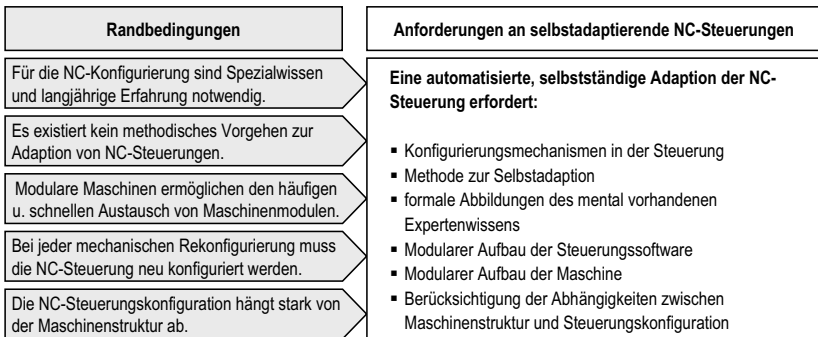


Bild 3.5: Anforderung an adaptierbare Steuerungssysteme für rekonfigurierbare Werkzeugmaschinen

4 Stand der Technik

4.1 Konfigurierbare und rekonfigurierbare Systeme

Allgemein setzt sowohl im Hard- als auch im Softwarebereich die **Wandlungsfähigkeit** immer **modulare Systemkonfigurationen** voraus, welche aus austauschbaren Bausteinen bestehen. In der Praxis finden Rekonfigurationen nur auf der Ebene von Produktionssystemen durch **Austausch** kompletter Maschinen oder im Bereich von Montageanlagen durch Austausch von Montagestationen statt. Im ersten Fall stellen die Maschinen als ganzes ein Modul des rekonfigurierbaren Produktionssystems dar, während im zweiten Fall eine Montagestation das Modul der rekonfigurierbaren Montageanlage ist.

Die beiden modularen Systeme sind dadurch gekennzeichnet, dass flexible und steckbare **Schnittstellen** für die elektrische und pneumatische Energieversorgung und die Informationstechnik (Bussystem) verwendet werden (Bild 4.1). Damit ist ein Anschluss oder Austausch von Modulen einfach und schnell durchführbar. Eine Rekonfiguration lässt sich so durch Umstellen oder Austauschen einzelner Bearbeitungseinheiten beziehungsweise Maschinen zu neuen Verbänden erreichen. Damit können beispielsweise neue Anforderungen bezüglich der Arbeitsabfolgen und der Kapazität durch Parallel- oder Serienanordnung der einzelnen Bearbeitungseinheiten erreicht werden.

Im Bereich **Montage** ist die Modularisierung beispielsweise einfach, da jede Station **autark** ist und ihre eigene Steuerung hat. Eine Verkopplung der Stationen wird über ein gemeinsames Transportsystem realisiert, die Synchronisation erfolgt über eine Kopfsteuerung. Ansonsten sind die Stationen entkoppelt und bilden eine **mechatronische, austauschbare Einheit** /81, 82/.

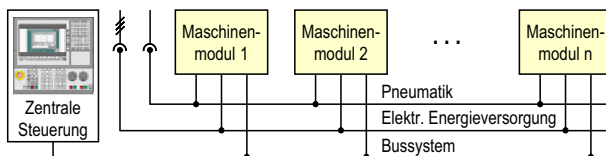


Bild 4.1: Schematischer Aufbau rekonfigurierbarer Systeme mit steckbaren und standardisierten Modulschnittstellen

Folgt man dem Prinzip der Modularisierung und Standardisierung von Schnittstellen können **Baukästen für Maschinen** erstellt werden. Diese sind allerdings meist nur für die einmalige Konfiguration entworfen. Rekonfigurierbare Maschinen basieren zwar auch auf modularen Strukturen, ihre Module sind allerdings speziell für die Rekonfiguration ausgelegt, indem

die Schnittstellenvielfalt durch Integration funktional zusammengehörender Komponenten reduziert wurde.

4.1.1 Baukastensysteme für konfigurierbare Werkzeugmaschinen

Im Bereich der Werkzeugmaschinen sind **Baukastensysteme** bei den Komponenten von Werkzeugmaschinen, wie beispielsweise bei den so genannten Zukaufteilen sowie bei den Bearbeitungswerkzeugen und den Vorrichtungssystemen ebenso bekannt wie **modulare Transferstraßen** und aus Baukästen gestaltete Verkettungseinrichtungen /26, 27, 28/. Allen diesen Baukastensystemen liegen unternehmensspezifisch definierte, mechanische Schnittstellen zugrunde, die einen Austausch innerhalb eines bestimmten Lieferprogramms zulassen.

Damit lassen sich Baukastensysteme für **Planungs- und Auslegungswerkzeuge** aufbauen. Mit deren Hilfe können Werkzeugmaschinen dann in einem vorgedachten Rahmen durch Auswahl einzelner Module kundenindividuell und softwareunterstützt konfiguriert werden. Die ausgewählten Module werden anschließend gefertigt und zum Gesamtsystem montiert, sofern die Module nicht schon vormontiert sind.

Die **Modulbildung** unterliegt den unterschiedlichen Sichtweisen der Fachdisziplinen Mechanik, Elektrik und Steuerungstechnik. Sie spiegeln sich in der Gestaltung der Maschinentechnik wider, so dass während des Konstruktionsprozesses eine **bauliche Trennung** von funktional zusammengehörenden Teilkomponenten entsteht. Die elektrischen Komponenten sind im Schaltschrank untergebracht, das Grundgestell der Maschine bildet eine Einheit, abgeschlossen durch die Maschinenabdeckung, die Nebenaggregate (z.B. Hydraulikaggregat) sind meist neben der Maschine untergebracht. Der Zusammenbau der Module aus einem Baukasten erfolgt in der Regel von „innen“ nach „außen“. Das heißt, die auf dem Maschinenbett montierten bewegungsbestimmenden Module werden ergänzt durch Prozessmodule (z.B. Spindel), Zusatzaggregate (z.B. Hydraulikaggregat), Werkzeug- und Palettenwechselsysteme, Maschinenverkleidung, Abdeckung usw. Die Energieversorgung für Elektrik, Hydraulik und Pneumatik werden fest verlegt und installiert. Damit sind viele Maschinenkomponenten miteinander starr gekoppelt. Die Steuerung wird anschließend speziell an die Maschinenkonfiguration angepasst. Dazu werden erfahrene Spezialisten für die Inbetriebnahme benötigt, die mit Hilfe von Konfigurierungswerkzeugen die Steuerungsanpassung vornehmen.

Bild 4.2 zeigt eine typische schematische Darstellung einer Werkzeugmaschine mit zentralem Schaltschrank. Zusätzlich zu den gezeichneten Schnittstellen gibt es auch mechanische Schnittstellen (z.B. Flansche) und fest installierte Schnittstellen für die Versorgung der Achseinheit mit Hilfsstoffen (z.B. Schmierstoffkreislauf, Luftdruck).

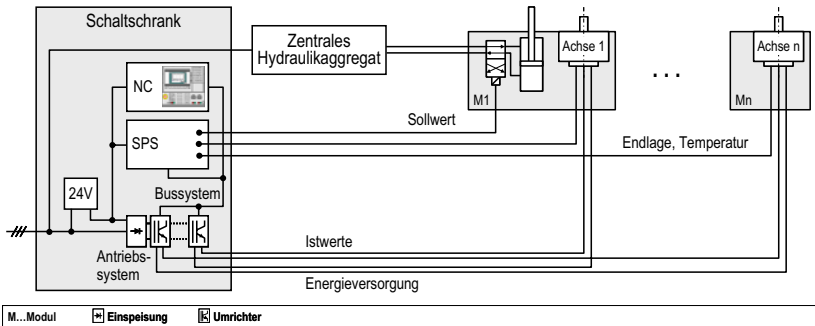


Bild 4.2: Schematische Darstellung einer typischen Verkabelung einer Werkzeugmaschine

Da die Energieversorgung durch zentrale Aggregate realisiert ist und die Steuerungskomponenten im Schaltschrank untergebracht sind, müssen viele **Verbindungen** (Kabel, Schläuche und Rohre) zur Maschine geschaffen werden. Da spätere Erweiterungen an Maschinen nicht vorhersehbar sind, muss Platz für zusätzliche Komponenten im Schaltschrank vorgehalten werden.

Aufgrund der verwendeten Schnittstellen und der das System durchziehenden Leitungen und Schläuche sind heute selbst als Baukastenmaschinen aufgebaute Werkzeugmaschinen kaum oder nur mit großem **Aufwand** veränderbar. Hilfs- und Nebenfunktionen werden überwiegend solitär an der Peripherie erzeugt, wobei die Aggregate in ihrer Leistung auf die Verbraucher abgestimmt werden. Änderungen der Art oder Anzahl der Verbraucher bei einer späteren Erweiterung werden dadurch erschwert. Insbesondere bei unterschiedlichen Maschinenkonfigurationen muss außerdem die Steuerung jedes Mal **erneut konfiguriert** werden.

4.1.2 Rekonfigurierbare Werkzeugmaschinen

Die Herausforderung im Bereich rekonfigurierbarer Werkzeugmaschinen liegt in der Definition geeigneter **Module** und deren **Schnittstellen**, damit der Austausch einzelner Module schnell erfolgen kann.

Die Erforschung und **Systematisierung** der mechanischen, elektrischen, kommunikationstechnischen **Schnittstellen von Modulen** wurde bisher allerdings lediglich aus **disziplinspezifischen Blickrichtungen** durch Definitionen von Referenzarchitekturen vorangetrieben. In der Wissenschaft gibt es daher zahlreiche Ansätze, Maschinen aus mechanischer Aufbausicht zu modularisieren und als Referenzarchitektur zu beschreiben. Darauf aufbauend wurden meist **Werkzeuge** für die Unterstützung von Konfigurationen und Rekonfigurationen ent-

worfen.

Auf internationaler Ebene wurden in den Forschungsprojekten MAREA (Study and Definition of Machining Workstation Reference Architecture) und MOSYN (Modular Synthesis of advanced Machine Tools) /83, 84/ bestehende Werkzeugmaschinenkonzepte analysiert und eine neue **modulare Referenzarchitektur** für eine Werkzeugmaschine mit Peripherieaggregaten definiert. Die Modellierung beschränkt sich auf Fräsmaschinen. **Bild 4.3** zeigt die Referenzarchitektur einer modularen Werkzeugmaschine, wie sie im EU-Projekt MOSYN erarbeitet wurde. Die Modellierung ist objektorientiert. Eine Menge von Objekten und deren Eigenschaften werden in abstrakten, physikalisch nicht existierenden Objekten (abstrakte Klassen) zusammengefasst. Subtypen erben deren Eigenschaften. Abstrakte Klassen müssen bei der Konfigurierung durch konkrete Subtypen ersetzt werden. Für die Verwendung der Modellierung in Konfigurierungswerkzeugen besitzen alle Modellelemente eine formale Beschreibung ihrer Eigenschaften. Bei Modulen sind dies z.B. Schnittstellenbeschreibungen und die physikalischen und geometrischen Eigenschaften.

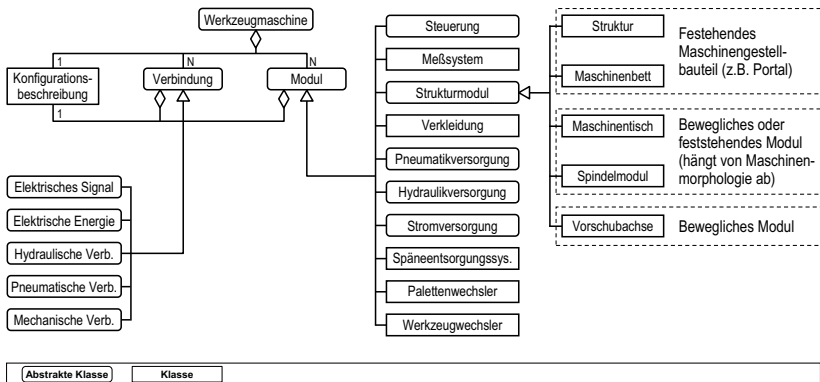


Bild 4.3: Referenzarchitektur einer modularen Werkzeugmaschine

Die MOSYN-Referenzarchitektur definiert, dass eine Werkzeugmaschine aus Modulen und Verbindungen besteht und eine **Konfigurationsbeschreibung** hat. Module und Verbindungen sind abstrakte Klassen und werden durch Subtypen konkretisiert. Die wesentlichen Module sind das Strukturmodul und die Steuerung. Die Steuerung wird weiter konkretisiert, indem sie gemäß der OSACA-Referenzarchitektur /85/ modularisiert wird. Strukturmodule liegen im Kraftfluss zwischen Werkzeug und Werkstück und setzen die Relativbewegung um. Sie teilen sich in Vorschubachse, Spindelmodul, Maschinentisch und -bett, sowie Struktur ein. Eine Struktur ist ein feststehendes Maschinengestellbauteil, wie z.B. ein Portal einer Portalfräsmaschine.

schine. Das Maschinenbett ist ebenfalls feststehend. Der Maschinentisch und das Spindelmodul können je nach Maschinenmorphologie (z.B. Ständer- oder Konsolbauweise) entweder feststehend oder beweglich sein. Vorschubachsen sind immer bewegliche Module.

Zur **simultanen Konfigurierung** von Mechanik und Steuerung wurden im EU-Projekt HIPARMS (Higly productive and reconfigurable manufacturing systems) /86, 87/ Methoden zur Erhöhung der Produktivität und Flexibilität von Fertigungssystemen in einer modularen Transferlinie erarbeitet. Aufbauend auf diesen Projekten entstanden Arbeiten die sich mit der **werkzeugunterstützten Konfigurierung** von modular gestalteten Maschinen /88/ und deren Steuerung befassen /89, 90/.

Im Rahmen des SFB 467 „Wandlungsfähige Unternehmensstrukturen für die variantenreiche Serienproduktion“ wurden in der ersten Förderperiode im Teilprojekt C2 /91, 92/ die Grundlagen für die Realisierung wandelbarer Werkzeugmaschinen geschaffen. Hierbei wurden sinnvolle **Systemgrenzen** für die Module von Werkzeugmaschinen erarbeitet, um die Funktionen festzulegen, die innerhalb eines eigenständigen Moduls vorhanden sein müssen. Des Weiteren wurden Schnittstellen entworfen, um den Austausch von Energie und Informationen über die Systemgrenzen hinweg möglichst ohne großen Aufwand zu gestalten und Wandlungsvorgänge zu vereinfachen. Basierend auf den entwickelten Schnittstellen und Normen für die Einheiten von Transferstraßen entwickelte das Institut für Werkzeugmaschinen (IfW) der Universität Stuttgart einen Baukasten für mechanische Baugruppen für rekonfigurierbare Bearbeitungssysteme (Bild 4.4) /93, 94, 95/.

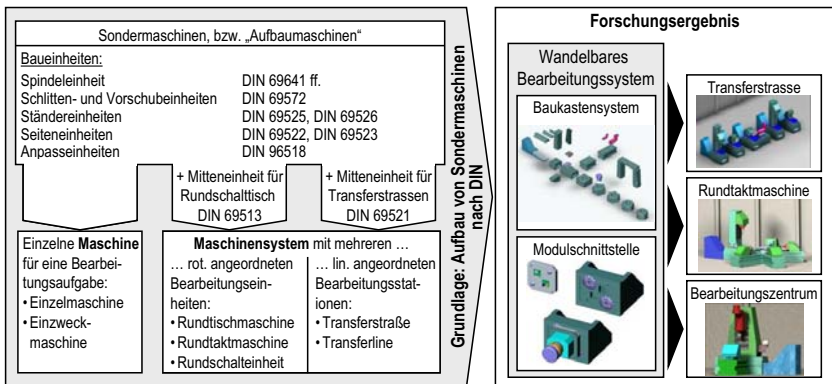


Bild 4.4: Baukastensystem für rekonfigurierbare Bearbeitungssysteme (Quelle: IfW)

4.1.3 Werkzeugunterstützte Konfiguration von rekonfigurierbaren Maschinen

In /88/ werden bestehende Strukturen von Fräsmaschinen analysiert, um mit einem top-down Vorgehen die Maschinen nach funktionalen Kriterien zu modularisieren. Aus dieser Analyse wurden dreizehn unabhängige **Basisfunktionen** abgeleitet (Bild 4.5), die hinsichtlich ihrer Integration in den Kraftfluss des Bearbeitungsprozesses in **Haupt- und Nebenfunktionen** unterschieden werden. Als Hauptfunktionen werden die Funktionen bezeichnet, die innerhalb des Kraftflusses liegen. Dazu gehören die Basisfunktionen Verfah- und Schnittbewegung, Werkzeug- und Werkstückführung. Nebenfunktionen liegen außerhalb des Kraftflusses. Dazu gehören die Basisfunktionen Werkzeug- und Werkstückbereitstellung, Prozesskühlung, Späneentsorgung, Prozessabschirmung, Energieversorgung, Aggregatekühlung, Schmierung, Steuerung/Überwachung.

Die Basisfunktionen weisen keine funktionalen Überschneidungen auf. Sie sind die Grundlage für die Struktur der **Referenzarchitektur**. Für jede der Basisfunktionen werden in der Arbeit physikalische Komponenten und deren Schnittstellen spezifiziert und in einem Konfigurierungswerkzeug abgebildet. Das Werkzeug unterstützt durch eine Konfigurierungsmethode die automatisierte Konfigurierung einer Maschine.

Basisfunktionen	Physikalische Komponenten
Hauptfunktionen (HF) HF1: Verfahrbewegung (lin./rot. Bewegung) HF2: Schnittbewegung HF3: Werkzeugführung HF4: Werkstückführung	Trägereinheit, Antriebseinheit, Übertragungseinheit, Bewegungseinheit, Meßsystem Trägereinheit, Antriebseinheit, Übertragungseinheit, Meßsystem Führungseinheit, Übertragungseinheit, Spanneinheit Spannvorrichtung
Nebenfunktionen (NF) NF1: Werkzeugbereitstellung NF2: Werkstückbereitstellung NF3: Prozesskühlung NF4: Späneentsorgung NF5: Prozessabschirmung NF6: Energieversorgung (elektrisch, hydraulisch, pneumatisch) NF7: Aggregatekühlung NF8: Schmierung NF9: Steuerung/Überwachung	Werkzeugspeicher, Werkzeugwechsler Werkstückspeicher, Werkstückwechsler Tank, Pumpe, Übertragungseinheit, Aufbereitungseinheit Späneförderer, Spänebehälter Kabine Schaltschrank, Verteilerbaugruppen, Tank, Pumpe/Kompressor, Übertragungseinheit, Aufbereitungseinheiten Eingabeeinheit, NC-Kern, SPS, Informationsbus

Bild 4.5: Basisfunktion einer Werkzeugmaschine (nach /88/)

In dieser Arbeit wird die **Steuerung** nur als eine abgeschlossene Komponente betrachtet, die nach der OSACA-Referenzarchitektur modular aufgebaut sein kann. Die Abhängigkeiten zwischen Maschinenkonfiguration und den Modulen der Steuerungskonfiguration werden nicht berücksichtigt. Das **Konfigurierungswerkzeug** beschränkt sich ausschließlich auf die Konfigurierung der mechanischen Baugruppen.

In /52/ wird ein **Modulbaukasten** für rekonfigurierbare Maschinen auf Basis von Funktionseinheiten für die Hauptfunktionen einer rekonfigurierbaren Maschine erarbeitet. Die Rekonfiguration wird durch ein **Planungswerkzeug** unterstützt, in dem die Module abgebildet sind. Es ermöglicht, die dynamischen Eigenschaften von rekonfigurierten Maschinen vorher durch Simulation abzuschätzen. Die Steuerungstechnik wird als modular und anpassbar vorausgesetzt und ist nicht Gegenstand der Arbeit. Sie wird auch nicht im Konfigurator berücksichtigt.

In /89/ wird ein **Konfigurationsmanager** für offene Steuerungen auf Basis der OSACA-Referenzarchitektur entworfen. Der Ansatz basiert auf der Idee, den **Zusammenhang** zwischen Maschinenstruktur und Steuerungsfunktionen formal zu beschreiben, um somit die Konfiguration durch ein Werkzeug zu erleichtern. Dazu werden **Referenzarchitekturen** für die Maschinen- und Steuerungskonfiguration modelliert, miteinander verknüpft und im Konfigurationsmanager abgebildet. Das Modell der mechanischen Referenzarchitektur basiert auf den Ergebnissen des EU-Projektes MOSYN (**Bild 4.3**). Das Modell für die Steuerungsreferenzarchitektur entspricht der Architektur der offenen OSACA-Steuerung. Für die Konfiguration der Steuerung muss die Maschinenkonfiguration in einer maschinenlesbaren Form beschrieben und als Datei auf der Steuerung abgelegt werden. Die vorgeschlagene **Methode zur Umkonfiguration** basiert auf dem Enabler-Konzept bei dem die Steuerungsstruktur vorkonfiguriert und die Steuerungsmodule vorparametriert sind und abhängig von der Maschinenbeschreibung aktiviert bzw. deaktiviert werden. Der Konfigurationsmanager ermöglicht die dialogunterstützte Inbetriebnahme. Die Lösung wird am Beispiel eines Werkzeugwechselsystems demonstriert. Dabei wird die Struktur des SPS-Anwenderprogramms automatisch angepasst und die NC-SPS-Schnittstelle entsprechend adaptiert. Die Lösung ist nur für **Peripherieaggregate** gültig, da die Zusammenhänge zwischen Maschinenstruktur und Steuerungsfunktionen nicht allgemeingültig und durchgängig beschrieben sind. Eine Verallgemeinerung im Rahmen der Arbeit bleibt aus. Eine Übertragung auf rekonfigurierbare Maschinen ist nicht möglich, da das Strukturteilmodell aus dem MOSYN-Projekt im Vergleich zum Steuerungsmodell zu detailliert ist. Eine **automatische Erkennung** im Sinne von Plug-and-Play und eine funktionale Klassifizierung von Maschinenmodulen ist nicht vorgesehen.

Zur Konfiguration von offenen NC-Steuerungen (OSACA) für **Transfermaschinen** wird in /90/ ein computerunterstütztes Konfigurierungswerkzeug (Computer Aided Configuration Tool – CAC) entworfen. Es basiert auf einer **modellbasierten Beschreibung** von Abhängigkeiten zwischen der Struktur der Transfermaschine und den NC-Steuerungsmodulen. Die beiden Systeme werden so modelliert, dass eine 1:1 Zuordnung zwischen Transfermodulen und Steuerungsmodulen möglich wird. Die Anpassung der Steuerung an die Module des Ferti-

gungssystems erfolgt nicht **automatisch** sondern muss mit Hilfe des CAC von einem Inbetriebnehmer vorgenommen werden. Eine automatische Erkennung von Maschinenmodulen ist nicht vorgesehen. Der Ansatz beschränkt sich auf **modulare Transferlinien** und kann nicht für Werkzeugmaschinen angewandt werden, ohne ein komplett neues Strukturmodell für Werkzeugmaschinen zu beschreiben und neue Beziehungen zwischen Steuerungsstrukturmodell und Werkzeugmaschinenmodell zu definieren.

4.1.4 Zusammenfassung der Arbeiten im Bereich rekonfigurierbarer Maschinen

In allen genannten Projekten über rekonfigurierbare Maschinen wurde daran gearbeitet, Maschinen zu modularisieren und mechanische Schnittstellen zu definieren und zu standardisieren. Eine **herstellerübergreifende Modularisierung** von Maschinen und eine Standardisierung von mechanischen Modulen und deren Schnittstellen ist allerdings bis heute nicht gelungen, da die entwickelten **Referenzarchitekturen** eine stark disziplinspezifische Ausprägung haben und hauptsächlich aus mechanischer Aufbaussicht strukturiert sind. Eine einheitliche, durchgängige mechanische und steuerungstechnische Referenzarchitektur für rekonfigurierbare Werkzeugmaschinen fehlen bisher genauso, wie Methoden zur Rekonfigurierung von Steuerungssystemen.

Werkzeuge zur Unterstützung von Rekonfigurationen beschränken sich auf die Rekonfigurierung von Maschinenmodulen. Sie unterstützen die Auswahl und Auslegung von Maschinenmodulen /52, 88/, nicht jedoch die Rekonfigurierung der Steuerung. Lediglich für rekonfigurierbare Transfermaschinen /90/ wurde ein Werkzeug zur Konfiguration der NC-Steuerung entwickelt. Das **Informationsmodell**, das als Basis für das Werkzeug dient, ist allerdings speziell für Transfermaschinen entwickelt und berücksichtigt nicht die Rekonfigurierungsfreiräume von rekonfigurierbaren Werkzeugmaschinen.

4.2 Modularisierung von Steuerungssystemen

Im wissenschaftlichen Umfeld gab es auf internationaler Ebene in Japan, den USA und Europa drei wesentliche Ansätze zur **Modularisierung** und **Strukturierung** von Steuerungssystemen und Schaffung einer **Referenzarchitektur** für offene, konfigurierbare Steuerungssysteme /96/. Dies waren Arbeiten der Interessensgemeinschaft OSEC /97/ und später JOP /98, 99/ in Japan, von der Interessensgemeinschaft OMAC /100, 101/ in Nordamerika und dem OSACA-Projektconsortium /70, 102/ in Europa. Das Ziel aller Aktivitäten war die Spezifikation einer Referenzarchitektur für Steuerungssysteme.

Die Aktivitäten der Interessensgemeinschaft OMAC und die Aktivitäten der JOP befassten sich hauptsächlich mit der **Vereinheitlichung** der Steuerungsschnittstellen innerhalb eines

Produktionssystems aus Anwendersicht /103/. Beide Aktivitäten vernachlässigen den internen Aufbau und das Kommunikationssystem der Steuerung. Bisher sind keine Produkte auf Basis von OMAC- oder JOP-Spezifikationen entstanden.

In Europa wurde im Rahmen des Projektes OSACA eine **Referenzarchitektur** für ein offenes, objektorientiertes und konfigurierbares Steuerungssystem definiert /69, 85, 104/. Zentraler Bestandteil der OSACA-Idee ist eine **modulare Systemplattform** mit einem Kommunikationssystem, das eine definierte Schnittstelle (API – Application Programming Interface) für die **Architekturobjekte** der modularen Steuerungssoftware und definierte, einheitliche Kommunikationsmechanismen hat. Darüber hinaus verfügt das Steuerungssystem über ein Konfigurationslaufzeitsystem. Es ermöglicht eine automatische Konfigurierung der Steuerung beim Hochlauf durch Interpretieren einer textbasierten Konfigurationsdatei /48, 70, 71, 105/.

Trotz der zahlreichen Beteiligung von Industriefirmen konnten sich die OSACA-Ideen nur begrenzt durchsetzen. Es gibt im industriellen Einsatz keine Steuerung, die sich im Sinne von OSACA als offen bezeichnen lässt. Gründe für die geringe Akzeptanz der OSACA-Offenheit liegen vor allem in der Furcht der Steuerungshersteller, durch die Offenlegung von Schnittstellen Know-how preiszugeben und durch die Austauschbarkeit von Komponenten, das eigene System für Software konkurrierender Firmen zu öffnen und dadurch Markt- und Gewinnanteile zu verlieren. Außerdem gelang es nicht, das spezifizierte Kommunikationssystem echtzeitfähig umzusetzen.

Im europäischen OSACA-Nachfolgeprojekt OCEAN /106, 107/ wurde versucht, diese Akzeptanzschwierigkeiten zu überwinden. Neben der Restrukturierung der Referenzarchitektur mit dem Ziel, die in OSACA gewählten Architekturobjekte auf wenige abgeschlossene Softwarekomponenten zu reduzieren, war die Schaffung einer echtzeitfähigen Kommunikationsschicht ein wesentlicher Schwerpunkt der Arbeiten. Dazu wurde die echtzeitfähige Kommunikationsplattform auf der Open-Source-Implementierung von RT-CORBA (Real-Time Corba) /108/ verwendet und auf das Open-Source-Betriebssystem Linux mit Echtzeiterweiterung zurückgegriffen.

Im deutschen OSACA-Nachfolgeprojekt HÜMNOS wurden Architekturobjekte für die OSACA-Plattform spezifiziert und ein grafisch unterstütztes **Softwarewerkzeug** für die **Konfigurierung** der OSACA-Plattform entwickelt. Dazu wurden baukastenbasierte Methoden für eine modellgestützte Erzeugung der Konfigurationsdateien entwickelt /109, 110/.

Alle Arbeiten im Bereich Modularisierung von Steuerungen beschränken sich in ihrer Sichtweise auf das numerische Steuerungssystem. Das heißt, eine Selbstadaption ist nicht möglich, da die Betrachtung des mechanischen Aufbaus der Werkzeugmaschine unberücksichtigt bleibt

und der Konfigurierungsvorgang des Eingriffs des Anwenders bedarf. Es findet keine automatische Erkennung neu zu konfigurierender Steuerungsfunktionen statt.

4.3 Adaption von Steuerungssystemen

4.3.1 Adaption von NC-Steuerungen

Bei der Inbetriebnahme wird die Funktion der NC-Steuerung durch **Konfigurierung** an die spezifischen Randbedingungen der Maschine und des Prozesses angepasst. Die Anpassung der NC erfolgt über **Maschinendaten**. Grundlegende wissenschaftliche Arbeiten zur Schaffung von konfigurierbaren Steuerungen auf Basis einer modularen Steuerungsarchitektur, die sich aus Funktionalen Einheiten zusammensetzt, wurden von /48/ und /111/ durchgeführt. In /111/ werden die Schnittstellen und die Konfigurationsräume am Beispiel der zentralen Funktionalen Einheit Geometriedatenverarbeitung eines NC-Kerns definiert.

Die Arbeiten von /48/ zeigen, wie die Funktionalität einer auf der OSACA-Plattform basierenden NC-Steuerung über Konfigurierung und Parametrierung **dynamisch** beim Steuerungshochlauf **angepasst** werden kann. Dazu werden Adaptionsschnittstellen der Software und ein **Konfigurationslaufzeitsystem** definiert. Bild 4.6 zeigt ein Softwaremodell einer solchen dynamisch konfigurierbaren NC-Steuerung. Wesentlicher Bestandteil ist ein Konfigurationslaufzeitsystem, das beim Steuerungshochlauf eine ASCII-basierte Konfigurierungsliste interpretiert und die erforderlichen Funktionalen Einheiten instanziiert und parametriert, die Kommunikationsverbindungen untereinander herstellt und initialisiert.

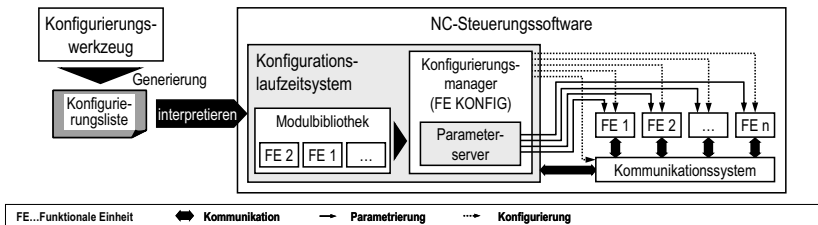


Bild 4.6: Softwaremodell einer dynamisch konfigurierbaren NC-Steuerung

Mit einer in der Arbeit entwickelten Sprache, die an die Sprache strukturierter Text der IEC 61131-3 /37/ angelehnt ist, kann die NC-Steuerungskonfiguration in einer Konfigurierungsliste **formal beschrieben** werden. Sie enthält die Beschreibung der Funktionalen Einheiten (FE) der Steuerung, deren Kommunikationsverbindungen und ihre Parameter. Über einen Parameterserver, der Teil des Konfigurationslaufzeitsystems ist, fordern FE ihre erforderlichen Parameter an. Die Parametrierungsdaten werden eingeteilt in achsbezogene Parame-

ter zur Anpassung der Funktion, Parameter zur Anpassung an die gerätetechnische Umgebung der Steuerung und Parameter zur Zuordnung von steuerungsinternen Achsen und Maschinenachsen.

Durch die Umsetzung der Grundlagen aus /111/ und den Methoden von /48/ ist eine dynamische Konfigurierung heutiger NC-Steuerung beim Steuerungshochlauf über das Einlesen von Konfigurationsdaten möglich. Die Anpassung der Parameter erfolgt über Parametrierungsdaten (Maschinendaten).

Die Maschinendaten können dabei direkt in Listenform, oder wesentlich komfortabler mit Windows-orientierten Inbetriebnahme-Werkzeugen eingegeben werden /112, 113, 114/. Die Werkzeuge verfügen über themenorientierte Eingabemasken mit denen sich die Anzahl der Parameter einschränken lassen. Das Editieren der Werte wird somit übersichtlicher.

Bei der Inbetriebnahme wird zunächst die Kanalstruktur definiert, indem die Achsen angelegt und zu einem NC-Kanal gruppiert werden. Dabei wird für jede gemeinsam interpolierte Achsgruppe eine Transformation festgelegt. Zusätzlich werden Spindeln und Zusatzachsen angelegt. Anschließend wird die Steuerungskonfiguration parametrierung und an die Maschineneigenschaften angepasst. Dazu werden die Achsen und ihre Kommunikationsschnittstellen, Spindeln, Handräder, Messsysteme, Regelkreise, Transformationen, Geschwindigkeits- und Beschleunigungsgrenzwerte und Interpolationsarten parametrierung.

4.3.2 Adaption von speicherprogrammierbaren Steuerungen

SPS-Steuerungen sind aufgrund ihrer freien Programmierbarkeit individuell anpassbar. Der Aufwand für die Anpassung kann sich durch **baukastenbasierte** Programmerstellung und eine Programmstruktur, die über Funktionsbausteine mit definierten Schnittstellen eine Hardwareabstraktion schafft, erheblich reduzieren. Dazu werden sogenannte Softwarebusse benutzt, die die Funktionsbausteine untereinander und mit der Hardware verbinden (Bild 4.7). Sie haben die Funktion eines Kommunikationssystems. Die Anpassung erfolgt dann durch Auswahl von vorgefertigten Programmbausteinen in einem baukastenbasierten Projektierungswerkzeug und automatisierter Genierung des Quellcodes für die SPS-Zielpattform /115, 116/.

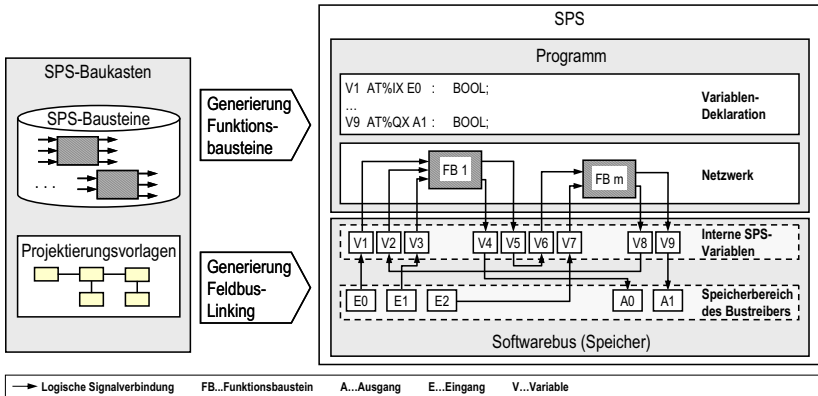


Bild 4.7: Generierung von SPS-Programmen

Wenn das SPS-Programm so strukturiert ist, dass kein direkter Zugriff auf den Speicher erfolgt, sondern dies über eine Schnittstelle (Softwarebus) realisiert ist, dann reduziert sich der Anpassungsaufwand auf die Verknüpfung von Hardwareadressen des Speichers mit den internen Variablen des SPS-Programms (Feldbus-Linking). Dies kann, wenn bestimmte Konventionen (meist bei der Namensgebung von Variablen) eingehalten werden, auch automatisch durch entsprechende Projektierungswerkzeuge generiert werden /79/.

4.3.3 Adaption des Interaktionssystems

Neben Konfigurierungswerkzeugen für Benutzungsoberflächen mit Masken und vorkonfigurierten Einstellungen bieten Steuerungshersteller auch Entwicklungswerkzeuge mit Bibliotheken für die Gestaltung von kundenindividuellen Oberflächen durch Neuprogrammierung an /117/. In der Forschung gibt es Ansätze für die automatisierte **Generierung** von Benutzungsoberflächen /118/ und selbstkonfigurierenden Interaktionssystemen, um die Benutzungssysteme an unterschiedliche Werkzeugmaschinenarten, beziehungsweise dazugehörige Steuerungen automatisch anzupassen /119/. Zur systematischen Erstellung von Interaktionssystemen für verschiedene Steuerungen und Steuerungskonfigurationen wird in /120/ ein **selbstkonfigurierendes Interaktionssystem** entworfen (Bild 4.8). Es passt sich bei jedem Hochfahren selbständig an die Steuerung an. Dazu müssen die Funktionalität und Konfiguration der NC-Steuerung bekannt und die Ergonomie durch Gestaltungsrichtlinien (Styleguides) vorgegeben sein /121/. Ein erneutes Anpassen bei jeder Rekonfigurierung entfällt damit.

Bild 4.8 zeigt die zeitliche Abfolge der Funktionsweise eines selbstkonfigurierenden Interaktionssystems. Mit dem Start des Interaktionssystems wird die zugehörige Steuerung auf ihre

aktuelle Konfiguration abgeprüft. Anschließend werden mit dieser Information Abbildungsregeln für die Zuordnung von Steuerungssystemzuständen und Anzeigeelementen angewandt. Aus ihnen ergibt sich schließlich eine Ausprägung des Interaktionssystems, die genau zur aktuellen Steuerungskonfiguration passt und den Vorgaben des Styleguide entspricht.

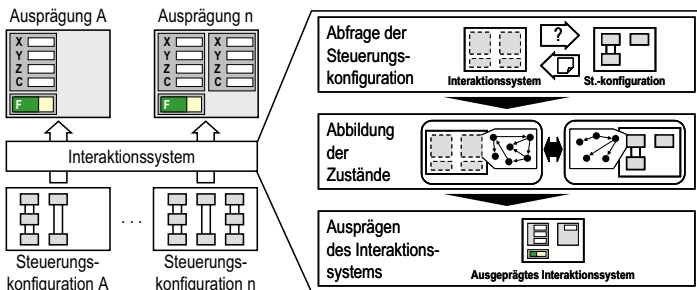


Bild 4.8: Selbstkonfigurierenden Interaktionssystems (nach /120/)

Solange die Steuerungskonfiguration offengelegt und in geeigneter Form beschrieben ist und eine offene Kommunikationsschnittstelle für den Zugriff auf diese Daten bereitgestellt wird, kann mit diesem Konzept eine Benutzungsoberfläche automatisch konfiguriert und an die spezifische Steuerungskonfiguration angepasst werden. Am Beispiel einer offenen NC-Steuerung konnte gezeigt werden, dass ein selbstkonfigurierendes Interaktionssystem für verschiedene Steuerungskonfigurationen verwendet werden kann und der immer wiederkehrende Aufwand für die Anpassung reduziert wird /107/.

4.3.4 Werkzeugunterstütztes Konfigurieren von Steuerungssystemen

Unterstützung bei der Inbetriebnahme wird inzwischen in Form von Programmen zur Parametereingabe, zur automatisierten Reglereinstellung und zur Aufzeichnung von Achsiswertern angeboten. Die Konfigurierung wird meist durch herstellerspezifische **Softwarewerkzeuge** unterstützt /112, 114/. Sie unterstützen allerdings nur bei der Dateneingabe und übernehmen die Datenaufbereitung und -darstellung, sowie die Datenübertragung auf die Steuerung (Schreiben der Konfiguration auf das Zielgerät).

Obwohl die meisten Steuerungshersteller Softwarewerkzeuge mit **Eingabemasken** für die Konfigurierung und Parametrierung anbieten, kann jedoch insbesondere die Eingabe der Parameter schnell unübersichtlich werden, da NC-Steuerungen bis zu mehreren hundert Parametern anbieten /122, 123, 124, 125/. Hinzu kommt, dass bei der Konfigurierung und Parametrierung die Bedeutung der Parameter und deren gegenseitige Beeinflussung und Zuordnung

meist in Handbüchern nachgeschlagen werden müssen. Erleichterung schaffen nur Tooltip-Texte und online Hilfesysteme /112/. Diese Maßnahmen können aber keine funktionalen Abhängigkeiten abbilden und Zusammenhänge darstellen. Problematisch und zeitaufwändig ist zudem die Eingabe und Änderung redundanter Informationen an unterschiedlichen Stellen.

Für die **Erzeugung** einer gültigen **Konfiguration** ist es notwendig, dass die Kombinationsmöglichkeiten vorgegeben und durch Konfigurierungswerkzeuge überprüfbar sind. Dies ist allerdings in heutigen Systemen nicht der Fall, so dass erst zur Laufzeit während des Hochlaufs der Steuerung die Gültigkeit der Konfiguration überprüft wird. Fehlerhaft parametrierte Steuerungen werden daher erst zur Laufzeit überprüft, so dass ein iteratives Korrigieren der fehlerhaften Parameter nach dem Trial-and-Error-Prinzip an der Maschine üblich ist.

Die Konfigurierung und Parametrierung von NC-Steuerungen erfordert daher, abhängig vom eingesetzten Steuerungstyp und Steuerungshersteller, langjähriges Expertenwissen. Aufgrund der jeweils herstellerspezifischen Konfigurierung und Parametrierung fehlt es außerdem an Werkzeugen, die eine **einheitliche Form** der Konfigurierung und Parametrierung unterstützen /48/. Das Wissen über die korrekte Erstellung einer NC-Steuerungskonfiguration ist daher nicht auf andere NC-Steuerungen übertragbar. Das bedeutet, dass jede NC-Steuerung eines anderen Herstellers eine komplett neue und andere Konfigurierung und Parametrierung erfordert.

In den Projekten HÜMNOS /63/ und MoWiMa /126/ wurden **Grundlagen** für Konfigurierungswerkzeuge für offene Steuerungen entwickelt. Wesentlicher Bestandteil der Forschungsprojekte und Forschungsarbeiten war eine **baukastenbasierte, objektorientierte Konfigurierungsmethode** (Bild 4.9).

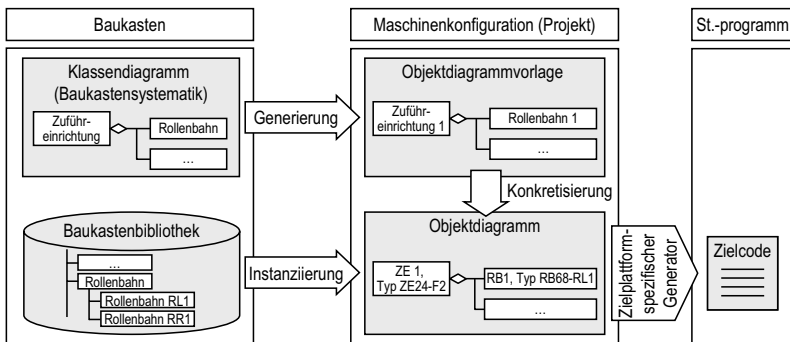


Bild 4.9: Baukastenbasierte Konfigurierungsmethode

Der **Baukasten** besteht aus einer **Bibliothek** von verfügbaren Komponenten sowie der **Baukastensystematik**. Sie beschreibt, wie die einzelnen Komponenten kombiniert werden können. In der Bibliothek werden die verfügbaren Komponenten als Klassen strukturiert gespeichert. Zur Strukturierung werden die Prinzipien der Abstraktion und Vererbung verwendet.

Die **Baukastensystematik** stellt einen Bauplan für bestimmte Maschinentypen dar und beschreibt projektneutral den Grundaufbau einer Fertigungseinrichtung und das Zusammenwirken der einzelnen Komponenten. Die Beschreibung erfolgt mit **UML-Klassendiagrammen**. Als **Beziehungstypen** werden die „besteht-aus“ (Aggregation) Beziehung verwendet sowie zwei neue Beziehungstypen „alternativ“ (entweder...oder...) und „restriktion“ (wenn...dann...), die in der Praxis des Maschinenbaus häufig erforderlich sind, eingeführt.

Aus dem Baukasten werden Objektdiagrammvorlagen erzeugt. Sie sind noch projektneutral und enthalten nur abstrakte Platzhalter (**Klassen**) für Komponenten aus dem Baukasten. Die Objektdiagrammvorlage entspricht einem projektneutralen Bauplan. Durch Instanzieren und Ersetzen der Platzhalter durch vorgefertigte Bausteine aus der Bibliothek wird aus der Objektdiagrammvorlage ein Objektdiagramm mit konkreten Komponenten (**Instanzen**). Das Objektdiagramm beschreibt eine projektspezifische konkrete Konfiguration. Abhängig von der Zielplattform lassen sich dann daraus SPS-Steuerungsprogramme und Konfigurierungslisten für offene Steuerungen generieren /109, 110, 127/.

Für maschinennahe SPS-Steuerungsfunktionen bei Fertigungseinrichtungen wurde die baukastenbasierte Konfigurierungsmethode weiterentwickelt, indem ein **Informationsmodell** erarbeitet wurde, das aus einem Teilmodell für die Maschinenstruktur und einem Teilmodell für die Steuerungssoftwarestruktur besteht (**Bild 4.10**) /128/. Dazu werden die Steuerungsfunktionen und die bauliche Struktur nach einer objektorientierten Analyse so strukturiert, dass sie zueinander passen, d.h. die SPS-Softwarestrukturen sind dann adäquat zur Maschinen- und Anlagenstruktur aufgebaut. Für beide Strukturen werden Partialmodelle definiert. Ihre Zusammenhänge werden in einem Baukasten als Systemklassendiagramm für Fertigungseinrichtungen abgebildet.

Es besteht aus den zwei Partialmodellen für das **Anlagenstrukturmodell** und das **Steuerungsmodell**. Aufgrund dieser formal abgebildeten Beziehungen können Generatoren aus der baulichen Struktur SPS-Steuerungsfunktionen ableiten, die entweder implementiert sind oder einen Coderumpf mitbringen und implementiert werden müssen. Damit wird bei der Konfigurierung der Zusammenhang zwischen Maschinenkonfiguration und Steuerungskonfiguration berücksichtigt und formal beschrieben.

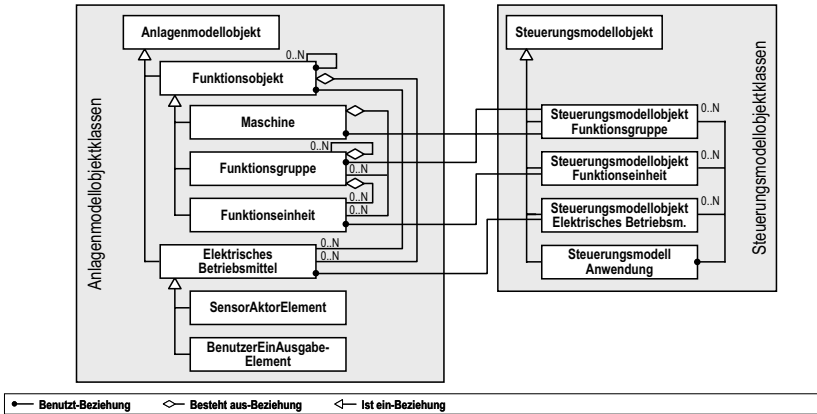


Bild 4.10: Systemklassendiagramm für den Anwendungsbereich der maschinennahen Steuerungsfunktionen an Fertigungseinrichtungen

4.3.5 Zusammenfassung der Arbeiten im Bereich Adaption von Steuerungssystemen

NC-Steuerungen haben **Mechanismen zur Anpassung** von außen über Konfigurationsdaten /48/. Dazu werden herstellerspezifische Werkzeuge eingesetzt. Sie berücksichtigen nur eingeschränkt Querbeziehungen zwischen Parametern. Zusammenhänge zwischen Maschinenkonfiguration und Steuerungskonfigurationen werden nur für Transfermaschinen /90/, nicht für Werkzeugmaschinen modelliert. Für die Konfiguration von NC-Steuerungen gibt es daher keine durchgängige Methodik. Das Vorgehen ist individuell und hängt vom Expertenwissen des Inbetriebnehmers ab. Er muss die Verknüpfung zwischen Mechanik und notwendigen Softwarestrukturen und Parametern als Erfahrungswissen im Kopf haben (mentales Modell).

Für die **Adaption des Interaktionssystems** gibt es Lösungen, die sich automatisch an die NC-Steuerungskonfiguration anpassen /120/. Die Arbeiten beschränken sich allerdings auf Interaktionssysteme und gehen nicht auf die Adaption der NC-Steuerung ein.

Für die automatisierte **Anpassung der SPS-Programme** wurden Generierungsmethoden entwickelt. Damit lassen sich SPS-Programme mit baukastenbasierten Werkzeugen in einem vorgegeben Rahmen vollständig generieren /128/.

Für die **Konfiguration von offenen NC-Steuerungen** wurden baukastenbasierte Methoden und Werkzeuge entwickelt /127/. Diese berücksichtigen allerdings nicht die funktionalen Zusammenhänge zwischen Maschinen- und Steuerungskonfiguration. Lediglich in /89/ werden zur automatisierten, werkzeuggestützten Konfiguration von offenen Steuerungen funktio-

nale Zusammenhänge in einem Konfigurationsmanager abgebildet. Die Arbeit beschränkt sich letztlich allerdings auf die automatisierte Anpassung von SPS-Anwenderprogrammen. Die Übertragung dieser Mechanismen auf die Konfigurierung der NC-Steuerung wird nicht gezeigt.

Insgesamt zeigt sich, dass die **Konfigurierung** und **Rekonfigurierung** von **NC-Steuerungen** bisher nicht methodisch unterstützt wird (Bild 4.11). Die Arbeiten im Bereich **Konfigurierungswerkzeuge** fokussieren sich bisher stark auf die Methodik zur Konfigurierung und Generierung von SPS-Anwenderprogrammen. Da geeignete Ansätze zur formalen Beschreibung der Abhängigkeiten zwischen Maschinen- und Steuerungskonfiguration fehlen, gibt es außerdem keine Softwarewerkzeuge zur Unterstützung der NC-Steuerungskonfigurierung. Daher wird heutzutage bei einer Umkonfigurierung der Maschine die Steuerung neu konfiguriert.

	Karler C. (ISM, Stuttgart)	Ulrich G. (WZL, Aachen)	Bauer G. (ISM, Stuttgart)	Las R. (ISM, Stuttgart)	Geppert M. (ISM, Stuttgart)	Mehaus J. (WZL, Aachen)
	/48/	/90/	/120/	/128/	/127/	/89/
Selbstständige Adaption des Interaktionssystems	○	○	●	○	○	○
Adaption der NC (Mechanismen in der Steuerung)	●	○	○	○	○	○
Adaption der NC (Methode zur Konfigurierung)	○	●	○	○	●	○
Berücksichtigung der Abhängigkeiten zw. Maschinen- und Steuerungskonfiguration	○	◐	○	○	○	◐
Selbstständige Adaption der NC	○	○	○	○	○	◐
Selbstständige Adaption der SPS (Methode)	○	○	○	○	○	●
Generierung von SPS-Programmen (Methode)	○	○	○	●	○	○

Bild 4.11: Bewertung der Arbeiten im Bereich Konfigurierung von Steuerungssystemen

4.4 Informationsmodellierung

Zur Abbildung von Objekten der Realität in Baukastensystemen werden **strukturierte Informationsmodelle** eingesetzt. In der Praxis ist keine einheitliche Strukturierung bekannt. Existierende Strukturierungsprinzipien orientieren sich an **fertigungs-** und **montagetechnischen, gerätetechnischen** oder **softwaretechnischen** und **funktionalen Sichtweisen** (Bild 4.12). Baukästen sind daher immer anwendungsfallspezifisch erstellt und für eine Domäne definiert.

Zur Unterstützung der Wiederverwendung wird in der Softwaretechnik von der Object Management Group (OMG) /62/ eine allgemeingültige Architektur von Informationsmodellen mit

vier **Abstraktionsebenen** als Meta Object Facility (MOF) standardisiert /129/.

Die **Objekt-Ebene** (M0-Ebene) beschreibt konkrete Objekte der Realität mit konkreten Daten. Beschreibungsmittel können Objektdiagramme der UML-Notation sein /60, 61/.

Die **Modell-Ebene** (M1-Ebene) definiert die Objekte und Struktur der Objekt-Ebene. Damit beschreibt sie die anwenderspezifische Ausprägung eines Modells der M2-Ebene. Beschreibungsmittel auf der M1-Ebene können Klassendiagramme der UML-Notation sein.

Die **Meta-Ebene** (M2-Ebene) definiert, wie Modelle aufgebaut und strukturiert sind. Die zur Beschreibung dieses Meta-Modells verwendeten Klassen werden als Systemklassen bezeichnet. Das Systemklassen-Diagramm legt durch die Systemklassen und ihre Beziehungen untereinander die Grundstruktur für die Modelle der M1-Ebene fest. Beschreibungsmittel kann die UML-Notation sein.

Die **Meta-Meta-Ebene** (M3-Ebene) definiert die Beschreibungsmittel für Meta-Modelle. Damit legt die M3-Ebene die Semantik und Syntax eines Beschreibungsmittels anwendungsfallsspezifisch fest. Bei der bekannten UML-Notation sind dies die Beschreibungselemente Klasse und Beziehungsart.

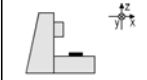
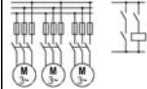
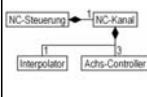
Fachdisziplin	Sichtweise	Strukturierungselemente	Beschreibungsmittel	Strukturierungskriterium
Mechanik 	Montageorientiert	<ul style="list-style-type: none"> - Baugruppe - Modul - Elemente, Normteile 	<ul style="list-style-type: none"> - Stückliste - Konstruktionszeichnung 	<ul style="list-style-type: none"> - Bauart - Freiheitsgrad - Dyn. Eigenschaften
Elektrik 	Geräteorientiert	<ul style="list-style-type: none"> - Geräte (Sensor, Aktor, Ein-/Ausgabebaugruppe) - Komponenten 	<ul style="list-style-type: none"> - Schaltplan - Konstruktionszeichnung für Schaltschrank 	<ul style="list-style-type: none"> - Anzahl I/O - Anzahl/Art Antriebe - Einbauart /-ort - Leistung
Steuerungstechnik 	Funktional	<ul style="list-style-type: none"> - Funktionsbaustein - Funktionale Einheiten - Softwarekomponenten 	<ul style="list-style-type: none"> - UML (Objekt-, Klassendiagramm) - Zustandsdiagramm, Petri-Netz 	<ul style="list-style-type: none"> - Zykluszeit - Schnittstelle (Semantik, Syntax) - Datentyp - Kommunikation

Bild 4.12: Unterschiedliche Sichtweisen und Strukturierungsprinzipien für Baukastensysteme

Da die Ebenen der MOF-Architektur sehr allgemeingültig sind, befassen sich im Bereich Fertigungstechnik einige Arbeiten mit der **anwendungsfallsspezifischen Anpassung** von Informationsmodellen /58, 57, 128, 130/. **Bild 4.13** zeigt rechts ein Informationsmodell für die Abbildung von Fertigungssystemen. Es basiert auf einem semantischen Netz, das auf der Meta-Meta-Ebene benutzt wird, um die Elemente der Meta-Ebene festzulegen.

Zur **Vereinheitlichung unterschiedlicher Sichtweisen** werden in /57, 58/ Ansätze vorgestellt, um in einem durchgängigen Informationsmodell den Zusammenhang zwischen Teilmo-

In /130/ wird die Idee des Föderal-Projektes weiter entwickelt und ein **generischer Ansatz** für die baukastenbasierte Erstellung von Fertigungseinrichtungen auf Basis eines **Baukasteninformationsmodells** erarbeitet. Dazu wird das Informationsmodell des Baukastens in vier Abstraktionsebenen gegliedert (Bild 4.14).

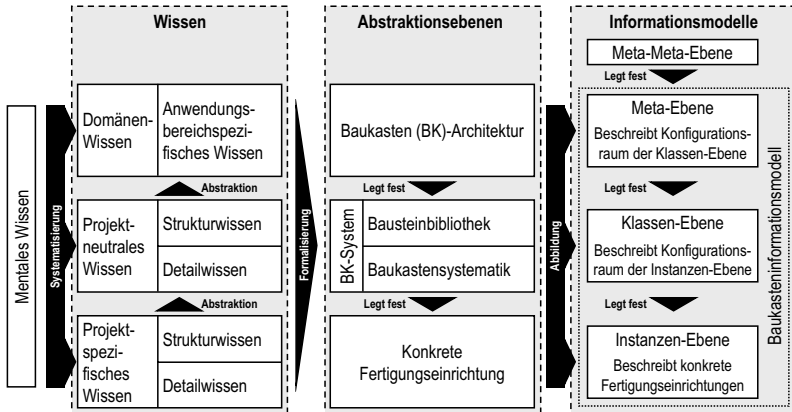


Bild 4.14: Baukastenbasierte Informationsmodellierung und Abstraktionsebenen eines allgemeingültigen Informationsmodells für adaptierbare Informationssysteme

Die **Instanzen-Ebene** formalisiert Betrachtungseinheiten konkreter Fertigungseinrichtungen durch Strukturobjektinstanzen (SOI) und deren Beziehungen zueinander. Auf der **Klassen-Ebene** wird das Baukastensystem durch die Verallgemeinerung der SOI zu Strukturobjekt-klassen (SOK) und Beziehungen zwischen SOK beschrieben. Die **Meta-Ebene** dient der Formalisierung der Baukastenarchitektur durch Strukturobjektmetaklassen und deren Beziehungen. Die **Meta-Meta-Ebene** ist ein Metamodell für die Meta-Ebene. Sie definiert Regeln für den Aufbau von Baukastensystemen, um Baukästen generisch beschreiben zu können.

4.5 Bewertung bestehender Arbeiten und Konzepte

Bisherige Forschungsarbeiten, dargestellt in Bild 4.15 und Bild 4.16, sind **fachdisziplinspezifisch beschränkt** und betrachten entweder nur die Modularisierung der Steuerung (Diss. Sperling, Projekt JOP, OMAC, OSACA) /71, 99, 100, 102/ oder der Maschine (Projekt MAREA u. MOSYN) /83, 84/. Arbeiten die sich mit der **werkzeugunterstützten Konfiguration** von Maschinen oder Steuerungen und der damit verbundenen Informationsmodellierung beschäftigen sind ebenfalls fachdisziplinspezifisch und stützen sich entweder auf Maschinenreferenzarchitekturen (Projekt HIPARMS) /86, 135/, die unter maschinenbaulichen

Aspekten erstellt wurden, oder auf Steuerungsreferenzarchitekturen (Projekt HÜMNOS) /63/, die unter softwaretechnischen Aspekten erstellt wurden.

In Forschungsarbeiten entwickelte **Informationsmodelle und baukastenbasierte Konfigurierungsmethoden** als Grundlage für Werkzeuge beschränken sich einseitig entweder auf die Konfigurierung und Modularisierung von Maschinen (Diss. Michaelis u. Böger) /52, 88/ und Anlagen (Diss. Ullrich) /90/, die Erzeugung von SPS-Programmen (Diss. Lutz) /128/ für Fertigungseinrichtungen oder die Konfigurierung von NC-Steuerungen (Diss. Daniel, Arbeit Seyfarth) /48, 127/. Der Zusammenhang zwischen mechanischer Maschinenkonfiguration und Steuerungskonfiguration steht in keiner der Arbeiten im Mittelpunkt und bleibt ungelöst. Eine mechatronische Methode zur Konfigurierung von Steuerungen, welche diesen Zusammenhang berücksichtigt, fehlt.

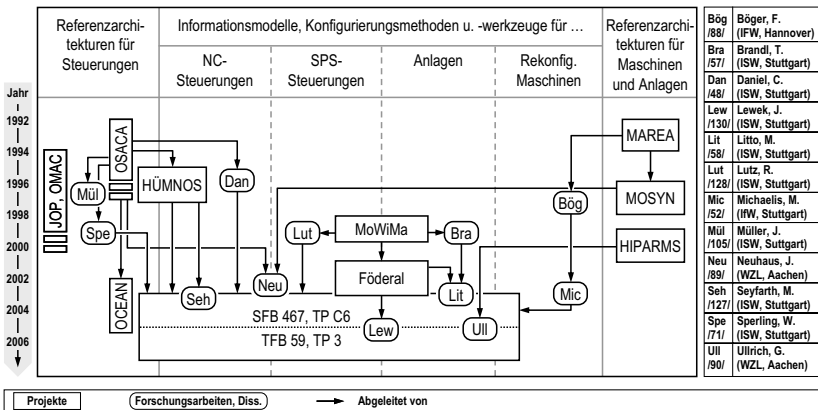


Bild 4.15: Übersicht über Projekte und Forschungsarbeiten zum Thema Modularisierung und Konfigurierung von Maschinen und Steuerungen

Einige Arbeiten berücksichtigen **fachdisziplinspezifische Zusammenhänge**, indem die Anlagenkonfigurierung zusammen mit der SPS-Programmentwicklung betrachtet wird (Projekt MoWiMa, Diss. Lutz) /126, 128/, und entwerfen Methoden zur Generierung fachdisziplinübergreifender (mechatronischer) Entwurfsunterlagen aus Baukastensystemen (Diss. Lewek, Projekt Föderal) /130, 131/. Informationsmodelle, die als Basis für Konfigurierungswerkzeuge dienen, berücksichtigen nur die Softwarestruktur oder sind speziell für die Konfigurierung von Fertigungseinrichtungen modelliert /110/. Es konnte bisher nicht gezeigt werden, ob und wie sich diese Methoden auf die Konfigurierung von NC-Steuerungen übertragen lassen.

Arbeiten, die sich mit der **Informationsmodellierung** befassen, entwerfen Erstellungssysteme-

matiken und Modelle für die Verknüpfung von maschinenbaulichen Strukturbaugruppen mit Dokumenten als Basis für Anlagen- und Störungsinformationssysteme (Diss. Brandl u. Litto /57, 58/. Sie beschreiben dabei Zusammenhänge zwischen Maschinenstruktur und Steuerungsfunktion, beschränken sich allerdings auf ihre Anwendungsfälle und berücksichtigen nicht die Konfigurierung von Steuerungen.

	Brandl T. (GSW, Stuttgart)	Reger F. (IWI, Hannover)	Daniel C. (GSW, Stuttgart)	Lewert J. (GSW, Stuttgart)	Litt M. (GSW, Stuttgart)	Lutz R. (GSW, Stuttgart)	Mechaus J. (WZL, Aachen)	Michaels M. (IWI, Stuttgart)	Müller J. (GSW, Stuttgart)	Schwahn M. (GSW, Stuttgart)	Spangler W. (GSW, Stuttgart)	Wirth G. (WZL, Aachen)
	/57/	/88/	/48/	/130/	/58/	/128/	/89/	/52/	/105/	/127/	/71/	/90/
Konfigurierungsmechanismen in der Steuerung	○	○	●	○	○	○	●	○	○	●	●	●
Methodik zur Selbstadaption	○	○	○	○	○	○	○	○	○	○	○	◐
Formale Abbildungen des mental vorhandenen Expertenwissens	●	◐	○	●	●	◐	●	○	○	◐	○	○
Modularer Aufbau der Steuerungssoftware	◐	○	●	○	◐	●	●	○	●	●	●	●
Modularer Aufbau der Maschine	●	●	○	◐	●	●	●	○	○	○	○	●
Berücksichtigung der Abhängigkeiten zw. Maschinen- u. Steuerungskonfiguration	●	○	○	○	●	●	◐	○	○	●	○	◐

Bild 4.16: Abschließende Bewertung relevanter Arbeiten

Der in /89/ beschriebene Ansatz zur **Umkonfigurierung von Werkzeugmaschinen** basiert als einziger auf zwei modellierten und miteinander verknüpften Referenzarchitekturen. Die Verknüpfung der Teilmodelle ist allerdings nicht allgemeingültig sondern auf das exemplarische Modul Werkzeugwechsler zugeschnitten. Der Ansatz ist daher nur für Peripherieaggregate gültig und lässt sich nicht auf die NC-Steuerungskonfigurierung übertragen. Darüber hinaus ist das Teilmodell der Mechanik für die werkzeugunterstützte Konfigurierung von Maschinen und daher sehr stark unter mechanischen Gesichtspunkten erstellt worden. Es eignet sich deshalb nicht für eine mechatronische Betrachtungsweise. Dies zeigt sich an der, im Vergleich zum Steuerungsreferenzmodell, detaillierteren Modellierung der mechanischen Bauteile und deren Schnittstellen.

Erst mit der Einrichtung des SFB 467 /136/ und dem anschließenden TFB 059 /137/, wurde der **Zusammenhang** zwischen modularer **Maschinenreferenzarchitektur** und **Steuerungsreferenzarchitektur** in den Mittelpunkt der Forschungsarbeiten gestellt. Die Ergebnisse dieser Forschungsarbeit sind Grundlage der vorliegenden Arbeit und werden daher noch nicht als Stand der Technik betrachtet.

4.6 Zusammenfassung und Zielsetzung der Arbeit

Moderne Werkzeugmaschinen führender und innovativer Maschinenhersteller sind inzwischen modular aufgebaut. Dazu haben etliche Forschungsprojekte beigetragen /52, 83, 84, 86, 87, 88, 91, 92, 93, 94, 135, 138, 139/. Die Module sind unternehmensintern standardisiert, so dass sich **firmenspezifische Baukastensysteme** für die Planung und Auslegung bilden lassen. Mit Hilfe von Konfigurierungswerkzeugen lässt sich so die Maschine in einem vorge-dachten Rahmen durch Auswahl einzelner Module kundenindividuell und softwareunterstützt konfigurieren. Die **Konfigurierbarkeit** bezieht sich allerdings nur auf den mechanischen Aufbau der Maschinen.

Alle Ansätze zur Gestaltung **modularer, rekonfigurierbarer** Werkzeugmaschinen beschränken sich auf die Modularisierung des mechanischen Maschinenaufbaus und lassen das Steuerungssystem unberücksichtigt. Im Bereich der Steuerungstechnik gibt es zwar Ansätze für **modulare Steuerungssysteme**, diese wurden allerdings nie im Zusammenhang mit rekonfigurierbaren Maschinen betrachtet. Derzeitige Werkzeugmaschinen, die einen gewissen Grad an Konfigurierbarkeit beziehungsweise Rekonfigurierbarkeit zulassen, sind gekennzeichnet durch eine **begrenzte Adaptionfähigkeit** der Steuerungstechnik an denkbare Konfigurationsänderungen der Maschinen. Zusammenfassend lassen sich heutige rekonfigurierbare Werkzeugmaschinen charakterisieren durch:

- einen modulareren Aufbau. Die **Modularisierung** beschränkt sich allerdings auf den Maschinenaufbau, die NC-Steuerung ist nicht berücksichtigt.
- eine **schlechte Erweiterbarkeit der NC-Steuerung**, da die bei der Erstkonfigurierung festgelegte Steuerungsstruktur eine spätere Erweiterung der Maschine erschwert und eine erneute Konfigurierung der Steuerung erfordert. Erweiterungen sind daher oftmals nur unter hohem Kosten- und Zeitaufwand im Bereich von mehreren Tagen durchzuführen.
- ein **unflexibles Steuerungssystem**, das speziell an die Maschinenkonfiguration angepasst werden muss. Da es **kein methodisches Vorgehen** zur Adaption von Steuerungen gibt, hat die Anpassung Projektcharakter, ist nicht reproduzierbar und hängt von der Erfahrung des Inbetriebnehmers ab.

Bei der Inbetriebnahme wird bisher die Steuerung nach einer Konfigurierung des Maschinenaufbaus speziell an die Maschinenkonfiguration angepasst. Insbesondere bei unterschiedlichen Maschinenkonfigurationen muss die Steuerung jedes Mal neu konfiguriert werden. Da die Modularisierung der Steuerung nicht der Modularität des Maschinenaufbaus folgt und die **funktionalen Zusammenhänge** zwischen Maschinenstruktur und Steuerungsstruktur nicht

formal beschrieben sind, sondern nur als Fachwissen in einem so genannten mentalen Modell vorhanden sind, erfordert die Adaption der Steuerung Spezialwissen und Erfahrung. Oftmals werden vorhandene Konfigurierungsdaten ähnlicher Maschinen wiederverwendet, indem sie kopiert und angepasst werden. Dadurch werden Fehler mit übertragen. Das Ergebnis ist dann nicht reproduzierbar und von einzelnen Fachspezialisten abhängig.

Für eine rekonfigurierbare Werkzeugmaschine fehlen daher ein **methodisches, durchgängiges Vorgehen** zur Adaption von NC-Steuerungen und Mechanismen und Werkzeuge zur **selbstständigen, automatisierten Adaption** von NC-Steuerungen nach einer Rekonfiguration, welche die Zusammenhänge zwischen Maschinen- und Steuerungskonfiguration berücksichtigen.

Daraus leitet sich die Zielsetzung der vorliegenden Arbeit ab (Bild 4.17). Es soll ein durchgängiges, methodisches Vorgehen zur **automatischen Adaption** von NC-Steuerungen entwickelt werden, indem allgemeine Konfigurierungsobjekte für NC-Steuerungen definiert werden, die als Datenobjekte für herstellerspezifische Konfigurierungs- und Parametrierungsdaten dienen. Diese Methode soll mechanische und steuerungstechnische Aspekte von rekonfigurierbaren Werkzeugmaschinen umfassen.

Ziele	Methoden	Vorgehensweise	
<ul style="list-style-type: none"> • Automatische Adaption von NC-Steuerungen nach einem Umbau der Maschine • Reduzierung der Inbetriebnahmezeiten • Vereinfachung der Steuerungskonfiguration 	Systematisierung und Formalisierung des NC-Konfigurierungsprozesses	<ul style="list-style-type: none"> • Analyse des Konfigurierungsprozesses und der funktionalen Zusammenhänge zwischen Maschinen- u. Steuerungskonfiguration • Analyse der Anpassbarkeit von NC-Steuerungen • Definition eines Strukturmodells einer adaptierbaren NC-Steuerung 	Kapitel 5
	Automatisierung der NC-Konfigurierung durch Übertragung von Plug-and-Play-Prinzipien	<ul style="list-style-type: none"> • Analyse von Plug-and-Play-Systemen und Definition von Anforderungen an Plug-and-Play-fähige Maschinenmodule • Ableitung der Voraussetzungen für die Übertragbarkeit • Übertragung von Plug-and-Play-Prinzipien auf die Konfigurierung von NC-Steuerungen 	Kapitel 6
	Modellbasierte Konfigurierung zur automatisierten Adaption der NC-Steuerung	<ul style="list-style-type: none"> • Einführung eines modularen Maschinenkonzepts • Definition eines Informationsmodells zur Beschreibung der Zusammenhänge zwischen Maschinen- und Steuerungskonfiguration • Erarbeitung einer Methode zur modellbasierten NC-Konfigurierung • Konzeption einer selbstadaptierenden NC-Steuerung 	Kapitel 7

Bild 4.17: Zielsetzung, Methoden und Vorgehensweise der Arbeit

Zur selbstständigen, automatisierten Konfigurierung der NC-Steuerung soll in Kapitel 5 in einem ersten Schritt der NC-Konfigurierungsprozess systematisiert und automatisiert werden, indem der bestehende Prozess und die **herstellerübergreifende Anpassbarkeit** von NC-Steuerungen analysiert werden. Dabei sollen die Abhängigkeiten zwischen Maschinen- und Steuerungskonfiguration untersucht und ein **Strukturmodell einer adaptierbaren NC-**

Steuerung erarbeitet werden.

Anschließend soll in Kapitel 6 eine **Methode zur Selbstadaption** von NC-Steuerungen entworfen werden. Die Methode soll sich an Prinzipien des aus der Computertechnik bekannten Plug-and-Play orientieren. Dazu sollen die Eigenschaften von Plug-and-Play-Systemen untersucht werden. Aus dieser Analyse sollen Anforderungen an Plug-and-Play-fähige Maschinenmodule und Voraussetzungen für die Übertragbarkeit von Plug-and-Play-Prinzipien auf die NC-Konfigurierung abgeleitet werden. Anschließend soll in Kapitel 6 durch **Übertragung von Plug-and-Play-Prinzipien** eine Methode zur Selbstadaption erarbeitet werden.

In Kapitel 7 soll die Methode zur Selbstadaption detailliert und anschließend ein Konzept für eine selbstadaptierende NC-Steuerung erarbeitet werden. Da die Anwendung von Plug-and-Play-Prinzipien intelligente Module voraussetzt, die über Kommunikationsschnittstellen und eigene Prozessoren verfügen, wird zuerst ein **modulares Maschinenkonzept** eingeführt, welches im Rahmen des SFB 467 vom Institut für Werkzeugmaschinen (IfW) und dem Institut für Steuerungstechnik der Werkzeugmaschinen und Fertigungseinrichtungen (ISW) der Universität Stuttgart entwickelt wurde. Dieses Maschinenkonzept basiert auf **mechatronischen Modulen**, welche die Voraussetzungen für die Anwendung der Plug-and-Play-Prinzipien zur Automatisierung der NC-Konfigurierung herstellen.

Als Grundlage für den Selbstadaptionsmechanismus soll in einem weiteren Schritt ein **Informationsmodell** entworfen werden, in dem das Expertenwissen über den Konfigurierungsvorgang modelliert wird. Das Informationsmodell soll die **Abhängigkeiten** zwischen Maschinenkonfiguration und Steuerungskonfiguration formalisieren. Dazu werden Teilmodelle für die Maschinenstruktur und die NC-Steuerungskonfiguration definiert und die **funktionalen Zusammenhänge** zwischen den Objekten dieser Strukturen abgebildet.

Abschließend soll das Konzept einer **selbstadaptierenden NC-Steuerung** die erarbeitete modellbasierte Konfigurierungsmethode integrieren und die Komponenten einer selbstadaptierenden NC-Steuerung beschreiben.

5 Analyse und Systematisierung des NC-Konfigurierungsprozesses

Die **Anpassung der NC-Steuerung** an die Maschinenkonfiguration erfolgt, indem Konfigurierungslisten erstellt und angepasst werden. Das Vorgehen zur Erstellung und Anpassung dieser Listen ist nicht systematisiert und formalisiert. Damit fehlt die Grundlage für einen Algorithmus zur Automatisierung dieses Vorganges.

Im folgenden Kapitel werden daher das Vorgehen zur Adaption von **NC-Steuerungen analysiert** und die Abhängigkeiten zwischen Maschinen- und Steuerungskonfiguration **systematisiert**. Anschließend wird die **herstellerübergreifende Anpassbarkeit** von NC-Steuerungen analysiert. Dazu werden mehrere NC-Steuerungen und ihre Konfigurierungslisten dahingehend untersucht, ob es gemeinsame und immer wiederkehrende Parameter (bei allen Steuerungen) gibt. Aus der Analyse und dem Vergleich der bestehenden NC-Konfigurierungslisten werden Gemeinsamkeiten abgeleitet und ein verallgemeinertes **Strukturmodell** einer **anpassbaren NC-Steuerung** definiert.

5.1 Analyse des Vorgehens bei der Adaption von NC-Steuerungen

Während der (Re-)Konfigurierung müssen **Abhängigkeiten, Restriktionen** und **Verträglichkeiten** zwischen Maschinenmodulen und Steuerungssoftwarekomponenten berücksichtigt werden. Linearachsen müssen beispielsweise anders konfiguriert und parametrieren werden als Rundachsen. Dadurch sind die (Re-)Konfigurierungsfreiräume eingeschränkt, da es beispielsweise Konfigurationen gibt, deren Module nicht miteinander harmonieren. Ebenso hat der Maschinenaufbau, d.h. die geometrische Anordnung der Achsen und die vorgesehenen Freiheitsgrade, einen wesentlichen Einfluss auf die Konfigurierung der Steuerung. Abhängig von den verbauten Maschinenmodulen muss anschließend die Steuerungskonfiguration erstellt und die NC-Steuerungssoftware parametrieren werden.

5.1.1 Abhängigkeiten zwischen Maschinen- und Steuerungskonfiguration

Die **NC-Steuerungskonfiguration** hängt von dem mechanischen Aufbau (Maschinenkonfiguration), den verwendeten Antriebskomponenten, deren geometrischer Anordnung und Kopplung, den physikalischen Randbedingungen des Gesamtsystems, sowie der eingebauten Einzelkomponenten ab (**Bild 5.1**).

Der mechanische Aufbau und die Freiheitsgrade der Mechanik legen die Anzahl der notwendigen **Achsen** und **Kanäle** fest. Sie müssen in den NC-Konfigurationsdaten angelegt werden. Je nach verwendetem Achstyp müssen sie als Linear- oder Rundachse definiert werden. Werden mehrere **Achsgruppen** (gemeinsam interpolierte Achsen) angesteuert, so muss dies

bei der Definition der NC-Kanäle berücksichtigt werden. Die geometrische Anordnung der Achsen und die durch die Konstruktion vorgegebene logische Kopplung wirken sich auf die **Kanalstruktur** und die **Transformation** aus.

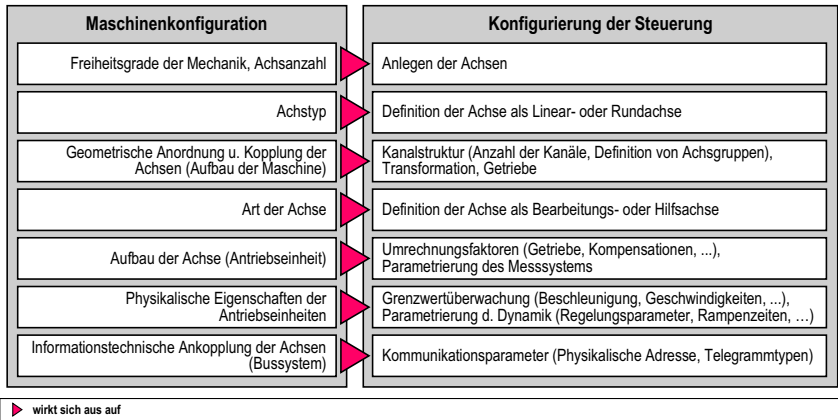


Bild 5.1: Abhängigkeiten zwischen Maschinen- und Steuerungskonfiguration

Bild 5.2 zeigt ein Beispiel einer **mehrkanaligen NC-Struktur** für die Steuerung von zwei voneinander unabhängigen 3-achsigen Bearbeitungseinheiten einer Drehmaschine für die Mehrschlittenbearbeitung. Die Achsen werden jeweils getrennt nach Bearbeitungseinheit in Achsgruppen in einem eigenen NC-Kanal zusammengefasst. Die Transformation berücksichtigt die kinematischen Zusammenhänge.

Die Konstruktion legt außerdem die Art der Achsen fest. Mit **Bearbeitungsachsen** („synchrone Achsen“) wird durch eine Relativbewegung zwischen Werkzeug und Werkstück die programmierte Werkstückgeometrie erzeugt. Ob hierbei das Werkstück oder das Werkzeug bewegt wird, ist von der Maschinenkonstruktion abhängig. Bearbeitungsachsen stehen in einem geometrischen Zusammenhang und bilden bei kartesisch aufgebauten Maschinen ein rechtwinkliges Koordinatensystem (2D oder 3D). Für die Bewegungserzeugung werden sie als Achsgruppe in einem NC-Kanal zusammengefasst und gemeinsam mit Sollwerten versorgt (Interpolationsverbund). Bei **Hilfsachsen** („asynchrone Achsen“) besteht im Gegensatz zu Bearbeitungsachsen kein geometrischer Zusammenhang. Sie sind daher keinem NC-Kanal zugeordnet. Sie werden in der Regel für Handling- oder Zubringeraufgaben an der Maschine eingesetzt.

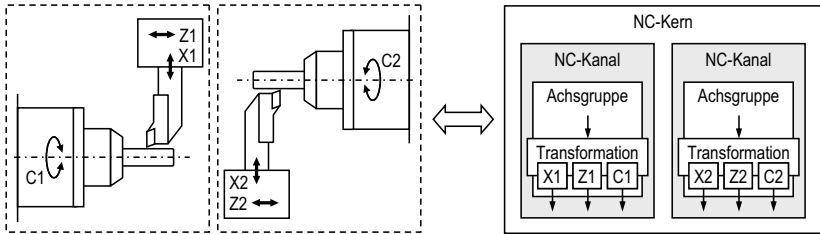


Bild 5.2: Beispiel für mehrere unabhängige Achsgruppen und eine mehrkanalige NC-Struktur

Für die **Kommunikation** zwischen Steuerung und Antriebssystem müssen bei der Konfiguration Kommunikationsparameter eingestellt werden. Wichtig ist bei allen Bussystemen die feste Einstellung der eindeutigen physikalischen Adresse und die Vorgabe der Zykluszeit. Sie muss mit der Einstellung des Interpolationstaktes harmonisieren, da sonst ein Speicherüberlauf droht. Des Weiteren müssen je nach Bussystem Parameter für den Telegrammaufbau vorgegeben werden. Dazu zählen Einstellungen zum Antriebstyp und dessen Funktionen (z.B. antriebsbasiertes Referenzieren, Feininterpolation auf dem Antrieb, schnelle Messeingänge und Messtaster, Art des Messsystems)

Der Aufbau der Achse muss bei der Konfiguration berücksichtigt werden, indem Umrechnungsfaktoren für die Getriebe eingestellt und die indirekten Messsysteme parametrisiert werden.

Über **Parametrierungsdaten** werden die physikalischen Eigenschaften der Maschine berücksichtigt. Die Parameter werden bei der Erzeugung von Sollgrößen durch die steuerungsisernen Algorithmen bei der Informationsverarbeitung im NC-Kanal berücksichtigt. Dies sind Grenzwertvorgaben (maximale Geschwindigkeiten und Beschleunigungen) aber auch Werte für die Parametrierung des dynamischen Verhaltens der Maschine (Regelungsparameter, Rampenzeiten für den Beschleunigungsauf- und -abbau).

5.1.2 Anpassung der NC-Steuerung auf Basis mentaler Modelle

Das Wissen über das Erstellen einer funktionsfähigen Konfiguration einer Maschine und deren Steuerung ist bisher Experten vorbehalten und nicht formal, also von Algorithmen prüfbar, beschrieben. Der Experte weiß aufgrund seiner Ausbildung und Erfahrung welche Softwarekomponenten konfiguriert werden müssen (**mentales Modell**) und wie, abhängig vom Aufbau der Maschine, die Steuerungssoftware parametrisiert werden muss (**Bild 5.3**). Bei der NC-Konfiguration hat er die Maschinenkonfiguration im Kopf und leitet an Hand der ver-

bauten Maschinenmodule die notwendigen Softwarekomponenten und deren Strukturierung ab. Bei Nichtbeachtung dieser Abhängigkeiten kommt es später, während der Inbetriebnahme, zu teuren Änderungen an der Steuerungskonfiguration oder gar an der Mechanik.

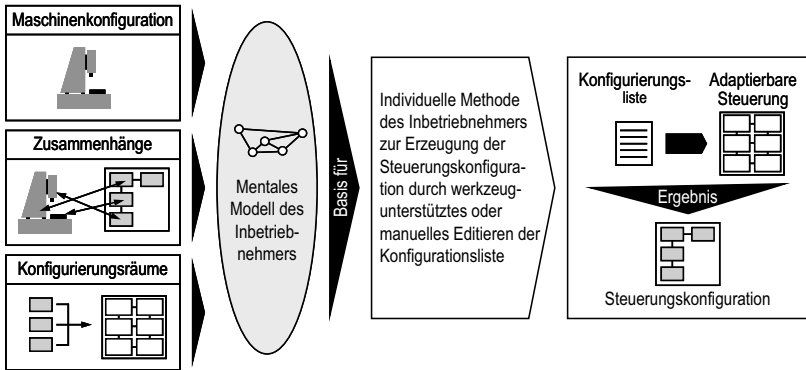


Bild 5.3: Mentales Modell

Durch **Baukästen für Maschinenmodule** (mechanischer Baukasten) wird die Konfiguration der Maschine durch Auswahl von Modulen in einem Softwarewerkzeug unterstützt /52, 88/ (Bild 5.4). Der Steuerungsinbetriebnehmer leitet aus den vorhandenen Maschinenmodulen und deren Anordnung die notwendigen Informationen für die Steuerungskonfigurierung ab. In Gedanken hilft ihm ein **Muster**, wie er die NC-Steuerung zu konfigurieren hat.

Die Anpassung der Steuerung bei monolithisch aufgebauten, anpassbaren Steuerungen erfolgt durch Vorgabe von Parameterwerten in einer Konfigurationsliste. Bei offenen Steuerungen wird in einer Konfigurationsliste die Auswahl der benötigten Softwarekomponenten, die Definition von Kommunikationsverbindungen und die Parametrierung der Steuerungssoftware festgeschrieben /120/. Da es keine herstellerübergreifende und formale Beschreibung dieser Referenzarchitekturen gibt, ist die Konfiguration in verfügbaren Werkzeugen auch nicht grafisch unterstützt, sondern erfolgt durch Eingabemasken oder Editieren von ASCII-basierten Textdateien. Die Vorgehensweise ist daher unstrukturiert und unsystematisch.

Die Anpassung erfolgt oft durch **Kopieren** und **Anpassen** existierender Konfigurationseinstellungen von Vorgängerprojekten. Dadurch werden Fehleinstellungen mitkopiert und das Ergebnis hängt sehr stark von der persönlichen Qualifikation des einzelnen Mitarbeiters ab, so dass die Ergebnisse nicht immer reproduzierbar sind.

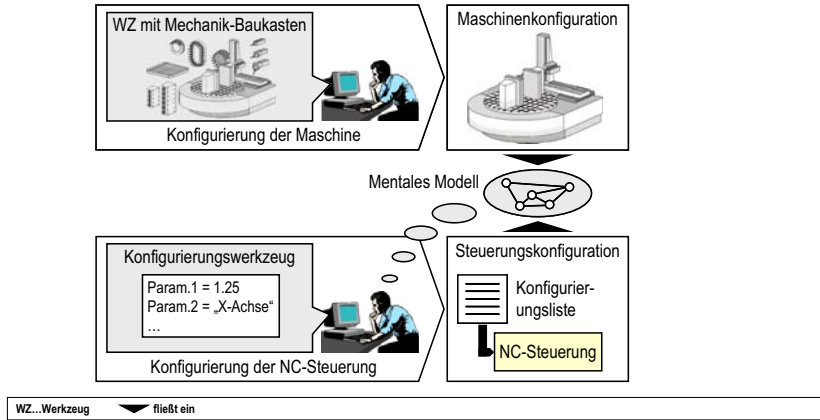


Bild 5.4: Vorgehen beim Konfigurieren von Steuerungssystemen

5.2 Systematisierung des Vorgehens bei der NC-Konfiguration

Das Vorgehen bei der Konfiguration lässt sich in vier Schritte unterteilen und folgendermaßen systematisieren (Bild 5.5):

1. **Analyse** der Maschinenkonfiguration.
2. **Ableiten** der für die NC-Konfiguration notwendigen **NC-Konfigurationsobjekte** (Achsen, Achsgruppe, Getriebe).
3. **Festlegen** der Struktur der NC-Steuerung (z.B. Kanäle und Achsen anlegen).
4. **Parametrieren** der Steuerungssoftware.

Bei der Analyse der Maschinenkonfiguration muss der Inbetriebnehmer die verbauten Maschinenmodule erkennen und die bewegungsbestimmenden Module, die Hilfsachsen und die von der SPS gesteuerten Aggregate identifizieren. Zusätzlich ist für die Konfiguration die geometrische Anordnung der Achsen wichtig, da sie die Art der Transformation festlegt. Diese Information ist nicht direkt ablesbar, sondern erfordert den Vergleich von Baumustern von Maschinenaufbauten. Der Inbetriebnehmer klassifiziert dazu gedanklich die Maschinentypen.

Aus dem Ergebnis dieser Analyse leitet der Inbetriebnehmer die notwendigen, logischen **NC-Konfigurationsobjekte** ab, da er weiß, welche Konfigurationmöglichkeiten die Steuerung hat. Als Konfigurationsobjekte werden in dieser Arbeit alle in der NC-Struktur zu definierenden, logischen Datenobjekte bezeichnet, die bei der Konfiguration in der NC angelegt werden müssen, damit Algorithmen darauf arbeiten können. Da es keine allgemeingültige

Beschreibung von NC-Konfigurierungsobjekten gibt, sind diese nur als Modellvorstellung im **mentalen Modell** des Inbetriebnehmers vorhanden. Eine Vorschubachse wird in der textbasierten Konfigurierungsliste beispielsweise nicht als ein Parameter abgebildet. Hierzu gehört eine Menge von Parametern (z.B. physikalische Grenzwerte, Softwareendschalter, usw.), die eingestellt werden müssen. Der Inbetriebnehmer bildet daher im Kopf **logische Einheiten** (Konfigurierungsobjekte), die einen funktionalen Bezug zur Mechanik haben und sich in einer Menge von logisch zusammengehörenden Parametern abbilden.

Die benötigten Konfigurierungsobjekte werden von außen durch Vorgabe von Parametern ausgewählt. Die **formale Beschreibung** der Auswahl erfolgt in einer ASCII-textbasierten Konfigurierungsliste. Es gibt auch Konfigurierungswerkzeuge /112, 113, 114/, die zur Konfigurierung von NC-Steuerungen Masken anbieten, in denen Konfigurierungsobjekte mit ihren Parametern angelegt werden können. Diese sind allerdings herstellerspezifisch und erfordern immer, dass der Inbetriebnehmer im Kopf den logischen Bezug zur Mechanik herstellt. Diese mentale Leistung des Inbetriebnehmers kann nicht von diesen Werkzeugen übernommen werden.

Beim Anlegen der Konfigurierungsobjekte muss der Inbetriebnehmer daher eine funktionale Klassifikation der Maschinenmodule im Kopf haben. Er unterscheidet dabei zwischen Maschinenmodulen, die für die Konfigurierung der NC-Steuerung relevant sind und Modulen, die unrelevant sind. Zu den relevanten Modulen zählen alle bewegungsbestimmenden Module (Achsen).

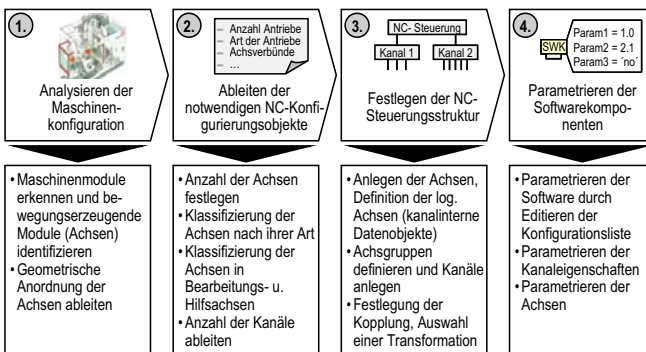


Bild 5.5: Systematisiertes Vorgehen bei der NC-Konfigurierung

Wenn die Mechanik spezielle Anforderung an die Steuerung stellt (z.B. Steuern einer Sonderkinematik), dann fehlen die möglichen Konfigurierungsobjekte und die Steuerung kann nicht

von außen über Parameter konfiguriert werden. Aufgrund der Offenheit einiger NC-Steuerungen ist es aber in begrenztem Umfang auch möglich, komplett neue Steuerungssoftware zu entwickeln und in die Steuerung zu integrieren. Allerdings muss dazu ggf. auch die Referenzarchitektur angepasst werden, wenn die neue Softwarekomponente in ihrer Art noch nicht in der Architektur vorgesehen war. Dieser Fall benötigt allerdings mehr Zeit und ist für eine schnelle Rekonfigurierung nicht geeignet. Daher beschränkt sich diese Arbeit auf die Auswahl vorhandener Komponenten. Die Neuentwicklung von Softwarekomponenten für eine Rekonfigurierung ist nicht vorgesehen.

Da die Steuerungsobjekte herstellerspezifisch sind und in der Fachliteratur nicht systematisch beschrieben sind, muss der Inbetriebnehmer Parameter einstellen, deren Zuordnung zu logischen und mechanischen Einheiten nicht eindeutig herzustellen ist. In den Handbüchern sind zwar detaillierte Beschreibungen zu den einzelnen Parametern zu finden, sie stellen aber keinen funktionalen Bezug zu Modulen der Maschine her, sondern nur zu dem einzelnen Steuerungshardwaregerät. Eine sinnvolle logische und herstellerübergreifende Zusammenfassung zu NC-Konfigurierungsobjekten, die einen funktionalen Bezug zu mechatronischen Modulen haben, ist bis jetzt noch nicht bekannt.

Für jedes bewegungsbestimmende Maschinenmodul muss eine Achse in der NC-Steuerung angelegt werden. Dies kann eine Linear- oder Rundachse, eine Spindel oder eine so genannte C-Achse (synchronisierte Spindel in Lageregelung, z.B. für das Gewindeschneiden) sein. Dazu muss der Inbetriebnehmer gedanklich eine **Klassifizierung** an Hand der eingebauten Achsen vornehmen. Zusätzlich müssen die Achsen als Bearbeitungs- oder Hilfsachse klassifiziert werden. Dies hängt davon ab, ob sie an der Erzeugung der geometrischen Relativbewegung beteiligt sind oder nicht. Bearbeitungsachsen sind an der Erzeugung der Relativbewegung beteiligt, Hilfsachsen sind nicht beteiligt.

Nachdem die NC-Steuerungsobjekte definiert sind, wird die **NC-Steuerungsstruktur** festgelegt. Hierbei ist wichtig, welche Achsen gemeinsam interpoliert werden und somit zu einer Achsgruppe gehören. Dazu wird ein **NC-Kanal** in der Steuerung angelegt. Ihm werden alle Bearbeitungsachsen einer Achsgruppe und die Hilfsachsen zugewiesen. Außerdem muss für jeden NC-Kanal durch die Auswahl einer Transformation der **geometrische Zusammenhang** der Bearbeitungsachsen festgelegt werden. Die Algorithmen des NC-Kanals arbeiten mit sogenannten **logischen Achsen** oder auch Kanalachsen. Diese können den **physikalischen Achsen** frei zugeordnet werden.

Bild 5.6 zeigt beispielhaft die steuerungsinterne Zuordnung von logischen und physikalischen Achsen in einem NC-Kanal. Die Motoren sind an Leistungsteile angeschlossen. Ihre physika-

liche Kommunikationsadresse ergibt sich fest durch die Platznummer (Index), das heißt sie ist durch den Einbau der Leistungsteile festgelegt. Um beim Aufbau einer Maschine eine Verdrahtungsfreiheit zu erreichen, wird die physikalische Achse von der steuerungsinternen Repräsentation entkoppelt. Dazu werden in der NC-Steuerung für alle physische Achsen **Maschinenachsen** mit sprechenden Namen gemäß ihrer Funktion (Achsrichtung) definiert.

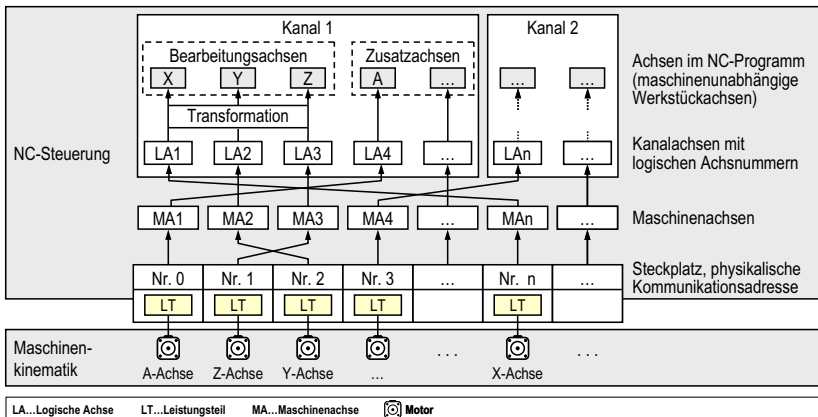


Bild 5.6: Beispielhafte Darstellung der internen Zuordnung von Achsen in einem NC-Kanal

Damit die Algorithmen der NC-Steuerung maschinenunabhängig entwickelt werden können, werden im NC-Kanal logische Achsen verwendet (Kanalachsen). Erst durch die Zuordnung von logischen Achsen zu den Maschinenachsen ergibt sich ein Zusammenhang zwischen steuerungsinternen logischen Achsen und physikalischen Achsen.

Die Bearbeitungsachsen stehen physikalisch durch ihren **kinematischen Aufbau** in einem geometrischen Zusammenhang (meist in kartesischer Anordnung). Dieser Zusammenhang wird in der NC-Steuerung durch eine Transformation abgebildet. Die Bearbeitungsachsen, die im NC-Programm angesprochen werden, sind daher durch eine Transformation mathematisch miteinander verknüpft.

Nachdem die NC-Struktur mit ihren Achsen und Kanälen konfiguriert ist, müssen die NC-Steuerungsobjekte **parametriert** werden. Wesentliche Einstellungen sind die Bahn- und Achsdynamik.

Für jeden **NC-Kanal** müssen Parameter für die maximale Bahnvorschub- und Eilgeschwindigkeit, sowie die maximale Beschleunigung vorgegeben werden. Mit der Definition von Beschleunigungsrampen wird der Bahnruck festgelegt. Außerdem kann der Override für jeden

NC-Kanal individuell voreingestellt werden.

Für jede **Achse** müssen die physikalische Adresse und die Kommunikationseigenschaften (Zykluszeit, Telegrammtyp) festgelegt werden. Die Dynamik der Achse im Vorschub und im Eilgang wird über Grenzwerte für die maximale Geschwindigkeit und Beschleunigung festgelegt. Rampenzeiten für den Beschleunigungsaufbau und den Beschleunigungsabbau begrenzen den Ruck. Betriebsartenspezifische Maximalgeschwindigkeiten und -beschleunigungen (Tipp-Geschwindigkeit, Vorschub- und Eilganggeschwindigkeit, Not-Stop-Verzögerung). Darüber hinaus werden achsspezifische Überwachungsfunktionen parametrierbar. Hierzu gehören z.B. Softwareendschalter-, Schleppabstand- und Temperaturüberwachung.

Bei der Parametrierung der Achse spielt außerdem deren mechanischer Aufbau eine Rolle. Abhängig davon müssen Getriebestufen und Messsysteme parametrierbar werden. Bei den Messsystemen spielen die Art der Messwerterfassung (absolut, inkrementell), die Auflösung und der Messort (direkte oder indirekte Messung) eine Rolle.

Ein Spezialfall der Parametrierung ist die Einstellung der **Regelparameter**. Hierfür gibt es aber Methoden und Werkzeuge zur automatisierten Identifizierung der Regelstrecke /140/. Damit ist es möglich, Regelparameter automatisiert abzuleiten und einzustellen. Aus diesem Grund wird die automatisierte Einstellung der Regelparameter in dieser Arbeit nicht detaillierter betrachtet.

Für die Optimierung der Achsdynamik stehen für jede Achse Parameter für Achsfilter zur Verfügung. Hierzu gehören beispielsweise Funktionen zur Losekompensation und Vorsteuerfunktionen.

Der Speicherbereich der NC-SPS-Schnittstelle muss ebenfalls parametrierbar werden. Dazu werden Speicherbereiche für globale, kanal- und achsspezifische Variablen definiert. Außerdem wird im Speicher Platz für die Werkzeugverwaltung angelegt.

5.3 Analyse der herstellerübergreifenden Anpassbarkeit von NC-Steuerungen

Wegen der Abhängigkeiten zwischen der Maschinenkonfiguration und der Steuerungskonfiguration sind die **Modularität** der Maschinenkonfiguration und die **Konfigurierungsfreiräume** des mechanischen Aufbaus eine wesentliche Grundlage für die Adaption des Steuerungssystems.

Gängige, auf dem Markt verfügbare NC-Steuerungen sind zwar adaptierbar und über Maschinendaten an Maschinenkonfigurationen anpassbar, der Aufbau ihrer Steuerungssoftware ist allerdings nicht identisch und im Sinne einer offenen, konfigurierbaren Steuerung nicht modular. Die **Konfigurierung** der NC-Steuerungen ist deshalb **herstellerspezifisch**. Daher soll

für die Entwicklung einer Selbstadaptionsmethode ein Strukturmodell einer adaptierbaren Steuerung definiert werden, welches die herstellerspezifischen Konfigurierungsmöglichkeiten abstrahiert (Bild 5.7).

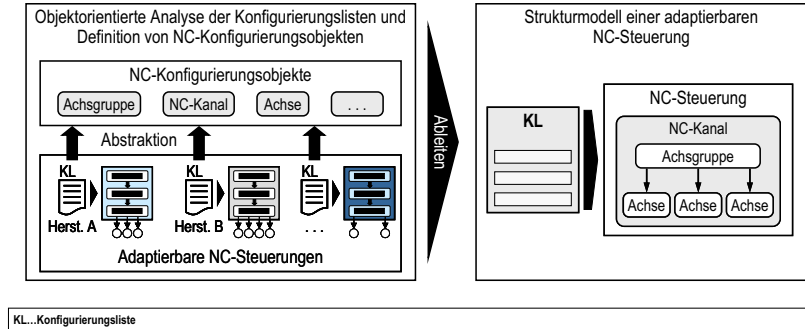


Bild 5.7: Vorgehen zur Definition des Strukturmodells einer adaptierbaren NC-Steuerung

Da die Steuerungssoftware gängiger, auf dem Markt verfügbarer NC-Steuerungen von außen gesehen nicht modular ist, lehnt sich das **Strukturmodell** an das OSACA-Systemkonzept an. Dort wurde im Rahmen einer objektorientierten Analyse eine modulare Steuerungsreferenzarchitektur auf Basis von Architekturobjekten definiert. Architekturobjekte sind die Elemente der NC-Steuerungskonfiguration und bilden die Konfigurierungsinformationen ab /85/.

Die Analyse der **Anpassbarkeit** der NC-Steuerungen und ihrer Konfigurierungsmethoden ist nicht durch einen Vergleich der NC-Steuerungsstrukturen und ihrer Modularität möglich, weil der interne Aufbau der NC-Steuerung nicht öffentlich dokumentiert ist. Da aber die Anpassung der als Maximalkonfiguration ausgelieferten NC-Steuerung immer über Parametrierung durch Aktivierung bzw. Deaktivierung von Optionen von außen über Konfigurationslisten realisiert wird, zeigt sich in den Parametern dieser Liste eine gewisse Modularität und Ähnlichkeit. Daher sollen die NC-Steuerungen von außen, im Sinne einer **Black-Box-Betrachtung**, durch den Vergleich ihrer Anpassbarkeit an Hand ihrer Konfigurationslisten analysiert werden und allgemeine NC-Konfigurierungsobjekte definiert werden. Damit wird eine abstrakte Modularität geschaffen, obwohl die NC-Steuerungen von außen betrachtet monolithisch aufgebaut sind.

5.3.1 Analyse der Konfigurierungsmöglichkeiten

Die Untersuchung der **Konfigurierungsmöglichkeiten** von vier gängigen NC-Steuerungen /141, 142/ zeigt, dass sich die NC-Steuerungen durch **Maschinendaten** an den spezifischen

Aufbau der Maschine anpassen lassen. Die Maschinendaten werden in ASCII-basierten Textdateien auf der Steuerung abgelegt. Beim Hochlauf der Steuerung werden diese Dateien interpretiert und die Software der Steuerung durch ein **Konfigurationslaufzeitsystem** angepasst.

Das Editieren der Maschinendaten kann entweder über ein herstellerspezifisches Werkzeug mit Eingabemasken erfolgen oder über die Vorgabe eines vorgefertigten Maschinendatensatzes, der herstellerspezifisch formatiert und an einer vorbestimmten Speicherstelle der NC-Steuerung abgelegt ist.

Da Steuerungen unabhängig von der Maschinenkonstruktion ausgeliefert werden und erst beim Maschinenhersteller über Maschinendaten angepasst werden, haben alle am Markt üblichen Steuerungen ähnliche Konfigurierungsmöglichkeiten. Die Maschinendaten enthalten deshalb ähnliche Informationen. Die Beschreibung der Maschinendaten ist allerdings herstellerspezifisch. Das heißt, die **Syntax** der Maschinendaten wird vom Steuerungshersteller festgelegt. Ebenso wird die Struktur des Maschinendatensatzes vorgegeben. Es gibt Maschinendaten, die in einer Datei abgelegt sind. Es gibt aber auch Maschinendaten, die über mehrere Dateien verteilt sind. Bild 5.8 zeigt beispielhafte Auszüge aus Konfigurierungslisten. Obwohl die Syntax herstellerspezifisch ist, sind die einstellbaren Parameter dieselben.

	840D (Siemens)	TwinCAT CNC (Beckhoff)	IndraMotion MTX (Bosch Rexroth)
Software- endlagen	<pre> ... \$MA_FIX_POINT_POS[1,AX1] = 1100 \$MA_FIX_POINT_POS[0,AX1] = -1100 ... </pre>	<pre> ... kenngr.swe_pos 1000 kenngr.swe_neg -1000 ... </pre>	<pre> ... <Dr> <Pos> <TrvLim> <PoTrvLim>1000.0</PoTrvLim> <NeTrvLim>-1000.0</NeTrvLim> </TrvLim> </Pos> </Vel> ... </pre>
Achsgeschwindig- keit von Achse 1	<pre> ... \$MA_MAX_AX_VELO[AX1] = 40000 ... </pre>	<pre> ... getriebe[0].dynamik.vb_max 40000 ... </pre>	<pre> ... <MaxVel>40000.0</MaxVel> <FeedCalc>0</FeedCalc> </Vel> ... </pre>
Eilgang-Override	<pre> ... \$MC_OVR_RAPID_VEL = 200 ... </pre>	<pre> ... kenngr.max_vb_override 2000 ... </pre>	<pre> ... </Dr> ... <Path> <Ovr> <MaxChOvr>200</MaxChOvr> </Ovr> ... </pre>
maximale Bahn- beschleunigung	<pre> ... \$SD_MAX_PATH_ACCEL = 1 ... </pre>	<pre> ... getriebe[0].slope_profil_a_beschl 1000 ... </pre>	<pre> ... <Acc> <MaxChAcc>1.0</MaxChAcc> </Acc> ... </Path> ... </pre>

Bild 5.8: Beispielhafte Auszüge aus Konfigurierungslisten

Für eine Methode zur Rekonfigurierung und automatisierten Generierung von Konfigurierungslisten ist nur der Informationsgehalt der Maschinendaten wichtig, da über so genannte Generatoren und entsprechende Generierungsvorgaben sowohl die Struktur der Maschinendaten (Speicherung im Dateisystem als eine Datei oder verteilt auf mehrere Dateien), als auch

die Syntax der Maschinendaten automatisch erzeugt werden kann. Wichtig ist daher die Festlegung **herstellerübergreifender Konfigurierungsobjekte** als Informationsträger für die Konfigurierungsdaten.

Die NC-Konfigurierungslisten der untersuchten NC- Steuerungen sind alle nach einem ähnlichen **Schema** aufgebaut und als ASCII-Textdatei formiert. Obwohl sich die Syntax der Parameterdefinition in den Konfigurierungslisten unterscheidet, lässt sich verallgemeinern, dass es einen **Bezeichner**, eine herstellerepezifische **Zuweisung**, einen **Wert** und gegebenenfalls einen **Kommentar** gibt. Bild 5.9 zeigt am Beispiel der Definition einer Softwareendlage den schematischen Aufbau von drei unterschiedlichen ASCII-Textdateien. Bei Konfigurierungslisten, die mit XML strukturiert sind (Beispiel Bosch Rexroth), erfolgt die Zuweisung durch das Einschließen des Wertes zwischen einem sogenannten öffnenden und schließenden Tag mit dem Bezeichner.

	Bezeichner	Zuweisung	Wert	Kommentar
Beispiel Siemens	...			
	\$MA_FIX_POINT_POS[1,AX1]	=	1000	
	\$MA_FIX_POINT_POS[0,AX1]	=	-1000	
	...			

Beispiel Beckhoff	...			
	kenngr.swe_pos	„Leerzeichen“	1000	# Positive Endlage
	kenngr.swe_neg	„Leerzeichen“	-1000	# Negative Endlage
	...			

Beispiel Bosch Rexroth		...		
		<PoTrvLim>	1000	
		</PoTrvLim>		
		<NeTrvLim>	-1000	
		</NeTrvLim>		
	...			

Bild 5.9: Schema zum Aufbau von textbasierten Konfigurierungslisten im ASCII-Format

Des Weiteren zeigt der Vergleich, dass sich die Maschinendaten logisch in **allgemeine, kanal- und achsspezifische Maschinendaten** abstrahieren und einteilen lassen (Bild 5.10). Allgemeine Maschinendaten legen im Wesentlichen die Struktur der NC-Steuerung fest. Dies geschieht durch Festlegung der Anzahl der Kanäle und der Achsen. Für jeden NC-Kanal und für jede Achse gibt es kanal- bzw. achsspezifische Maschinendaten. In den kanalspezifischen Maschinendaten werden die in den allgemeinen Maschinendaten angelegten Achsen referenziert. Die Achsparametrierung erfolgt in den achsspezifischen Maschinendaten.

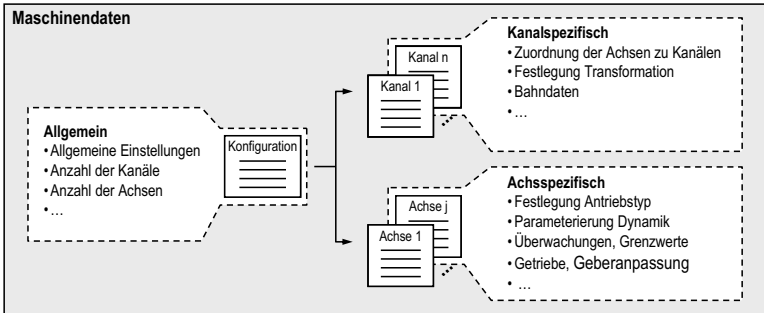


Bild 5.10: Einteilung der Maschinendaten

Die Speicherung der Maschinendaten in Dateien ist herstellerspezifisch. Oftmals wird für jede Achse und jeden NC-Kanal eine separate Datei angelegt. Die Klammer um die Dateien bilden die allgemeinen Maschinendaten. Durch sie wird festgelegt wie viel weitere kanal- und achs-spezifische Maschinendaten es gibt. Bild 5.11 zeigt beispielhaft einen Auszug der kanal- und achsspezifischen Maschinendaten einer Beckhoff NC-Steuerung.

	Bezeichner	Wert	Kommentar	Speicherung in Dateien
Kanalparameter	grp_anzahl	1	# Festlegung der Achsgruppen	SDA_Kanal1.lis
	gruppe[0].bezeichnung	IPO_1	# Name der Achsgruppe im Kanal	
	gruppe[0].achs_anzahl	6	# Anzahl der Achsen	
	gruppe[0].achse[0].log_achs_nr	1	# 1. Achse	
	gruppe[0].achse[0].bezeichnung	X	# 1. Achse	
	gruppe[0].achse[1].log_achs_nr	2	# 2. Achse	
	gruppe[0].achse[1].bezeichnung	Y	# 2. Achse	
kinematik_id	52	# weitere Achsen bis zur 6. Achse # Auswahl der kinematische Transformation Nr. 52		
Achspanparameter Achse1	kopf.log_achs_name	X		X_Kanal1.lis
	kenngr.swe_pos	2500000	# [0.1um] Positive Softwareendlage	
	kenngr.swe_neg	-2500000	# [0.1um] Negative Softwareendlage	
	kenngr.max_vb_override	1000	# Maximaler Wert des Geschwindigkeitsoverride	
	kenngr.getriebe_stufe	1	# Default-Getriebestufe	
	getriebe[0].nummer	1	# Nummer der Getriebestufe	
	getriebe[0].slope_profil_a_beschl	1000	# [mm/s ²] Beschleunigung bei Bearbeitungsverschub G1, ...	
	getriebe[0].slope_profil_a_brems	1000	# [mm/s ²] Verzögerung bei Bearbeitungsverschub G1, ...	
	getriebe[0].dynamik_vb_max	100000	# (500000)[um/s] Maximal zulässige Achsgeschwindigkeit	
	getriebe[0].dynamik_a_max	2000	# (5000)[mm/s ²] Maximal zulässige Achsbeschleunigung	
Achspanparameter Achse 2, usw.				..._Kanal1.lis

Bild 5.11: Beispiel für eine in mehreren Dateien gespeicherte NC-Konfigurierungsliste

Die Konfigurierungsdaten sind in unterschiedlichen Dateien gespeichert. Die kanalspezifischen Maschinendaten sind in der Datei „SDA_Kanal1.lis“ gespeichert. Dort wird die Achsgruppe IPO_1 bestehend aus sechs Achsen definiert, indem für jede Achse eine logische

Achsnummer und einen Bezeichnung angelegt wird. Unter anderem wird über den Bezeichner *kinematik_id* die Transformation für den NC-Kanal definiert. Für jede in der Datei „*SDA_Kanall.lis*“ definierten Achse gibt es eine separate Datei. Der Name der Datei mit den achsspezifischen Parametern beginnt mit dem Bezeichner der Achse (im Beispiel *X*) und wird dann mit „*Kanall.lis*“ ergänzt, da die Achse dem Kanal 1 zugeordnet ist. In der Datei „*X_Kanall.lis*“ werden die Parameter der X-Achse definiert. Dies sind zum Beispiel Getriebeparameter, Grenzwerte für den Bearbeitungsvorschub und Grenzwerte für die Dynamik der Achse.

5.3.2 Strukturmodell einer adaptierbaren NC-Steuerung

Da gängige, auf dem Markt verfügbare NC-Steuerungen zwar adaptierbar und über Maschinendaten an Maschinenkonfigurationen anpassbar sind, ihre Steuerungssoftware von außen gesehen aber nicht modular ist, soll für die weitere Arbeit ein verallgemeinertes **Strukturmodell** einer **adaptierbaren Steuerung** entwickelt werden. Mit dem Strukturmodell sollen die herstellerspezifischen Konfigurierungsmöglichkeiten einer NC-Steuerung aus Kapitel 5.3.1 abstrahiert und eine virtuelle Modularität geschaffen werden, indem **verallgemeinerte NC-Konfigurierungsobjekte** (NC-KO) definiert werden. Damit kann eine allgemeingültige Lösung für eine selbstadaptierende NC-Steuerung auf Basis dieses Strukturmodells beschrieben werden, ohne jedes Mal auf herstellerspezifische Implementierungen eingehen zu müssen. Für einen herstellerübergreifenden Ansatz sollen daher allgemeingültige, abstrakte **NC-KO** definiert werden, die bei der Konfigurierung als Platzhalter und Datencontainer für herstellerspezifische NC-Konfigurierungsdaten dienen. Über diese NC-KO lassen sich auf einer allgemeingültigen Ebene die Beziehungen zu den Maschinenmodulen formal abbilden.

Für die Definition verallgemeinerter Maschinendaten einer adaptierbaren NC-Steuerung müssen die Adaptionsmöglichkeiten der NC-Funktionen betrachtet werden. Dazu wird im Folgenden der in Kapitel 3.1.2 grundlegend eingeführte Aufbau eines NC-Kerns bestehend aus den Funktionalen Einheiten Decoder, Werkzeuggeometriekorrektur, Bahnvorbereitung, Bahnausführung und Lageeinstellung detailliert ([Bild 5.12](#)).

Der **Decoder** interpretiert das NC-Programm, prüft die Syntax, teilt die Befehle in Verfahrbefehle (G-Funktionen, Vorschubgeschwindigkeit), in Maschinen- und Technologiefunktionen (M- und T-Funktionen) auf. Die M- und T-Funktionsaufrufe werden an die SPS weitergeleitet. Die Verfahrbefehle werden im NC-Kanal weitergeleitet. Da sie in der Steuerung verändert (z.B. Werkzeuggeometriekorrektur) und zusätzlichen Informationen hinzugefügt werden (z.B. Einfügen von Kurvensegmenten beim Eckenüberschleifen), wird der NC-Satz in ein steuerungsisinternes Datenformat umgewandelt (NC-Steuerdaten).

In der **Werkzeuggeometriekorrektur** werden Nullpunkt- und Koordinatenverschiebungen berücksichtigt. Damit die Programmierung unabhängig vom Werkzeug erfolgen kann, korrigiert die NC-Steuerung in der Werkzeuggeometriekorrektur die Bahn und berechnet Äquidistanten.

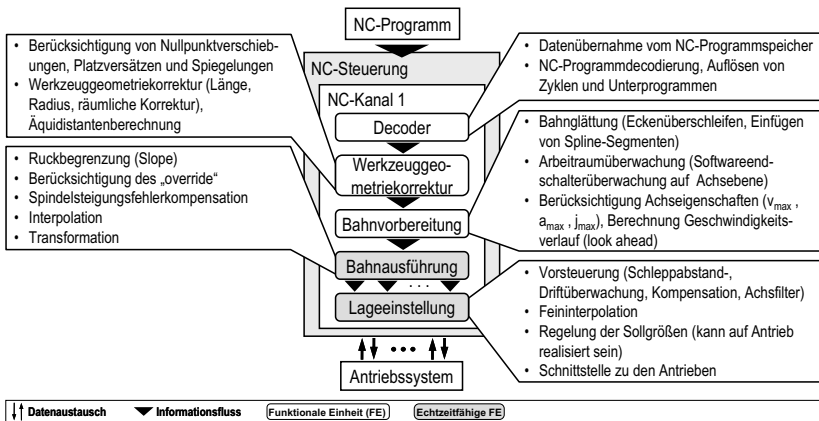


Bild 5.12: Steuerungsfunktionen und interner Informationsfluss einer NC-Steuerung

Um eine Anregung der Mechanik zu vermeiden, werden in der **Bahnvorbereitung** an Ecken Übergänge eingefügt. Für die internen Berechnungen wird die Bahn in Polynomschreibweise überführt. Anschließend wird durch die look-ahead-Funktion ein Geschwindigkeitsprofil geplant, welches ein rechtzeitiges Halten am Satzende ermöglicht. Dabei werden die physikalischen Randbedingungen der Achsen berücksichtigt. Als maximale Beschleunigung gilt das Beschleunigungsvermögen der schwächsten Achse. Als Standardstrategie zur Planung von Geschwindigkeitsprofilen wird das 7-Phasenmodell verwendet [73, 74]. Zusätzlich überwacht die Bahnvorbereitung den zulässigen Arbeitsraum der Maschine durch Softwareendschalter auf Achsebene.

In der **Bahnausführung** wird das Geschwindigkeitsprofil durch eine Slope-Funktion (Ruckbegrenzung) so angepasst, dass auch der Ruck ein Profil hat und nicht konstant ist. Dies vermeidet Anregungen der Mechanik, was sich insbesondere bei kurzen NC-Sätzen, wie sie bei der Freiformflächenbearbeitung üblich sind, auf die Oberflächengenauigkeit, die Konturtreue und die Fertigungszeit auswirkt. Zusätzlich werden in der Bahnausführung zeitgerecht (im Interpolationstakt, im Allgemeinen 1 ms) Lagesollwerte für die Bahn gebildet. Die interpolierten Werte werden dann durch eine Transformation in achsweise Vorgabewerte für die Lageeinstellung umgerechnet. Danach können für jede Achse Genauigkeitsfehler (fertigungsbe-

dingt) durch positionsabhängige Kompensationen ausgeglichen werden (z.B. Spindelsteigungsfehler-, Lose- und Durchbiegungskompensation). Diese Funktionen können bei der Inbetriebnahme parametrierbar werden. Wichtig ist dies insbesondere bei der Transformation, da dadurch fertigungs- und montagebedingte Fehler ausgeglichen werden können.

Die **Lageeinstellung** übernimmt bei geschwindigkeitsgeregelten Antriebssystemen die Lage-
regelung für jede Achse /73, 143/. Die Lageeinstellung bildet außerdem die Schnittstelle zum
Bussystem. Über einen Treiberbaustein versorgt die Lageeinstellung zyklisch den Bus mit
Sollwerten für die Antriebe. Dafür müssen für jede Achse die Kommunikationsparameter
(Zykluszeit, physikalische und logische Adresse) eingestellt werden.

Da die Implementierung der NC-Funktionen im NC-Kern als FE herstellerspezifisch ist und
die Softwarestruktur daher keine einheitliche, herstellerübergreifende Modularität aufweist,
werden im Strukturmodell der adaptierbaren NC-Steuerung die FE mit denselben Rechenzeit-
anforderungen zu abstrahierten Softwarekomponenten zusammengefasst.

Zur Abstraktion werden daher die FE Decoder, FE Werkzeuggeometriekorrektur, FE Bahn-
vorbereitung und FE Bahnausführung zur Softwarekomponente **Steuerdatenaufbereitung**
(SDA) und die FE Bahnausführung und FE Lageeinstellung zur Softwarekomponente **Geo-
metriedatenverarbeitung** (GEO) zusammengefasst (Bild 5.13). SDA und GEO bilden einen
NC-Kanal in dem alle gemeinsam interpolierten Achsen zusammengefasst sind. In der Geo-
metriedatenverarbeitung werden die Sollwerte für die Bahn erzeugt. Eine **Transformation**
berücksichtigt dabei die geometrische Anordnung der Achsen und berechnet aus der pro-
grammierten Bahn im Raum Sollwerte für jede einzelne Achse im so genannten Achskoordi-
naten-system.

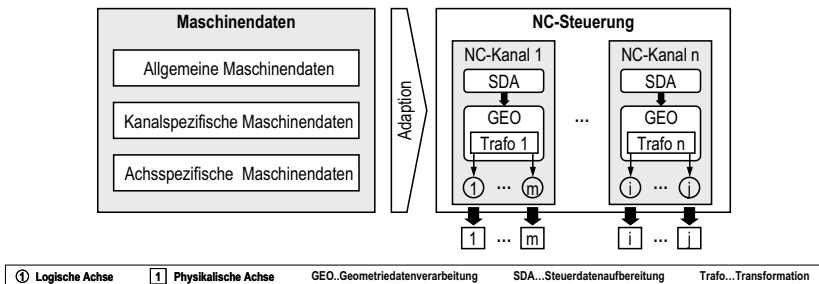


Bild 5.13: Strukturmodell einer adaptierbaren Steuerung

Die Algorithmen der NC-Steuerung arbeiten mit **logischen Achsen**, denen bei der Konfigu-
rierung physikalische Achsen zugeordnet werden. Die logischen Achsen sind Datenobjekte

für die Algorithmen im NC-Kanal. Sie repräsentieren die Eigenschaften der physikalischen Achsen. Dies sind beispielsweise physikalische Grenzwerte für Geschwindigkeit und Beschleunigung, Fertigungsungenauigkeiten wie zum Beispiel Spindelsteigungsfehler oder achstypspezifische Informationen zur Art der Achse (z.B. Rund- oder Linearachse).

Zur Adaption der NC-Steuerung müssen die Algorithmen, Softwarekomponenten und Datenobjekte parametrisiert werden. Dies erfolgt über **Maschinendaten**. Dort sind alle Parameter der NC-Konfigurierungsobjekte beschrieben. Die Maschinendaten lassen sich in allgemeine, achs- und kanalspezifische Maschinendaten einteilen. **Bild 5.14** zeigt zusammenfassend eine detaillierte Auflistung der wichtigsten Maschinendaten zur Anpassung der NC-Steuerung.

Maschinendaten		
allgemein	kanalspezifisch	achsspezifisch
<ul style="list-style-type: none"> • Zykluszeit (Interpolationstakt) • Maßeinheiten • Anzeigfeinheit • Normierungsfaktoren für Einheiten und Darstellung der steuerungsisnternen Genauigkeit • Festlegung von Achsen • Anlegen der NC-Kanäle • Speicherkonfigurierung <ul style="list-style-type: none"> • Kanal- u. achsspezifische M-Funktionen • Globale Variablen • Werkzeugverwaltungs- und Werkzeugkorrekturdaten 	<ul style="list-style-type: none"> • Bahndaten <ul style="list-style-type: none"> • Vorschubgeschwindigkeit • Eilganggeschwindigkeit • Max. einstellbarer Override • Dynamik der Bahn (Ruck, Beschleunigungsprofil) • Anlegen der Koordinatenachsen (logische Achsen) • Kinematische Transformation <ul style="list-style-type: none"> • Auswahl einer Transformation • Parametrierung der kin. Transformation 	<ul style="list-style-type: none"> • Kommunikation (physikal. Adresse, Telegrammtyp) • Kanalzuordnung • Antriebstyp (linear, rotatorisch, Spindel, C-Achse) • Referenzierung (Nocken, Referenziergeschw., ...) • Geberanpassung (Auflösung, direkt, inkrementell, ...) • Getriebe (Getriebestufen, Gantrykopplung, ...) • Regelparameter <ul style="list-style-type: none"> • Dynamik (jeweils für Vorschub u. Eilgang) <ul style="list-style-type: none"> • Max. (neg. u. pos.) Beschleunigung (Slope) • Rampenzeit f. Beschleunigungsauf-/abbau • Überwachungen (SW-Endschalter, Schleppabstand, ...) • Grenzwerte (Not-Stopp-Verzögerung, Vorschub- u. Eilganggeschwindigkeit, Override) • Betriebsartenspezifische Parameter (Tipp-Geschw., ...) • Achsfilter u. Vorsteuerung (Lose, ...) • Spindelparameter

Bild 5.14: Maschinendaten zur Anpassung der NC-Steuerung

5.4 Zusammenfassung

Ausgehend von der Analyse des Vorgehens bei der Konfigurierung von NC-Steuerungen, wurden in diesem Kapitel die Tätigkeiten des Inbetriebnehmers abstrahiert und systematisiert. Damit wurde die Grundlage für eine Automatisierung der bisher in unsystematischer Weise durchgeführten Steuerungskonfigurierung geschaffen.

Im Rahmen der Analyse der herstellerübergreifenden Anpassbarkeit von NC-Steuerungen zeigte sich, dass sich die Konfigurierungsinformationen der NC-Konfigurierungslisten verallgemeinern lassen. Dadurch konnte ein herstellerübergreifendes Strukturmodell einer adaptierbaren NC-Steuerung geschaffen werden, obwohl NC-Steuerungen unterschiedliche und öffentlich nicht dokumentierte innere Softwarestrukturen haben. Diese Abstraktion ermöglicht die Entwicklung von herstellerübergreifenden Selbstadaptionmethoden.

6 Methode zur Selbstadaption von NC-Steuerungen

Methoden zur selbständigen Konfigurierung von Systemen sind aus dem Bereich der Computertechnik als **Plug-and-Play** (PnP) bekannt /144/. Als Plug-and-Play wird dort ein Prinzip bezeichnet, das die Anwendung von zusätzlichen Geräten am Computer dadurch vereinfacht, dass die Konfigurierung und Bereitstellung der Gerätefunktion automatisch abläuft. Zur Automatisierung der NC-Konfigurierung sollen die aus der Computertechnik bekannten PnP-Prinzipien analysiert und auf die NC-Steuerungstechnik übertragen und angepasst werden.

Ausgehend von dem systematisierten Vorgehen zur NC-Konfigurierung und dem Strukturmodell einer adaptierbaren Steuerung aus Kapitel 5 wird in diesem Kapitel eine **Methode zur Selbstadaption** von NC-Steuerungen entwickelt, indem die Plug-and-Play-Prinzipien auf die Steuerungstechnik und der PnP-Gerätegedanke auf **modulare Werkzeugmaschinen** übertragen werden (Bild 6.1).

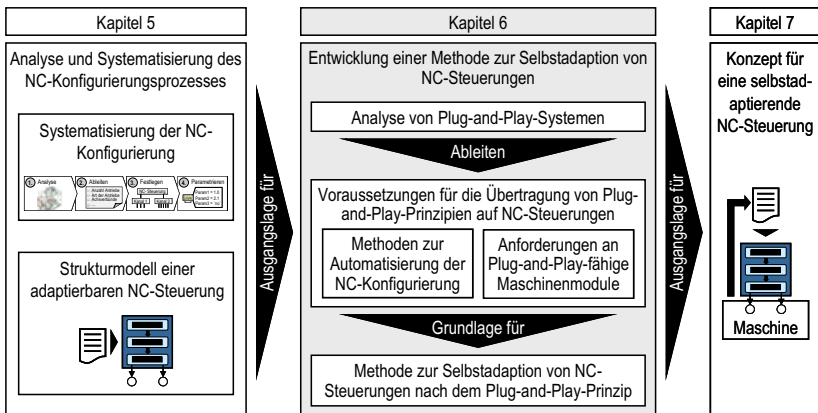


Bild 6.1: Vorgehen zur Entwicklung der Methode zur Selbstadaption von NC-Steuerungen

Aus der Analyse der Eigenschaften und Methoden von PnP-Systemen werden die notwendigen Voraussetzungen für die Anwendung des PnP-Prinzips im Bereich der Steuerungstechnik abgeleitet. Als Grundlage für die Methode zur Selbstadaption von NC-Steuerungen nach dem PnP-Prinzip werden Methoden zur **Automatisierung der NC-Konfigurierung** und Anforderungen an **PnP-fähige Maschinenmodule** definiert.

In Kapitel 7 wird schließlich ein Konzept für eine selbstadaptierende NC-Steuerung entworfen. Dabei werden die in Kapitel 6 definierten Voraussetzungen für die Übertragung der PnP-Prinzipien geschaffen, indem die Methoden zur Automatisierung der NC-Konfigurierung de-

tailliert erarbeitet und die Anforderungen an PnP-fähige Maschinenmodule in einem modularen Maschinenkonzept umgesetzt werden.

6.1 Analyse von Plug-and-Play-Systemen aus der Computertechnik

Beim Plug-and-Play in der Computertechnik können Geräte (z.B. Drucker, externe Festplatten, Fotokameras) über eine definierte Kommunikationsschnittstelle an den Computer angeschlossen werden. Das System konfiguriert sich anschließend von selbst, so dass die Funktionalität des Gerätes genutzt werden kann. Das heißt, die Geräte werden erkannt, die Kommunikation wird konfiguriert und die notwendigen Treiber werden geladen. Dafür sind spezielle, PnP-fähige Geräte notwendig. Diese Geräte besitzen eine **Kommunikationsschnittstelle**, einen **internen Speicher** für Parameter- und Identifizierungsinformation und erfüllen eine **standardisierte** und **klassifizierte Funktion**. Als Kommunikationsschnittstelle zwischen Geräten und Computer hat sich inzwischen USB (Universal Serial Bus) /145/ und der unter der Spezifikation IEEE 1394 /146/ bekannte FireWire-Bus /147, 148/ durchgesetzt.

Der Vergleich der beiden Systeme USB und FireWire in /149/ und /150/ zeigt, dass sie ähnliche funktionale Eigenschaften haben und der grundlegende PnP-Ablauf derselbe ist. Die für das PnP notwendigen Funktionen sind in den Bussystemen in Protokollschichten implementiert, die zwar systemspezifisch aufgebaut sind, aber dennoch verallgemeinert werden können.

6.1.1 Selbstkonfigurierung von Plug-and-Play-Systemen

Beim Plug-and-Play-Vorgang wird ein zusätzliches Gerät über eine definierte Kommunikationsschnittstelle an den Computer angeschlossen. Der Computer erkennt die Art des Gerätes und teilt dem Gerät eine Adresse für die Kommunikation zu. Dafür haben diese Systeme einen so genannten **Master**, der die Konfigurierung des Gesamtsystems vornimmt. Bei USB ist das der PC. Bei dem FireWire-Bus ist die Funktion des Masters nicht auf ein bestimmtes Gerät festgelegt. Jedes so genannte masterfähige FireWire-Gerät kann sie übernehmen. Sie muss aber auf jeden Fall von einem Gerät erbracht werden. Die Masterfunktionalität umfasst hauptsächlich die Konfigurierung des Systems (automatische Adressvergabe, Vergabe von Bandbreiten, Reaktion bei Umkonfigurierungen, ...).

Zur **Identifikation** haben alle PnP-Geräte der beiden Systeme eine eindeutige Kennung. Diese Kennung wird zusammen mit Zusatzinformation wie zum Beispiel Angaben zum Hersteller, Versionsnummer oder benötigten Treibern im Gerät selbst auf einem Konfigurierungsspeicher (Configuration ROM) gespeichert. Der Informationsgehalt im Konfigurierungsspeicher variiert und ist im jeweiligen Standard zwingend vorgeschrieben /145,146, 151/.

Zum Auslesen des Konfigurierungsspeichers haben die PnP-fähigen Geräte eine **standardi-**

sierte Zugriffsmethode. Sie ermöglicht dem Master bei der Identifizierung aller Kommunikationsteilnehmer, die für die Konfigurierung der Kommunikation notwendige Information aus dem Konfigurierungsspeicher auszulesen. Dazu ist die Information formal beschrieben und an einer festgelegten Stelle im Speicher abgelegt, so dass der Master standardisiert darauf zugreifen kann. Bei FireWire wird dies zum Beispiel durch eine vorgeschriebene **Speicherstruktur**, der sogenannten CSR (Control and Status Register)-Architektur erreicht. Im CSR-Standard /151/ wird der logische Aufbau eines Gerätes (Knoten) und dessen Speicherstruktur definiert. Im Speicher sind bestimmte Bereiche für Konfigurierungsinformationen reserviert. Die CSR-Architektur kann von verschiedenen Bussystemen implementiert und erweitert werden.

In FireWire wird der 64 Bit breite Adressraum aufgeteilt in die drei Teiladressräume für den Bus, den Geräteknoten und die Register des Gerätes (Bild 6.2). Jeder Knoten wird als Speicher betrachtet. Ein Teil dieses Speichers wird von FireWire selbst verwendet und ein Teil ist durch die CSR-Architektur vorgegeben. Sie legt damit einen Mindestumfang an Informationsgehalt fest, weitere Informationen können in einem freien Datenbereich beschrieben werden. FireWire definiert dazu im Rahmen der CSR-Architektur eine Beschreibungssprache. Mit ihr kann der individuelle Datenbereich strukturiert werden.

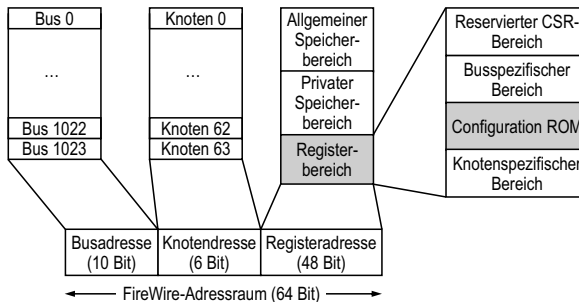


Bild 6.2: CSR-Architektur am Beispiel FireWire

Mit den ersten 16 Bit der Adresse wird ein bestimmter Knoten adressiert. Die restlichen 48 Bit teilen sich in drei Blöcke ein. Aus dem allgemeinen Speicherbereich können beliebige Daten gelesen werden. Ebenso können Daten in den allgemeinen Speicherbereich geschrieben werden. Der private Speicherbereich ist für herstellerspezifische Angaben über den Knoten (z.B. Hersteller- und Gerätenamen, Firmwareversion) reserviert. Der Registerbereich teilt sich in die vier Bereiche CSR-Bereich, busspezifischer Bereich, Configuration ROM und knotenspezifischer Bereich auf. Im CSR-Bereich befinden sich die Kontroll- und Statusregister, über

die der Knoten konfiguriert und gesteuert wird. Im busspezifischen Bereich sind Informationen über das Bussystem enthalten. Im Configuration ROM wird die Architektur des Knotens beschrieben. Es enthält außerdem Informationen über die logischen Funktionseinheiten des Gerätes und deren hierarchische Gliederung. Der knotenspezifische Bereich beinhaltet Information für den Bus oder herstellerspezifische Angaben zu den logischen Funktionseinheiten und deren Softwareschnittstellen.

Bei PnP-Systemen erfolgt der Zugriff einer Applikation auf eine Gerätefunktion über **Treiber**. Sie steuern den Zustand des Gerätes und bieten zur Applikation hin eine definierte Funktionsschnittstelle an. Für bestimmte Geräteklassen hat der Master (Computer) eine vordefinierte, standardisierte Funktionsschnittstelle mit einheitlicher Ansteuerung.

PnP-Systeme nutzen für die Auswahl eines Treibers zur Ansteuerung von PnP-Geräten eine **Klassifikation der Geräte**. Durch eine Zuordnung von Geräteklassen zu Gerätefunktionen wird die Funktions- und Steuerschnittstelle des PnP-Gerätes festgelegt und kann so standardisiert im System angesprochen werden. Die Identifizierung erfolgt durch festgelegte und vordefinierte Kennungen (ID) und Geräteklassennamen, die in dem Konfigurationsspeicher der PnP-Geräte abgelegt sind.

Wesentlich ist die **Standardisierung der Gerätefunktionen**, so dass der Master (Computer) mit bereits vordefinierten Gerätetreibern unterschiedliche Gerätetypen von verschiedenen Herstellern bedienen kann. Diese Geräte haben durch den einheitlichen Treiber dieselbe Funktionsschnittstelle und sie funktionieren alle mit derselben Zustandssteuerung. Dafür werden im Treiber sogenannte logische Geräte angelegt (**Bild 6.3**). Sie sind ein Abbild des realen Gerätes und besitzen einheitliche Funktions- und Steuerschnittstellen. Dazu sind die PnP-Geräte klassifiziert (z.B. Massenspeichermedium, Web-Cam, Eingabegerät, ...), so dass für jede Geräteklasse eine Mindestmenge an Funktionen und Eigenschaften (Zykluszeit, maximale Bandbreite, ...) vorausgesetzt werden kann.

Alle PnP-Geräte, die der gleichen Klasse angehören, werden auf die **logischen Geräte** abgebildet. Sie implementieren dadurch dieselbe funktionale Schnittstelle (Syntax und Semantik). Die Applikation arbeitet nur mit der Schnittstelle der logischen Geräte des Treibers. Dadurch ist sie unabhängig von den realen Geräten. Die realen PnP-Geräte sind aufgrund der definierten funktionalen Schnittstelle austauschbar und können einfach integriert werden. So wird durch die Zuordnung von Geräteklasse und entsprechenden Standardtreibern erreicht, dass mehrere Geräte denselben Treiber benutzen. Der Applikation und dem Nutzer steht damit eine standardisierte Zugriffsmöglichkeit auf die Gerätefunktionen zur Verfügung.

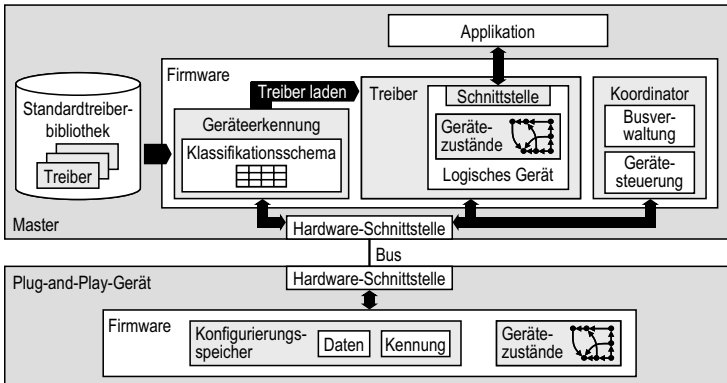


Bild 6.3: Einheitliche Geräteschnittstelle durch logische Geräte

Die für die Konfiguration des PnP-Systems notwendigen Funktionen sind in der Firmware der Geräte implementiert. Dazu hat der Master einen Koordinator mit einem Busverwalter und einer Gerätesteuerung. Im PnP-Gerät beinhaltet die Firmware im Wesentlichen den Gerätezustand und den Konfigurationsspeicher mit einem Datenspeicher und einer ID zur Kennung des Gerätes.

Der **Busverwalter** im Master ermöglicht die Kommunikation und die Datenübertragung zwischen zwei Geräten, indem er die Übertragung über das Bussystem konfiguriert und herstellt. Die wichtigste Funktion des Busverwalters ist die automatische Konfiguration der Kommunikation. Hierzu werden als erstes die am Bus angeschlossenen Geräte erkannt und initialisiert ([Bild 6.4](#)). Dies erfolgt entweder beim Einschalten (Hochfahren) des Systems oder bei laufendem Betrieb (Hotplug), indem ein Signal beim Einstecken den Bus in den Initialisierungszustand versetzt.

Nach dem Hochfahren oder Hotplug wird jedem Gerät eine eindeutige Adresse zugewiesen. Bei FireWire erfolgt dies aufgrund seiner Position in der Bustopologie. Bei USB wird dem Gerät zuerst eine Standardadresse (Adresse) zugeteilt bis der Master eine Neudressierung der Teilnehmer vornimmt.

Nachdem auf diese Weise eine PnP-Verbindung zum Gerät hergestellt ist, kann mit dem Gerät kommuniziert werden. Dafür stehen vorerst nur einfache Dienste zur Verfügung, die von jedem PnP-Gerät unterstützt werden. Anschließend wird die zyklische Kommunikation festgelegt, indem der Master über den Busverwalter die vorhandene Bandbreite verteilt und überwacht.

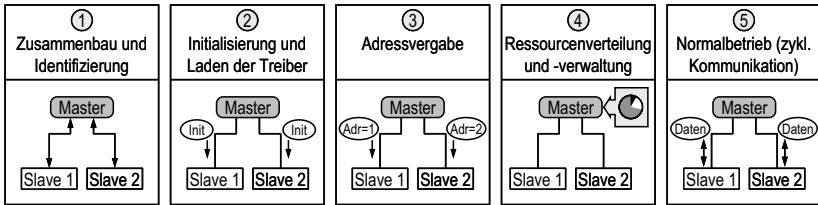


Bild 6.4: PnP-Ablauf

Um geräteklassenspezifische Funktionen nutzen zu können, wird durch die **Geräteerkennung** über die PnP-Verbindung die Geräteklasse ermittelt und der entsprechende Standardtreiber geladen. Durch die Klassifikation steht das Kommunikationsprotokoll und die Funktions- und Steuerschnittstelle fest, so dass die Daten des Gerätes interpretiert werden können. Werden an ein PnP-System Geräte angeschlossen, für die es keine generischen (herstellerunabhängige) Treiber gibt, müssen die entsprechenden Treiber geladen und die herstellereigenen Protokolle verwendet werden.

6.1.2 Eigenschaften von Plug-and-Play-Systemen

Die Analyse zeigt, dass die beiden PnP-Systeme USB und FireWire aus dem Computerbereich ähnliche Konzepte haben und sich die Eigenschaften der PnP-Geräte beider Systeme verallgemeinern lassen. Zusammenfassend sind die abstrahierten Eigenschaften und Prinzipien von PnP-Systemen im Bereich Computertechnik folgende:

- **Identifizierung und Klassifizierung** von Geräten,
- Standardisierte **Schnittstellen** (Treiber) für Geräteklassen durch Zuordnung von Funktionen zu Geräten über ein Klassifikationssystem,
- **Masterkonzept** zur Koordinierung des Geräte- und des Systemzustands und zur Koordinierung des Konfigurierungsablaufs,
- **Mechanismus zur Konfigurierung** der Kommunikation (automatische Adressvergabe, Vergabe und Überwachung der Ressourcen),
- Auslesbare und beschreibbare **Konfigurierungsspeicher** auf jedem Gerät.

PnP-Geräte haben eine **Kenntnis zur Identifikation**, die in einer Norm festgelegt ist. Darin sind sowohl die Art der Beschreibung (Syntax), deren Bedeutung (Semantik) und die Art und Weise der Zugriffsmöglichkeiten auf diese Kenntnis (Lesen und Schreiben) definiert.

Durch die Kenntnis wird auch die Art des Gerätes beschrieben. PnP-Systeme nutzen dafür

Klassifikationssysteme, um funktional gleichartige, gegeneinander austauschbare Geräte zu Mengen gleichartiger Geräte (Geräteklassen) zusammenzufassen. Den so definierten Geräteklassen können dadurch **Funktionen** zugeordnet werden. Das Verhalten von Geräten kann über eine Steuerschnittstelle beeinflusst werden, da sie **einheitliche Gerätezustände** haben. Durch die Normung der Geräteklassen, ihrer Funktions- und Steuerschnittstellen ist es möglich, auf einheitliche Weise auf die PnP-Geräte zuzugreifen und ihr Verhalten zu steuern.

PnP-Systeme haben einen **Mechanismus zur Konfigurierung der Kommunikation** und können somit eine grundlegende Kommunikationsverbindung zu den Geräten aufbauen. Dazu vergibt ein Master automatisch die Adresse und vergibt und überwacht die Ressourcen (Bandbreite auf dem Bus). Die für die Konfigurierung notwendigen Parameter werden aus dem **Konfigurierungsspeicher** der Geräte gelesen. Dort sind auch Informationen zur Identifizierung der Geräte und ihrer Geräteklassen und Informationen zur Konfigurierung gespeichert. Der Master steuert den **Systemzustand** und stellt dadurch sicher, dass die Phasen der Konfigurierung richtig durchlaufen werden, bevor die eigentliche Datenkommunikation beginnt.

6.2 Methoden zur Automatisierung der Inbetriebnahmetätigkeiten

Wenn die Adaption der NC-Steuerung selbstständig erfolgen soll, müssen die Tätigkeiten des Inbetriebnehmers bei der NC-Konfigurierung automatisiert werden. Hierzu sind geeignete Methoden zur Automatisierung des Vorgehens für die folgenden, der in Kapitel 5 systematisierten Konfigurierungsschritte zu entwickeln:

1. Analyse der Maschinenkonfiguration,
2. Ableiten der notwendigen NC-Konfigurierungsobjekte,
3. Festlegen der NC-Steuerungsstruktur,
4. Parametrieren der ausgewählten NC-Konfigurierungsobjekte.

Der Inbetriebnehmer beginnt mit der **Analyse der Maschinenkonfiguration** und erstellt gedanklich eine Liste der eingebauten Maschinenmodule. Eine selbstadaptierende Steuerung muss diese Module selbst erkennen. Dies ist möglich, wenn die Erkennungsmechanismen nach dem Plug-and-Play-Prinzip aus dem Bereich der Computertechnik im Bereich der Steuerungstechnik angewendet werden.

Aus der Analyse der Maschinenmodule leitet der Inbetriebnehmer die **NC-Konfigurierungsobjekte** ab, die für die Konfigurierung der NC angelegt und parametrieren müssen. Diese Auswahl soll automatisch erfolgen, indem ein modellbasierter Mechanismus zur Kon-

figurierung der NC-Steuerung angewendet wird. In einem Informationsmodell sollen die Zusammenhänge zwischen mechanischem Aufbau der Maschine und der Steuerungskonfiguration abgebildet sein. Durch die im ersten Schritt bestimmten Maschinenmodule lassen sich dann automatisch über die im Informationsmodell formalisierten Zusammenhänge die notwendigen NC-Konfigurierungsobjekte ableiten.

Nachdem der Inbetriebnehmer die notwendigen NC-Konfigurierungsobjekte festgelegt hat, muss er die **Struktur der NC-Steuerung** definieren. Hierbei ist wichtig, wie viele unabhängige Achsverbünde es gibt und wie sich die Achsen auf diese Verbünde aufteilen. Mit dieser Information können die NC-Kanäle angelegt und die Achsen zugeordnet werden. Dieser Vorgang kann durch Auswahl einer vordefinierten Steuerungsstruktur automatisiert werden. Dazu ist eine Referenzarchitektur notwendig bei der die NC-Konfigurierungsobjekte und die Steuerungsstruktur in Form von Mustervorlagen definiert sind. Sie legt fest, wie die ausgewählten NC-Konfigurierungsobjekte zu einer NC-Konfiguration kombiniert werden müssen.

Die **Parametrierung** der Steuerungssoftwarekomponenten durch den Inbetriebnehmer kann durch Anwendung von Plug-and-Play (PnP) Mechanismen aus dem Bereich der Computertechnik automatisiert werden. Bei PnP-fähigen Geräten sind in einem definierten Speicherbereich standardisierte Konfigurierungsinformationen abgelegt. Ein Master, der auf das PnP-Gerät zugreift, weiß dadurch wo die Information abgespeichert ist und was die Daten bedeuten. Nach diesem Prinzip sollen PnP-fähige Maschinenmodule einen definierten Speicherbereich besitzen, in dem die Konfigurierungsinformationen abgelegt sind.

Zur Automatisierung der Inbetriebnahmetätigkeiten müssen zusammenfassend für die folgenden Schritte Methoden entwickelt und umgesetzt werden Bild 6.5:

1. Automatische **Erkennung** der Maschinenmodule über den Bus,
2. Automatisches **Ableiten** der NC-Konfigurierungsobjekte durch einen modellbasierten Mechanismus zur Konfigurierung der NC-Steuerung,
3. Automatische NC-Konfigurierung durch **Auswahl** einer vorgegebenen Steuerungsstruktur,
4. **Auslesen** der Maschinenmodulparameter aus einem Konfigurierungsspeicher auf dem Maschinenmodul.

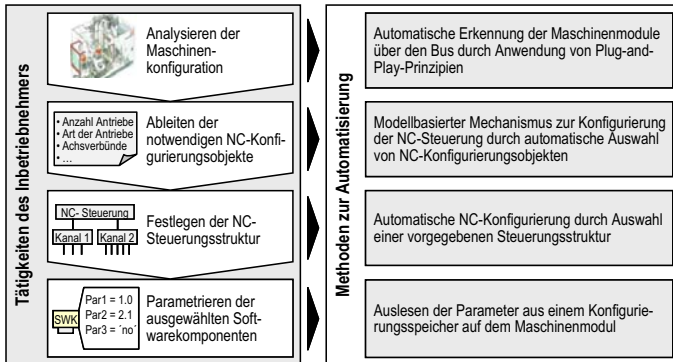


Bild 6.5: Methoden zur Automatisierung der Inbetriebnahmetätigkeiten

6.2.1 Automatische Erkennung der Maschinenmodule

Die selbstadaptierende Steuerung soll die Maschinenmodule selbst erkennen, indem **Erkennungsmechanismen** nach dem Plug-and-Play (PnP) Prinzip aus dem Bereich der Computertechnik auf die NC-Steuerungstechnik übertragen werden. Dies ist nur dann möglich, wenn PnP-fähige Maschinenmodule geschaffen werden. Sie müssen eine definierte Kommunikationsschnittstelle haben und eindeutig identifizierbar sein. Damit alle am Bussystem angeschlossenen Module erkannt werden können, wird ein Kommunikationssystem benötigt, das nach einer Initialisierungsphase grundlegende Kommunikationsmechanismen zur Verfügung stellt, mit denen die Module Daten zur Identifizierung austauschen können. Dazu muss es in der Lage sein, sich selbst zu konfigurieren.

Hier können **ethernetbasierte, echtzeitfähige** Kommunikationssysteme eingesetzt werden. Sie bieten neben der für die Automatisierung zwingend notwendigen Echtzeitkommunikation die Möglichkeit, in einem Nicht-Echtzeitkanal zusätzlich zwischen Teilnehmern über TCP/IP-basierte Kommunikationsdienste zu kommunizieren. Diese Bussysteme können sich beim Einschalten selbst initialisieren und durch Erkennungsmechanismen am Bus angeschlossene Teilnehmer erkennen. Der Bus ist dann während einer Initialisierungsphase in der Lage, die Kommunikation selbst zu konfigurieren. Dieser Mechanismus funktioniert aber nur für einzelne Geräte der Aktor-Sensor-Ebene und nicht für funktionale Einheiten, die eine logische Zusammenfassung aus funktional zusammengehörenden Aktoren und Sensoren im Sinne von mechatronischen Maschinenmodulen bilden.

Mit diesem Basismechanismus kann zwar eine Liste aller Kommunikationsteilnehmer abgefragt werden, wie ihre funktionalen Zusammenhänge sind und welche NC-Konfiguration

notwendig ist, lässt sich daraus aber nicht ableiten. Die Initialisierung des Bussystems kann aber im ersten Schritt der Selbstadaption genutzt werden, um eine Basiskommunikation für die nachfolgenden Selbstadaptionsmechanismen bereitzustellen. Für die automatische Adaption des Steuerungssystems wird dann über die einfache Liste der Kommunikationsteilnehmer hinaus eine Zuordnung von funktionalen Einheiten zu Konfigurierungsobjekten der NC-Steuerung notwendig, damit die NC-Konfigurierungsobjekte automatisch abgeleitet werden können.

6.2.2 Automatisches Ableiten von NC-Konfigurierungsobjekten und einer Steuerungsstruktur

Damit die Auswahl der NC-Konfigurierungsobjekte automatisch erfolgen kann, muss ein **Zusammenhang** zwischen **Maschinenmodulen** der Maschinenkonfiguration und **NC-Konfigurierungsobjekten** hergestellt werden. Für die Zuordnung von Funktionen zu Geräten nutzen PnP-Systeme ein Klassifikationssystem. Damit sind einer Klasse von Geräten ein definierter Funktionsumfang und eine definierte Schnittstelle (Treiber) zugewiesen.

Dieses Prinzip soll genutzt werden, um über ein **Klassifikationssystem** die funktionalen Zusammenhänge zu definieren, so dass Maschinenmodulen NC-Konfigurierungsobjekte zugeordnet werden können. Dies muss dann nicht für jedes Modul im Speziellen erfolgen, sondern kann für eine Klasse (Menge) von Modulen erfolgen, sofern es möglich ist, die Module anhand von gleichen Funktionen zusammenzufassen und zu hierarchisieren (Klassifizierung). Damit lassen sich einer Klasse von Modulen die notwendigen NC-Konfigurierungsobjekte zuordnen. Dazu muss dieser Zusammenhang formalisiert werden.

Informationsmodelle, wie sie zum Beispiel beim Engineering von Maschinen und Anlagen verwendet werden /132, 152, 153/, eignen sich für die formale Beschreibung von Maschinenmodulen und den Abhängigkeiten zwischen diesen Modulen und der notwendigen NC-Steuerungskonfiguration. Aufbauend auf diesem Informationsmodell lassen sich dann modellbasierte Konfigurierungen durchführen. Die NC-Konfigurierungslisten können dann nach dem Prinzip der modellbasierten Generierung von Entwurfsunterlagen aus dem Informationsmodell generiert werden.

Voraussetzung hierfür sind **Teilmodelle** für die **Mechanik** und die **Steuerungskonfiguration**, die die vorhandenen Maschinenmodule und die NC-Konfigurierungsobjekte abbilden. Zur formalen Beschreibung der funktionalen Zusammenhänge müssen die Teilmodelle miteinander verknüpft werden. Als Grundlage für das Teilmodell der Steuerung müssen die NC-Steuerungsobjekte definiert werden. Dadurch entsteht eine logische Verknüpfung zwischen Maschinenmodulen und NC-Konfigurierungsobjekten.

Über die **Zuordnung** von **Maschinenmodulen** zu **NC-Konfigurierungsobjekten** kann an Hand der automatisch erkannten Module der Maschinenkonfiguration abgeleitet werden, welche NC-Konfigurierungsobjekte benötigt werden. Zusätzlich legt eine **Referenzarchitektur** in einem Konfigurationsmuster fest, wie die NC-Konfigurierungsobjekte zu einer NC-Konfiguration kombiniert werden müssen. Hierzu ist durch Abstraktion von Steuerungskonfigurationen, eine Referenzarchitektur für eine adaptierbare modulare Steuerung (Mustervorlage) zu definieren. Grundlage für diese Referenzarchitektur wird das in Kapitel 5.3.2 entwickelte Strukturmodell einer adaptierbaren Steuerung sein.

6.2.3 Auslesen von Maschinenmodulparametern

PnP-fähige Geräte haben einen **definierten Speicherbereich** in dem standardisierte Konfigurierungsinformationen abgelegt sind. Bei dem FireWire-Bus wird beispielsweise der Aufbau des Datenbereichs nach der sogenannten CSR-Architektur genau spezifiziert (**Bild 6.2**). Dort ist festgelegt wie der Adressbereich des Speichers aufgebaut ist. Ein Gerät (Master), das auf das PnP-Gerät zugreift, weiß dadurch wo die Information abgespeichert ist und was die Daten bedeuten.

Für die Anwendung dieses Prinzips bei Maschinenmodulen muss daher der **PnP-Gerätegedanke** auf Werkzeugmaschinenmodule **übertragen** werden. Das bedeutet zum einen, dass Maschinenmodule einen auslesbaren Konfigurierungsspeicher benötigen und zum anderen, dass die Module über eine Kommunikationssystem miteinander verbunden sein müssen. Das Kommunikationssystem muss das Lesen und Schreiben von Daten in und aus dem Konfigurierungsspeicher während der Konfigurierung in der Initialisierungsphase der NC-Steuerung unterstützen. Diese Funktionen werden benötigt, damit die Parameter automatisch ausgelesen und geschrieben werden können, ohne dass davor schon die gesamte NC-Konfigurierung abgeschlossen sein muss. Die Parameter können dann mit einer Grundeinstellung in den NC-Konfigurierungsobjekten im Informationsmodell hinterlegt sein und können anschließend mit den Werten aus dem Speicher auf dem Maschinenmodul ergänzt und konkretisiert werden. Voraussetzung hierfür ist, dass sie schon vorher durch eine Voreinstellung im Maschinenmodul gespeichert sind.

6.3 Anforderungen an Plug-and-Play-fähige Maschinenmodule

In der Automatisierungstechnik gibt es lediglich sogenannte intelligente Feldgeräte. Damit werden Automatisierungskomponenten bezeichnet, die über eine Kommunikationsschnittstelle und einen Prozessor verfügen, der Informationsverarbeitungen dezentralisiert. Diese Geräte erfüllen aber keine der zehn Hauptfunktionen einer Werkzeugmaschine /88/. Dazu sind

mechatronische Module notwendig, die mechanische und steuerungstechnische Funktionen vereinen und eine abgeschlossene, logische Einheit bilden. Sie sind außerdem notwendig, damit eine Zuordnung zu NC-Konfigurierungsobjekten möglich ist.

Aus den Einsatzbedingungen im Werkzeugmaschinenbau und den Methoden zur Automatisierung der Inbetriebnahmetätigkeiten (Kapitel 6.2) leiten sich für das modulare Maschinenkonzept und seine Module folgende Anforderungen ab:

1. Die Module müssen **mechatronische, funktionale Einheiten** bilden damit ein Klassifikationssystem für die Zuordnung von Funktionen zu Modulen aufgebaut werden kann.
2. Module sollten mehrere **Betriebsarten** unterstützen. Sie sollten mindestens einen normalen Betriebszustand und einen **Konfigurierungszustand** unterstützen. Im laufenden Betrieb darf die Echtzeitkommunikation nicht durch Hinzufügen, Austauschen oder Entfernen von Modulen unterbrochen werden.
3. Das **Kommunikationssystem**, welches die Module verbindet, muss eine **Konfigurierungsphase** unterstützen. Im Betriebszustand muss die Echtzeitkommunikation sichergestellt sein. Im Konfigurierungszustand reicht eine einfache Kommunikationsverbindung zum Lesen und Schreiben von Konfigurierungsparametern für die Konfigurierung des Gesamtsystems aus.
4. Für die Speicherung von Konfigurierungsinformation müssen die Module einen **Konfigurierungsspeicher** haben. Über eine Kommunikationsschnittstelle muss dieser Konfigurierungsspeicher auslesbar und beschreibbar sein.
5. Die unterschiedlichen Modulzustände in einer modularen Maschinenkonfiguration müssen koordiniert werden. Daher ist eine zentrale Einheit (**Koordinator**) notwendig, der den Zustand der Maschinenkonfiguration steuert.

6.4 Voraussetzungen für die Übertragbarkeit von Plug-and-Play-Prinzipien

Aus jedem NC-Konfigurierungsschritt lässt sich ableiten, welche **Voraussetzungen** für die Anwendung der Methoden zur Automatisierung der Inbetriebnahmetätigkeiten (Kapitel 6.2) geschaffen werden müssen (**Bild 6.6**). Teilweise überschneiden sich diese, da die notwendigen, wesentlichen PnP-Systemeigenschaften in jeder Methode verwendet werden.

Die PnP-Prinzipien erfordern Geräte mit den in Kapitel 6.1.2 analysierten Eigenschaften. Das bedeutet, dass für die Übertragung der PnP-Prinzipien auf den Bereich Werkzeugmaschinen ein Maschinenkonzept basierend auf Modulen erforderlich ist. Damit die PnP-Methode zur

Identifizierung und Klassifizierung übertragen werden können, müssen **mechatronische Module mit PnP-Eigenschaften** geschaffen werden. Diese müssen über ein Kommunikationssystem miteinander verbunden sein, das eine Basiskommunikation zum Lesen und Schreiben von Parametern während der Konfigurierungsphase und eine Echtzeitkommunikation für den Normalbetrieb bereitstellt. Außerdem muss es Funktionen zur **Selbstkonfigurierung der Kommunikation** besitzen, da bereits nach dem Einschalten bei der Initialisierung des Gesamtsystems eine Kommunikation für die Selbstadaptionmechanismen der Steuerung benötigt wird.

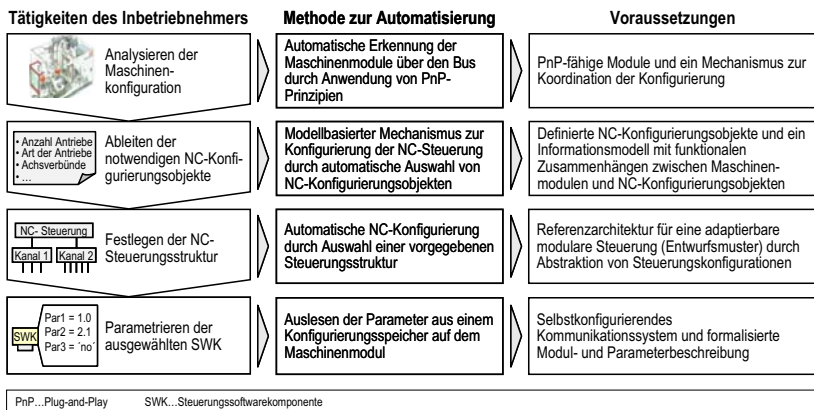


Bild 6.6: Notwendige Voraussetzungen zur Automatisierung der Inbetriebnahme

Bei einer Konfigurierung einer aus Modulen bestehenden Maschinenkonfiguration, müssen die einzelnen Modulzustände so gesteuert werden, dass das Gesamtsystem **koordiniert** konfiguriert und für den Normalbetrieb vorbereitet wird. Dies macht bisher der Inbetriebnehmer. Für die Übertragung der PnP-Prinzipien ist es daher notwendig, dass ein Mechanismus zur Koordinierung der Modul- und Systemkonfigurierung geschaffen wird. Hierzu ist ein **Konfigurierungskordinator** notwendig, der die Konfigurierung aller Module schrittweise so steuert, dass sie kontrolliert den Betriebszustand verlassen und in einen Konfigurierungs- und Initialisierungszustand wechseln. Dies setzt voraus, dass die Module einen einheitlichen Zustandsautomaten mit einem Konfigurierungszustand haben in dem sie konfiguriert werden können.

Für das automatische Ableiten von NC-Konfigurierungsobjekten und zur Auswahl einer Steuerungsstruktur muss eine Methode zur **modellbasierten Konfigurierung** von NC-Steuerungen entwickelt werden. Damit baukastenbasierte Methoden übertragen werden kön-

nen, müssen die funktionalen Zusammenhänge zwischen Maschinenmodulen und NC-Konfigurierungsobjekten in einem Informationsmodell beschrieben werden. Damit kann innerhalb eines vorgedachten Rahmes für eine Maschinenkonfiguration anhand von Maschinenmodulen automatisch eine NC-Konfiguration abgeleitet werden.

Dazu müssen zunächst **NC-Konfigurierungsobjekte** und eine **Steuerungsreferenzarchitektur** für eine adaptierbare NC-Steuerung definiert werden. Die Referenzarchitektur soll durch ein Konfigurationsmuster die Struktur der NC-Steuerung und die Kombination der NC-Konfigurierungsobjekte innerhalb der Struktur festlegen. Zusätzlich muss ein Klassifikationssystem für Module geschaffen werden, damit eine funktionale Zuordnung zwischen Maschinenmodulen und NC-Konfigurierungsobjekten aufgebaut werden kann. Die Klasse legt außerdem fest, welche Parameter das Modul in seinem Konfigurierungsspeicher für die Konfigurierung zur Verfügung stellt und welche Bedeutung diese haben.

Die NC-Konfigurierungsobjekte, die Referenzarchitektur, die funktionalen Zusammenhänge zwischen Maschinenmodulen und NC-Konfigurierungsobjekten und das Klassifikationssystem müssen zusammen in einem **Informationsmodell** abgebildet werden.

Für die Übertragung des Plug-and-Play-Prinzips auf die Konfigurierung von NC-Steuerungen sind zusammenfassend die in Bild 6.7 dargestellten Voraussetzungen notwendig.

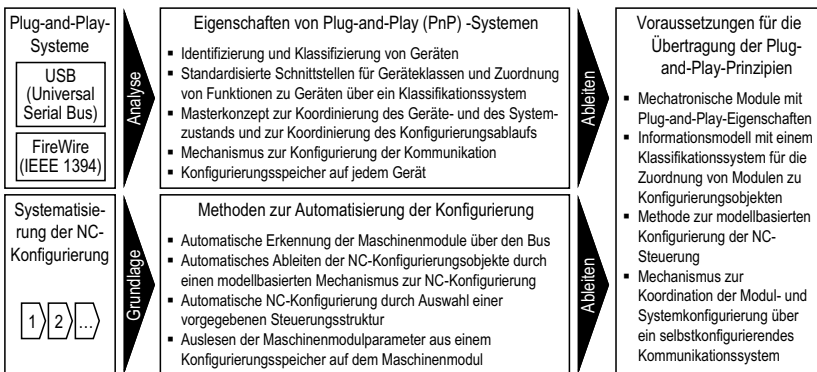


Bild 6.7: Notwendige Voraussetzungen für die Übertragung der PnP-Prinzipien

6.5 Methode zur Selbstadaption von NC-Steuerungen

Nach der Analyse der Eigenschaften von PnP-Systemen und der Definition der notwendigen Voraussetzungen für die Übertragbarkeit der PnP-Prinzipien wird in diesem Kapitel eine grundlegende und durchgängige **Methode zur Selbstadaption** von NC-Steuerungen nach

dem PnP-Prinzip entworfen, indem die bisherigen Vorüberlegungen zusammengeführt werden. Randbedingung dieses Entwurfs ist, dass die in Kapitel 6.3 definierten Voraussetzungen geschaffen wurden. Die Methode zur Selbstadaption wird anschließend in Kapitel 7 detailliert umgesetzt.

Bild 6.8 zeigt den Ablauf einer Selbstadaption. Nachdem in einer ersten Phase eine Maschine durch Kombination, Austausch oder Entfernen von PnP-fähigen Modulen konfiguriert wurde, beginnt das Kommunikationssystem nach dem Einschalten mit der **Initialisierung** der Kommunikation und stellt eine Basiskommunikation zur Verfügung.

Die Basiskommunikation erlaubt es, in einer zweiten Phase alle Kommunikationsteilnehmer zu ermitteln und zu **identifizieren**. Die Identifikation erfolgt über die Kennung der PnP-fähigen Maschinenmodule, die einen Konfigurierungsspeicher haben, in dem diese Information abgespeichert ist. Die Identifizierung der Module wird durch einen **Konfigurierungskordinator** gesteuert. Er erstellt eine Liste aller Maschinenmodule (Kommunikationsteilnehmer) mit ihren Kennungen. Anhand der Kennung bestimmt er modellbasiert die **Modulklasse**. Dazu nutzt er das Teilmodell Mechanik des **Informationsmodells** in dem das Klassifikationssystem formalisiert ist. Durch das Klassifikationssystem ist die Funktion des Maschinenmoduls innerhalb der Maschinenkonfiguration definiert.

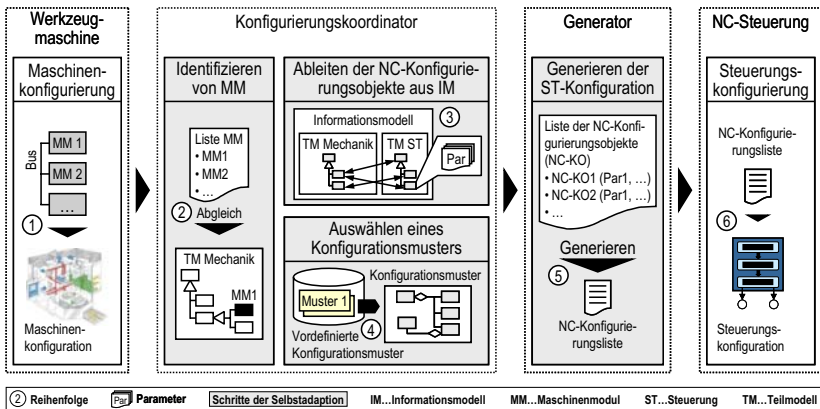


Bild 6.8: Ablauf einer Selbstadaption

Über die im Informationsmodell abgebildeten Zusammenhänge zwischen Maschinenmodulen und NC-Konfigurierungsobjekten leitet der Konfigurierungskordinator in einer dritten Phase die notwendigen **NC-Konfigurierungsobjekte** ab. In dem Informationsmodell müssen dazu NC-Konfigurierungsobjekte definiert werden, die als Datencontainer für herstellerepezifische

Konfigurierungsinformationen dienen. Sie stellen allgemeine Bausteine eines Baukastensystems dar und bilden die konfigurierbaren Objekte einer NC-Steuerung ab. Im Informationsmodell sind die NC-Konfigurierungsobjekte mit Parametern hinterlegt, die mit den ausgelesenen Werten aus dem Konfigurierungsspeicher des Moduls belegt werden.

Über die Zuordnungen im Informationsmodell werden alle notwendigen NC-Konfigurationselemente ermittelt. Ihr Zusammenwirken innerhalb der NC-Konfiguration muss in einer vierten Phase durch Auswahl eines **Konfigurationsmusters** festgelegt werden. Dies kann nicht automatisch erfolgen, da das Konfigurationsmuster abhängig von der geometrischen Anordnung der Achsen ausgewählt werden muss. Die Anordnung kann jedoch nicht vom Kommunikationssystem erkannt werden. Aufgrund der erkannten Anzahl Achsmodule können allerdings die Auswahlmöglichkeiten automatisiert vorsortiert und eingeschränkt werden, so dass sich die Vielfalt der Konfigurationsmuster auf die sinnvollen Muster beschränkt.

Anhand des ausgewählten Konfigurationsmusters erzeugt der Generator in der fünften Phase auf Basis des Informationsmodells die **NC-Konfigurierungsliste**. Dies erfolgt nach dem Prinzip der modellbasierten Konfigurierung und baukastenbasierten Generierung von Konstruktionsunterlagen. Durch den Generator werden die NC-Konfigurierungsobjekte des Informationsmodells in zielplattformspezifische NC-Konfigurierungslisten umgewandelt und mit den Parametern aus dem Informationsmodell parametrisiert. Der Generator berücksichtigt dabei die steuerungssystemspezifische Darstellung der Parameter in den Konfigurierungslisten (Syntax).

Eine Selbstadaption läuft zusammenfassend in den in [Bild 6.9](#) dargestellten Phasen ab.

Phase 1 Konfigurieren der Maschine	Phase 2 Identifizieren der Module	Phase 3 Ableiten der NC-KO	Phase 4 Auswählen eines KM	Phase 5 Generieren der KL	Phase 6 Adaptieren der NC-Steuerung				
<ul style="list-style-type: none"> • Konfigurierung der Maschine aus Modulen • Initialisierung des Kommunikationssystems 	<ul style="list-style-type: none"> • Auslesen der Modulkennung • Klassifizierung des Moduls im Informationsmodell 	<ul style="list-style-type: none"> • NC-KO über Zusammenhänge im IM ermitteln • Parameter auslesen und NC-KO parametrieren 	<ul style="list-style-type: none"> • Festlegung der Bauform ergibt Konfigurationsmuster im Informationsmodell 	<ul style="list-style-type: none"> • Aus NC-KO unter Verwendung einer Konfigurierungsvorlage Parameter in KL schreiben 	<ul style="list-style-type: none"> • Einlesen der KL • Internes Konfigurationslaufzeit-system adaptiert NC-Steuerung 				
<table border="1" style="width: 100%; text-align: center;"> <tr> <td>IM...Informationsmodell</td> <td>KL...Konfigurierungsliste</td> <td>KM...Konfigurationsmuster</td> <td>NC-KO...NC-Konfigurierungsobjekt</td> </tr> </table>						IM...Informationsmodell	KL...Konfigurierungsliste	KM...Konfigurationsmuster	NC-KO...NC-Konfigurierungsobjekt
IM...Informationsmodell	KL...Konfigurierungsliste	KM...Konfigurationsmuster	NC-KO...NC-Konfigurierungsobjekt						

Bild 6.9: Phasen der Selbstadaption

6.6 Zusammenfassung

In diesem Kapitel wurden, nach dem Vorbild der analysierten PnP-Prinzipien aus der Computertechnik, Methoden zur Automatisierung der Inbetriebnahmetätigkeiten entwickelt und Anforderungen an PnP-fähige Maschinenmodule definiert. Anschließend wurde eine durchgän-

gige Methode zur Selbstadaption von NC-Steuerungen nach dem PnP-Prinzip entworfen.

Zur Umsetzung der Selbstadaption werden zusammenfassend die in Bild 6.10 dargestellten Methoden und Konzepte angewendet.

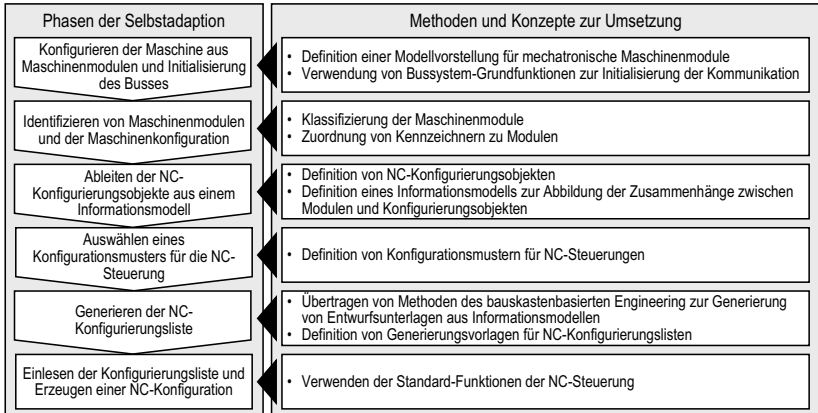


Bild 6.10: Methoden und Konzepte zur Umsetzung der Selbstadaption

Die Konfigurierung der Maschine aus Maschinenmodulen basiert auf einem **Baukastensystem** auf Basis mechatronischer Module mit PnP-Eigenschaften. Bei der Gestaltung des Baukastens muss der PnP-Gerätegedanke auf die Maschinenmodule übertragen werden. Dies führt zu funktional integrierten, mechatronischen Modulen mit wenigen Schnittstellen und abgeschlossenem Funktionsumfang. Zur Initialisierung des Busses nach dem Einschalten sollen Standard-Grundfunktionen des Bussystems genutzt werden.

Für das **Identifizieren** von Maschinenmodulen wird ein Klassifikations- und Kennzeichnungssystem benutzt. Dadurch wird eine Beziehung zum Informationsmodell geschaffen, in dem die logische Verknüpfung zwischen dem mechanischen Maschinenaufbau (Maschinenkonfiguration) und der NC-Steuerungsstruktur (NC-Konfigurierungsobjekte) abgebildet ist.

Das **Ableiten** der NC-Konfigurierungsobjekte erfolgt aus dem Informationsmodell. Aus den **formalisierten Zusammenhängen** zwischen Maschinenmodulen und NC-Konfigurierungsobjekten kann abgeleitet werden, welche NC-Konfigurierungsobjekte benötigt werden. Dazu sollen NC-Konfigurierungsobjekte definiert und ein Informationsmodell zur Abbildung der Zusammenhänge geschaffen werden.

Das Zusammenspiel der NC-Konfigurierungsobjekte in einer NC-Steuerungskonfiguration wird durch ein **Konfigurationsmuster** festgelegt. Für die modellbasierte Generierung sollen

als Vorlage Klassen von Konfigurationsmustern definiert werden.

Die **Generierung** der NC-Konfigurierungslisten soll nach den Methoden des baukastenbasierten Engineerings zur Generierung von Konstruktionsunterlagen erfolgen. Diese Methode ist auf die Generierung von NC-Konfigurierungslisten zu übertragen. Die NC-Konfigurierungslisten können aus dem Informationsmodell generiert werden, nachdem durch ein Konfigurationsmuster eine NC-Struktur festgelegt wurde. Die eigentliche Anpassung der NC-Steuerung erfolgt durch die internen Standardmechanismen des **Konfigurationslaufzeitsystems** der NC-Steuerung aufgrund der generierten NC-Konfigurierungsliste.

7 Konzept für eine selbstadaptierende NC-Steuerung

Nach der Analyse der Eigenschaften von PnP-Systemen und der Analyse der Voraussetzungen für die Übertragbarkeit des PnP-Prinzips in Kapitel 6 soll in diesem Kapitel das PnP-Prinzip auf die Steuerungstechnik übertragen und damit ein **Konzept** für eine **selbstadaptierende NC-Steuerung** erarbeitet werden (Bild 7.1). Dazu müssen in der Steuerungstechnik die in Kapitel 6.4 erarbeiteten **Voraussetzungen** für die Übertragung der PnP-Prinzipien geschaffen werden.

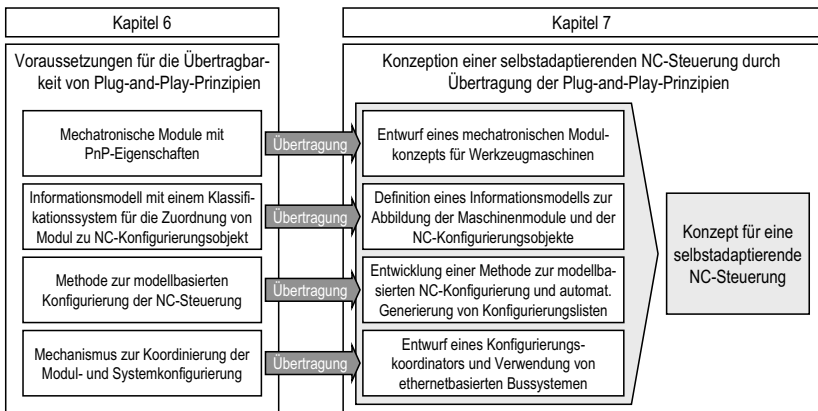


Bild 7.1: Übertragung der Plug-and-Play-Prinzipien

7.1 Modulkonzept für Werkzeugmaschinen auf Basis mechatronischer Module

Das im Rahmen dieser Arbeit vorgestellte modulare Maschinenkonzept wurde im Rahmen des SFB 467 von dem Institut für Werkzeugmaschinen (IfW) und dem Institut für Steuerungstechnik der Werkzeugmaschinen und Fertigungseinrichtungen (ISW) erarbeitet /136/. Die Grundlage dieses Maschinenkonzepts bilden **mechatronische, autarke Maschinenmodule** /25, 95/. Dieses Modulkonzept wurde im Rahmen des TFB 059 /137/ beispielhaft umgesetzt und demonstriert.

7.1.1 Mechatronische Module

Während des Konstruktionsprozesses entsteht eine **bauliche Trennung** von funktional zusammengehörenden Teilkomponenten. Die elektrischen Komponenten sind im Schaltschrank untergebracht, das Grundgestell der Maschine bildet eine Einheit, abgeschlossen durch die Maschinenabdeckung, die Nebenaggregate (z.B. Hydraulikaggregat) sind meist neben der

Maschine untergebracht. Für die Funktionserfüllung wird jedoch eine Kombination unterschiedlicher Teilkomponenten benötigt. Da sie nicht alle im selben Bauraum untergebracht sind, werden sie über Schnittstellen miteinander verkoppelt. Die baulichen Systemgrenzen unterscheiden sich dadurch stark von den funktionalen Systemgrenzen und stimmen nicht immer überein. Dadurch entstehen zusätzliche, oft unnötige Modulschnittstellen, die eine Rekonfigurierung der Maschine erschweren.

Durch eine **Harmonisierung** der funktionalen und baulichen Systemgrenze können **mechatronische, autarke Module** geschaffen werden, deren Systemgrenzen so gewählt sind, dass möglichst wenige Schnittstellen zur Umwelt notwendig sind. Harmonisierung der Systemgrenzen bedeutet ein weitestgehendes zur Deckung bringen von baulichen und funktionalen Modulgrenzen, indem Funktionen, die bisher peripher erbracht und zugeführt werden, im Modul integriert sind. Dadurch vereinfachen und reduzieren sich die Schnittstellen zwischen Modulen. Es entstehen mechatronische Einheiten, die ein abgeschlossenes autarkes System bilden. Alle Elemente, die für die Realisierung einer Modulfunktion notwendig sind, sind im Modul untergebracht. Steuerungs-, Kommunikations- und (alternativ) Leistungssteuerungskomponenten sind ebenso integriert, wie geschlossene autarke Kreisläufe für Schmierung und Druckluftversorgung. Die entsprechenden Versorgungsleitungen hin zum Modul können dann ebenso eingespart werden, wie die zentralen Hilfs- und Nebenfunktionen, da sie in den Modulen selbst erbracht werden /24, 154/.

Bild 7.2 skizziert, wie bei einer konsequenten Zusammenfassung von Haupt- und Nebenfunktionen in Modulen ein **rekonfigurierbares System** entsteht /53, 155/. Dabei kommt es auf die richtige Zusammenfassung der zusammengehörenden Teilfunktionen innerhalb eines Moduls an.

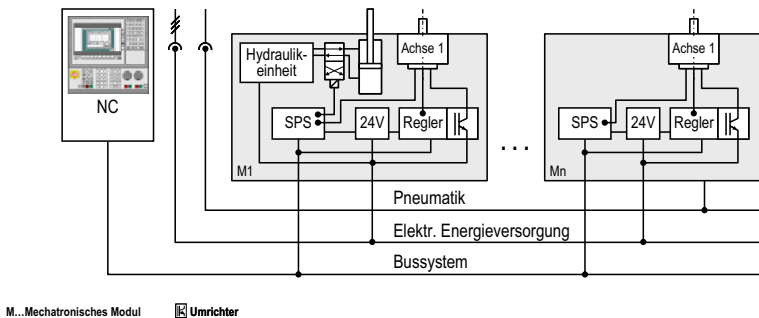


Bild 7.2: Modulares Maschinenkonzept auf Basis mechatronischer Module

Eine Maschinenkonfiguration setzt sich dann aus mechatronischen Modulen zusammen, die über schnell wechselbare mechanische, energie- und informationstechnische Schnittstellen miteinander verbunden sind. Da die hydraulische Energie über dezentrale Einheiten erzeugt wird, reduziert sich die Modulschnittstelle auf eine Leitung für Information (Bussystem) und je eine Leitung für die elektrische und die pneumatische Energie. Über das Bussystem werden die Module bei der Konfigurierung der Werkzeugmaschine informationstechnisch verknüpft, so dass die Steuerung mit den Modulen kommunizieren kann.

7.1.2 Steuerungstechnischer Aufbau von mechatronischen Modulen

Zur Anbindung an das Kommunikationssystem der Maschine hat die Hardware der mechatronischen Module eine Kommunikationsschnittstelle. Zusätzlich können mechatronische Module ein internes, nach außen gekapseltes Bussystem beinhalten, an dem Antriebe, Ein- und Ausgangsmodule, sowie Sensoren, Aktoren und dezentrale Feldgeräte angeschlossen werden (Bild 7.3) /156, 157/.

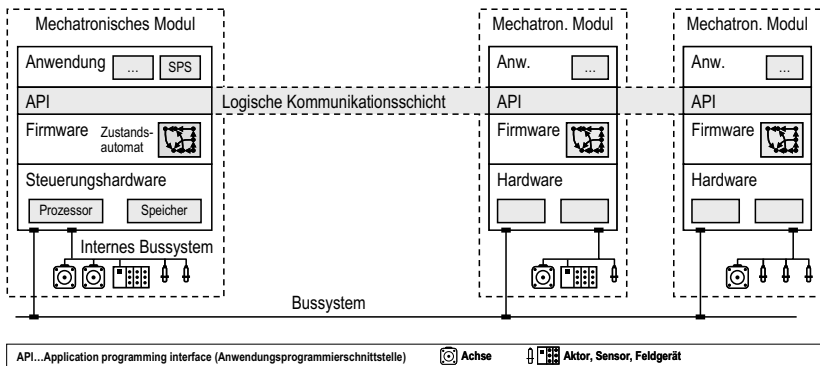


Bild 7.3: Mechatronische Module aus kommunikations- und steuerungstechnischer Sicht

Zur Erfüllung dezentraler steuerungstechnischer Funktionen, die direkt einem Modul zu zuordnen sind, besitzen mechatronische Module einen eigenen Prozessor. Damit können lokal Eingangs-, Ausgangs- und Sensorsignale verarbeitet werden. Im Konfigurierungsspeicher sind die Daten für die Konfigurierung und ein Kennzeichner (ID) für die Erkennung des Moduls gespeichert. Zusätzlich können dort modulspezifische Parameter abgespeichert werden, die für die Adaption der Steuerung notwendig sind (z.B. physikalische Grenzwerte einer Achse, wie max. Beschleunigung und Verfahrweg). Je nach erforderlicher Leistungsfähigkeit kann die Steuerungshardware ein industrietauglicher Mini-PC sein mit Soft-SPS und NC-Kern als Anwendung.

Das Bild 7.3 zeigt eine Maschinenkonfiguration aus **kommunikations-** und **steuerungstechnischer Sicht**. Jedes Modul hat eine Steuerungshardware mit Firmware. In der Firmware ist ein Zustandsautomat für das PnP-Verhalten implementiert (Kapitel 7.4.2). Außerdem bietet die Firmware eine definierte Schnittstelle für die Anwendungsprogrammierung (API) für die darüberliegenden Anwendungen an. Die Firmware sorgt dafür, dass das Netzwerk transparent und die physikalische Verteilung der mechatronischen Module abstrahiert ist. Die physikalische Kommunikation nach außen, zu anderen Modulen, erfolgt über ein Bussystem. Dies ist für die Anwendungen allerdings nicht sichtbar, da sie über die logische Kommunikationsschicht der Firmware kommunizieren, so als ob sie alle auf derselben Hardware abliefen.

7.1.3 Baukastensystem auf Basis mechatronischer Module

Im Rahmen des SFB 467 /136/ wurden bestehende Maschinenkonzepte unter **funktionalen Gesichtspunkten** analysiert und ohne Berücksichtigung der baulichen Einheit in **Haupt-** und **Nebenfunktionen** zerlegt (Bild 7.4). Hauptfunktionen sind für die Erzeugung der Relativbewegung zwischen Werkzeug und Werkstück notwendig und sind im Kraftfluss integriert und direkt an der Spannerzeugung beteiligt. Nebenfunktionen erbringen periphere Hilfsfunktionen und tragen nur indirekt zur Spannerzeugung bei, wie beispielsweise Schmierung. Nach der funktionalen Analyse wurden neue Modulgrenzen festgelegt und zwar so, dass sich funktional zusammengehörende Komponenten in einem Modul befinden und sich die Ein- und Ausgangsschnittstellen der Module reduzieren.

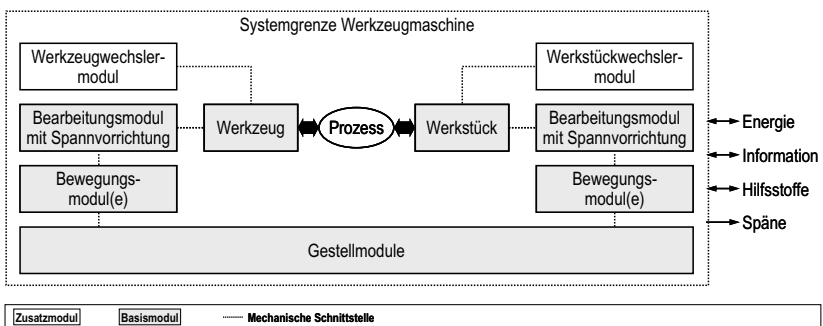


Bild 7.4: Funktionaler Aufbau und Schnittstellen einer Werkzeugmaschine (nach /136/)

Auf Basis dieses mechatronischen Modulgedankens wurde ein **Baukastensystem** für **rekonfigurierbare Werkzeugmaschinen** definiert. Die Module des Baukastens basieren auf DIN-Normen für Gestelleinheiten (DIN 69521 bis 69526), Schlitteneinheiten (DIN 69572) und Spindleeinheiten (DIN 69641 bis 69643). Die Schnittstellen der Module sind so konstru-

iert, dass sie schnell wechselbar und standardisiert aber dennoch genügend steif und genau sind /25, 93, 138, 139/. Medienflüsse über die Systemgrenze hinaus gibt es nicht. Diskrete hydraulische und pneumatische Druckerzeuger sind überall, wo nötig, in die Module integriert. Bild 7.5 zeigt das Baukastensystem für rekonfigurierbare Werkzeugmaschinen mit den folgenden Basis- und Zusatzmodulen:

- Gestellmodule (stationär, mobil oder koppelnd)
- Bewegungsmodule zur Erzeugung der Relativbewegung
- Bearbeitungsmodule mit Spann- und Hilfsvorrichtungen
- Entsorgungsmodule (z.B. Späneförderer)
- Zusatz- und Hilfsmodule (z.B. Werkzeug- und Werkstückwechslermodule)

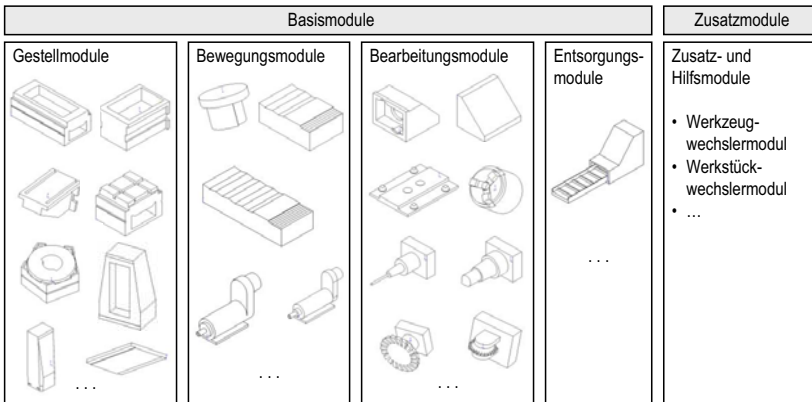


Bild 7.5: Baukastensystem für rekonfigurierbare Werkzeugmaschinen /136/

Gestellmodule können stationäre Einheiten (z.B. Gestell, Rundtisch), mobile Einheiten (z.B. Ständer) oder koppelnde Einheiten sein. Sie bilden die Grundstruktur der Maschine und haben funktional die Aufgabe, Kräfte aufzunehmen. Die **mobilen Einheiten** sind im Gegensatz zu den **stationären Einheiten** beweglich, weil sie auf Bewegungseinheiten angebracht werden. Koppelnde Einheiten stellen eine fixe Verbindung zwischen Gestellmodulen her. Zur Erzeugung einer Bewegung müssen in die stationären und mobilen Einheiten **Bewegungseinheiten** eingebaut werden. Durch diese funktionale Trennung ist ein modularer Aufbau möglich. Außerdem können die Bewegungseinheiten je nach Anforderung nach unterschiedlichen Prinzipien aufgebaut sein (z.B. Achse mit Kugelgewindetrieb oder Lineardirektantrieb). Spindeln

gehören zu den Bewegungsmodulen. Sie sind für die Bewegungserzeugung des Werkzeugs verantwortlich. Sie können auch im Interpolationsverbund mitinterpoliert werden, wenn eine synchronisierte Bewegung wie beispielsweise beim Gewindeschneiden benötigt wird. **Bearbeitungseinheiten** mit Spann- und Hilfseinheiten dienen zur Aufnahme und Fixierung von Werkzeugen und Werkstücken. **Entsorgungseinheiten** leiten Hilfsstoffe wie z.B. Kühlschmierstoff oder Späne aus der Maschine. Werkzeug- und Werkstückwechslersmodul gehören zu den **Zusatz-** und **Hilfsmodulen**. Sie erfüllen nur Nebenfunktionen und sind als Erweiterung zu den Basismodulen zu sehen, da sie den Automatisierungsgrad der Werkzeugmaschine erhöhen. Weitere Zusatzmodule könnten ein Werkzeugmagazin für die automatisierte Verwaltung und Bereitstellung des Werkzeugs und ein Verkettungsmodul zur Verkettung des Werkstückflusses sein.

7.2 Definition eines durchgängigen mechatronischen Informationsmodells

Die **Kombinationsregel** (Systematik), wie aus Modulen Maschinen konfiguriert werden, ist nicht formalisiert und individuell in einem mentalen Modell des Inbetriebnehmers abgebildet. Da sich die Maschine aus Modulen zusammensetzt, bildet der Inbetriebnehmer die Modularität der Maschinenkonfigurationen in seinem **mentalen Modell** selbst ab. Die Zuordnung von NC-Konfigurierungsobjekten zu Maschinenmodulen macht der Inbetriebnehmer im Kopf, indem er funktional gleiche Module zu Klassen zusammenfasst. So kann er gedanklich einer Menge funktional gleicher Module ein NC-Konfigurierungsobjekt zuordnen.

Für die Methode zur Selbstadaption von NC-Steuerungen muss ein **Informationsmodell** aufgebaut werden, welches das mentale Modell des Inbetriebnehmers **abstrahiert** und **formalisiert** abbildet ([Bild 7.6](#)).

Durch Modularisierung der Maschinenkonfigurationen und Abstraktion des mentalen Modells lässt sich ein Baukasten beschreiben aus dem sich durch Kombination von Maschinenmodulen nach Vorgabe der Bauform (Kinematik) Maschinen konfigurieren lassen. Durch funktionale Analyse der Module und Abstraktion lassen sich funktional gleiche Module zu Klassen zusammenfassen und Relationen zu NC-Konfigurierungsobjekten abbilden. Damit können dann aus dem Informationsmodell mit **baukastenbasierten Generierungsmethoden** NC-Konfigurierungslisten erzeugt werden.

Zur Abbildung im Informationsmodell auf der Meta-Meta-Ebene werden die realen Objekte zu Modellierungsobjekten abstrahiert ([Bild 7.7](#)). **Modellierungsobjekte** sind allgemein die Elemente des Informationsmodells auf der Meta-Meta-Ebene. Modellierungsobjekte sind **Steuerungsmodellobjekt**, **Mechanikmodellobjekt**, **Relation** und **Maschinenmodellobjekt**. Eine Werkzeugmaschine mit ihrer Maschinen- und Steuerungskonfiguration wird auf der Me-

ta-Meta-Ebene des Informationsmodells als Maschinenmodellobjekt abgebildet.

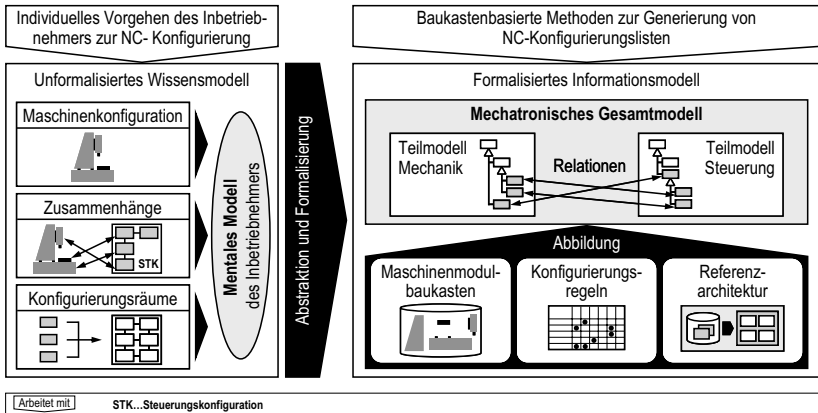


Bild 7.6: Vorgehen zur Formalisierung und Abbildung des mentalen Modells

Zur Abbildung der Zusammenhänge werden im Informationsmodell Relationen zwischen Steuerungs- und Mechanikmodellobjekten definiert. Dies können Kinematik- oder Funktionsrelationen zur Abbildung von kinematischen und der funktionalen Beziehungen sein. Über die **Kinematikrelation** wird der Zusammenhang zwischen NC-Steuerungsstruktur und Maschinenaufbau abgebildet. Der Zusammenhang zwischen Maschinenmodulen und NC-Konfigurierungsobjekten wird über **Funktionsrelationen** abgebildet.

Damit an Hand von Maschinenmodulen modellbasiert NC-Konfigurierungslisten erzeugt werden können, müssen in diesem Informationsmodell die Maschinenmodule funktional klassifiziert und die Abhängigkeiten zwischen dem mechanischen Aufbau der Maschine (Maschinenkonfiguration) und der NC-Steuerung (NC-Konfiguration) abgebildet werden. Dazu sind **Teilmodelle** für die **Mechanik** und die **Steuerung** zu entwerfen und miteinander zu verknüpfen.

Das Teilmodell der NC-Steuerung muss die logischen **NC-Konfigurierungsobjekte** des Strukturmodells einer adaptierbaren NC-Steuerung (Kap. 5.3.2) enthalten. Die NC-Konfigurierungsobjekte sind nicht nur Modellierungselemente sondern sind gleichzeitig Datencontainer für Konfigurierungsinformationen. So können herstellerspezifisch zu implementierende Generatoren daraus die NC-Konfigurierungslisten erzeugen. Zusätzlich müssen im Teilmodell Steuerung die möglichen NC-Konfigurationen, die der Inbetriebnehmer in seinem mentalen Modell kennt, als Regelwerk in einer **Referenzarchitektur** abstrahiert werden. In

dieser Referenzarchitektur müssen die Konfigurierungsmöglichkeiten der NC-Konfigurierungsobjekte in Form von **Konfigurierungsmustern** allgemein abgebildet sein. Im Informationsmodell wird dies auf der Meta-Ebene durch die Steuerungsmodellobjekte *Referenzarchitektur* und das Steuerungsmodellobjekt *NC-Konfigurierungsobjekt* abgebildet. Das Steuerungsmodellobjekt *Referenzarchitektur* setzt sich aus Steuerungsmodellobjekten *NC-Konfigurierungsobjekt* zusammen und legt die Struktur der NC-Steuerung fest.

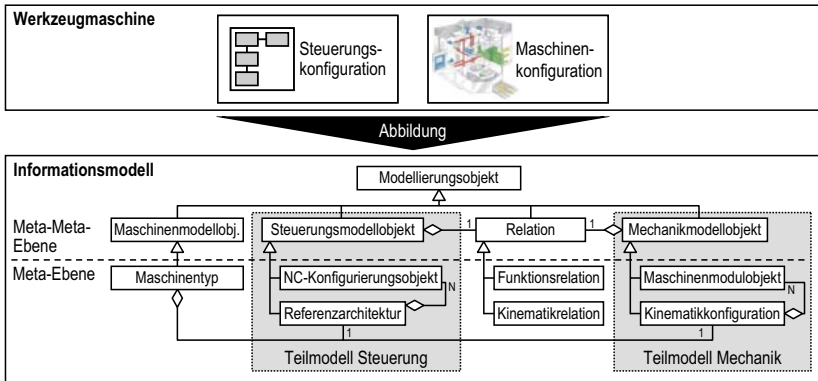


Bild 7.7: Metamodell einer Werkzeugmaschine

Das Teilmodell für die Mechanik muss die Maschinenmodule und deren Kombinationsmöglichkeiten zu einer Maschinenkonfiguration abbilden. Der Aufbau der Maschine wird im Informationsmodell durch das Mechanikmodellobjekt *Kinematikkonfiguration* festgelegt. Damit ist definiert, wie die Maschinenmodule einer Maschinenkonfiguration kombiniert und angeordnet werden können. Im Informationsmodell werden die Maschinenmodule als *Maschinenmodulobjekte* auf der Meta- Ebene abstrahiert und funktional klassifiziert.

Das Maschinenmodell bildet *Maschinentypen* der Meta-Ebene ab. Sie haben eine Referenzarchitektur und eine Kinematikkonfiguration und bilden logisch den Zusammenhang der Realität ab.

7.2.1 Definition von NC-Konfigurierungsobjekten

Die Analyse der herstellerübergreifenden Konfigurierbarkeit von NC-Steuerungen in Kapitel 5.3 zeigt, dass der Funktionsumfang der am Markt üblichen NC-Steuerungen und der interne Aufbau von NC-Steuerungen vergleichbar ist und die zu berücksichtigenden Abhängigkeiten zwischen Maschinen- und Steuerungskonfiguration herstellerunabhängig sind. Daher sind die Konfigurierungsmöglichkeiten der NC-Steuerungen ähnlich und die Maschinendaten

in den Konfigurierungslisten sind semantisch dieselben, obwohl sie herstellerspezifisch beschrieben sind.

Daher werden im Folgenden, ausgehend von der Analyse der herstellerübergreifenden Anpassbarkeit von NC-Steuerungen (Kapitel 5.3), allgemeingültige **NC-Konfigurierungsobjekte** für die NC-Steuerung definiert (NC-KO). Bild 7.8 zeigt das Vorgehen zur Definition von NC-Konfigurierungsobjekten und die Abbildung in einem Informationsmodell.

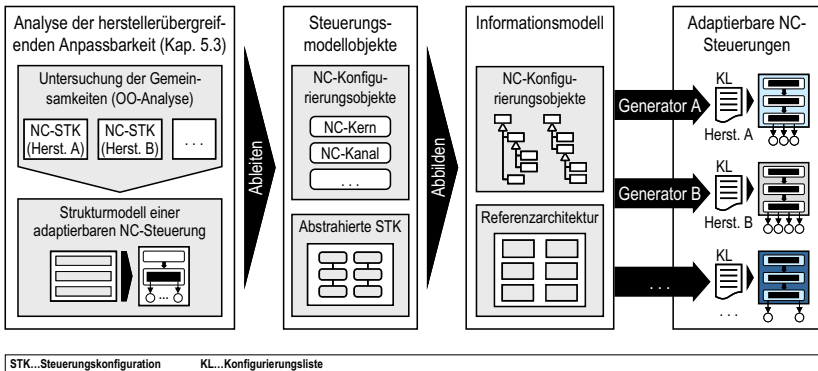


Bild 7.8: Vorgehen zur Definition und Abbildung von NC-Konfigurierungsobjekten

Aus dem Strukturmodell einer adaptierbaren NC-Steuerung (Kapitel 5.3.2) werden durch Formalisierung die **Steuerungsmodellobjekte** für das Informationsmodell abgeleitet. Im Informationsmodell werden die **NC-Konfigurierungsobjekte** (NC-KO) als Klassen und die abstrahierten Steuerungskonfigurationen als Konfigurationsmuster in Form einer **Referenzarchitektur** abgebildet. Aus NC-KO wird eine Steuerungskonfiguration erstellt, indem sie gemäß den Vorschriften der Referenzarchitektur konfiguriert und parametrieren werden. Die NC-KO sind nicht nur Strukturelemente der Steuerungskonfiguration sondern auch Datencontainer für die Konfigurierungs- und Parametrierungsinformation, die im Informationsmodell abgespeichert werden. Mit Hilfe von Generatoren lassen sich daraus zielplattformspezifische NC-Konfigurierungslisten erzeugen.

Das Strukturmodell einer adaptierbaren NC-Steuerung beschreibt allgemein welche Elemente eine adaptierbare NC-Steuerung hat und wie der interne Aufbau dieser adaptierbaren NC-Steuerung ist. Daraus lassen sich durch Abstraktion die NC-KO und die Referenzarchitektur als **Steuerungsmodellobjekte** für das Informationsmodell ableiten (Bild 7.9).

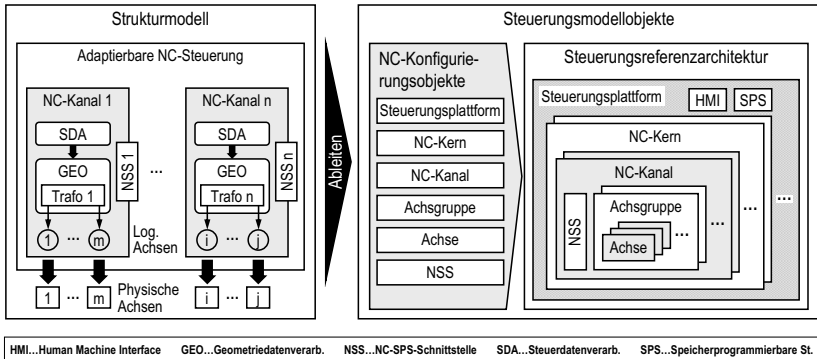


Bild 7.9: Ableitung der Steuerungsmodellobjekte aus dem Strukturmodell der adaptierbaren NC-Steuerung

Zur Konfiguration einer NC-Steuerung müssen die folgenden NC-KO im Informationsmodell abgebildet werden:

- Steuerungsplattform,
- NC-Kanal
- Achsguppe
- Achse
- NC-SPS-Schnittstelle

Für die Definition des Informationsmodells und die Entwicklung der modellbasierten Konfigurationsmethode wird vereinfachend davon ausgegangen, dass die NC-KO ähnliche Konfigurationsinformationen wie die Architekturobjekte in OSACA abbilden, da das Strukturmodell der adaptierbaren Steuerung an das modulare OSACA-Systemkonzept /85/ angelehnt ist. Im Fokus dieser Arbeit steht die Definition einer Methode zur Selbstkonfiguration und nicht die herstellerepezifische Detaillierung der Konfigurationsinformationen. Daher ist der genaue Umfang der Konfigurationsinformation der NC-KO nicht relevant. Das bedeutet auch, dass für einen herstellerepezifischen Ansatz, genau wie im OSACA-Projekt, durch eine objektorientierte Analyse die spezifischen Konfigurationsinformationen als Eigenschaften der NC-Konfigurationsobjekte definiert werden müssen. Eine Methode dafür wird in /105/ beschrieben.

Aus den NC-KO lässt sich eine Struktur einer NC-Steuerung aufbauen (Bild 7.10). Das NC-

KO Steuerungsplattform abstrahiert die Steuerungshardware auf der die Steuerungssoftware NC-Kern, HMI (Human Machine Interface) und der SPS als Anwendung ausgeführt wird. Der NC-Kern beinhaltet die NC-Steuerungssoftware. Wesentliches Strukturierungselement ist das **NC-KO NC-Kanal** mit dem **NC-KO Achsgruppe** und **NC-KO Achse**, sowie **NC-KO NC-SPS-Schnittstelle**. Das NC-KO *Achsgruppe* bildet die kinematischen Zusammenhänge der Transformation im Informationsmodell ab. Mit dem NC-KO *Achsgruppe* werden Bearbeitungsachsen zusammengefasst, die gemeinsam interpoliert werden müssen. Das NC-KO *Achse* ist eine Klasse und abstrahiert Bearbeitungs- und Hilfsachsen, die wiederum Linear- oder Rotationsachsen sein können.

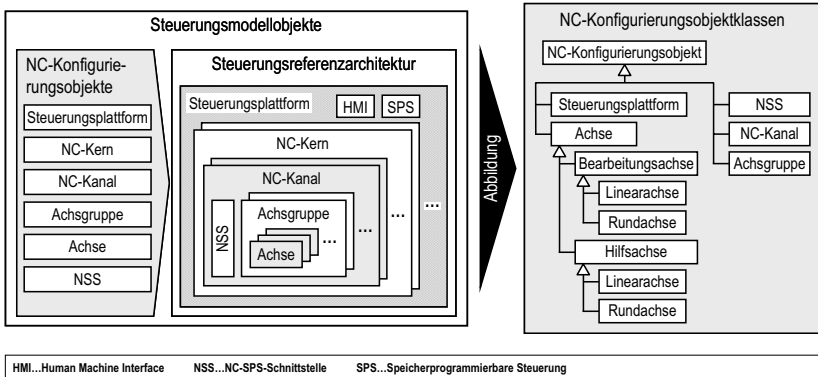


Bild 7.10: Abbildung der Steuerungsmodellobjekte als NC-Konfigurierungsobjektklassen

7.2.2 Definition von Referenzarchitekturen

Die **Referenzarchitekturen** werden nach der Anzahl ihrer Kanäle und dann nach der Anzahl ihrer Bearbeitungsachsen unterschieden. Dies wird auf der Klassen-Ebene des Informationsmodells beschrieben. Zusätzlich beschreibt die Referenzarchitektur, wie die NC-KO kombiniert werden müssen, damit die Steuerungskonfiguration korrekt ist. Dazu wird die auf der Meta-Ebene definierte „besteht-aus“-Beziehung detailliert.

Mit „besteht-aus“-Beziehungen wird ein **Konfigurationsmuster** für jede Architekturklasse beschrieben. Das Konfigurationsmuster beschreibt wie die NC-KO untereinander in Beziehung stehen. Allgemein schreibt die Referenzarchitektur vor, dass jede NC-Steuerung eine Steuerungsplattform mit einer Steuerungssoftware (NC-Kern) hat. Der NC-Kern enthält je nach Architekturklasse mindestens einen oder mehrere NC-Kanäle. Ein NC-Kanal hat eine NC-SPS-Schnittstelle und koordiniert Achsgruppen. Die für die Erzeugung der Relativbewe-

gung verantwortlichen Achsen werden als Bearbeitungsachsen bezeichnet. Sie werden in Achsgruppen zusammengefasst und synchronisiert. Bearbeitungsachsen einer Achsgruppe stehen über eine Transformation in einem geometrischen Zusammenhang. Ein NC-Kanal kann auch Hilfsachsen haben. Dies sind asynchrone Achsen, die nicht mit anderen Achsen synchronisiert werden.

Das [Bild 7.11](#) zeigt exemplarisch die Beschreibung des Steuerungsmodellobjekts *Referenzarchitektur* auf der Klassen-Ebene durch Detaillierung der „besteht-aus“-Beziehung der Meta-Ebene. Das Beispiel zeigt, wie sich die 3-achsige Architektur nach einem Konfigurationsmuster aus NC-KO zusammensetzt. Wesentliches Charaktermerkmal der 3-achsigen Architektur sind die drei, in der Achsgruppe zusammengefassten Bearbeitungsachsen.

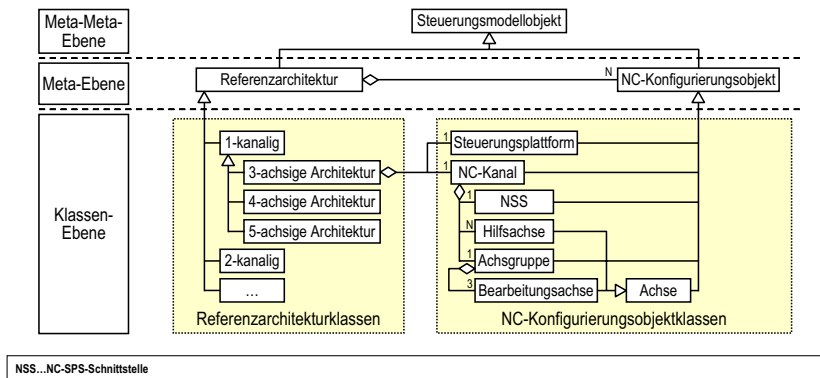


Bild 7.11: Referenzarchitektur mit Konfigurationsmuster für eine 3-achsige Architektur

7.2.3 Definition von Maschinenmodulobjekten

Bisherige Modellierungsansätze für Werkzeugmaschinen /84, 88/ definieren Werkzeugmaschinenkomponenten ausgehend vom Kraftfluss in der Maschine unter mechanischen und baulichen Gesichtspunkten ([Bild 4.3](#)). Da die Forschungsschwerpunkte dort im wesentlichen die Konfigurierbarkeit der Maschine und die Schnittstellen zwischen Modulen und nicht die **Zusammenhänge** zwischen Maschinen- und NC-Konfiguration sind, gibt es keine Unterscheidung der Modulklassen in bewegungserzeugende und feststehende Module. Diese Unterscheidung ist aber für die Modellierung von funktionalen Zusammenhängen wichtig, da die bewegungserzeugenden mechanischen Einheiten von der NC-Steuerung mit Sollwerten versorgt werden müssen und dabei ihr kinematischer Zusammenhang berücksichtigt werden muss. Daher basieren die Maschinenobjekte des Teilmodells Mechanik auf dem in Kapi-

tel 7.1.3 eingeführten Modulbaukasten (Bild 7.12).

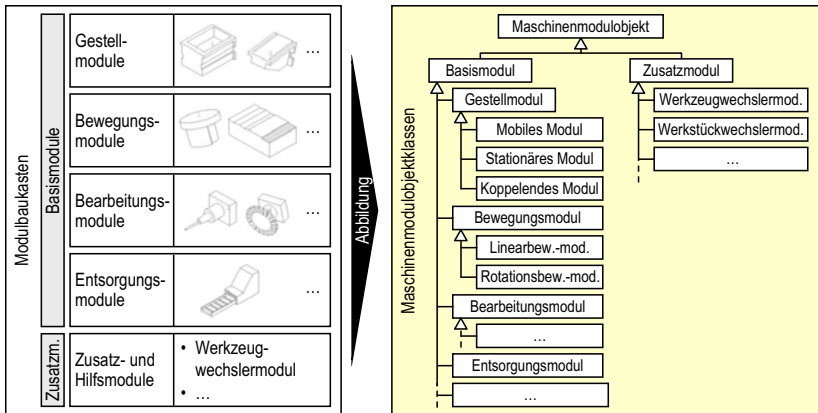


Bild 7.12: Abbildung des Modulbaukastens als Maschinenmodulobjektclassen

7.2.4 Definition von Kinematikkonfigurationen

Maschinen sind nach bestimmten und bewährten **Schemata** aufgebaut. Sie lassen sich nach der **Bauform** (z.B. Bett-, Portal-, Konsol- oder Ständerbauweise), der **Anzahl** ihrer Achsen (Freiheitsgrade der Bewegung) und der **Anordnung** ihrer Achsen (werkstück- oder werkzeugseitig) typisieren und einordnen (Bild 7.13).

Bauform	Anzahl der Achsen	Anordnung der Achsen
<ul style="list-style-type: none"> - Konsolbauweise - Bettbauweise <ul style="list-style-type: none"> - Kreuzbett - Kreuztisch - Fahrständer - Portal <ul style="list-style-type: none"> - Gantry - Tisch - ... 	<ul style="list-style-type: none"> - 3 Bearbeitungsachsen - 4 Achsen <ul style="list-style-type: none"> - 3 Bearbeitungsachsen und 1 Hilfsachse - 4 Bearbeitungsachsen - 5 Achsen <ul style="list-style-type: none"> - 5 Bearbeitungsachsen - 4 Bearbeitungsachsen und 1 Hilfsachse - ... 	<ul style="list-style-type: none"> - Werkstückseitig - Werkzeugseitig

Bild 7.13: Aspekte der Typologiebestimmung

Die Bauform legt durch ein **Muster** für die Kombination der Gestell- und Bewegungsmodul die Kinematik der Maschine fest. Je nach Anzahl der Achsen können unterschiedliche Freiheitsgrade der Bewegung erreicht werden. Der benötigte Freiheitsgrad lässt sich aus dem Bearbeitungsprozess ableiten /158/. Ein weiteres Unterscheidungskriterium für Maschinentypen ist die Aufteilung der Bearbeitungsachsen auf der Werkstück- und Werkzeugträgerseite. Bild

7.14 zeigt einen Überblick über die wichtigsten **Bauformen** von Fräsmaschinen und Bearbeitungszentren mit Werkzeugspindel.

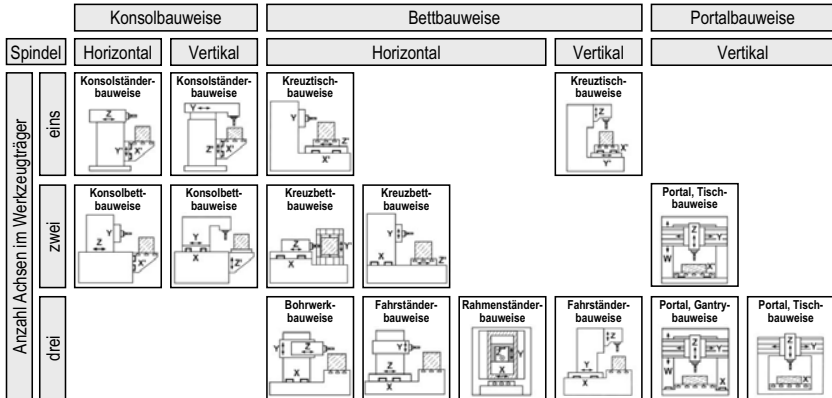


Bild 7.14: Bauformen vertikaler und horizontaler Fräsmaschinen (nach /159/)

In der schematischen Darstellung wird unterschieden zwischen der Anordnung der Werkzeugspindel (horizontal und vertikal) und der Anzahl der Achsen im Werkzeugträger. Die Vorschubbewegungen werden vom Werkzeugträger (Ständer, Spindelkasten) und/oder Werkstückträger (Tisch) ausgeführt. Je nach Aufteilung der Bewegungsachsen im Werkzeugträger und nach Anordnung der Bewegungsachsen lassen sich die Konstruktionsprinzipien in Konsol-, Bett- und Portalbauweise einteilen.

Für die Beschreibung der Bauformen als Mustervorlage im Informationsmodell als Modellierungsobjekt **Kinematikkonfiguration** müssen die Kombinationsmöglichkeiten ermittelt werden. Zur Systematisierung der Bauformen wird in /88, 90/ ein Verfahren vorgestellt bei dem anhand der werkzeug- und werkstückseitigen Anordnung der Achsen im Kraftfluss die **Kombinationsmöglichkeiten** für die Anordnung der Achsen bestimmt wird (Bild 7.15). Ausgehend vom Werkzeug, über die möglichen Anordnungen der Rotationsachsen und der möglichen Anordnungen der Linearachsen auf dem Maschinenbett kann eine Maschinenkonfiguration bestimmt werden. Die Rotationsachsen können entweder werkzeug- oder werkstückseitig angeordnet sein. Daraus ergeben sich die drei Kombinationsmöglichkeiten für die Anordnung der Rotationsachsen. Dasselbe gilt für die Anordnung der Linearachsen. Je nachdem auf welcher Seite sie angeordnet sind, taucht im Kraftfluss das Maschinenbett auf. Daraus ergeben sich vier Kombinationsmöglichkeiten. Jeder Kombination der Rotationsachsen kann mit jeder Kombination der Linearachsen kombiniert werden.

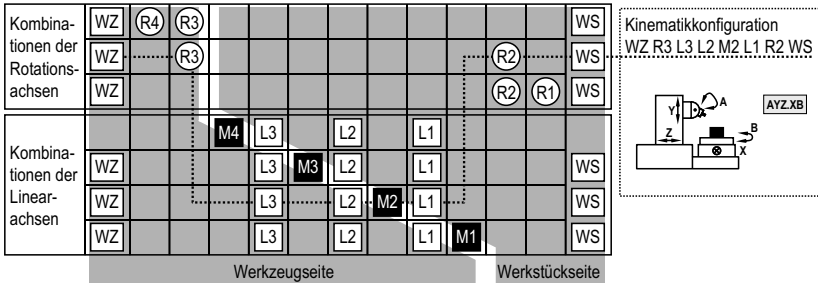


Bild 7.15: Systematik zur Bestimmung der Kombinationsmöglichkeiten der Achsanordnung (in Anlehnung an /90/)

Mit dieser Systematik können alle Kombinationsmöglichkeiten zur Anordnung von Achsen ermittelt werden. In der Praxis werden jedoch nicht alle Kombinationsmöglichkeiten eingesetzt. Dies reduziert die Anzahl auf die sinnvollen und praxistauglichen Kombinationen. Das **Bild 7.16** zeigt ausschnittsweise die Achsanordnungen am Beispiel von drei Linearachsen für die Bauformen Konsol-, Kreuztisch-/Kreuzbett-, Fahrständer- und Portalbauweise und listet die typischerweise in der Praxis verwendeten Anordnungen auf.

Bauformen				i. a. nicht verwendete Achsanordnungen
Konsolbauweise	Kreuztisch-/ Kreuzbettbauweise	Fahrständerbauweise	Portalbauweise	
				--- ZYX --- YZX --- XZY --- ZXY --- YXY --- XZZ --- X.ZY --- X.YZ --- XY.Z --- YX.Z --- ZX.Y --- XZ.Y YXZ XZY XYZ . = Maschinenbett
Achsenanordnungen				
-- Y . XZ _ -- Y . ZX _ -- YZ . X _ -- ZY . X _	-- Z . XY _ -- Z . YX _ -- ZY . X _	ZYX . _ _ ZXY . _ _ YZX . _ _	ZYX . _ _ _ZY . X _ = Maschinenbett

Bild 7.16: Kombinationsmöglichkeiten dreier Linearachsen (in Anlehnung an /88, 159/)

In der Schreibweise bezeichnet der Punkt die Trennstelle zwischen Werkstück- und Werkzeugseite durch das Maschinenbett. Diese reduzierte Anzahl der Anordnungen lässt sich für eine 5-achsige Maschine nach der Systematik in **Bild 7.15** mit den Anordnungsmöglichkeiten der Rotationsachsen kombinieren.

Für die Abbildung der Kombinationsmöglichkeiten ist eine Unterscheidung von Maschinen nach der Anzahl der Achsen nicht sinnvoll. Für die Steuerung ist in erster Linie interessant, wie viel synchronisierte Achsen (Bearbeitungsachsen) angesteuert und zu einer Achsgruppe zusammengefasst werden müssen. Zusätzlich kann in der Steuerung dann eine Hilfsachse angelegt werden. Daher wird in der **Klassifikation der Kinematikkonfigurationen** nur nach Bearbeitungsachsen unterschieden (Bild 7.17). Außerdem ist für die Steuerung die Anordnung der Achsen wichtig, da sie bei der Erzeugung der Sollwerte durch Transformation berücksichtigt werden muss. Daher wird im Informationsmodell weiter unterschieden nach den Achsanordnungen, die sich aus der Systematik aus Bild 7.15 ergeben.

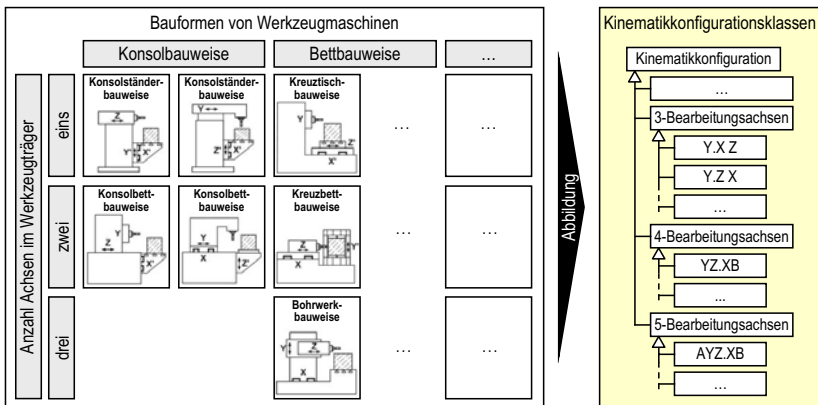


Bild 7.17: Abbildung der Bauformen als Kinematikkonfigurationsklassen

Zur Abbildung der Maschinenmodulobjektkombinationen zu Kinematikkonfigurationen wird die auf der Meta-Ebene des Informationsmodells definierte „besteht-aus“-Beziehung auf der Klassen-Ebene detailliert. Bild 7.18 zeigt exemplarisch die Detaillierung der 3-Bearbeitungsachsen-Kinematikkonfiguration Y.X.Z auf der Klassen-Ebene des Informationsmodells. Die Kinematikkonfiguration Y.X.Z besteht aus einem stationären Modul, drei mobilen Modulen, einem Bearbeitungs- und einem Entsorgungsmodul. Das stationäre und die mobilen Module sind Gestellmodule. Damit sich die mobilen Module bewegen können benötigen sie Bewegungsmodule. In diesem Fall bestehen die drei mobilen Module aus drei Linearbewegungsmodulen.

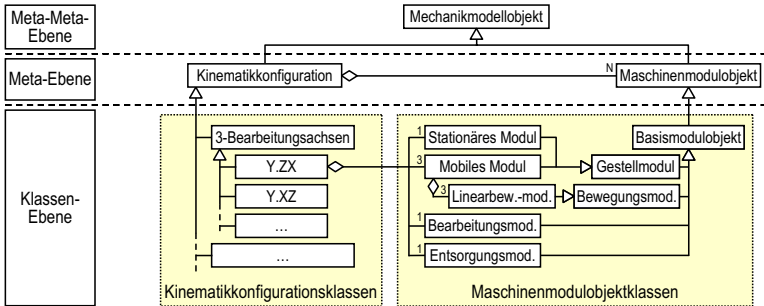


Bild 7.18: Detaillierung der Kinematikkonfiguration auf der Klassen-Ebene

7.2.5 Abbildung der Zusammenhänge zwischen den Teilmodellen

Aus Sicht der Steuerungstechnik sind nur die **Bewegungsachsen** und die **Kinematik** von Interesse. Die **Bewegungsmodule** sind bewegungserzeugende Module und müssen von der NC-Steuerung mit Sollwerten versorgt werden. Die Kinematik ist relevant, weil sie durch die Transformation mathematisch abgebildet wird und bei der Berechnung der Achssollwerte berücksichtigt werden muss. Die Kinematik wird aus den **Gestellmodulen** gebildet.

Bearbeitungs-, Zusatz- und Entsorgungsmodule haben keinen Anteil an der Erzeugung der Relativbewegung von Werkzeug und Werkstück und haben entweder keine anzusteuernenden Aktoren (z.B. stationäre und koppelnde Einheiten) oder sind nach dem Prinzip der mechatronischen, autarken Module funktional abgeschlossene Einheiten (z.B. Bearbeitungsmodule mit autarken Spann- und Hilfseinheiten, Entsorgungsmodule).

Für die **Kopplung der Teilmodelle** Steuerung und Mechanik müssen daher das NC-KO *Achse* mit dem Maschinenmodulobjekt *Bewegungsmodul* und das NC-KO *Achsguppe* mit dem Kinematikkonfigurationsobjekt *3-Bearbeitungsachsen* verknüpft werden. Die Verknüpfung zwischen NC-KO *Achse* mit dem Maschinenmodulobjekt *Bewegungsmodul* wird durch die Funktionsrelation *Steuerungsfunktion* im Informationsmodell abgebildet. Die Verknüpfung zwischen dem NC-KO *Achsguppe* mit dem Kinematikkonfigurationsobjekt *3-Bearbeitungsachsen* wird durch die Kinematikrelation *Kinematikzusammenhang* modelliert (Bild 7.19).

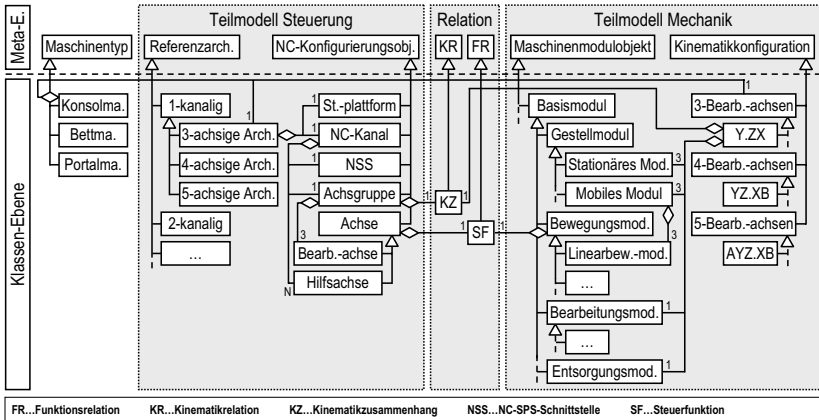


Bild 7.19: Zuordnung von NC-Konfigurierungsobjekten zu Maschinenmodulobjekten

7.2.6 Überblick über das Informationsmodell

Bild 7.20 gibt einen Überblick über das detaillierte **Informationsmodell** mit der Meta-Meta-, Meta- und Klassen-Ebene. Wegen der Übersichtlichkeit sind auf der Klassen-Ebene die „besteht-aus“-Beziehungen und die Klassen der Funktionsrelation und der Kinematikrelation nicht gezeichnet.

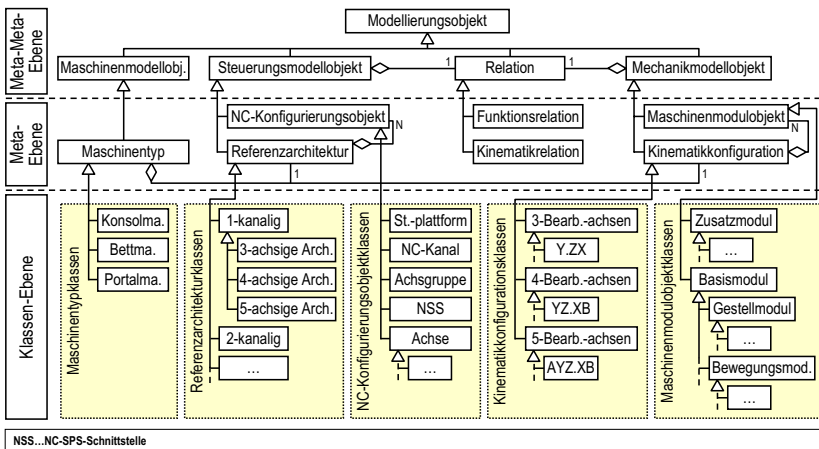


Bild 7.20: Detailliertes Informationsmodell mit der Meta-Meta-, Meta- und Klassen-Ebene

Auf der **Meta-Meta-Ebene** werden die Modellierungsobjekte des Informationsmodells definiert. Dazu gehören die Modellierungsobjekte *Maschinenmodellobjekt*, *Steuerungsmodellobjekt*, *Mechanikmodellobjekt* und *Relation*. Diese Modellierungsobjekte werden auf der **Meta-Ebene** weiter unterschieden. Steuerungsmodellobjekte werden unterteilt in *NC-Konfigurierungsobjekt* und *Referenzarchitektur*. Mechanikmodellobjekte werden unterteilt in *Maschinenmodulobjekt* und *Kinematikkonfiguration*. Relationen werden unterschieden in *Funktions- und Kinematikrelation*. Zur Unterscheidung von unterschiedlichen Maschinenbauformen wird das Maschinenmodellobjekt *Maschinentyp* auf der Meta-Ebene definiert.

Außerdem werden auf der Meta-Ebene **Beziehungen** definiert. Maschinentypen haben eine *Referenzarchitektur* und eine *Kinematikkonfiguration* und bilden so logisch den Zusammenhang der Realität ab. Zum Aufbau von Referenzarchitekturen werden NC-Konfigurierungsobjekte benutzt. Kinematikkonfigurationen bestehen aus Maschinenmodulen.

Auf der **Klassen-Ebene** werden die Modellierungselemente der Meta-Ebene verfeinert. Als Maschinentypen werden die Bauformen Konsol-, Bett- und Portalbauweise unterschieden. Des Weiteren werden Klassen für Referenzarchitekturen, NC-Konfigurierungsobjekte, Kinematikkonfigurationen und Maschinenmodulobjekte definiert.

7.3 Methode zur modellbasierten Konfigurierung der NC-Steuerung

Zur Konfigurierung der NC-Steuerung sollen **modellbasierte Methoden** eingesetzt werden, die aus dem baukastenbasierten, mechatronischen Engineering bekannt sind. Dort werden Informationsmodelle zur Abbildung von realen Objekten in Baukastensystemen und zur automatischen Generierung von Entwurfsunterlagen eingesetzt /132, 152, 153/.

Beim **baukastenbasierten Engineering** werden im Baukasten die Beziehungen zwischen den fachdisziplinspezifischen Entwurfsobjekten formal abgebildet, so dass mechatronische Entwurfsobjekte entstehen. Das mechatronische System wird aus Bausteinen des Baukastens konfiguriert (modellbasiertes Konfigurieren) und die fachdisziplinspezifischen Entwurfsunterlagen werden dann von Dokumentengeneratoren automatisch aus dem Informationsmodell erzeugt /116, 130/ (baukastenbasierte Generierung).

Die **Generatoren** arbeiten mit **Mustervorlagen**, so genannten Templates. Die Generierungsmethode auf der Basis von Templates ist im Bereich der automatisierten Generierung von SPS-Programmcode /79/, komponentenbasierter Simulationsmodelle /160, 161/ und Schaltplanunterlagen /115/ verbreitet und bewährt. Das Template gibt einen Rahmen vor in dem an bestimmten, durch Platzhalter festgelegten Stellen, zielplattformspezifischer Code eingefügt wird. Die Platzhalter sind als Schlüsselwörter gekennzeichnet und werden vom Generator

erkannt und entsprechend mit dem zielplattformspezifischen Code ersetzt. Das Template legt die Struktur und der Generator die Syntax des zu generierenden Codes fest. Somit ist es möglich, mit dieser Methode durch vordefinierte Templates und zielplattformspezifischen Generatoren unterschiedliche NC-Steuerungen zu konfigurieren.

Diese **baukastenbasierte Konfigurierungsmethode** für Konstruktionsunterlagen soll auf die Konfigurierung von NC-Steuerungen übertragen werden, so dass auf Basis des in Kapitel 7.2 definierten Informationsmodells NC-Konfigurierungslisten generiert (Bild 7.21) werden können. Wesentliche Voraussetzung hierfür ist, dass in dem Informationsmodell typische **Entwurfsmuster** für NC-Steuerungskonfigurationen die Konfigurierungsregeln formal abbilden. Dazu werden im Informationsmodell die Maschinenmodule und die NC-Konfigurierungsobjekte, sowie deren Zusammenhänge (Relationen) abgebildet.

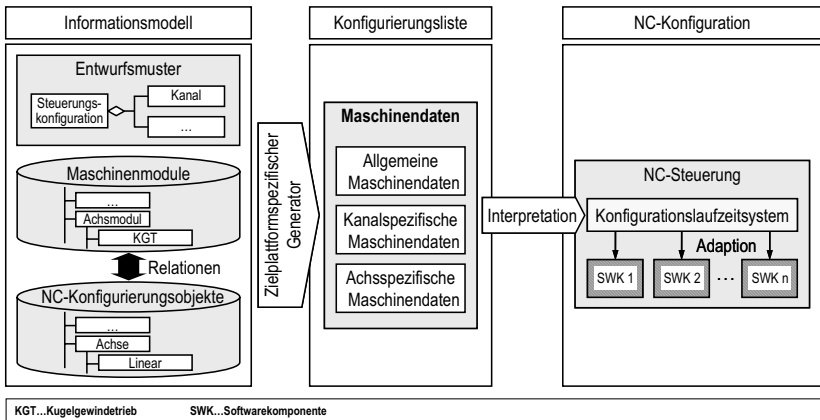


Bild 7.21: Modellbasierte NC-Konfigurierung nach dem Prinzip des baukastenbasierten Engineerings

Durch **zielplattformspezifische Generatoren** können aus dem Informationsmodell steuerungsspezifische Formate für Maschinendaten erzeugt werden. Die Anpassung der NC-Steuerung erfolgt durch die in Kapitel 5.3 untersuchten Adaptionsmechanismen der Steuerung. Das steuerungssinterne Konfigurationslaufzeitsystem liest dazu die Maschinendaten ein und interpretiert sie. Aufgrund der eingelesenen Maschinendaten werden die internen Steuerungssoftwarekomponenten durch das Konfigurationslaufzeitsystem adaptiert.

Das **modellbasierte Konfigurieren** erfolgt durch Instanziierung von Klassen aus der Klassen-Ebene des Informationsmodells /162/. Diese modellbasierte Konfigurierungsmethode

wird auch bei einer Rekonfiguration angewandt. Eine Rekonfiguration ist aus Sicht der Steuerung daher eine erneute Konfiguration.

Bei der modellbasierten Konfiguration werden die instanziierten Klassen zu Objekten mit konkreten Parameterwerten. Die Klassen geben im Sinne der Objektorientierung nur den Rahmen für die Eigenschaften (Parameter) der Objekte vor und können als „Platzhalter“ für konkrete Objekte gesehen werden. Die Konfiguration entspricht in dieser Sichtweise dann der Ausgestaltung der Platzhalter durch konkrete Objekte. Diese Objekte werden aus den NC-Konfigurationsobjektklassen und den Maschinenmodulobjektklassen der Klassen-Ebene des Informationsmodells abgeleitet.

Die modellbasierte Konfiguration läuft in zwei Schritten ab (Bild 7.22). Im ersten Schritt wird ein Maschinentyp ausgewählt und instanziiert. Durch seine „besteht-aus“-Beziehungen auf der Klassen-Ebene des Informationsmodells wird automatisch eine Steuerungskonfiguration als Instanz einer Referenzarchitekturklasse und eine Maschinenkonfiguration als Instanz einer Kinematikkonfigurationsklasse angelegt.

Die modellbasierte Methode kann nicht vollautomatisch ablaufen. Erst nachdem der Maschinentyp durch einen Benutzer festgelegt wurde, können die Beziehungen im Informationsmodell automatisch ausgewertet und Instanzen auf der Instanzen-Ebene angelegt werden. Ein Benutzer muss zusätzlich noch die Kinematikkonfiguration auswählen, da die geometrische Anordnung der Bearbeitungsachsen nicht automatisch erkannt werden kann.

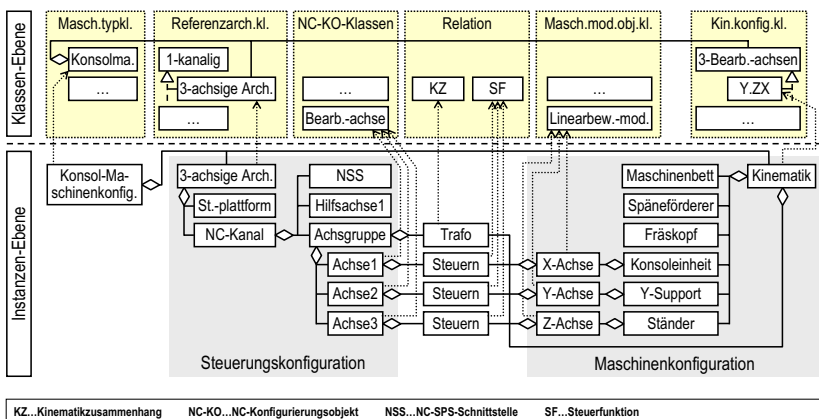


Bild 7.22: Modellbasiertes Konfigurieren am Beispiel einer 3-achsigen Maschine

Im zweiten Schritt werden die nach der Steuerungs- und Maschinenkonfiguration angelegten

Klassen durch konkrete NC-Konfigurierungs- und Maschinenmodulobjekte ersetzt. Diese Instanzen werden dann durch Parametrierung individuell ausgeprägt.

Das Bild 7.22 zeigt am Beispiel einer 3-achsigen Maschine in Konsolbauweise wie die Instanzebene bei der Konfigurierung im Informationsmodell angelegt wird. Aus der Klassenebene wird vom Maschinentyp *Konsolmaschine* eine Maschine abgeleitet. Durch die Beziehungen auf der Klassen-Ebene werden automatisch die Steuerungskonfiguration und die Maschinenkonfiguration abgeleitet.

Die Steuerungskonfiguration ist eine Instanz der Referenzarchitekturklasse *3-achsige Architektur* und hat als NC-KO eine *Steuerungsplattform* und einen *NC-Kanal*. Das NC-KO *NC-Kanal* besteht aus den NC-KO *NC-SPS-Schnittstelle*, *Achsgruppe* und *Hilfsachse*. Das NC-KO *Achsgruppe* hat drei NC-KO von der Klasse *Bearbeitungsachse*.

Die Maschinenkonfiguration *Kinematik* ist eine Instanz der Kinematikkonfigurationsklasse *Y.XZ* und hat als Maschinenmodulobjekte *Maschinenbett*, *Späneförderer*, *Fräskopf*, *Konsolereinheit*, *Y-Support* und *Ständer*. Die Maschinenmodulobjekte *Konsolereinheit*, *Y-Support* und *Ständer* sind mobile Module. Ihnen sind daher über eine „besteht-aus“-Beziehung die Linearbewegungsmodule *X*-, *Y*- und *Z-Achse* zugeordnet. Damit die Linearbewegungsmodule von der NC-Steuerung angesteuert werden können, sind sie über die Funktionsrelation *Steuern* mit den NC-KO *Achse1*, *Achse2* und *Achse3* verknüpft. Der kinematische Zusammenhang wird über den Kinematikzusammenhang *Trafo* zwischen dem NC-KO *Achsgruppe* und dem Kinematikkonfigurationsobjekt *Kinematik* abgebildet.

7.4 Mechanismus zur Koordination der Modul- und Systemkonfigurierung

In einer modularen Maschine sind die Module über ein Kommunikationssystem miteinander verbunden. Beim Konfigurieren werden die Module verbunden und müssen anschließend konfiguriert und initialisiert werden. Dazu muss ein **Koordinator** den Ablauf der **Konfigurierung und Initialisierung** steuern. Dies setzt voraus, dass jedes Modul ein **definiertes Verhalten** beim Einschalten und während der Konfigurierung hat, das in einem Zustandsautomat formalisiert ist. Da in der ersten Phase der Selbstkonfigurierung die Maschinenmodule identifiziert werden, müssen in einem ersten Schritt zum Auslesen der Modulkennungen aus dem Konfigurierungsspeicher der mechatronischen Module die Kommunikationsverbindungen hergestellt werden. Dafür sollen Grundmechanismen von ethernetbasierten Echtzeitkommunikationssystemen genutzt werden.

In diesem Kapitel werden daher am Beispiel von SERCOS III die **Initialisierungsmechanismen** eines ethernetbasierten Echtzeitkommunikationssystem dahingehend analysiert, ob

sie in der ersten Konfigurierungsphase genutzt werden können. Außerdem werden **Zustandsautomaten für das PnP-Verhalten** von mechatronischen Modulen und für die Steuerung des Konfigurierungsablaufs durch den **Koordinator** entworfen.

7.4.1 Selbstkonfigurierendes Kommunikationssystem

Wenn eine Maschine durch das Zusammenstecken von Modulen konfiguriert wurde, müssen als erstes die Kommunikationsverbindungen hergestellt werden, damit die Selbstadaption der NC-Steuerung beginnen kann. Dazu muss das Kommunikationssystem gemäß des Ablaufs einer Selbstadaption (Kapitel 6.5) eine **Initialisierungsphase** beim Starten haben, die es ermöglicht eine Basiskommunikation zwischen den Modulen aufzubauen und das Kommunikationsverhalten zu konfigurieren. Bild 6.9 zeigt in welcher Phase der Selbstadaption die Initialisierung des Kommunikationssystems stattfindet. In der ersten Phase wird das Kommunikationssystem initialisiert, so dass es eine Basiskommunikation bereitstellt mit der in den weiteren Phasen die Mechanismen zur NC-Konfigurierung mit den Modulen kommunizieren können bis die Maschine konfiguriert und der Betriebszustand erreicht ist.

Es gibt in der Automatisierungstechnik Bussysteme, die sich beim Einschalten **selbst initialisieren** können /34, 35/. Während der Initialisierungsphase können sie mit Hilfe von Erkennungsmechanismen die am Bus angeschlossenen Kommunikationsteilnehmer erkennen. Die Kommunikationsteilnehmer haben einen **Initialisierungszustand**, den sie beim Einschalten des Systems einnehmen. Für die selbstständige Konfigurierung der Kommunikation besitzen sie grundlegende Kommunikationsmechanismen zur Vergabe von Adressen und zur Zuteilung von Ressourcen. Ist die Konfigurierung der Kommunikation abgeschlossen, kann die Echtzeitkommunikation gestartet und der Normalbetrieb aufgenommen werden. Damit ist eine Konfigurierung von Kommunikationsteilnehmern auch ohne initialisiertes Echtzeit-Netzwerk möglich.

Am Beispiel von SERCOS III /34/ soll gezeigt werden, wie diese Grundfunktion zur Initialisierung des Kommunikationssystems in der ersten Phase der Selbstadaption genutzt werden können. Der SERCOS-Bus basiert auf einer Ethernet-Hardware und verwendet ein speziell auf die Anforderung der Automatisierungstechnik zugeschnittenes Echtzeitprotokoll, das in einen Standard Ethernet-Frame eingelagert ist (Bild 7.23).

Zur Koordination der Kommunikation arbeitet das Bussystem nach dem Master-/Slave-Prinzip /163/. Neben der Echtzeitkommunikation für zyklische Daten bietet das SERCOS-Bussystem einen nichtechtzeitfähigen Kommunikationskanal (NRT-Kanal) und einen Service-Kanal an. Der Service-Kanal wird für die Konfigurierung der Slave-Geräte verwendet. Über den NRT-Kanal (Non-Real-Time-Kanal) kann mit dem aus dem Internet bekannten

Standardkommunikationsprotokoll TCP/IP-Protokoll kommuniziert werden. Der NRT-Kanal setzt direkt auf der Ethernet-Schicht auf und nutzt den Zykluszeit-Anteil, der nicht von Echtzeit-Telegrammen verwendet wird. Im Echtzeitkanal werden jedem Slave-Gerät Gerätekanäle mit einem MDT (Master-Daten-Telegramm) und einem AT (Antwort-Telegramm) zugewiesen, aus dem es Daten ausliest bzw. in die es sie hineinschreibt. Das MDT nutzt der Master für die Vorgabe von Daten an die Slave-Geräte. Das AT ist das Antworttelegramm der Slave-Geräte.

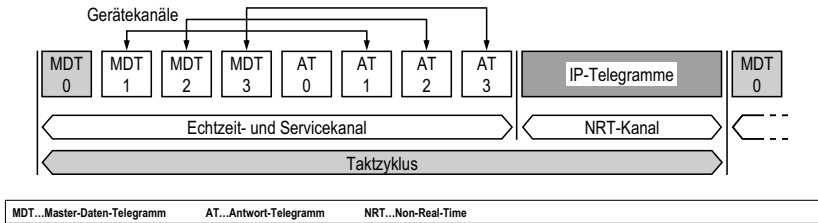


Bild 7.23: SERCOS III-Kommunikationszyklus (nach /34/)

In einer **Initialisierungsphase** werden die am Bus angeschlossenen Geräte erkannt, adressiert und für die Anwendung konfiguriert. Dazu wird eine automatische Geräteerkennung und Adresszuweisung vom Master durchgeführt. Während der Initialisierung synchronisiert sich das Netzwerk selbst. Während des laufenden Betriebs neu eingekoppelte Geräte werden automatisch in die Kommunikation und den Echtzeitdatenaustausch eingebunden (Hot-Plugging).

Aufgrund ihrer Eigenschaften, sich selbst zu initialisieren und alle Kommunikationsteilnehmer zu erkennen, eignen sich am Markt verfügbare ethernetbasierte Bussysteme, wie SERCOS III, als Kommunikationssystem für rekonfigurierbare Werkzeugmaschinen. Damit können für die Selbstadaption ethernetbasierte Standardbussysteme verwendet werden. Aufbauend auf deren Konfigurations- und Kommunikationsmechanismen können so die PnP-fähigen Maschinenmodule während der Selbstadaption miteinander kommunizieren und Konfigurationsdaten austauschen.

7.4.2 Zustandsautomat eines PnP-fähigen Moduls

Der **Zustandsgraph** in [Bild 7.24](#) beschreibt das Verhalten, welches alle mechatronischen Module implementieren müssen. Die Zustandsübergänge sind von außen durch Aufträge des Konfigurationskoordinators steuerbar, um Module kontrolliert und koordiniert in ihren Betriebszustand (Zustand „Betrieb“) zu bringen. Diese Koordination der Module von außen ist notwendig, damit die Maschine vor einer Konfiguration stets in einem sicheren Zustand ist

und die Modulzustände konsistent sind.

Vor einer Konfiguration müssen alle mechatronischen Module gestoppt werden. Sie befinden sich dann im Zustand „uninitialisiert“. Werden Module neu hinzugefügt („Einschalten“), sind sie nicht im Betriebszustand, sondern warten im Zustand „uninitialisiert“ bis der Koordinator alle Module auffordert ihre Kommunikationsverbindungen zu initialisieren (Zustand „Kommunikation initialisieren“).

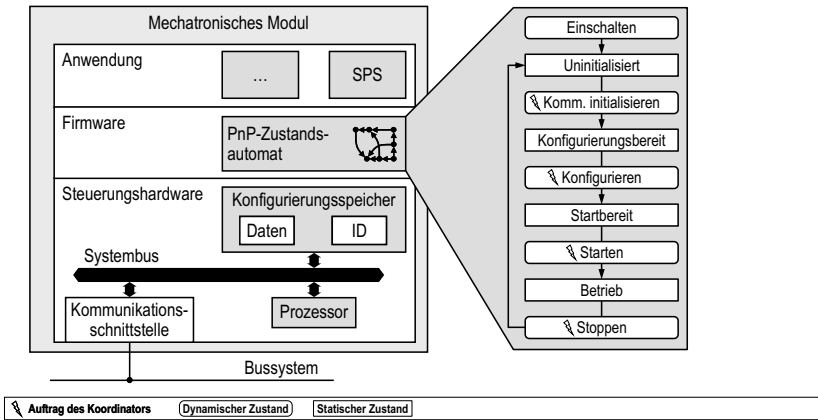


Bild 7.24: Steuerungstechnischer Aufbau und Zustandsautomat eines PnP-fähigen Moduls

Die Initialisierung der Kommunikation wird vom Kommunikationssystem selbst durchgeführt. Danach ist eine Basiskommunikation zur Konfiguration der Module möglich. Erst dann wechseln die Module in den Zustand „Konfigurierungsbereit“ bevor sie anschließend im Zustand „Konfigurieren“ konfiguriert werden. In diesem Zustand wird der Konfigurationsspeicher ausgelesen und gegebenenfalls mit Zustandsdaten beschrieben. Danach wechselt das Modul in den Zustand „startbereit“. Wenn alle Module im Zustand „startbereit“ sind, kann der Koordinator jedes einzelne Modul starten. Dabei interpretiert jedes Modul für sich die Daten aus dem Konfigurationsspeicher und die zyklische Echtzeitkommunikation wird hergestellt. Das Modul befindet sich danach im Zustand „Betrieb“.

7.4.3 Zustandsautomat des Konfigurationskoordinators

Bevor eine Maschine rekonfiguriert werden kann, müssen alle mechatronischen Module der Maschinenkonfiguration in einen Konfigurationzustand überführt werden. Dazu muss ein **Konfigurationskoordinator** die Modulzustände koordinieren. Die Funktion des Konfigurationskoordinators kann jedes Modul übernehmen, wenn das Kommunikationssystem ini-

tialisiert ist. Im einfachsten Fall übernimmt der Kommunikationsmaster, der auch das Kommunikationssystem initialisiert, diese Aufgabe und sorgt so dafür, dass sich alle Module im selben Zustand befinden. Bild 7.25 zeigt die Zustände des Konfigurierungskordinators zur Koordination der Modulzustände mehrerer Module.

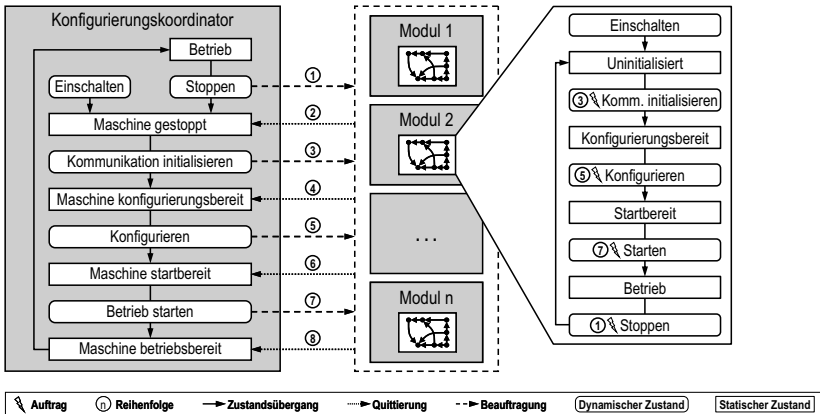


Bild 7.25: Zustände des Konfigurierungskordinators zur Koordination mehrerer Module

Wenn die Maschinenkonfiguration in Betrieb ist und rekonfiguriert werden soll, dann geht der Master in den Zustand „Betrieb stoppen“ und beauftragt alle angeschlossenen Module ihren Betriebszustand zu stoppen (Schritt 1). Jedes Modul geht daraufhin in den Zustand „Stoppen“ und beendet individuell seinen Betriebszustand. Die Module führen autark die noch notwendigen Funktionen sicher zu Ende, bevor sie in den Zustand „uninitialisiert“ wechseln und dies an den Koordinator quittieren (Schritt 2). Daraufhin nimmt der Koordinator den Zustand „Maschine gestoppt“ ein. In diesem Zustand befindet sich die Maschine auch, wenn sie nicht aus dem Betriebszustand heraus konfiguriert wird, sondern neukonfiguriert und eingeschaltet wurde. Im Zustand „Maschine gestoppt“ ist es möglich, Module auszutauschen, neue Module hinzuzufügen oder alte zu entfernen (Bild 7.26 zeigt den detaillierten Ablauf der Koordination der Modulzustände beim Hinzufügen).

Nachdem die Maschine gestoppt wurde und Module hinzugefügt, ausgetauscht oder entfernt wurden, muss die Kommunikation neu initialisiert werden (Schritt 3). Dazu werden die Standard-Initialisierungsmechanismen des Kommunikationssystems genutzt (Kapitel 7.3). Der Kommunikationsmaster des Bussystems erkennt die Kommunikationsteilnehmer und vergibt automatisch die Adressen. Alle Module werden anschließend durch den Kommunikationsmaster initialisiert, so dass eine Basiskommunikation möglich ist. D.h. die Parameter der Mo-

odule können gelesen und geschrieben werden. In dieser Phase führt der Koordinator eine Suche nach mechatronischen Modulen durch und erstellt eine Liste aller Module, die an der Kommunikation teilnehmen. Dann wechseln die Module in den Zustand „konfigurierungsbereit“ und quittieren dies (Schritt 4).

Erst wenn alle Module bereit für die Konfigurierung sind, geht der Koordinator in den Zustand „Maschine konfigurierungsbereit“ und anschließend in den Zustand „Konfigurieren“ über. In diesem Zustand beauftragt er alle Module, ebenfalls in den Konfigurierungszustand zu wechseln und die Konfigurierung durchzuführen (Schritt 5). Jedes Modul bereitet daraufhin die Konfigurierung vor, indem es das Beschreiben und Lesen des Konfigurierungsspeichers zulässt. Die Informationen für die NC-Konfigurierung werden aus dem Konfigurierungsspeicher des Moduls gelesen und als Parameter in den NC-KO des Informationsmodells abgelegt. Dazu wird die Basiskommunikation des Kommunikationssystems genutzt. In diesem Zustand wird die NC-Konfigurierungsliste durch die modellbasierte Konfigurierungsmethode erzeugt (Kapitel 7.3). Erst wenn alle Module ihre notwendigen Konfigurierungsinformationen bereitgestellt haben und die NC-Konfigurierungsliste erzeugt ist, wechselt der Master in den Zustand „Maschine startbereit“ (Schritt 6).

Danach kann der Koordinator in den Zustand „Betrieb starten“ gehen und die Maschine gestartet werden (Schritt 7). Dies führt dazu, dass alle Module in den Zustand „Starten“ übergehen. In diesem Zustand werden die Konfigurierungsinformationen übernommen und die Module initialisieren sich von selbst. Gleichzeitig liest die NC-Steuerung die Konfigurierungsliste ein und erzeugt eine NC-Konfiguration (Kapitel 4.3.1). Wenn die Modulinitialisierung abgeschlossen ist geht jedes Modul automatisch in den Zustand „Betrieb“ und quittiert dies dem Koordinator (Schritt 8). Der Koordinator wartet bis alle Module in Betrieb sind und wechselt dann in den Zustand „Maschine betriebsbereit“. Dabei geht das Kommunikationssystem in den zyklischen Echtzeitbetrieb über. Der Koordinator kann dann den Betrieb für die gesamte Maschinenkonfiguration wieder freigeben, indem er in den Zustand „Betrieb“ übergeht.

Bild 7.26 zeigt den **Ablauf der Koordination** der Modulzustände, wenn in diesem Beispiel das Modul 1 während des laufenden Betriebs hinzugefügt wird. Beim Einschalten geht ein Modul automatisch als erstes in den Zustand „uninitialisiert“ (Schritt 1). Dadurch ist es im gleichen Zustand wie alle Module der Maschine nachdem der Koordinator den Betrieb gestoppt hat. Über die Hot-Plugging-Funktion des Bussystems teilt es dem Koordinator mit, dass ein neues Modul da ist (Schritt 2). Der Betriebszustand der Maschine wird dadurch unterbrochen, indem der Koordinator alle anderen Module beauftragt, in den Zustand „stoppen“ zu gehen (Schritt 3).

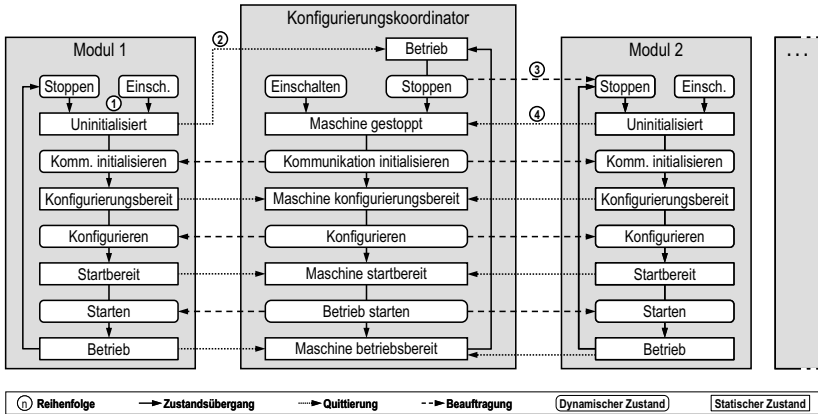


Bild 7.26: Koordination der Modulzustände durch den Master

Die Module verlassen daraufhin ihren Betriebszustand und wechseln in den Zustand „uninitialisiert“. Sobald sie diesen Zustand eingenommen haben, quittieren die Module dies und der Koordinator kann in den Zustand „Maschine gestoppt“ wechseln (Schritt 4). Jetzt befinden sich alle Module im Zustand „uninitialisiert“. Der weitere Ablauf entspricht dem in Bild 7.25 dargestellten Standardablauf. Der Koordinator initialisiert die Kommunikation, koordiniert die Konfiguration und startet dann die Module.

7.5 Konzept der selbstadaptierenden NC-Steuerung

Damit sich eine NC-Steuerung selbst adaptieren kann, müssen die in Kapitel 7 erarbeiteten Methoden zu einem Gesamtkonzept zusammengefasst werden. Daher soll in diesem Kapitel ein Konzept einer **selbstadaptierenden NC-Steuerung** erarbeitet werden, das die Selbstadaptionsmethoden integriert. Voraussetzung hierfür ist (Kapitel 6.4), dass

- sich die **Maschinenkonfiguration aus mechatronischen Modulen** mit PnP-Eigenschaften zusammensetzt,
- die Module und die selbstadaptierende Steuerung über ein **Bussystem** miteinander verbunden sind,
- die Module einen **Konfigurationsspeicher** besitzen, in dem die Parameter für die Konfiguration und ein Kennzeichner für die Identifizierung gespeichert sind,
- das PnP-Verhalten der Module in einem **Zustandsautomat** implementiert ist.

Eine selbstadaptierende NC-Steuerung besteht aus einem **Konfigurationskoordinator** und

einer **adaptierbaren NC-Steuerung** (Bild 7.27). Der Konfigurierungskordinator beinhaltet eine Ablaufsteuerung, einen Identifizierer, ein Informationsmodell und einen Generator.

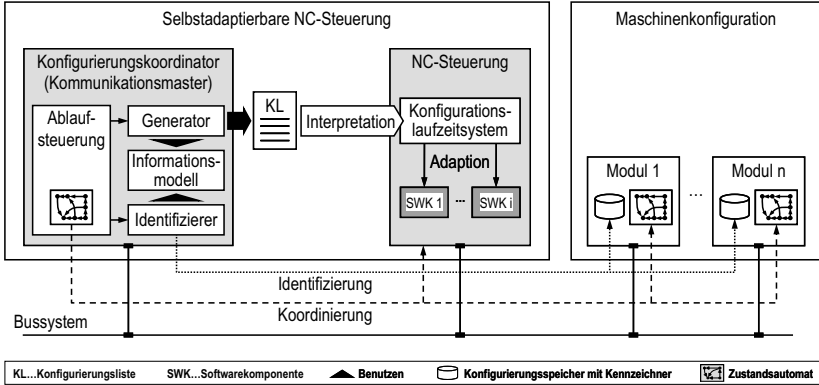


Bild 7.27: Elemente einer selbstadaptierenden NC-Steuerung

Der **Konfigurierungsvorgang** wird von der **Ablaufsteuerung** des Konfigurierungskordinators gesteuert (Kapitel 7.4.3). Der **Identifizierer** ist für die Erkennung und Klassifizierung der Module zuständig. Im Informationsmodell sind die Abhängigkeiten zwischen Maschinen- und Steuerungskonfiguration abgebildet. Der **Generator** nutzt das Informationsmodell, um modellbasiert NC-Konfigurierungen durchzuführen und NC-Konfigurationslisten zu erzeugen.

Nachdem eine Maschine aus PnP-fähigen Modulen konfiguriert wurde, startet die Ablaufsteuerung das Kommunikationssystem. Es beginnt mit der **Initialisierung** der Kommunikation und stellt eine Basiskommunikation zur Verfügung. Dazu vergibt das Kommunikationssystem automatisch die Adressen und baut einen Kommunikationskanal für Basisdienste auf (Kapitel 7.4.1). Nachdem das Kommunikationssystem initialisiert ist, befindet sich die Maschine im Zustand „Maschine konfigurierungsbereit“ (Bild 7.25) und geht dann in den Zustand „Konfigurieren“ über. In diesem Zustand findet die **Adaption der NC-Steuerung** statt, indem unter Benutzung des Informationsmodells modellbasiert Konfigurationslisten für die adaptierbare NC-Steuerung generiert werden. Den detaillierten Ablauf der Selbstadaption in diesem Maschinenzustand zeigt Bild 7.28.

Über die Basiskommunikation ermittelt und identifiziert der Identifizierer des Konfigurierungskordinators alle Kommunikationsteilnehmer. Die **Identifikation** erfolgt über die Kennung der PnP-fähigen Maschinenmodule, die im Konfigurationsspeicher gespeichert ist. Dabei wird durch den Identifizierer eine Liste aller Module (Kommunikationsteilnehmer) mit

ihren Kennungen erstellt (Bild 7.29, Schritt 1).

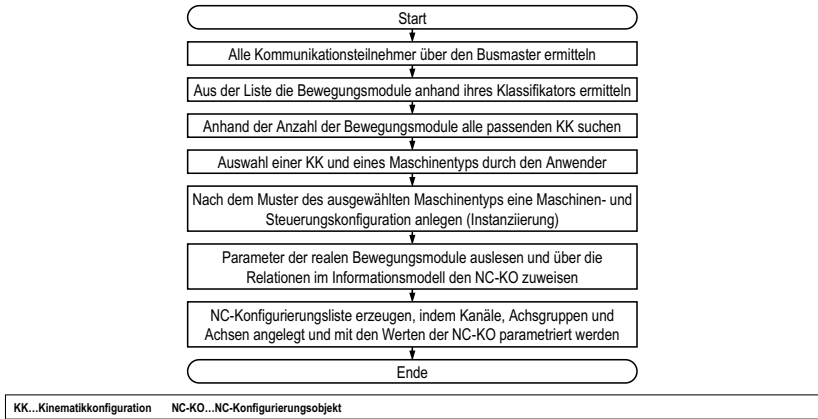


Bild 7.28: Algorithmus für den Ablauf der Selbstadaption

Aus der Liste der Module ermittelt der Identifizierer anhand der Klassifikatoren der Module die Bewegungsmodule (Bild 7.29, Schritt 2). Damit steht fest wie viele Freiheitsgrade die vorhandene Maschine hat. Der **kinematische Zusammenhang** dieser Achsen wird durch die Kinematikkonfiguration im Informationsmodell festgelegt (Kapitel 7.2.4). Da der Identifizierer nur ermitteln kann, wie viele Bewegungssachsen in der Maschine vorhanden sind und nicht erkennen kann wie ihre Anordnung ist, ist für diese Festlegung die Hilfe eines Inbetriebnehmers notwendig. Er muss eine Kinematikkonfiguration auswählen. Dabei wird er aber von dem Konfigurierungskordinator unterstützt, da er nur die passenden Kinematikkonfigurationen mit den richtigen Bewegungsfreiheitsgraden zur Auswahl anzeigt. Zusätzlich muss der Inbetriebnehmer den **Maschinentyp** festlegen, weil mit der Achsanordnung noch nicht eindeutig feststeht, um welche Bauform es sich handelt (Bild 7.29, Schritt 3).

Nachdem ein Maschinentyp feststeht, führt der Konfigurierungskordinator eine **modellbasierte Konfigurierung** (Kapitel 7.3) durch, indem eine Maschinen- und Steuerungskonfiguration nach dem Muster des gewählten Maschinentyps instanziiert wird (Bild 7.29, Schritt 4). Dabei werden Instanzen von Maschinenmodulobjekten und NC-Konfigurierungsobjekte aus der Klassenebene des Informationsmodells angelegt. Die Maschinenmodulobjekte haben eine reale Entsprechung und können einem echten Modul zugeordnet werden. Die NC-Konfigurierungsobjekte im Informationsmodell sind logische, nicht real existierende Einheiten. Sie dienen als Datencontainer für herstellerepezifische Konfigurierungsinformationen. Durch die Relationen im Informationsmodell haben sie über die Maschinenmodulobjekte des

Informationsmodells einen Bezug zu den realen Modulen. Über diese Beziehung werden den NC-Konfigurierungsobjekten die Modulparameter der mechatronischen Module zugewiesen (Bild 7.29, Schritt 5).

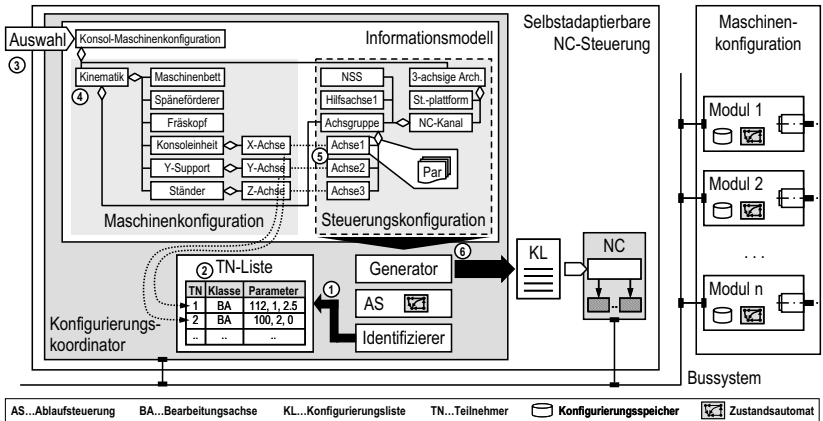


Bild 7.29: Ablauf der Selbstadaption

Auf Basis der Steuerungskonfiguration im Informationsmodell erzeugt der **Generator** nach dem Prinzip der modellbasierten Konfiguration (Kapitel 7.3) eine Konfigurationsliste für die NC-Steuerung (Bild 7.29, Schritt 6) und berücksichtigt dabei die steuerungssystemspezifische Darstellung der Parameter (Syntax). Dadurch werden zielplattformspezifisch NC-Konfigurationslisten erzeugt und mit den Parametern aus den NC-KO parametrierbar.

7.6 Zusammenfassung

In diesem Kapitel wurden die Voraussetzungen für die Übertragung der PnP-Prinzipien auf die Konfiguration von NC-Steuerungen geschaffen, indem ein Modulkonzept für Werkzeugmaschinen auf Basis mechatronischer Module entworfen und ein Informationsmodell für die Abbildung der funktionalen Zusammenhänge und der Konfigurationen erarbeitet wurde. Aufbauend auf dem Informationsmodell wurde eine modellbasierte Konfigurationemethode für NC-Steuerungen entwickelt, mit der NC-Konfigurationslisten generiert werden können. Abschließend wurden die erarbeiteten Methoden und Prinzipien in einem Konzept für eine selbstadaptierende NC-Steuerung zusammengefasst.

8 Ergebnisse der Arbeit

Bild 8.1 zeigt das Vorgehen und die Ergebnisse der Arbeit. Ausgehend von der **Analyse des NC-Konfigurierungsprozesses** und der Analyse der herstellerübergreifenden Anpassbarkeit von NC-Steuerungen /141, 142/ wurde in Kapitel 5 der NC-Konfigurierungsprozess systematisiert und ein herstellerübergreifendes **Strukturmodell einer adaptierbaren NC-Steuerung** entworfen /46, 136/.

In Kapitel 6 wurde eine Methode zur Selbstadaption von NC-Steuerungen /51, 157, 163/ nach dem PnP-Prinzip entwickelt. Dazu wurden die **Eigenschaften von Plug-and-Play-Systemen** aus der Computertechnik analysiert, um daraus **Voraussetzungen für die Übertragbarkeit** von PnP-Prinzipien auf die Konfigurierung von NC-Steuerungen abzuleiten /149, 150/. Durch Übertragung von PnP-Prinzipien wurden anschließend **Methoden zur Automatisierung des NC-Konfigurierungsprozesses** erarbeitet /157, 162/. Die Ergebnisse dieses Kapitel wurden abschließend in einer einheitlichen **Methode zur Selbstadaption von NC-Steuerungen** zusammengefasst /51, 157, 163/.

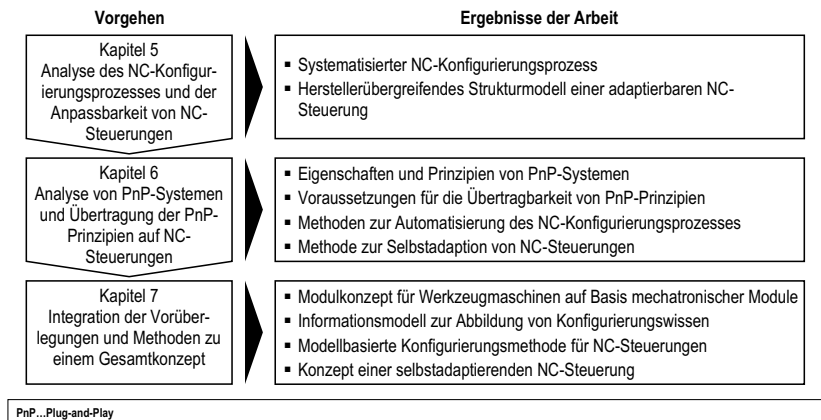


Bild 8.1: Vorgehen und Ergebnisse der Arbeit

Ausgehend von den Vorüberlegungen der Kapitel 5 und 6 wurden in Kapitel 7 die Voraussetzungen für die Übertragbarkeit der PnP-Prinzipien auf die NC-Konfigurierung geschaffen und ein **Konzept für eine selbstadaptierende NC-Steuerung** erarbeitet /157, 163/. Dazu wurde ein **Modulkonzept für Werkzeugmaschinen** auf Basis mechatronischer, PnP-fähiger Module entwickelt /24, 25, 53, 137, 154/. Diese Module wurden anschließend zusammen mit dem Strukturmodell einer adaptierbaren NC-Steuerung aus Kapitel 5 in einem **Informationsmo-**

dell abgebildet. Zur Formalisierung von Konfigurierungswissen wurden in dem Informationsmodell durch Abstraktion von Steuerungs- und Maschinenkonfigurationen **Konfigurierungsmuster** erarbeitet /156/. Sie legen die Kombinationssystematik für modulare Werkzeugmaschinen fest. Darauf aufbauend wurde eine **modellbasierte Konfigurierungsmethode** für NC-Steuerungen entwickelt /156, 163/. Sie nutzt die im Informationsmodell abgebildeten funktionalen Zusammenhänge zwischen Maschinenmodulen und NC-Konfigurierungsobjekten, um automatisiert NC-Konfigurierungslisten zu generieren. Abschließend wurden die in dieser Arbeit erarbeiteten Grundlagen und Methoden in einem **Konzept für eine selbstadaptierende NC-Steuerung** zusammengefasst /163, 165/.

Im Rahmen des SFB 467 und anschließend im Rahmen des TFB 59 wurden zu Demonstrationszwecken **Komponenten** einer **selbstadaptierbaren NC-Steuerungen** prototypenhaft umgesetzt /136, 137, 165/.

8.1 Komponentenweise Realisierung einer selbstadaptierenden NC-Steuerung

Ziel der Arbeiten im SFB 467 war es, die Rekonfiguration der Maschine und der NC-Steuerung zu vereinfachen und zu beschleunigen. Dies wurde durch die Definition von **autarken Maschinenmodulen** mit einfachen, standardisierten und reduzierten Modulschnittstellen und einer Automatisierung der **NC-Steuerungsadaption** erreicht (Bild 8.2).

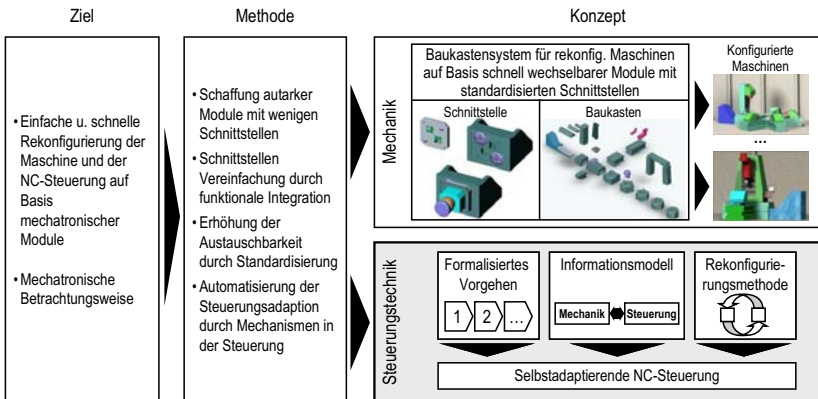


Bild 8.2: Fokus der SFB-Arbeiten und Grundlagen für die selbstadaptierende NC

Dazu wurden bestehende Maschinenkonzepte unter **funktionalen Gesichtspunkten** analysiert und in funktional zusammengehörende Einheiten zerlegt. Nach der funktionalen Analyse wurden neue Modulgrenzen so festgelegt, dass sich funktional zusammengehörende Kompo-

nennten in einem Modul befinden. Dadurch reduzieren sich die Ein- und Ausgangsschnittstellen der Module und es entstehen **mechatronische, autarke Module**. Auf Basis dieser Module wurde ein Baukastensystem für rekonfigurierbare Werkzeugmaschinen definiert (Kapitel 7.1). Dazu wurden die Schnittstellen der Module so konstruiert, dass sie schnell wechselbar und standardisiert aber dennoch genügend steif und genau sind. Zur Anpassung der NC-Steuerung wurden eine Methode (Kapitel 6) und ein Konzept für ein selbstadaptierendes System erarbeitet. Dazu wurde das Vorgehen zur NC-Konfigurierung formalisiert (Kapitel 5), ein Informationsmodell zur Abbildung der Zusammenhänge zwischen Mechanik und Steuerung definiert (Kapitel 7.2) und eine Methode zur **modellbasierten Konfigurierung und Rekonfigurierung** von NC-Steuerungen entwickelt (Kapitel 7.3).

8.1.1 Modulkonzept für Werkzeugmaschinen

Am Beispiel eines Gegenspindelmoduls einer Drehmaschine konnte im TFB 59 /137/ das **mechatronische Modulkonzept** /25, 46, 165, 166, 167/ exemplarisch umgesetzt werden (Bild 8.3).

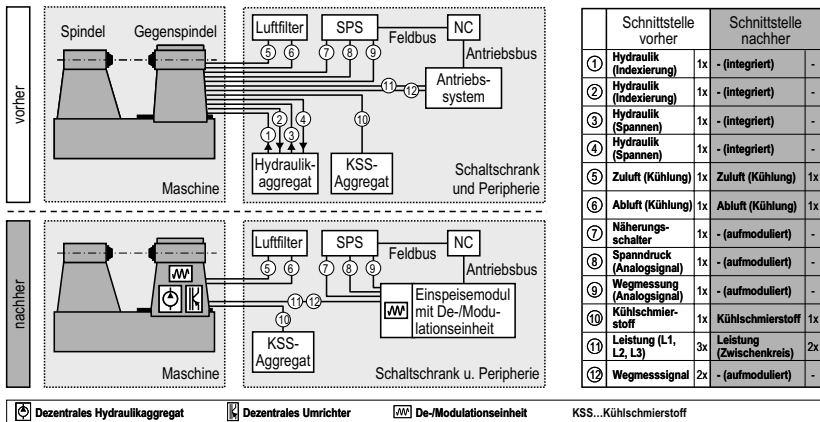


Bild 8.3: Schnittstellen eines mechatronischen Moduls vor und nach der Dezentralisierung

Durch den Einsatz dezentraler Peripherie, die Integration des lastseitigen Frequenzumrichters und des Hydraulikaggregats in das Gegenspindelmodul und den Einsatz von Aufmodulationstechniken zur Echtzeitkommunikation reduzieren sich die elektrischen Versorgungsleitungen auf zwei Energieversorgungsleitungen für den Zwischenkreis. Die Leitungen für Hydraulik können entfallen. Die Signale für die SPS werden über ein Aufmodulationsverfahren übertragen /168/. Außerdem wurden zur schnellen Auswechslung des Moduls bei einer

Rekonfigurierung die mechanischen Schnittstellen vereinfacht und eine Ausrichtmethode entwickelt.

Darüber hinaus wurde am Beispiel eines modularen, rekonfigurierbaren Verkettungssystems gezeigt, dass dezentrale Steuerungstechnik in Modulen eingesetzt und über ethernetbasierte Echtzeitkommunikationssysteme miteinander verknüpft werden kann /169/. In diesem Beispiel wurde zwar die Modulgröße zu Demonstrationszwecken so gewählt, dass die Systemgrenze eine ganze Maschine ist, die Realisierung mittels dezentraler Steuerungstechnik konnte aber dennoch gezeigt werden. Sie ist übertragbar auf mechatronische Module wie sie im Sinne dieser Arbeit in Kapitel 7.1.1 definiert wurden.

8.1.2 Informationsmodell und modellbasierte Konfigurierungsmethode

Im Rahmen des SFB 467 wurde im Teilprojekt C6 /136/ die Kernkomponente der selbstadaptierenden NC-Steuerung, das **Informationsmodell**, prototypisch in einem Softwarewerkzeug zur modellbasierten Rekonfigurierung von Bearbeitungssystemen (**Soft-moRe**) umgesetzt. Mit dem Softwarewerkzeug ist es möglich, nach der Methode aus Kapitel 7.3, **modellbasiert Konfigurierungen** und **Rekonfigurierungen** von NC-Steuerungen durchzuführen und NC-Konfigurierungslisten zu generieren.

In dem Softwarewerkzeug ist ein Informationsmodell mit Teilmodellen für Steuerung und Mechanik in Form von Komponentenbibliotheken und Konfigurationsmusterbibliotheken abgebildet. Zusätzlich sind die Abhängigkeiten zwischen den NC-Steuerungskonfigurationsobjekten und den Maschinenmodulobjekten formal beschrieben. Bild 8.4 zeigt einen Screenshot des Konfigurierungswerkzeugs Soft-moRe.

Im linken unteren Fenster ist die Komponentenbibliothek zu sehen, in der die Maschinenmodule abgebildet sind. Zur Strukturierung wurden die Module klassifiziert. Z.B. wurden Bearbeitungsadapter, Konsol- und Mitteneinheiten als Gestell abstrahiert und zusammengefasst. Gestell ist hier eine abstrakte Klasse, da es weitere Unterklassen gibt. Die Mitteneinheiten lassen sich in Mitten-Einheit-Standard und Polygoneinheit weiter unterscheiden. Hier im Beispiel ist das Modul Mitten-Einheit-Standard eine konkrete Klasse, die direkt im Konfigurationsmuster eingebaut werden kann.

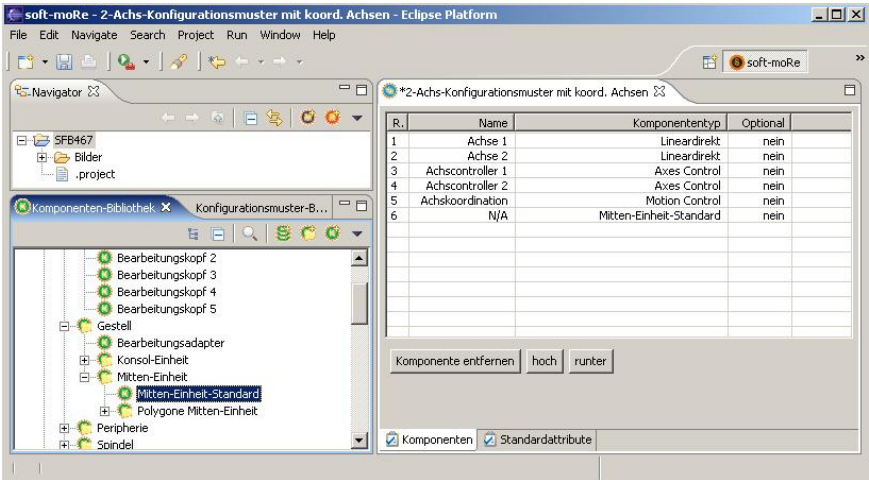


Bild 8.4: Screenshot von Soft-moRe

Neben der Komponentenbibliothek ist in [Bild 8.4](#) der Reiter für die Konfigurationsmusterbibliothek zu sehen. Im rechten Fenster ist der Aufbau eines Konfigurationsmusters dargestellt. In der Spalte „Komponententyp“ stehen die Modulklassen, die in die Platzhalter des Konfigurationsmusters aus der Bibliothek ausgewählt und eingefügt werden müssen (z.B. ist das Modul „Achskoordination“ ein Modul von der Klasse „Motion Control“). Im Bild ist zu erkennen, dass im Konfigurationsmuster noch ein Modul vom Typ „Mitten-Einheit-Standard“ fehlt (N/A) und aus der Bibliothek ausgewählt werden muss.

Die Kopplung zwischen Werkzeug und NC-Steuerung sieht so aus, dass die Funktionalität des Werkzeugs mit dem Informationsmodell komplett in dem Konfigurierungskordinator integriert ist (Kapitel 7.5). Als NC-Steuerung kommt jede am Markt verfügbare NC-Steuerung in Frage, die über Konfigurierungslisten adaptierbar ist.

8.2 Bewertung der Realisierung

Die während der Realisierung im Rahmen des SFB 467 und TFB 59 gewonnenen Erkenntnisse zeigen, dass eine Selbstadaption nicht vollautomatisch ablaufen kann. Ein Inbetriebnehmer muss bei der NC-Konfigurierung unterstützen und eingreifen, da die Selbstadaptionmethoden die Anordnung der Bearbeitungsachsen und die Bauform der Maschine nicht automatisch erkennen können.

Da die Modularität einer Werkzeugmaschine herstellerspezifisch ist, lassen sich keine allge-

meingültigen Module beschreiben. Damit die in dieser Arbeit beschriebene Selbstadaption funktioniert, muss daher jede Maschinenkonfiguration herstellerspezifisch im Informationsmodell abgebildet werden. Dadurch ist der Aufwand für die Modellerstellung sehr hoch und die modellbasierte Methode der selbstadaptierenden NC-Steuerung nur bei vorher modellierten Maschinenkonfigurationen anwendbar. Die Definition von Maschinenmodellobjektclassen im Informationsmodell ermöglicht zwar eine einfache Erweiterung des Informationsmodells, bevor allerdings eine neue, vorher noch nicht abgebildete Maschinenkonfiguration erstellt werden soll, muss sie modelliert werden.

Das Strukturmodell einer adaptierbaren NC-Steuerung wurde in Kapitel 5.3.2 eingeführt, um die herstellerspezifischen Konfigurierungsinformationen zu abstrahieren. Für eine herstellerspezifische, konkrete Umsetzung der selbstadaptierenden NC-Steuerung müssen die NC-Konfigurierungsobjekte steuerungsspezifisch durch eine objektorientierte Analyse definiert und im Informationsmodell abgebildet werden.

Zusammenfassend hat die erarbeitende Selbstadaptionmethode folgende Einschränkungen:

- Damit die Konfigurierung nach dem PnP-Prinzip funktioniert, müssen die **mechatronischen Maschinenmodule** vorparametriert sein.
- Für die modellbasierte Konfigurierung muss die **herstellerspezifische Modularität** im Informationsmodell abgebildet werden.
- Für die modellbasierte Generierung von NC-Konfigurierungslisten müssen die **NC-Konfigurierungsobjekte steuerungsspezifisch** definiert und im Informationsmodell abgebildet werden.

Dennoch kann das Konzept der selbstadaptierenden NC-Steuerung eine Erleichterung der Konfigurierung bringen. Im Gegensatz zu Lösungen einiger Steuerungshersteller, bei denen lediglich einzelne Steuerungskomponenten nach dem Einschalten der Steuerung erkannt werden, geht das in dieser Arbeit vorgestellte Konzept der selbstadaptierenden NC-Steuerung weiter. Es stellt Methoden bereit, bei denen **logische Zusammenhänge** zwischen Maschinenmodulen und Steuerungskomponenten genutzt werden, um eine NC-Steuerung automatisiert zu konfigurieren.

9 Zusammenfassung

Produzierende Unternehmen müssen zukünftig stärker als bisher auf Turbulenzen, ausgelöst durch die Globalisierung und Massenindividualisierung, reagieren können. Hinzu kommt, dass wegen den sich rasch ändernden Marktbedürfnissen die Lebenszyklen von Konsumgütern kürzer sind als die Amortisationszeit der Maschinen und Anlagen auf denen sie gefertigt werden. Deshalb müssen Maschinen und Anlagen wiederverwendbar und an neue Produktionsprozesse anpassbar sein. Für eine marktgerechte, schnelle Reaktion auf unvorhersehbare Fertigungsaufgaben werden daher zukünftig Werkzeugmaschinen benötigt, die an veränderte und nicht planbare Situationen rasch anpassbar, das heißt durch Umbau oder Erweiterung rekonfigurierbar sind. Dies betrifft sowohl den mechanischen Aufbau als auch das Steuerungssystem.

Bei einer Rekonfigurierung von Werkzeugmaschinen muss neben dem Umbau der Maschinenstruktur durch Austausch von Modulen auch die NC-Steuerung an die geänderte Maschinenkonfiguration angepasst werden. Dabei sind Abhängigkeiten zwischen der maschinenbaulichen Struktur der Maschine und der NC-Steuerung zu beachten. Sie werden bisher von Spezialisten, die ein mentales Modell dieser Abhängigkeiten besitzen, bei der einmaligen Konfigurierung der NC-Steuerung während der Auslegung und Inbetriebnahme berücksichtigt. Dadurch ist die NC-Steuerung fest an eine Maschinenkonfiguration angepasst und muss bei einer Rekonfigurierung jedes Mal aufwändig neu konfiguriert werden.

Für die einfache, schnelle und wirtschaftliche Rekonfigurierung von Werkzeugmaschinen muss die Adaption der NC-Steuerung deshalb weitestgehend automatisch ablaufen. In dieser Arbeit wurde daher durch Übertragung von Plug-and-Play-Prinzipien auf die Konfigurierung von NC-Steuerungen eine selbstadaptierende NC-Steuerung entworfen, die sich automatisiert an die Maschinenkonfiguration anpasst.

Zusammenfassend wurden folgende wesentliche Neuerungen gegenüber dem Stand der Technik erarbeitet:

- Aus der **Systematisierung der NC-Konfigurierung** wurde ein **herstellerunabhängiges Strukturmodell einer adaptierbaren NC-Steuerung** abgeleitet. Es basiert auf allgemeingültigen NC-Konfigurierungsobjekten über welche die NC-Steuerung konfiguriert wird.
- In einer Analyse von Plug-and-Play-Systemen wurden Merkmale von Plug-and-Play-Geräten herausgearbeitet und **Voraussetzungen für die Übertragbarkeit von Plug-and-Play-Prinzipien** auf die Steuerungstechnik definiert.

- Die Übertragung der Plug-and-Play-Merkmale auf Werkzeugmaschinen führte zur Definition eines Konzepts für eine **selbstadaptierende NC-Steuerung** und eines **Modulkonzepts für Werkzeugmaschinen** auf Basis **mechatronischer** Maschinenmodule.
- Zur Abbildung von Maschinen- und Steuerungskonfigurationen, sowie ihrer **funktionalen Zusammenhänge** wurde ein **Informationsmodell** geschaffen. Darin sind die NC-Konfigurierungsobjekte einer adaptierbaren NC-Steuerung abgebildet. Zusammen mit den Maschinenmodulen stellen sie Bausteine eines Baukastensystems dar. Zwischen diesen Bausteinen sind Beziehungen definiert, welche die funktionalen Abhängigkeiten zwischen Mechanik und Steuerung abbilden.
- Für die Selbstadaption der NC-Steuerung wurde eine **modellbasierte Konfigurierungsmethode entwickelt**. Dabei werden im Informationsmodell über eine Kennung die Klasse eines jeden Maschinenmoduls und seine Funktion innerhalb der Maschinenkonfiguration ermittelt. Über die funktionalen Zuordnungen im Informationsmodell werden anschließend alle notwendigen NC-Konfigurierungsobjekte abgeleitet. Ihr Zusammenwirken innerhalb der NC-Konfiguration wird durch die Auswahl eines Konfigurationsmusters festgelegt. Ein Generator erzeugt auf Basis des Informationsmodells eine NC-Konfigurierungsliste, die mit den Parametern aus dem Informationsmodell parametrisiert und anschließend von der NC-Steuerung interpretiert wird.

Das in dieser Arbeit vorgestellte Konzept zeigt, welche Voraussetzungen für die Umsetzung selbstadaptierender NC-Steuerungen zu schaffen und welche Randbedingungen einzuhalten sind. Im Rahmen des SFB 467 und dem TFB 59 konnte exemplarisch gezeigt werden, dass Komponenten einer selbstadaptierenden NC-Steuerung umsetzbar sind.

10 Literaturverzeichnis

- /1/ Westkämper, E.; Turbulenz in der PPS – eine Analogie.
Wiendahl, H.-H.; In: wt Werkstattstechnik online 90 (2000), Nr. 5, S. 203–207
Pritschow, G;
u.a.:
- /2/ Wiendahl, H.-H.; Turbulenzen erschweren die Planungssicherheit: Analogien aus
Rempp, B.;’ der Physik erklären das Entstehen und Beherrschen von Turbulen-
Schanz, M.: zen.
In: io management 69 (2000), Nr. 5, S. 38–43
- /3/ Westkämper, E.; Wandlungsfähigkeit: Ausprägungen und Anforderungen Ergebnis-
u.a.: se einer empirischen Untersuchung. Studie „Turbulenz und Wand-
lungsfähigkeit“.
Stuttgart: Eigenverlag Fraunhofer IPA, 2003
- /4/ Uhly, V.: Anforderungen an die Produktion komplexer Maschinen – Zu-
nehmende Schwankung der Auslastung erhöhen Zwang zur Flexi-
bilität.
In: VDMA Nachrichten (2003), Nr. 12, S. 46-48
- /5/ Bullinger, H.-J.: Forschung und Entwicklung in der Produktion – Kernkompetenz
für Marktwachstum.
In: Brecher, C.; Klocke, F.; Schmitt, R.; Schuh, G. (Hrsg.): Wett-
bewerbsfaktor Produktionstechnik – Aachener Perspektiven. Aa-
chener Werkzeugmaschinen-Kolloquium, 5. und 6. Juni 2008,
Aachen. Aachen: Apprimus Verlag, 2008, S. 1-11
- /6/ Brecher, C; Integrative Produktionstechnik für Hochlohnländer.
u.a.: In: Brecher, C.; Klocke, F.; Schmitt, R.; Schuh, G. (Hrsg.): Wett-
bewerbsfaktor Produktionstechnik – Aachener Perspektiven. Aa-
chener Werkzeugmaschinen-Kolloquium, 5. und 6. Juni 2008,
Aachen. Aachen: Apprimus Verlag, 2008, S. 13-39

- /7/ Drabow, G.: Die Fabrik der Zukunft.
In: Computer&AUTOMATION (2004), Nr. 2, S. 21- 25
- /8/ Heger, C. L.: Bewertung der Wandlungsfähigkeit von Fabrikobjekten.
Garbsen: PZH Produktionstechnisches Zentrum GmbH. 2007 (Berichte aus dem IFA, 01/2007). Hannover, Univ., Diss. 2006
- /9/ Wiendahl, H.-P.: Wandlungsfähigkeit – Schlüsselbegriff der zukunftsfähigen Fabrik.
In: wt Werkstattstechnik online 92 (2002), Nr. 4, S. 122-127
- /10/ Hernández Morales, R.: Systematik der Wandlungsfähigkeit in der Fabrikplanung.
Düsseldorf: VDI Verlag. 2003 (Fortschr.-Ber. VDI Reihe 16 Nr. 149). Hannover, Univ., Diss., 2002
- /11/ Kirchner, S.; Winkler, R.; Westkämper, E.: Unternehmensstudie zur Wandlungsfähigkeit von Unternehmen.
In: wt Werkstattstechnik online 93 (2003), Nr. 4, S. 254-260
- /12/ Westkämper, E.; Zahn, E.; Balve, P.; Tilebein, M.: Ansätze zur Wandlungsfähigkeit von Produktionsunternehmen – Ein Bezugsrahmen für die Unternehmensentwicklung im turbulenten Umfeld.
In: wt Werkstattstechnik online 90 (2000), Nr. 1/2, S. 22-26
- /13/ Armbruster, P.; u.a.: Werkzeugmaschine 2010. Untersuchung zum Forschungsbedarf bei spanenden Werkzeugmaschinen bis zum Jahr 2010.
In: wt Werkstattstechnik online 92 (2002), Nr. 7/8, S. 389-391.
- /14/ Gausemaier, J.; Binger, V.; Dreher, C.; Kinkel, S.: WZM 20XX – Initiative für die Werkzeugmaschine von morgen.
In: ZWF Zeitschrift für wirtschaftlichen Fabrikbetrieb 99 (2004), Nr. 4, S. 146-151

- /15/ Verl, A.;
Kremer, M.;
Kircher, C.;
Sekler, P.: Steuerungstechnik – Quo Vadis?
In: Gesellschaft für Fertigungstechnik (Hrsg.): Fertigungstechnik für die Zukunft. Fertigungstechnisches Kolloquium, 20. und 21. September 2006, Stuttgart. Stuttgart: Gesellschaft für Fertigungstechnik, 2006, S. 77-98
- /16/ National
Research
Council: Visionary manufacturing challenges for 2020.
Washington, D.C.: National Academy Press, 1998
- /17/ N.N. Report of the High-Level Group.
ManuFuture – A Vision for 2020. Lissabon, November 2004
- /18/ N.N. ManuFuture 2006: “Implementing the ManuFuture Strategy”.
4th annual conference on the Future of Manufacturing in Europe. Konferenz, Tampere (Finnland), 9. und 10. Oktober 2006
- /19/ Mellor, C.: Quick-change artists – Why techs must get ready for reconfigurable manufacturing.
In: The Ontario Technologist (2002), Nr. 1/2, S. 12-15
- /20/ Boër, C. R.;
Jovane F.: Flexible and Reconfigurable Manufacturing Automation for Mass Customisation.
15th IFAC World Congress on Automatic Control. Konferenz Barcelona, Spanien, 21. Juli 2002.
- /21/ Fili, W.: Längere Laufzeit für teures Investgut.
In: Industrieanzeiger (2008), Nr. 1/2, S. 26-29
- /22/ Stanik, M.: Rekonfigurierbarkeit schafft Flexibilität für die Großserienfertigung.
In: MM MaschinenMarkt (2008), Nr. 1/2, S. 20-22
- /23/ Möhring, H.-C.: Lebensdauererhöhung durch Anpassung.
In: phi (2003), 2, S. 10-11

- /24/ Wurst, K.-H.;
Heisel, U.;
Kircher, C.: (Re)konfigurierbare Werkzeugmaschinen – notwendige Grundlage für eine flexible Produktion.
In: wt Werkstattstechnik online 96 (2006) Nr. 5, S. 257-265
- /25/ Kircher, C.;
Meitzner, M.;
Heisel, U.;
Wurst, K.-H.: Wandelbare, zielvariable Bearbeitungssysteme.
In: Industrie Management 20 (2004) Nr. 2, S. 17-20
- /26/ Drabow, G.;
Woelk, P.-O.: Wandlungsfähige Fabrik der Zukunft – Modulare Struktur- und Steuerungskonzepte.
In: ZWF Zeitschrift für wirtschaftlichen Fabrikbetrieb 99 (2004), Nr. 3, S. 90-94
- /27/ N.N. Mauser-Werke Oberdorf Maschinenbau GmbH: Das modulare Produktionssystem – PS Baureihe.
Oberndorf a.N., 2006 – Firmenschrift
- /28/ N.N. Cross Hüller Deutschland: Agile Fertigungssysteme für Zylinderköpfe.
Ludwigsburg, 2004 – Firmenschrift
- /29/ N.N. ELHA-Maschinenbau Liemke KG: Fertigungsmodule. Vom Entwickler der Fertigungsmodule – Die neue Generation der rekonfigurierbaren Werkzeugmaschinen.
Hövelhof, 2005 – Firmenschrift
- /30/ N.N. INDEX-Werke GmbH & Co. KG Hahn & Tessky: RatioLine G200/G300 - Dreh-Fräszentren.
Esslingen, 2008 – Produktkatalog

- /31/ Heisel, U.: Innovationen und Trends im Werkzeugmaschinenbau.
In: Gesellschaft für Fertigungstechnik (Hrsg.): Fertigungstechnik für die Zukunft. Fertigungstechnisches Kolloquium, 10. und 11. September 2008, Stuttgart. Stuttgart: Gesellschaft für Fertigungstechnik, 2008, S. 125-158
- /32/ N.N. Norm DIN ISO 2806 1996-04:
Numerische Steuerung von Maschinen – Begriffe
- /33/ N.N. Norm DIN 66025 Teil 2 1988-09:
Programmaufbau für numerisch gesteuerte Arbeitsmaschinen – Wegbedingungen und Zusatzfunktionen
- /34/ N.N. SERCOS International e.V.: SERCOS interface – Universelle Echtzeit-Kommunikation mit Ethernet.
Süssen, 2007 – Broschüre
- /35/ N.N. EtherCAT Technology Group: The Ethernet Fieldbus.
Nürnberg, 2007 - Broschüre
- /36/ N.N. DIN IEC 61158 Teil 500 2009-06:
Industrielle Kommunikationsnetze – Feldbusse - Teil 500: Dienstfestlegungen des Application Layer (Anwendungsschicht);
- /37/ N.N. Norm DIN EN 61131 Teil 3 2003-12:
Speicherprogrammierbare Steuerungen – Programmiersprachen
- /38/ N.N. DIN IEC 61784-2 2009-06:
Industrielle Kommunikationsnetze – Profile – Teil 2: Zusätzliche Feldbusprofile für Echtzeitnetzwerke basierend auf ISO/IEC 8802-3

- /39/ N.N. Wikipedia: OLE for Process Control.
http://de.wikipedia.org/wiki/OLE_for_Process_Control
(13.01.2009)
- /40/ Pritschow, G.: Einführung in die Steuerungstechnik.
München, Wien: Carl Hanser Verlag, 2006
- /41/ N.N. Norm DIN 19226 Teil 6 1997-09:
Leittechnik, Regelungstechnik und Steuerungstechnik Teil 6: Begriffe zu Funktions- und Baueinheiten
- /42/ Selig, A.;
Wetter, O.;
Rüppel, W.: Innovatives Gerätemodell und zukunftsweisendes I/O-Geräteprofil für SERCOS III.
In: Computer & Automation (2007), Nr. 11
- /43/ N.N. DIN EN 61800-7 2008-11:
Elektrische Leistungsantriebssysteme mit einstellbarer Drehzahl - Teil 7-303: Generisches Interface und Nutzung von Profilen für Leistungsantriebssysteme (PDS) - Abbildung von Profil-Typ 3 auf Netzwerktechnologien
- /44/ Lechler, A.;
Verl, A.: Einheitliche Kommunikationsschnittstelle für den funktionalen Zugriff auf ethernetbasierte Bussysteme.
In: Verl, A.; Bender, K.; Schumacher, W. (Hrsg.): Tagungsband SPS/IPC/Drives/Elektrische Automatisierung, Systeme und Komponenten, Kongress, 24. November 2009, Nürnberg. Berlin: VDE Verlag, 2009
- /45/ Verl, A.;
Pritschow, G.: Durchgängige Kommunikation für modulare Maschinenkonzepte.
In: SERCOS News (2008), Nr. 2, S. 2
- /46/ Kircher, C.: Einheitlichen Systemen gehört die Zukunft.
In: elektro AUTOMATION (2008), Nr. 4, S. 38-41

- /47/ Felser, M.: Vom Feldbus-Krieg zur Feldbus-Koexistenz.
In: Bulletin SEV/VSE (2002), Nr. 9, S. 1-3
- /48/ Daniel, C.: Dynamisches Konfigurieren von Software für offene Systeme.
Berlin: Springer-Verlag, 1997 (ISW Forschung und Praxis; 112).
Stuttgart, Univ., Diss., 1996
- /49/ N.N. Norm DIN 19226 Teil 5 1994-01:
Leittechnik, Regelungstechnik und Steuerungstechnik Teil 5:
Funktionelle Begriffe
- /50/ Koren, Y.; Reconfigurable Manufacturing Systems.
Jovane, F.; In: Annals of the CIRP 48 (1999), Nr. 2, S. 527-540
Pritschow, G.;
u.a.:
- /51/ Pritschow, G.; Control of Reconfigurable Machine Tools.
Wurst, K.-H.; In: Maraghy, H. (Hrsg.): Changeable and Reconfigurable Manu-
Kircher, C.; facturing Systems. London: Springer, 2008, S. 71-100
Seyfarth, M.:
- /52/ Michaelis, M.: Flexibilitätssteigerung in der Fertigung durch rekonfigurierbare
Bearbeitungssysteme
Tübingen: Medien Verlag Köhler. 2002 (Berichte aus dem Institut
für Werkzeugmaschinen; 23). Stuttgart, Univ., Diss., 2002
- /53/ Wurst, K.-H.; Conception of Reconfigurable Machining Systems – Design of
Kircher, C.; Components and Controllers.
Seyfarth, M.: In: Bramley, A.; Brissaud, D.; Coutellier, D.; McMahon, C.
(Hrsg.): Advances in Integrated Design and Manufacturing in Me-
chanical Engineering. Dordrecht: Springer, 2005, S. 417-433
- /54/ Rumbaugh, J.: Objektorientiertes Modellieren und Entwerfen.
München, Wien: Carl Hanser Verlag, 1993

- /55/ Jacobson, I.;
u.a.: Object-Oriented Software Engineering.
New York: Addison-Wesley, 1994
- /56/ Kohlhase, N.: Strukturieren und Beurteilen von Baukastensystemen: Strategien, Methoden, Instrumente.
Düsseldorf: VDI Verlag, 1997 (Fortschr.-Ber. VDI Reihe 1 Nr. 275). Darmstadt, Techn. Hochsch., Diss., 1996
- /57/ Brandl, T.: Anlageninformationsmodell – Informationsmodell und Erstellungssystematik.
Heimsheim: Jost Jetter Verlag, 1999 (ISW Forschung und Praxis; 130). Stuttgart, Univ., Diss. 1999
- /58/ Litto, M.: Störungsinformationssystem – Informationsmodell und Erstellungssystematik.
Heimsheim: Jost Jetter Verlag, 2002 (ISW Forschung und Praxis; 144). Stuttgart, Univ., Diss. 2002
- /59/ Balzert, H.: Lehrbuch der Software-Technik, Software-Entwicklung, 2. Auflage.
Heidelberg, Berlin: Spektrum Akademischer Verlag, 2000
- /60/ N.N. Object Management Group: Unified Modeling Language (UML)
<http://www.uml.org/> (20.08.2009)
- /61/ Jeckle, M.;
u.a.: UML 2 glasklar.
München: Carl Hanser Verlag, 2004
- /62/ N.N. The Object Management Group (OMG).
<http://www.omg.org/> (03.02.2009)

- /63/ N.N. Trendwende in der Steuerungstechnik – Herstellerübergreifend offene Systeme.
Frankfurt a. Main: VDW-Verlag, 1998 (VDW-Forschungsberichte)
- /64/ N.N. MPST Systembeschreibung.
Frankfurt: IFAO, MPST-Kreis e.v. (Hrsg.), Stuttgart, 1987
- /65/ N.N. DIN 66264 Teil 1 1988-08:
Mehrprozessor-Steuersystem für Arbeitsmaschinen (MPST); Parallelbus
- /66/ N.N. DIN 66264 Teil 2 1988-08:
Mehrprozessor-Steuersystem für Arbeitsmaschinen (MPST); Regeln für den Informationsaustausch
- /67/ Spieth, U.: Numerische Steuerungssysteme – Hardwareaufbau und Ablaufsteuerung eines Mehrprozessorsteuerungssystems.
Berlin: Springer-Verlag, 1982 (ISW Forschung und Praxis; 40).
Stuttgart, Univ., Diss., 1982
- /68/ Plasch, D.: Standardisierte Softwareschnittstellen in Mehrprozessorsteuerungssystemen
Berlin: Springer-Verlag, 1983 (ISW Forschung und Praxis; 46).
Stuttgart, Univ., Diss., 1983
- /69/ Lutz, P.;
Sperling, W.: OSACA – the Vendor Neutral Control Architecture.
In: Fichtner, D. (Hrsg.): Facilitating Deployment of Information and Communications Technologies. Proceedings of the Conference on Integration in Manufacturing, 24.-26. September 1997, Dresden, S. 247-256

- /70/ Pritschow, G.;
Koren, Y.;
Jovane, F.: Status of Specification and Implementation within OSACA/HÜMNOS Project.
In: Koren, Y., Jovane, F.; Pritschow, G. (Hrsg.): Open Architecture Control Systems – Summary of Global Activity. ITIA Series Vol. 2 (1998), S. 15-38
- /71/ Sperling, W.: Modulare Systemplattformen für offene Steuerungssysteme.
Berlin: Springer-Verlag, 1999 (ISW Forschung und Praxis; 128).
Stuttgart, Univ., Diss., 1998
- /72/ Bürger, T.: Durchgängige analytische Qualitätssicherung für numerische Steuerungssoftware.
Heimsheim: Jost Jetter Verlag, 2005 (ISW Forschung und Praxis; 153). Stuttgart, Univ., Diss. 2005
- /73/ Weck, M.;
Brecher, C.: Werkzeugmaschinen 4 – Automatisierung von Maschinen und Anlagen. 6. Aufl.
Berlin, u.a.: Springer, 2006
- /74/ Verl, A.: Steuerungstechnik der Werkzeugmaschinen und Industrieroboter.
Institut für Steuerungstechnik der Werkzeugmaschinen und Fertigungseinrichtungen, Stuttgart, 2009 – Skript zur Vorlesung
- /75/ N.N. Bosch Rexroth AG: Rexroth IndraMotion MTX - Die CNC-Systemlösung für perfektes Zerspanen und Umformen.
Lohr, 2008 – Produktbeschreibung
- /76/ N.N. World Wide Web Consortium: Extensible Markup Language (XML).
<http://www.w3.org/XML/> (02.01.2010)
- /77/ N.N. SINUMERIK Open Architecture - Durchgängig offen bis in den Kern.
Siemens AG, Erlangen, 2003 – Firmenschrift

- /78/ N.N. Typ3 osa. Dokumentation der Kundenschnittstelle Bewegungserzeugung.
Robert Bosch GmbH, Erbach, 2001 – Firmenschrift
- /79/ Angerbauer, R.: SPS-Software aus dem Baukasten – Ein Erfahrungsbericht.
In: VDMA Nachrichten (2002), Nr. 1, S. 45-46
- /80/ Scheffels, G.: Steuerungsprogramm automatisch generiert.
In: SPS Magazin (2007), Nr. 11, S. 11-14
- /81/ N.N. teamtechnik Maschinen und Anlagen GmbH: TEAMOS – Prozessmodulares System für flexible Montageanlagen.
Freiberg, 2008 – Firmenschrift
- /82/ N.N. Karl Utz Sondermaschinen GmbH: MOMOSYS – Modulares Montagesystem.
Limbach-Oberfrohna, 2008 – Firmenschrift
- /83/ N.N. European Association of the Machine Tool Industries (CECIMO):
MOSYN – Modular Synthesis of Advanced Machine Tools.
<http://www.cecimo.be/TEAMS/Projects/mosyn/mosyn.htm>
(16.05.2002)
- /84/ Tönshoff, H. K.; An Approach for the Concurrent Development and Manufacturing
Mey, M.; of Modular Machine Tools.
Schnülle, A.: In: Production Engineering V (1998), Nr. 1, S. 63-66
- /85/ N.N. OSACA Association: OSACA Handbook Version 2.0.
Stuttgart, 2001

- /86/ Tönshoff, H. K.; Schnülle, A.; Drabow, G.: Highly flexible and reconfigurable manufacturing systems due to a modular design.
In: Proceedings of the CIRP 1st International Conference on Agile and Reconfigurable Manufacturing, 21.-22. Mai 2001, Ann Arbor, USA
- /87/ Weck, M.; Ullrich, G.; Neuhaus, J.: Modulare Transfermaschinen – hochproduktiv und (re-)konfigurierbar.
In: wt Werkstattstechnik online 90 (2000), Nr. 3, S. 102-104
- /88/ Böger, F. H.: Herstellerübergreifende Konfiguration modularer Werkzeugmaschinen.
Düsseldorf: VDI Verlag, 1998 (Fortschr.-Ber. VDI Reihe 2 Nr. 462). Hannover, Univ., Diss., 1998
- /89/ Neuhaus, J.: Umkonfigurieren von Werkzeugmaschinen durch Plug & Play mechatronischer Module.
Aachen: Shaker Verlag, 2003 (Berichte aus der Produktionstechnik; 16/2003). Aachen, Techn. Hochsch., Diss. 2003
- /90/ Ullrich, G.: Konfigurierbare Steuerungssoftware für modulare Transfermaschinen auf Basis offener Steuerungssysteme.
Aachen: Shaker Verlag, 2006 (Berichte aus der Produktionstechnik; 11/2006). Aachen, Techn. Hochsch., Diss. 2005
- /91/ Heisel, U.; Michaelis, M.: Rekonfigurierbare Werkzeugmaschinen.
In: ZWF Zeitschrift für wirtschaftlichen Fabrikbetrieb 93 (1998), Nr. 10, S. 506-507

- /92/ Heisel, U.;
Michaelis, M.: Rekonfigurierbare Bearbeitungssysteme zur Steigerung der Produktivität.
In: Neugebauer, R. (Hrsg.): Offensivkonzepte wirtschaftlicher Produktionstechnik, 3. Chemnitzer Produktionstechnisches Kolloquium, 6. und 7. September 2001, Chemnitz. Zwickau: Verlag Wissenschaftliche Skripten, 2001, S. 269-285
- /93/ Heisel, U.;
Meitzner, M.: Rekonfigurierbare Bearbeitungssysteme – Modulare Werkzeugmaschinen-systeme zur Anpassung an den Produktlebenszyklus.
In: wt Werkstattstechnik online 94 (2004), Nr. 10, S. 517-520
- /94/ Heisel, U.;
Meitzner, M.: Progress in Reconfigurable Manufacturing Systems.
In: The International Journal for Manufacturing Science & Production 6 (2004), Nr. 1-2, S. 1-8
- /95/ Heisel, U.;
Meitzner, M.: Wandlungsfähige Produktionssysteme für die spanende Bearbeitung.
Symposium „Produktionssysteme in Zeiten des Wandels“ auf der internationalen Ausstellung für Metallbearbeitung, 16. September 2004, Messe Stuttgart
- /96/ Koren, Y.: Open Architecture Controllers for Manufacturing Systems.
In: Jovane, F.; Pritschow, G.; Koren, Y. (Hrsg.): Open Architecture Control Systems – Summary of Global Activity. Mailand: ITIA Series (1998), Vol. 2, S. 85-102
- /97/ Fujita, S.;
Yoshida, T.: OSE: Open System Environment for Controller.
In: Proceedings of the 7th International Machine Tool Engineers Conference, 16. und 17. November 1996, Tokyo, Japan, S. 234-243

- /98/ Ueno, S.;
Chino, S.
Hoshino, Y.;
Uneme, M.:
Development of the Standard Application Program Interface (API)
for Open FA controller in Japan.
In: ASPE Proceedings of the 15th Annual Meeting, 22.-27. Oktober
2000, Scottsdale, USA, S. 296-299
- /99/ Ueno, S.;
Mitsubishi, M.;
Muto, K.;
Takata, S.:
Standard API for Open-architecture CNC and It's Application to
HMI and Operation Monitoring.
In: Inasaki, I. (Hrsg.): Initiatives of Precision Engineering at the
Beginning of a Millennium. International Conference on Precision
Engineering (ICPE), 18.-20. Juli 2001, Yokohama (Japan). New
York, u.a.: Kluwer Academic Publishers, 2002, S. 789-793
- /100/ N.N.
OMAC: Organization for Machine Automation and Control.
<http://www.omac.org/> (13.01.2009)
- /101/ Yen, J.:
Overview of OMAC Activities and Global Open Architecture
Control Development Effort.
Deutsche Messe AG (Hrsg.): Konferenz zur "Trendwende in der
Steuerungstechnik", 1997, EMO Hannover
- /102/ N.N.
OSACA Association: OSACA - Open System Architecture for
Controls within Automation Systems.
<http://www.osaca.org/> (20.08.2009)
- /103/ Pritschow, G.;
Altintas, Y.;
Jovane, F.;
u.a.:
Open Controller Architecture – Past, Present and Future.
In: Annals of the CIRP 50 (2001), Vol. 1, S. 463-470
- /104/ Lutz, P.:
Offenheit als oberste Maxime – OSACA/HÜMNOS: Trotz PC-
Welt wird weiter an der offenen Steuerung gebaut.
In: Fertigung 9 (1999), S. 40-42

- /105/ Müller, J.: Objektorientierte Softwareentwicklung für offene numerische Steuerungen.
Berlin: Springer-Verlag, 1997 (ISW Forschung und Praxis; 120).
Stuttgart, Univ., Diss., 1997
- /106/ Pritschow, G.; Offene, echtzeitfähige Steuerungsarchitektur.
u.a.: In: Werkstattstechnik online 93 (2003) Nr. 5, S.374-378
- /107/ Pritschow, G.; OCEAN – Open Controller Enabled by an Advanced Real-Time
Weck, M.; Network.
u.a.: In: Production Engineering XI (2004), Nr. 1, S. 171-74
- /108/ Schmidt, D. C.: Real-time CORBA with TAO. (The ACE ORB)
<http://www.cs.wustl.edu/~schmidt/TAO.html> (14.01.2008)
- /109/ Lutz, R.; Konfigurierungswerkzeuge für offene Steuerungen.
Seyfarth, M.: In: Gesellschaft für Fertigungstechnik (Hrsg.): Stuttgarter Impulse.
Innovation durch Technik und Organisation. Fertigungstechnisches Kolloquium, 1997, Stuttgart. Berlin, u. a.: Springer, 1997, S. 500-506
- /110/ Müller, J.; Flexible Konfigurierung von Steuerungssystemen.
Schäfer, U.; In: Trendwende in der Steuerungstechnik – Herstellerübergreifend
Seyfarth, M.: offene Systeme. Frankfurt a. M.: VDW Verlag, 1998, Forschungsberichte, S. 45-58
- /111/ Walker, B.: Konfigurierbarer Funktionsblock Geomtriedatenverarbeitung für
numerische Steuerungen.
Berlin: Springer-Verlag, 1987 (ISW Forschung und Praxis; 68).
Stuttgart, Univ., Diss., 1987
- /112/ N.N. Bosch Rexroth AG: Rexroth IndraWorks 10VRS Engineering.
Lohr, 2010 – Bedien- und Programmieranleitung

- /113/ N.N. Siemens AG: STARTER – Die Antriebs-/Inbetriebnahmesoftware.
Erlangen, 2008 - Produktpräsentation
- /114/ N.N. Beckhoff Automation GmbH: TwinCAT CNC: Überblick
Verl, 2010 – Software Dokumentation
- /115/ Angerbauer, R.; Sondermaschinen effizient engineert – SPS-Programme und
Lewek, J.; Schaltpläne auf Knopfdruck
Litto, M.: In: IEE 48 (2003), Nr. 04, S. 32-34
- /116/ Angerbauer, R.; Funktionales Engineering – Baukasten für disziplinübergreifendes
Lewek, J.; Projektieren.
Litto, M.: In: IEE 49 (2004), Nr. 3, S. 48-50
- /117/ N.N. HMI-Software.
In: Computer & Automation (2001) H. 1, S. 60 ff
- /118/ Weisbecker, A.: Ein Verfahren zur automatischen Generierung von software-
ergonomisch gestalteten Benutzungsoberflächen.
Berlin: Springer-Verlag, 1995 (IPA-IAO Forschung und Praxis,
221). Stuttgart, Univ., Diss., 1995
- /119/ Bauer, G.; Offen und flexibel. Konfigurierbares Bediensystem für offene
Müller, M.; Steuerungen auf PC-Basis.
Schittenhelm, W.: In: Elektronik 45 (1997), H. 16, S. 52–56
- /120/ Pritschow, G.; Adaptives HMI für konfigurierbare offene Steuerungen.
Bauer, G.: In: wt Werkstattstechnik online 94 (2004), Nr. 5, S. 172- 178
- /121/ N.N. OSACA Association: Styleguide Werkzeugmaschinen – Ein
Handbuch zur Gestaltung von Benutzungsoberflächen für Werk-
zeugmaschinen.
Stuttgart: Fraunhofer IRB Verlag, 1997

- /122/ N.N. Siemens AG: SINUMERIK 840D/810D Inbetriebnahmehandbuch (IADC) – Ausgabe 03/2006.
Erlangen, 2006 – Technische Dokumentation
- /123/ N.N. Siemens AG: IBN CNC: NCK, PLC, Antrieb - Inbetriebnahmehandbuch, 03/2009.
Erlangen, 2009 – Technische Dokumentation
- /124/ N.N. Bosch Rexroth AG: Rexroth IndraMotion MTX 09VRS - Inbetriebnahme.
Lohr, 2008 – Inbetriebnahmeanleitung
- /125/ N.N. Industrielle Steuerungstechnik GmbH: Dokumentation Hochlauf-
liste.
Stuttgart, 2006 – Technische Dokumentation
- /126/ Projektge- Abschlussbericht des BMBF-Verbundprojekts MoWiMa (Model-
meinschaft lierung und Wiederverwendung objektorientierter Maschinensoft-
MoWiMa: ware).
Frankfurt am Main: VDMA Maschinenbau Institut GmbH, 1998
- /127/ Pritschow, G.; Werkzeuggestützte Konfigurierung offener Steuerungen.
Seyfarth, M.: In: wt Werkstattstechnik online 93 (2003), Nr. 5, S. 351-354
- /128/ Lutz, R.: Softwaretechnik für maschinennahe Steuerungsfunktionen bei
Fertigungseinrichtungen.
Heimsheim: Jost Jetter Verlag, 1999 (ISW Forschung und Praxis; 132). Stuttgart, Univ., Diss. 1999
- /129/ N.N. Object Management Group: OMG's MetaObject Facility (MOF).
<http://www.omg.org/mof/> (03.02.2009)

- /130/ Lewek, J.: Adaptierbares Informationssystem zur Erstellung baukastenbasierter Fertigungseinrichtungen.
Heimsheim: Jost Jetter Verlag, 2005 (ISW Forschung und Praxis; 151). Stuttgart, Univ., Diss. 2004
- /131/ N.N. FÖDERAL. Projekt-Homepage.
<http://www.foederal.org/index.htm> (03.02.2009)
- /132/ Litto, M.;
u.a.: Baukastenbasiertes Engineering mit Föderal – Ein Leitfaden für Maschinen- und Anlagenbauer.
Frankfurt a. M.: VDMA Verlag, 2004
- /133/ Angerbauer, R.: Baukastensystematik in der Steuerungstechnik auf Basis einer Föderalen Informationsarchitektur.
Institut für Steuerungstechnik der Werkzeugmaschinen und Fertigungseinrichtungen: Workshop „Fit für die maßgeschneiderte Massenfertigung durch agile, rekonfigurierbare Fertigungssysteme“, 20. März 2002, Stuttgart
- /134/ N.N. Mind8 GmbH: Mind8 engineering center.
Stuttgart, 2009 – Firmenschrift
- /135/ Tönshoff, H. K.;
Schnülle, A.: Highly Productive and Re-configurable Modular Manufacturing Systems.
Tagungsband, CIRP Konferenz Haifa, Israel, Mai 2000
- /136/ Heisel, U.;
Wurst, K.-H.;
Kircher, C.;
u.a.: Wandelbare, zielvariable Bearbeitungssysteme.
In: Westkämper, E. (Hrsg.) u.a.: SFB 467 – Abschlussbericht.
Stuttgart: Eigenverlag, 2006

- /137/ Heisel, U.;
Wurst, K.-H.;
Kircher, C.
u.a.:
Konzeption und Entwicklung mechatronischer Module für wandelbare Bearbeitungssysteme und Untersuchung deren Auswirkungen auf die Gestaltung von Fertigungseinrichtungen Wandelbare, zielvariable Bearbeitungssysteme.
In: Westkämper, E. (Hrsg.) u.a.: TFB 059 – Abschlussbericht. Stuttgart: Eigenverlag, 2009
- /138/ Heisel, U.;
Meitzner, M.:
Reconfigurable Manufacturing Systems.
In: Academic Journal of Manufacturing Engineering (2003), Nr. 1, S. 6-9
- /139/ Heisel, U.;
Meitzner, M.:
Flexibilitäts- und Produktivitätssteigerung durch rekonfigurierbare Werkzeugmaschinen-systeme.
In: Modern trends in manufacturing. Proceedings of the first international CAMT conference, 7. und 8. Februar 2002, Wrocław, S. 179-196
- /140/ Pruschek, P.:
Verfahren zur anwendungsgerechten Parametrierung der Steuerung und Regelung von Vorschubachsen.
Heimsheim: Jost Jetter Verlag, 2009 (ISW Forschung und Praxis; 172). Stuttgart, Univ., Diss. 2009
- /141/ Lapper, S.:
Untersuchung der Konfigurierungsmöglichkeiten marktgängiger NC-Steuerungssysteme.
Institut für Steuerungstechnik der Werkzeugmaschinen und Fertigungseinrichtungen, Studienarbeit (Betreuer: C. Kircher, M. Seyfarth), Stuttgart, 2007
- /142/ Wöhner, M.:
Untersuchung der Konfigurierungsmöglichkeiten marktgängiger NC-Steuerungssysteme.
Institut für Steuerungstechnik der Werkzeugmaschinen und Fertigungseinrichtungen, Studienarbeit (Betreuer: C. Kircher, M. Seyfarth), Stuttgart, 2007

- /143/ Verl, A.: Angewandte Regelungstechnik in Produktionsanlagen.
Institut für Steuerungstechnik der Werkzeugmaschinen und Fertigungseinrichtungen, Stuttgart, 2009 – Skript zur Vorlesung
- /144/ N.N. Plug and Play.
http://de.wikipedia.org/wiki/Plug_and_Play (06.01.2010)
- /145/ N.N. USB Implementers Forum, Inc.: Universal Serial Bus.
<http://www.usb.org/home> (20.01.2010)
- /146/ N.N. Standard IEEE 1394 1996:
IEEE Standard for a High Performance Serial Bus. The Institute of Electrical And Electronics Engineers, New York.
- /147/ N.N. FireWire
<http://de.wikipedia.org/wiki/Firewire> (03.02.2009)
- /148/ Anderson, D.: FireWire System Architecture, 2. Auflage.
München: Addison Wesley, 1999
- /149/ Selig, A.; Kircher, C.: Grundlegende Betrachtung zu Plug and Play-Konzepten im Maschinen- und Anlagenbau am Beispiel SERCOS III.
2. Rexroth Doktoranden Kolloquium, Lohr a. Main, 2006
- /150/ Blind, R.: Plug-and-Play Konzepte.
Institut für Steuerungstechnik der Werkzeugmaschinen und Fertigungseinrichtungen, Studienarbeit (Betreuer: C. Kircher), Stuttgart, 2005
- /151/ N.N. Norm ISO/IEC 13213 (ANSI/IEEE 1212) 1994:
Information technology – Microprocessor systems – Control and Status Registers (CSR) Architecture for microcomputer buses.

- /152/ Lewek, J.;
Lutz, R.;
Storr, A.:
Konfigurierung von Steuerungssoftware für modulare Maschinenkonzepte.
In: Pritschow, G. (Hrsg.), u.a.: Modulare flexible Systeme in der Fertigung. Düsseldorf: VDI Verlag, 2001 (Fortschr.-Ber. VDI Reihe 2 Nr. 578)
- /153/ Klemm, P.;
Rüdele, H.;
Weimer, T.:
Durchgängiges Engineering von mechatronischen Systemen.
In: Gesellschaft für Fertigungstechnik (Hrsg.): Fertigungstechnik für die Zukunft. Fertigungstechnisches Kolloquium, 21. und 22. September 2006, Stuttgart. Stuttgart: Gesellschaft für Fertigungstechnik, 2006, S. 101-131
- /154/ Kircher, C.;
Wurst, K.-H.:
Modulkonzept für rekonfigurierbare Bearbeitungs- und Steuerungssysteme.
In: ZWF Zeitschrift für wirtschaftlichen Fabrikbetrieb 101 (2006), Nr. 6, S. 360-365
- /155/ Pritschow, G.;
Wurst, K.-H.;
Seyfarth, M.;
Bürger, T.:
Requirements for Controllers in Reconfigurable Machining Systems.
CIRP 2nd International Conference on Reconfigurable Manufacturing, University of Michigan, Ann Arbor, USA, 19. August 2003
- /156/ Kircher, C.;
Seyfarth, M.;
Wurst, K.-H.:
Modellbasiertes Rekonfigurieren von Werkzeugmaschinen.
In: wt Werkstattstechnik online 94 (2004), Nr. 5, S. 179-183
- /157/ Kircher, C.;
Seyfarth, M.;
Wurst, K.-H.;
Pritschow, G.:
Self-Adapting Control Systems for RMS.
CIRP-sponsored 3rd International Conference on Reconfigurable Manufacturing, University of Michigan, Ann Arbor, USA, 11. Mai 2005
- /158/ Li, R.:
Agentenbasierte NC-Planung für die Komplettbearbeitung auf Dreh-/ Fräszentren.
Heimsheim: Jost Jetter Verlag, 2001 (ISW Forschung und Praxis; 137). Stuttgart, Univ., Diss. 2000

- /159/ Weck, M.;
Brecher, C.:
Werkzeugmaschinen 2 – Konstruktion und Berechnung. 8. Aufl.
Berlin, u.a.: Springer, 2006
- /160/ Haubelt, A.;
Verl, A.:
Der Weg zur automatischen Generierung von Simulationsmodellen aus mechatronischen Baukästen.
In: Brecher, C. (Hrsg.); u.a.: Effiziente Methodiken und durchgängige Werkzeuge zur Modellerstellung digitaler Produktionseinrichtungen. Fortschritt-Berichte VDI, Reihe 2, Nr. 671. Düsseldorf: VDI-Verlag, 2009
- /161/ Verl, A.;
Müller, V.;
Haubelt, A.:
Baukastenbasiertes simulationsgestütztes Engineering.
In: A&D-Kompodium 2009/2010, S.32-34
- /162/ Kircher, C.;
Wurst, K.-H.:
Steuerungssysteme für rekonfigurierbare Werkzeugmaschinen.
AMB-Messe, SFB 467-Industriekolloquium, Stuttgart, 16. September 2004
- /163/ Pritschow, G.;
Kircher, C.;
Kremer, M.;
Seyfarth, M.:
Control Systems for RMS and Methods of their Reconfiguration.
In: Dashchenko, A. I. (Hrsg.): Reconfigurable Manufacturing Systems and Transformable Factories. Berlin, u.a.: Springer, 2006, S. 195-209
- /164/ Lutz, P.;
Schneider, B.:
Präzise im Takt.
In: Computer&AUTOMATION (2007), Nr. 8, S. 74-79
- /165/ Kircher, C.;
u.a.:
Das Stuttgarter Unternehmensmodell in der Theorie – Technik und Strukturen.
In: Westkämper, E.; Zahn, E. (Hrsg.): Wandlungsfähige Produktionsunternehmen – Das Stuttgarter Unternehmensmodell. Berlin, u.a.: Springer, 2009, S. 62-199

- /166/ Wurst, K.-H.;
Schmitz, S.;
Lechler, A.: Integrierte mechatronische Module für rekonfigurierbare Bearbeitungssysteme.
Internationales Forum Mechatronik, 12. und 13. März 2007, Winterthur, Schweiz.
- /167/ Lechler, A.,
Schmitz, S.,
Kircher, C.,
Wurst, K.-H.: Flexible Reconfiguration and Maintenance through Mechatronic Moduls in Manufacturing Systems.
ICPR, 19th International Conference in Production Research, Valparaiso. Chile, 2007
- /168/ Schmitz, S.;
Yang, D.;
Wurst, K.-H.: A new Communication System for Reconfigurable Mechatronic Modules.
15th International Conference on Mechatronics and Machine Vision in Practice, Massey University, Auckland, New Zealand, 2. Dezember 2008
- /169/ Lechler, A.;
Kircher, C.;
Verl, A.: Simulationsgestütztes mechatronisches Engineering – Entwurf und Inbetriebnahme mechatronischer Module und flexibler Produktionssysteme.
In: wt Werkstattstechnik online 98 (2008), Nr. 5, S. 377-383

Lebenslauf

Persönliche Daten:	Name	Christian Kircher
	Geburtstag	01.07.1977
	Geburtsort	Stuttgart-Bad Cannstatt
	Eltern	Wolfgang und Eva Kircher
Schulbildung:	1984 – 1988	Grundschule, Leonberg
	1988 – 1997	Gymnasium, Leonberg
Studium:	1997 – 2002	Automatisierungstechnik in der Produktion, Schwerpunkt Mechatronik, Universität Stuttgart
Berufstätigkeit:	2003 – 2006	Wissenschaftlicher Mitarbeiter am Institut für Steuerungstechnik der Werkzeugmaschinen und Fertigungseinrichtungen (ISW) der Universität Stuttgart
	2006 – 2009	Leiter des Fachbereichs Steuerungstechnik am Institut für Steuerungstechnik der Werkzeugmaschinen und Fertigungseinrichtungen (ISW) der Universität Stuttgart
	2006 – 2009	Geschäftsführer der FISW Steuerungstechnik GmbH, Stuttgart
	2009 – 2011	Leiter Steuerungstechnik bei der Fa. StarragHeckert AG, Rorschacherberg (Schweiz)
	Seit 01.04.11	Gruppenleiter in der Forschung und Vorausbildung bei der Fa. Robert Bosch GmbH, Schwieberdingen

ISW/IPA Forschung und Praxis

Berichte aus dem Institut für Steuerungstechnik der Werkzeugmaschinen und Fertigungseinrichtungen der Universität Stuttgart und dem Fraunhofer-Institut für Produktionstechnik und Automatisierung IPA

Herausgegeben bis Band 57 von Prof. Dr.-Ing. G. Stute †
ab Band 58 von Prof. Dr.-Ing. Dr. h. c.mult. Dr.-Ing. E.h G. Pritschow
ab Band 161 von Prof. Dr.-Ing. Dr. h.c. A. Verl

Erschienen bei Springer-Verlag:

- 1 Schmid, D.: Numerische Bahnsteuerung, 1973.
- 2 Schwegler, H.: Fräsbearbeitung gekrümmter Flächen, 1972.
- 3 Eisinger, J.: Numerisch gesteuerte Mehrachsenfräsmaschinen, 1972.
- 4 Nann, R.: Rechnersteuerung von Fertigungseinrichtungen, 1972.
- 5 Augsten, G.: Zweiachsige Nachformeinrichtungen, 1972.
- 6 Karl, B.: Die Automatisierung der Fertigungsvorbereitung durch NC-Programmierung, 1972.
- 7 Eitel, H.: NC-Programmiersystem, 1973.
- 8 Knorr, E.: Numerische Bahnsteuerung zur Erzeugung von Raumkurven auf rotationssymmetrischen Körpern, 1973.
- 9 Bumiller, S.: Viskohydraulischer Vorschubantrieb, 1974.
- 10 Maier, K.: Grenzregelung an Werkzeugmaschinen, 1974.
- 11 Waelkens, J.: NC-Programmierung, 1974.
- 12 Bauer, E.: Rechnerdirektsteuerung von Fertigungseinrichtungen, 1975.
- 13 König, H.: Entwurf und Strukturtheorie von Steuerungen für Fertigungseinrichtungen, 1976.
- 14 Damsohn, H.: Fünfachsiges NC-Fräsen, 1976.
- 15 Jetter, H.: Programmierbare Steuerungen, 1976.
- 16 Henning, H.: Fünfachsiges NC-Fräsen gekrümmter Flächen, 1976.
- 17 Boelke, K.: Analyse und Beurteilung von Lagesteuerungen für numerisch gesteuerte Werkzeugmaschinen, 1977.

- 18 Götz, F.-R.: Regelsystem mit Modellrückkopplung für variable Streckenverstärkung, 1977.
- 19 Tränkle, H.: Auswirkungen der Fehler in den Positionen der Maschinenachsen beim fünfachsigem Fräsen, 1977.
- 20 Stof, P.: Untersuchungen über die Reduzierung dynamischer Bahnabweichungen bei numerisch gesteuerten Werkzeugmaschinen, 1978.
- 21 Wilhelm, R.: Planung und Auslegung des Materialflusses flexibler Fertigungssysteme, 1978.
- 22 Kappen, N.: Entwicklung und Einsatz einer direkten digitalen Grenzregelung für eine Fräsmaschine mit CNC, 1979.
- 23 Klug, H. G.: Integration automatisierter technischer Betriebsbereiche, 1978.
- 24 Binder, D.: Interpolation in numerischen Bahnsteuerungen, 1979.
- 25 Klingler, O.: Steuerung spanender Werkzeugmaschinen mit Hilfe von Grenzregeleinrichtungen, 1979.
- 26 Schenke, L.: Auslegung einer technologisch-geometrischen Grenzregelung für die Fräsbearbeitung, 1979.
- 27 Wörn, H.: Numerische Steuersysteme – Aufbau und Schnittstellen eines Mehrprozessorsteuersystems, 1979.
- 28 Osofisan, P. B.: Verbesserung des Datenflusses beim fünfachsigem NC-Fräsen, 1979.
- 29 Berner, J.: Verknüpfung fertigungstechnischer NC-Programmiersysteme, 1979.
- 30 Böbel, K.-H.: Rechnerunterstützte Auslegung von Vorschubantrieben, 1979.
- 31 Dreher, W.: NC-gerechte Beschreibung von Werkstücken in fertigungstechnisch orientierten Programmiersystemen, 1980.
- 32 Schurr, R.: Rechnerunterstützte Projektsteuerung hydrostatischer Anlagen, 1981.
- 33 Sielaff, W.: Fünfachsiges NC-Umfangfräsen verwundener Regelflächen. Beitrag zur Technologie und Teileprogrammierung, 1981.
- 34 Hesselbach, J.: Digitale Lageregelung an numerisch gesteuerten Fertigungseinrichtungen, 1981.
- 35 Fischer, P.: Rechnerunterstützte Erstellung von Schaltplänen am Beispiel der automatisierten Hydraulikplanzeichnung, 1981.
- 36 Ackermann, U.: Rechnerunterstützte Auswahl elektrischer Antriebe für spanende Werkzeugmaschinen, 1981.

- 37 Döttling, W.: Flexible Fertigungssysteme – Steuerung und Überwachung des Fertigungsablaufs, 1981.
- 38 Firnau, J.: Flexible Fertigungssysteme – Entwicklung und Erprobung eines zentralen Steuersystems, 1982.
- 39 Herrscher, A.: Flexible Fertigungssysteme – Entwurf und Realisierung prozeßnaher Steuerungsfunktionen, 1982.
- 40 Spieth, U.: Numerische Steuersysteme – Hardwareaufbau und Ablaufsteuerung eines Mehrprozessorsteuersystems, 1982.
- 41 Schimmele, A.: Rechnerunterstützter Entwurf von Funktionssteuerungen für Fertigungseinrichtungen, 1982.
- 42 Sanzenbacher, M.: NC-gerechte Beschreibung von Werkstücken mit gekrümmten Flächen, 1982.
- 43 Walter, W.: Interaktive NC-Programmierung von Werkstücken mit gekrümmten Flächen, 1982.
- 44 Huan, J.: Bahnregelung zur Bahnerzeugung an numerisch gesteuerten Werkzeugmaschinen, 1982.
- 45 Erne, H.: Taktile Sensorführung für Handhabungseinrichtungen – Systematik und Auslegung der Steuerungen, 1982.
- 46 Plasch, D.: Numerische Steuersysteme – Standardisierte Softwareschnittstellen in Mehrprozessor-Steuersystemen, 1983.
- 47 Wang, Z. L.: NC-Programmierung – Maschinennaher Einsatz von fertigungstechnisch orientierten Programmiersystemen, 1983.
- 48 Schwager, J.: Diagnose steuerexterner Fehler an Fertigungseinrichtungen, 1983.
- 49 Klemm, P.: Strukturierung von flexiblen Bediensystemen für numerische Steuerungen, 1984.
- 50 Runge, W.: Simulation des dynamischen Verhaltens elektrohydraulischer Schaltungen – Einsatz von geräteorientierten, universellen Simulationsbausteinen, 1984.
- 51 Steinhilber, H.: Planung und Realisierung von Werkzeugversorgungssystemen für die NC-Bearbeitung, 1984.
- 52 Ohnheiser, R.: Integrierte Erstellung numerischer Steuerdaten für flexible Fertigungssysteme, 1984.
- 53 Keppeler, M.: Führungsgrößenerzeugung für numerisch bahngesteuerte Industrieroboter, 1984.
- 54 Kohler, P.: Automatisiertes Messen mit NC-Werkzeugmaschinen, 1985.

- 55 Rieger, K.-H.: Rechnerunterstützte Projektierung der Hardware und Software von Speicherprogrammierten Steuerungen, 1985.
- 56 Vogt, G.: Digitale Regelung von Asynchronmotoren für numerisch gesteuerte Fertigungseinrichtungen, 1985.
- 57 Chmielnicki, S.: Flexible Fertigungssysteme – Simulation der Prozesse als Hilfsmittel zur Planung und zum Test von Steuerprogrammen, 1985.
- 58 Renn, W.: Struktur und Aufbau prozeßnaher Steuergeräte zur Verkettung in flexiblen Fertigungssystemen, 1986.
- 59 Harig, K.: Quantisierung im Lageregelkreis numerisch gesteuerter Fertigungseinrichtungen, 1986.
- 60 Frank, H.: Programmier- und Überwachungsfunktionen für teileartbezogene NC-Werkzeugmaschinen, 1986.
- 61 Möller, H.: Integrierte Überwachungs- und Diagnose-Systeme für numerische Steuerungen, 1986.
- 62 Fink, H.: Einsatz speicherprogrammierbarer Steuerungen in der Fertigungstechnik, 1986.
- 63 Fleckenstein, J.: Zustandsgraphen für SPS – Grafikunterstützte Programmierung und steuerungsunabhängige Darstellung, 1987.
- 64 Wagner, E.: Steuerungen von Koordinatenmeßgeräten mit schaltenden und messenden Tastsystemen, 1987.
- 65 Grimm, W.: Diagnosesystem für steuerungsperiphere Fehler an Fertigungseinrichtungen, 1987.
- 66 Swoboda, W.: Digitale Lageregelung für Maschinen mit schwach gedämpften schwingungsfähigen Bewegungsachsen, 1987.
- 67 Gruhler, G.: Sensorgeführte Programmierung bahngesteuerter Industrieroboter, 1987.
- 68 Walker, B.: Konfigurierbarer Funktionsblock Geometriedatenverarbeitung für numerische Steuerungen, 1987.
- 69 Mayer, J.: Werkzeugorganisation für flexible Fertigungszellen und -systeme, 1988.
- 70 Lederer, R.: Programmierung von NC-Drehmaschinen mit mehreren Werkzeugschlitten, 1988.
- 71 Häberle, G.: NC-Musterprogrammierung für rechnerintegrierte Textilfertigung, 1988.
- 72 Pfeiffer, D.: Kompensation thermisch bedingter Bearbeitungsfehler durch prozeßnahe Qualitätsregelung, 1988.

- 73 Schmidt, W.: Grafikunterstütztes Simulationssystem für komplexe Bearbeitungsvorgänge in numerischen Steuerungen, 1988.
- 74 Egner, M.: Hochdynamische Lageregelung mit elektrohydraulischen Antrieben, 1988.
- 75 Schittenhelm, W.: Konfigurierbares Bedienungssystem für Steuerungen an Fertigungseinrichtungen, 1988.
- 76 Scheifele, D.: Grafisch dynamische Simulation des Bearbeitungsvorgangs für Doppelschlittendrehmaschinen, 1988.
- 77 Keuper, G.: Automatisierte Identifikation der Streckenparameter servohydraulischer Vorschubantriebe, 1989.
- 78 Kayser, K.-H.: Kollisionserkennung in numerischen Steuerungen mit der Distanzfeldmethode, 1989.
- 79 Viefhaus, R.: Fräsergeometriekorrektur in Numerischen Steuerungen für das fünfachsige Fräsen, 1989.
- 80 Zirbs, J.: Fertigungsgerechte Aufbereitung von Flächenverbänden bei der NC-Programmierung im Formenbau, 1989.
- 81 Ruoff, W.: Optische Sensorsysteme zur On-line-Führung von Industrierobotern, 1989.
- 82 Jantzer, M.: Bahnverhalten und Regelung fahrerloser Transportsysteme ohne Spurbindung, 1990.
- 83 Schumacher, H.: Einheitliche Programmierung von Automatisierungskomponenten roboterbestückter Bearbeitungs- und Montagezellen, 1991.
- 84 Schimonyi, J.: NC-Programmierung für das Werkzeugschleifen, 1991.
- 85 Wurst, K.-H.: Flexible Robotersysteme – Konzeption und Realisierung modularer Roboterkomponenten, 1991.
- 86 Hagl, R.: Erhöhung der Verfügbarkeit von Vorschubantrieben mit selbstanpassender Lageregelung, 1991.
- 87 Krebsler, G.: Betriebssystem für NC mit einheitlichen Schnittstellen, 1992.
- 88 Lei, W.-T.: Flächenorientierte Steuerdatenaufbereitung für das fünfachsige Fräsen, 1992.
- 89 Diehl, G.: Steuerungsperipheres Diagnosesystem für Fertigungseinrichtungen auf Basis überwachungsgerechter Komponenten, 1992.
- 90 Nepustil, U.: Offene NC-Schnittstellen zur Korrektur von Fertigungsfehlern, 1992.
- 91 Bauder, M.: Konfigurierbare Robotersteuerung mit allgemeiner Transformation, 1992.

- 92 Philipp, W.: Regelung mechanisch steifer Direktantriebe für Werkzeugmaschinen, 1992.
- 93 Härdtner, G. M.: Wissensstrukturierung in Diagnoseexpertensystemen für Fertigungseinrichtungen, 1992.
- 94 Wiedmann, H.: Objektorientierte Wissensrepräsentation für die modellbasierte Diagnose an Fertigungseinrichtungen, 1993.
- 95 Rudloff, H.: Hochgenaue Konturerzeugung bei Bewegungsachsen mit einer dominanten mechanischen Resonanzstelle, 1993.
- 96 Brantner, K.: Adaptierbares Leitsteuerungssystem für flexible Produktionssysteme, 1993.
- 97 Kugler, W.: Kommunikationsmechanismen für offene Numerische Steuerungssysteme, 1994.
- 98 Schnurr, B.: Elektrodynamisches Antriebssystem zur Unrundbearbeitung, 1994.
- 99 Schneider, J.: Fehlerreaktion mit Speicherprogrammierbaren Steuerungen – ein Beitrag zur Fehlertoleranz, 1994.
- 100 Siewert, U.: Systematische Erstellung adaptierbarer Leitsteuerungssoftware am Beispiel der Durchsetzungsplanung, 1994.
- 101 Heger, G. F. J.: Maschinenferner Qualitätsregelkreis in flexiblen Fertigungssystemen, 1994.
- 102 Hofmeister, W.: Objektorientiert strukturiertes Programmiersystem für NC-Mehrschlittenmaschinen, 1994.
- 103 Horn, A.: Optische Sensorik zur Bahnführung von Industrierobotern mit hohen Bahngeschwindigkeiten, 1994.
- 104 Rentschler, U.: Fehlertolerantes Präzisionsfügen, 1995.
- 105 Junghans, G.: Modulares grafikunterstütztes Simulationssystem für Bearbeitungs- und Handhabungsvorgänge, 1995.
- 106 Heller, J.: Sensorgestützte Bewegungserzeugung leitlinienloser Transporthfahrzeuge, 1995.
- 107 Wieland, E.: Anwendungsorientierte Programmierung für die robotergestützte Montage, 1995.
- 108 Ketterer, G.: Automatisierte Inbetriebnahme elektromechanischer, elastisch gekoppelter Bewegungsachsen, 1995.
- 109 Reibetanz, Th.: Situationsorientierte Bearbeitungsmodellierung zur NC-Programmierung, 1995.

- 110 Frager, O.: Durchgängige Programmierung von Fertigungszellen, 1996.
- 111 Ordenewitz, R.: Betriebsweite Bereitstellung von Werkzeuginformationen, 1996.
- 112 Daniel, C.: Dynamisches Konfigurieren von Steuerungssoftware für offene Systeme, 1996.
- 113 Angerbauer, R.: Anwenderorientierte Programmierung fahrerloser Transportsysteme, 1996.
- 114 Krauß, F.: Splinverarbeitung in numerischen Steuerungen für das fünfachsiges Fräsen, 1996.
- 115 Schittenhelm, K.-M.: Einsatz vorgefilterter Führungsgrößen für Bewegungsachsen zur Bahnerzeugung, 1997.
- 116 Häberle, U.: Einheitliche Anwenderschnittstelle für Feldbussysteme, 1997.
- 117 Strassacker, D.: Testumgebung für die Implementierung und Inbetriebnahme eines adaptierbaren Leitsteuerungssystems, 1997.
- 118 Renz, B.: Hochdynamische Strahlagekorrektursysteme zur Erhöhung der Bahnengenauigkeit von CO₂-Laserbearbeitungsmaschinen, 1997.
- 119 Itterheim, C.: Objektorientiertes Bearbeitungsmodell für Freiformflächen – Erstellung und maschinengebundene Modifikation –, 1997.
- 120 Müller, J.: Objektorientierte Softwareentwicklung für offene numerische Steuerungen, 1997.
- 121 Glöckler, M.: Verbesserung des Störverhaltens elektrohydraulischer lage geregelter Zylinderantriebe, 1998.
- 122 Uhl, J.: Entwurfssystematik für ein dezentral strukturiertes, objektorientiertes Fertigungsleitsystem, 1998.
- 123 Hammann, G.: Modellierung des Abtragsverhaltens elastischer, robotergeführter Schleifwerkzeuge, 1998.
- 124 Scholich-Tessmann, W.: Direktantriebe für Industrieroboter, 1998.
- 125 Wagner, R.: Robotersysteme zum kraftgeführten Entgraten grobtolerierter Leichtmetallwerkstücke mit Fräswerkzeugen, 1998.
- 126 Anders, C.: Adaptierbares Diagnosesystem bei Transferstraßen, 1998.
- 127 Fahrbach, C.: Regelung hochdynamischer elektrischer Servo-Direktantriebe in Fertigungseinrichtungen, 1999.
- 128 Sperling, W.: Modulare Systemplattformen für offene Steuerungssysteme, 1999.

Erschienen bei Jost-Jetter Verlag:

- 129 Kehl, G.: Gestaltung von Formgedächtnis-Aktorsystemen für sensorgeführte Inspektionsgeräte, 1999.
- 130 Brandl, Th.: Anlageninformationssystem - Informationsmodell und Erstellungssystematik, 1999.
- 131 Gronbach, H.: Simulationswerkzeug für die Gestaltung modularer CO₂-Laserbearbeitungsmaschinen, 1999.
- 132 Lutz, R.: Softwaretechnik für Maschinennahe Steuerungsfunktionen bei Fertigungseinrichtungen, 1999.
- 133 Tran, T. L.: Allgemeine Transformation für Maschinen mit Parallelkinematiken, 2000.
- 134 Bretschneider, J.: Reglerselbsteinstellung für digital geregelte, elektromechanische Antriebssysteme an Werkzeugmaschinen, 2000.
- 135 Schoenberg, M.: Zuverlässiger Fertigungsprozess bei Transferstraßen durch präventive Maßnahmen, 2000.
- 136 Uhl, A.: Flexibles Telerobotersteuerungssystem auf der Basis offener numerischer Steuerungen, 2000.
- 137 Rui Li: Agentenbasierte NC-Planung für die Komplettbearbeitung auf Dreh-/Fräszentren, 2001.
- 138 Wildermuth, D.: Bahnvorbereitung in numerischen Steuerungen für Parallelkinematiken, 2001.
- 139 Handel, D.: Werkergerechte NC-Programmierung zur Komplett-Schleifbearbeitung von Bohrwerkzeugen, 2001.
- 140 Kosiedowski, U.: Adaptive Vorsteuerverfahren für elektromechanische Bewegungsachsen an Werkzeugmaschinen, 2001.
- 141 Haug, K.: Laser-Lichtschnittsensorik für die Automatisierung von Metall-Schutzgasschweißprozessen, 2002.
- 142 Hohenadel, J.: Einheitliches Steuerungssystem für NC und RC, 2002.
- 143 Wälde, K.: Sicherstellung der Softwarequalität von Anwendungsmodulen und Systemplattformen in offenen Steuerungssystemen, 2002.
- 144 Litto, M.: Störungsinformationssystem – Informationsmodell und Erstellungssystematik, 2002.
- 145 Schweiker, A.: Offene numerische Steuerungen für prozeßabhängige Bearbeitungen – vereinheitlichte Struktur, Funktionen und Schnittstellen – , 2003.
- 146 Rempp, B.: Regelungstechnische Untersuchung durchsatzgeregelter Produktionssysteme, 2003.

- 147 Eppler, C.: Kompensation fremderregter Schwingungen in Antriebssystemen mit Umlaufgetrieben, 2003.
- 148 Weiner, M.: Wiederverwendungsgerechte Entwicklungssystematik von Feinplanungssoftware für die flexible Fertigung, 2004.
- 149 McCormac, St.: Lageregelung hydraulischer Manipulatoren unter Einsatz eines Ferraris-Sensors, 2004.
- 150 Lehner, W.-D.: Regelung von Vorschubachsen unter Verwendung der Relativbeschleunigung, 2005.
- 151 Lewek, J.: Adaptierbares Informationssystem zur Erstellung baukastenbasierter Fertigungseinrichtungen, 2005.
- 152 Laible, U.: Aufbau numerischer Steuerungssysteme für sicherheitskritische Anwendungen, 2005.
- 153 Bürger, Th.: Durchgängige analytische Qualitätssicherung für numerische Steuerungssoftware, 2005.
- 154 Brinzer, B.: Produktionsregelung für die variantenreiche Serienfertigung, 2005.
- 155 Heusinger, S.: STEP-NC-basierter Korrekturkreis für die Schlichtbearbeitung von Freiformflächen, 2005.
- 156 Kirchberger, R.: Verbesserte Auswertung inkrementeller Messsysteme durch schnelle Signal-Vorverarbeitung, 2005.
- 157 Conrath, M.: Systematische Gestaltung von frequenzadaptierbaren Ultraschall-Werkzeugsystemen zum Einsatz in fertigungstechnischen Prozessen, 2005.
- 158 Wadehn, W.: Gestaltung von Antriebssystemen für formadaptive Strukturen, 2005.
- 159 Horber, H.: Fugendetektion bei Lichtbogenschweißprozessen mit robuster Signalverarbeitung für optische Sensoren, 2006.
- 160 Dreyer, J.: Situative Informationsbereitstellung an Fertigungseinrichtungen Informationsmodell und Erstellungssystematik, 2006.
- 161 Fritz, S.: Rekonstruktion von Prozesskräften aus Antriebssignalen von Werkzeugmaschinen, 2006.
- 162 Staudt, S.: Spezifikation und Konformitätstest zur Interoperabilität von automatisierten Produktionsmaschinen, 2006.
- 163 Reichle, R.: Einsatz von Internet-Werkzeugen und -Diensten in numerischen Steuerungssystemen für Werkzeugmaschinen, 2006.
- 164 Kaiser, L.: Systematik für das Qualitäts- und Projektmanagement bei der Abwicklung von Unikatprojekten, 2006.

- 165 Garber, Th.: Nutzung des redundanten Freiheitsgrades von sechsachsigen Parallelkinematik-Maschinen, 2007.
- 166 Altenburger, R.: Dynamische Eigenschaften und Regelung mechanisch verkoppelter Antriebssysteme, 2007.
- 167 Korajda, B.: Steuerungstechnische Verfahren zur echtzeitfähigen Kompensation der Fräserabdrängung, 2007.
- 168 Röck, S.: Echtzeitsimulation von Produktionsanlagen mit realen Steuerungssystemen, 2007.
- 169 Prieße, J.: Verfahren zur durchgehenden dezentralen Planung in Werkstattstrukturen, 2007.
- 170 Boye, T.: Vorhersage der kinematischen Kalibriergüte von Parallelkinematiken, 2008.
- 171 Joannides, M.: Ein Beitrag zur volumenorientierten 3D-Objektrekonstruktion aus digitalen Daten, 2009.
- 172 Pruschek, P.: Verfahren zur anwendungsgerechten Parametrierung der Steuerung und Regelung von Vorschubachsen, 2009.
- 173 Fritsch, D.: Steuerung selbstorganisierender Multi-Roboter-Systeme für dynamische Sammelaufgaben am Beispiel der Bekämpfung maritimer Ölverschmutzungen, 2009.
- 174 Schmitz, S.: Industrielle Powerline-Kommunikation für Antriebseinheiten in Werkzeugmaschinen, 2010.
- 175 Bengel, M.: Workpiece-centered Approach to Reconfiguration in Manufacturing Engineering, 2010
- 176 Weimer, T.: Informationsmodell für die durchgängige Datennutzung in Fabrikplanung und -betrieb, 2010
- 177 Oglochin, V.: Maschinenübergreifender agentenbasierter Informationsaustausch für die Störungsbeseitigung, 2010
- 178 Kramer, C.: Offene Antriebsreglerplattform, 2011
- 179 Stotz, M.: Adaptive Segmentierung von Tiefenbildern für die 3-D-Objektlageerkennung auf Basis von kombinierten regelgeometrischen Elementen, 2011.
- 180 Selig, A.: Informationsmodell zur funktionalen Typisierung von Automatisierungsgeräten, 2011
- 181 Meyer, C.: Aufnahme und Nachbearbeitung von Bahnen bei der Programmierung durch Vormachen von Industrierobotern, 2011
- 182 Meier, M.: Verfahren zum emulationsgestützten MES-Engineering für die Photovoltaikindustrie, 2011

- 183 Walther, M.: Antriebsbasierte Zustandsdiagnose von Vorschubantrieben, 2011
- 184 Lechler, A.: Konzeption einer funktional einheitlichen Applikationsschnittstelle für Ethernet-basierte Bussysteme, 2011.
- 185 Kircher, C.: Selbstadaptierende NC-Steuerung für rekonfigurierbare Werkzeugmaschinen, 2011.

