

Brigitte Bärnreuther

*Ein Beitrag zur Bewertung des Kommunikationsverhaltens  
von Automatisierungsgeräten in flexiblen Produktionszellen*





Brigitte Bärnreuther

*Ein Beitrag zur Bewertung des  
Kommunikationsverhaltens von  
Automatisierungsgeräten in  
flexiblen Produktionszellen*

Herausgegeben von

Professor Dr.-Ing. Klaus Feldmann,

Lehrstuhl für

Fertigungsautomatisierung und Produktionssystematik

**FAPS**



Carl Hanser Verlag München Wien

Als Dissertation genehmigt von der Technischen Fakultät  
der Friedrich-Alexander-Universität Erlangen-Nürnberg

Tag der Einreichung:	04. 06. 1992
Tag der Promotion:	21. 09. 1992
Dekan:	Prof. Dr. rer. nat. H. Mughrabi
Berichterstatter:	Prof. Dr.-Ing. K. Feldmann Prof. Dr.-Ing. U. Herzog

Die Deutsche Bibliothek - CIP-Einheitsaufnahme

**Bärnreuther, Brigitte:**

Ein Beitrag zur Bewertung des Kommunikationsverhaltens von  
Automatisierungsgeräten in flexiblen Produktionszellen /

Brigitte Bärnreuther. - München; Wien; Hanser 1992

(Fertigungstechnik - Erlangen; 32)

Zugl.: Erlangen, Nürnberg, Univ., Diss., 1992

ISBN 3-446-17451-6

NE: GT

Dieses Werk ist urheberrechtlich geschützt.

Alle Rechte, auch die der Übersetzung, des Nachdrucks  
und der Vervielfältigung des Buches oder Teilen daraus,  
vorbehalten.

Kein Teil des Werkes darf ohne schriftliche Genehmigung des  
Verlages in irgendeiner Form (Fotokopie, Mikrofilm oder ein  
anderes Verfahren), auch nicht für Zwecke der Unterrichts-  
gestaltung - mit Ausnahme der in den §§ 53, 54 URG ausdrücklich  
genannten Sonderfälle - , reproduziert oder unter Verwendung  
elektronischer Systeme verarbeitet, vervielfältigt oder  
verbreitet werden.

© Carl Hanser Verlag München, Wien 1992

Herstellung: Copy Center 2000, Erlangen-Eltersdorf

Printed in Germany

## **Vorwort**

Die vorliegende Dissertation entstand während meiner Tätigkeit als wissenschaftliche Mitarbeiterin am Lehrstuhl für Fertigungsautomatisierung und Produktionssystematik der Friedrich-Alexander-Universität Erlangen-Nürnberg.

Herrn Professor Dr.-Ing. K. Feldmann, dem Leiter dieses Lehrstuhls am Institut für Fertigungstechnik, danke ich für die wohlwollende Förderung bei der Durchführung dieser Arbeit.

Herrn Professor Dr.-Ing. U. Herzog danke ich für die Übernahme des Korreferates, für die fachlichen Anregungen und Verbesserungsvorschläge.

Bedanken möchte ich mich bei allen Kollegen des Lehrstuhls, vor allem denen aus der Steuerungs- und Sensorikgruppe. Ferner gilt mein Dank den Kollegen der Informatik Forschungsgruppe E für die praktischen Hilfeleistungen.

Einen besonders herzlichen Dank möchte ich Herrn Dr. Heinrich Dietsch aussprechen. Die fruchtbare Kooperation im Sonderforschungsbereich "Multiprozessor- und Netzwerkkonfigurationen" (SFB 182) hat ganz wesentlich zum Gelingen der Arbeit beigetragen.

Das letzte Dankeschön gilt meinem Mann Werner vor allem für die tatkräftige Unterstützung bei der Betreuung unseres Sohnes.

Brigitte Bärnreuther



# **Ein Beitrag zur Bewertung des Kommunikationsverhaltens von Automatisierungsgeräten in flexiblen Produktionszellen**

## **Inhaltsverzeichnis**

<b>1 Einleitung und Zielsetzung</b> .....	<b>1</b>
<b>2 Zugrundeliegende Begriffe</b> .....	<b>6</b>
2.1 Verwendete Begriffe aus der Fertigungsautomatisierung und Produktions- technik .....	7
2.2 Verwendete Begriffe aus der Technischen Informatik und Kommunikations- technik .....	12
<b>3 Darstellung und Wertung derzeitiger Kommunikationssysteme in der     rechnerintegrierten Produktion</b> .....	<b>26</b>
3.1 Kommunikationssysteme in der Produktions- und Fertigungsleitung ...	27
3.2 Feldbusse .....	31
3.2.1 Allgemeine Eigenschaften .....	31
3.2.2 Der PROFIBUS als exemplarischer Vertreter .....	33
3.3 Kommunikationsschnittstellen für Sensoren und Aktoren .....	39
3.4 Anwendungsorientierung .....	41
3.4.1 Manufacturing Message Specification .....	41
3.4.2 MMS-Companion-Standards .....	45
3.4.3 Semiconductor-Equipment-Communication-Standard .....	51
3.5 Echtzeitunterstützung .....	53
3.5.1 MAP-Enhanced-Performance-Architecture und Mini-MAP .....	53
3.5.2 Factory-Automation-Interconnection-System .....	55
3.5.3 Time-Critical-Communication-Architecture .....	57
3.5.4 Real-Time-Manufacturing-Message-Specification .....	61

#### **4 Kommunikationskonzepte für die Anwendung in flexiblen Produktions-**

<b>zellen</b>	<b>65</b>
4.1 Zielsetzung	65
4.2 Fragen bzgl. der Kommunikationsfähigkeit von Automatisierungsgeräten	66
4.2.1 Offene Kommunikation	68
4.2.2 Stationstypen	72
4.2.3 Datentransfermodi	75
4.2.4 Dynamische Topologieänderung	76
4.2.5 Teilnetzbildung	77
4.2.6 Echtzeit	79
4.2.7 Leistungsbewertung	83
4.2.8 Alarmbehandlung	83
4.2.9 Fehlertoleranz	87
4.2.10 Netzdiagnose	88
4.2.11 Projektierungshilfen	91
4.3 Exemplarische Anwendung des Klassifikationsschemas	93
4.3.1 Allgemeine Vorgehensweise	93
4.3.2 Anwendung auf den PROFIBUS	94
4.3.3 Anwendung auf den DIN-PDV-Bus	102
4.3.4 Anwendung auf den DIN-Meßbus	102
4.3.5 Zusammenstellung der Ergebnisse	105

#### **5 MAP-konforme Kommunikationsarchitekturen für das**

<b>Gerätespektrum flexibler Produktionszellen</b>	<b>107</b>
5.1 Gerätespektrum in flexiblen Produktionszellen	107
5.2 LABORINTH als konzeptueller Rahmen für prozeßnahe Kommunikationssysteme	109
5.2.1 Die Hard- und Softwarebausteine des LABORINTH-Konzepts und ihre Kommunikationsbeziehungen	109
5.2.2 Die Schichtenstruktur des LABORINTH-Kommunikationssystems	112
5.3 Repräsentative Kommunikationsarchitekturen	114
5.3.1 Prinzipien des Architekturkonzepts	114
5.3.2 Überblick über die Architekturvarianten	115
5.4 Beispielhafte Implementierung ausgewählter Architekturvarianten	117

5.5 Leistungsbetrachtung	120
5.5.1 Leistungsdaten der LABORINTH-Funktionen	122
5.5.2 Leistungsdaten ausgewählter MMS-Funktionen	123
5.5.3 Ausblicke auf Optimierungsmöglichkeiten	125
5.6 Zusammenfassung	126
<b>6 Companion-Standard-Entwurf für Sensoren und Aktoren</b>	<b>128</b>
6.1 Exemplarisches Einfachgerät	128
6.2 Ausführung	130
6.2.1 Einleitende Bemerkungen zum Standpunkt und zur Einordnung	132
6.2.2 Referenzen	132
6.2.3 Definitionen	132
6.2.4 Symbole und Abkürzungen	134
6.2.5 Beschreibung der Anwendung	134
6.2.5.1 Anwendungsspezifisches Modell	134
6.2.5.2 Anwendungsspezifische Funktionen	137
6.2.6 Abbildung der Anwendungsspezifika	139
6.2.6.1 Definition des anwendungsspezifischen VMD-Objekts	140
6.2.6.2 Anwendungsspezifische Abbildung auf Variable-Objekte	141
6.2.6.2 Anwendungsspezifische Abbildung auf Event-Management-Objekte	142
6.2.7 Verwendung der MMS-Dienste	148
6.2.7.1 Definition und Verwendung von anwendungsspezifischen Diensten	148
6.2.7.2 Initiate-Dienst und -Protokoll	150
6.2.8 Standard-Objekte	151
6.2.9 Konformität	153
6.2.9.1 Basis-Konformitätsklassen	153
6.2.9.2 Höhere Konformitätsklassen	155
6.2.10 Anwendungsbeispiel	157
<b>7 Zusammenfassung und Ausblick</b>	<b>159</b>
<b>8 Literaturverzeichnis</b>	<b>162</b>





# 1 Einleitung und Zielsetzung

Der Einsatz rechnerintegrierter Produktionssysteme ist für die Wettbewerbsfähigkeit der Industrie von wachsender Bedeutung. Charakteristisch für moderne Produktionsbetriebe ist, daß die Produktionsbereiche rechnergesteuert und datentechnisch miteinander verbunden sind. Das Ziel "*Computer Integrated Manufacturing*" (CIM) soll dadurch erreicht werden, daß man diese rechnergestützten Abläufe auf den verschiedenen Ebenen zu durchgängigen "Verfahrensketten" verknüpft. /1/

Die Thematik der vorliegenden Arbeit ist der *technischen* Produktionsteuerung zugeordnet und beschäftigt sich mit der Kommunikation in rechnerintegrierten Produktionssystemen. Betrachtet wird das Kommunikationsverhalten bei den verschiedenen Komponenten der Produktionsausführung. Dies betrifft einerseits die Steuerung und Überwachung von Werkzeugmaschinen, Handhabungsgeräten, Transportsystemen usw. und andererseits die Erfassung des Prozeß-Istzustandes in Form von Betriebs- und Maschinendaten. Hinsichtlich des Informationsaustausches zwischen den Automatisierungskomponenten stehen Aspekte der technischen Realisierung im Vordergrund: fehlertolerantes Verhalten des Kommunikationssystems, Netzdiagnose, Aufwand für die Anschaltung, Leistungsbetrachtungen usw.

Die *dipositiven* Aufgaben der Auftragsabwicklung mit betriebswirtschaftlichen Überlegungen zur Kapazitäts-, Terminplanung, Auslastungsoptimierung usw. werden nicht betrachtet. Außer acht gelassen wird ferner die Problematik des Informationsaustausches zwischen verschiedenen anwendungsabhängigen Softwarepaketen zur Produktionssteuerung, z.B. zwischen CAD-Systemen, NC-Programmierung, Produktdatenbanken usw.

Auf dem Gebiet der Montage und speziell der Elektronikproduktion bemühen sich verschiedene Hersteller und Institute um eine einheitliche Lösung für die informationstechnische Verknüpfung der verschiedenen Anwendungssysteme /2, 3/, z.B. für das Leiterplatten-Layout, die Programmierung der Bestückungsautomaten, Lötanlagen usw. Daneben arbeiten Normungsgremien derzeit international an der Definition eines einheitlichen *Produktmodells*, das nicht nur geometrische und technologische Eigenschaften, sondern alle im Produktlebenszyklus entstehenden Informationen enthält /4/.

Ausgangspunkt der vorliegenden Untersuchungen ist die physische CIM-Hierarchie mit der Anordnung der Automatisierungskomponenten, wie sie sich in Bild 1.1 darstellt. Die umfangreichen Planungs-, Konstruktions- und Optimierungsaufgaben auf Produktions- und Fertigungsleitebene werden von leistungsfähigen Rechnern mit Unterstützung von innovativen Werkzeugen aus der Informatik, wie z.B. Expertensystemen, Non-Standard-Datenbanken, Neuronalen Netzen, Fuzzy-Logik, bearbeitet. In Richtung zum Prozeß überwiegen Funktionen der *technischen* Steuerung. Bei den dort verwendeten Komponenten werden elementare Eigenschaften wie *Ausfallsicherheit*, *Störungsunempfindlichkeit*, *Wartungsfreundlichkeit* usw. bei gleichzeitig *kostengünstiger Realisierung* angestrebt. Das Themenumfeld "Fertigungsautomatisierung" impliziert, daß als Prozeßklasse Stückgutprozesse in der Teilefertigung und Montage und keine Fließ- oder Folgeprozesse /5/ betrachtet werden. Besondere Maßnahmen zum Kommunikationsbetrieb in explosionsgefährdeten Umgebungen werden i.allg. nicht vorgesehen. Der Unterschied zu den höheren Ebenen der CIM-Hierarchie liegt ferner im geforderten Echtzeitverhalten bei insgesamt sehr kurzen Reaktionszeiten.

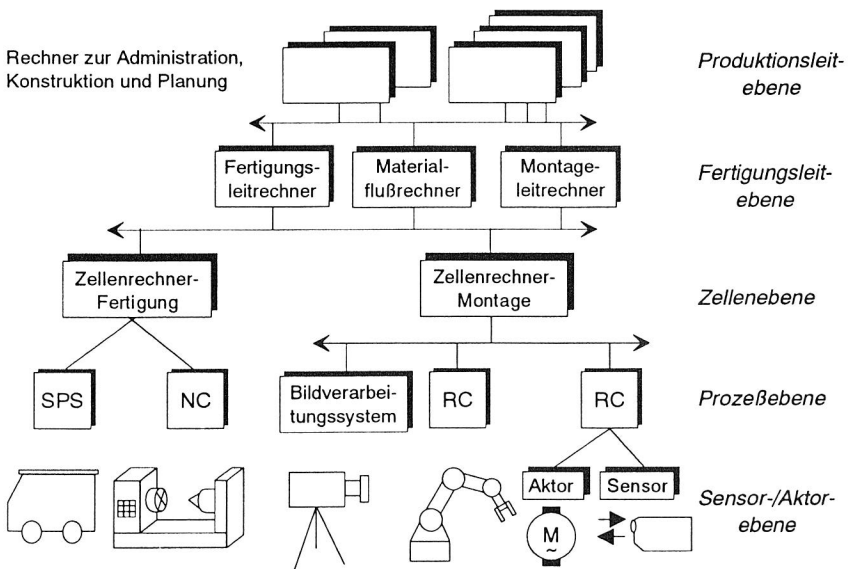


Bild 1.1: Komponenten in der physischen CIM-Hierarchie

Somit sind auf den beiden untersten Hierarchieebenen - innerhalb flexibler Produktionszellen - vorwiegend Geräte mit dedizierter Hardware angesiedelt: speicherprogrammierbare Steuerungen, numerische Steuerungen und Robotersteuerungen für die Maschinen und Handhabungsgeräte auf der Prozeßebene und einfache, preiswerte Sensoren und Aktoren für Steuerungs-, Regelungs- und Überwachungsaufgaben auf der Sensor-/Aktorebene.

Die kommunikationstechnische Verknüpfung der Komponenten auf den übergeordneten Ebenen geschieht auf der Basis Lokaler Netze in verschiedenen Ausprägungen. Der derzeit bekannteste Vorschlag für ein Kommunikationssystem in der automatisierten Fertigung wurde anfangs der 80er Jahre von der Firma General Motors vorgelegt: "Manufacturing Automation Protocol" (MAP) ist eine Protokollarchitektur gemäß den Richtlinien des ISO-OSI-Basisreferenzmodells /6/. Sie ist richtungsweisend für Kommunikationssysteme in der Produktion, zumal die darin enthaltenen Protokolle den Stand internationaler Normen erlangt haben. Der wichtigste MAP-Bestandteil ist das Anwendungsprotokoll "Manufacturing Message Specification" (MMS) mit seinen Erweiterungen ("Companion Standards"), mit dem verteilte Anwendungen in der Fertigungsumgebung objektbasiert modelliert werden können. Trotz der aktuellen Feldbus-Bestrebungen (PROFIBUS, FIP u.a) hat sich auf den unteren Ebenen der CIM-Hierarchie kein vergleichbarer internationaler Standard durchgesetzt. Vielmehr werden zahlreiche, einfache Kommunikationslösungen praktiziert: Die Übermittlung von Prozeßdaten erfolgt über serielle Punkt-zu-Punkt-Verbindungen und spezifische Protokolle sowie durch den Austausch von binären oder analogen Signalen.

Die Problematik der Kommunikation auf den unteren CIM-Ebenen liegt zum einen in der Heterogenität der Geräte, die innerhalb flexibler Produktionszellen vertreten sind und sich hinsichtlich Verarbeitungsleistung und Kommunikationsfähigkeit stark unterscheiden. Zum anderen existiert eine Vielfalt von Anwendungen, bei denen Aufgaben der spanenden Fertigung, der mechanischen Montage, der Elektronikproduktion und Halbleiterfertigung sowie Meß- und Prüfvorgänge bearbeitet werden müssen. Deshalb sind in Produktionszellen Speziallösungen entstanden, die in Abhängigkeit von der jeweiligen Anwendung und unter Berücksichtigung der Kommunikationsausstattung der

Geräte an die spezifischen Kommunikationsbedürfnisse angepaßt und optimiert sind. Bild 1.2 zeigt eine typische Kommunikationsstruktur von Bearbeitungszellen für die spanende Fertigung. Weder ist eine einheitliche Kommunikationsarchitektur mit flexibler Austauschbarkeit der Komponenten noch eine kompatible Anbindung an die übergeordnete Kommunikations-Welt gegeben. Die Verwendung von MAP innerhalb flexibler Produktionszellen scheitert daran, daß der Protokollaufwand durch die volle Ausprägung aller 7 ISO/OSI-Schichten ("Full MAP") erheblich ist und das damit verbundene Leistungsverhalten auf der Anwendungsebene nicht ausreicht.

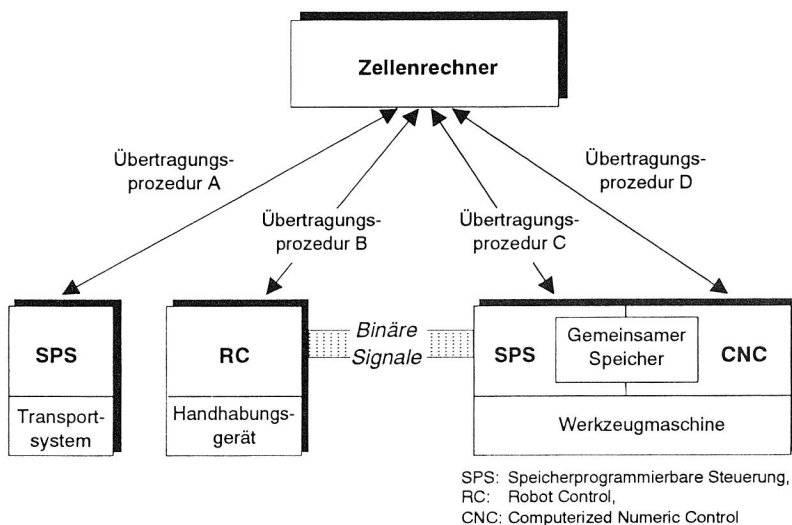


Bild 1.2: Kommunikationsstruktur von Bearbeitungszellen in der Teilefertigung

Die vorliegende Arbeit behandelt deshalb im Kern die Kommunikation innerhalb flexibler Produktionszellen unter Berücksichtigung des durchgängigen Informationsaustausches im gesamten Produktionsverbund. Um das Kommunikationsverhalten in Produktionszellen generell systematisieren zu können, wurde ein Klassifikationsschema erarbeitet, mit dessen Hilfe zelleninterne Kommunikationssysteme bzw. kommunizierende Automatisierungskomponenten auf relevante Kriterien hin überprüft werden können. "Offene Kommunikation" mittels standardisierter Protokolle und die

"Integration von Einfachstationen" stehen besonders im Vordergrund.

Neben der Kommunikationsfähigkeit wird die Gesamtleistungsfähigkeit eines Automatisierungsgerätes durch die eigentliche Anwendungsfunktionalität einhergehend mit der entsprechenden Rechenleistung bestimmt. Weiteres Ziel war es deshalb, repräsentative Kommunikationsarchitekturen zu konzipieren, die hinsichtlich dieser beiden Maßstäbe (Kommunikationsfähigkeit, Rechenleistung) stark variieren: Leistungsfähige Zellenrechner und einfache Sensor-/Aktorstationen sollten MAP-konform kommunizieren. Dabei wurde einerseits das Prinzip verfolgt, aufwendige Anwendungsfunktionen aus der Einfachstation auszulagern. Andererseits lassen die aktuellen Entwicklungen auf dem Gebiet der "smart transmitter" /7/ "intelligente" Sensoren (und Aktoren) erwarten, die mit Unterstützung der Mikroelektronik nicht nur umfangreiche Algorithmen zur Signalverarbeitung sondern auch für die Kommunikation abarbeiten können. Von Seiten der Kommunikationstechnik kommt man dem Ziel der "integrierten", MAP-kommunizierenden Einfachstation dadurch näher, daß man die Mächtigkeit der anwendungsorientierten Protokolle reduziert. Zu diesem Zweck wird ein MMS-Companion-Standard vorgeschlagen, der die MMS-Modelle auf ein für die Sensor-/Aktorumgebung notwendiges Mindestmaß begrenzt.

Insgesamt soll ein Beitrag dazu geleistet werden, die kommunikationstechnischen Anforderungen von flexiblen Produktionszellen zu erarbeiten, um den durchgängigen Informationsverbund im Produktionsbetrieb bis in die unmittelbare Nähe zum technischen Prozeß zu ermöglichen.

## 2 Zugrundeliegende Begriffe

Die vorliegende Arbeit bewegt sich thematisch im Grenzgebiet zwischen Fertigungsautomatisierung/Produktionstechnik und Technischer Informatik/Kommunikationstechnik. Das Zusammenwachsen beider Bereiche zeigt sich u.a. darin, daß die physischen Komponenten im hierarchischen CIM-Modell heutzutage durch verschiedene Verbindungs- und Übertragungssysteme verknüpft werden (s. Bild 2.1): Glasfaser-Systeme, MAP-Token-Busse und ETHERNET-LANs werden auf den drei höheren Ebenen eingesetzt. Unterhalb der Fertigungsleitebene herrschen unterschiedliche Topologien vor: Konventionell sind die beteiligten Stationen sternförmig mit einer zentralen Komponente in der nächst höheren Hierarchiestufe verbunden. Neuere Bestrebungen wollen die Geräte an eine gemeinsame Übertragungsleitung ankoppeln ("Feldbus").

Bevor im Abschnitt 3 der Stand der Kommunikationstechnik in der rechnerintegrierten Produktion dargestellt und diskutiert wird, sollen in diesem Abschnitt zunächst grund-

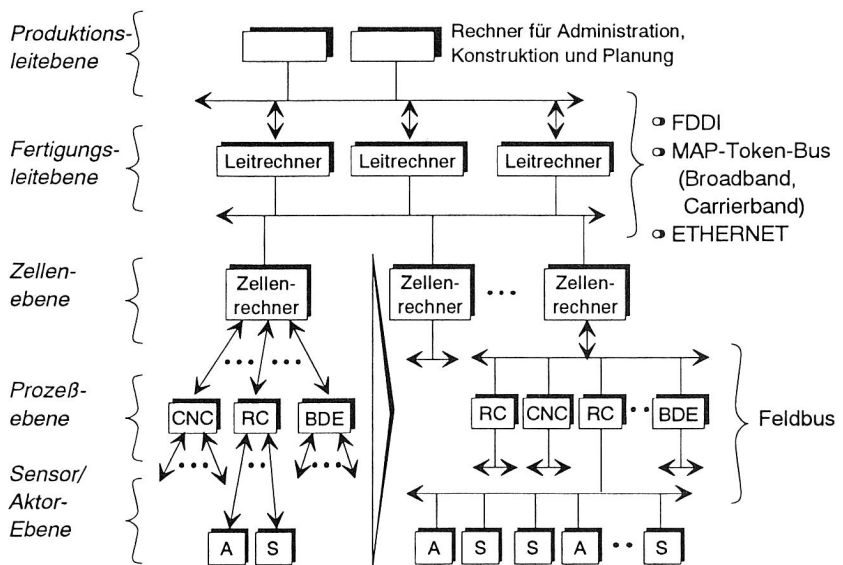


Bild 2.1: Physische Verbindungsstrukturen in rechnerintegrierten Produktionssystemen

liegende Begriffe aus den beiden Schwerpunktbereichen erläutert werden.

## **2.1 Verwendete Begriffe aus der Fertigungsautomatisierung und Produktionstechnik**

Das Thema der "rechnerintegrierten Produktion" stellt den großen Rahmen für die vorliegende Arbeit dar. Nach diesbezüglichen, einleitenden Bemerkungen wird das Hauptaugenmerk auf die physischen Komponenten zur Prozeßsteuerung, nämlich die unterschiedlichen Automatisierungsgeräte, die in flexiblen Produktionszellen zusammengefaßt sind, gelegt.

### Rechnerintegrierte Produktion

Der Themenkomplex der *rechnerintegrierten Produktion* hat unter dem Schlagwort *CIM* (*Computer Integrated Manufacturing*) weite Verbreitung gefunden:

"CIM beschreibt den integrierten Einsatz der Datenverarbeitung in allen mit der Produktion zusammenhängenden internen und externen Unternehmensfunktionen. Hierbei soll die Integration der technischen und organisatorischen Funktionen zur Produktplanung und -erstellung erreicht werden. Dies bedingt die gemeinsame, bereichsübergreifende Nutzung von Informationen über geeignete Schnittstellen, Datenbasen und die Kommunikation über Netze" /8/.

Der durchgängige Informationsaustausch für alle Produktionsbereiche (Konstruktion, Arbeitsvorbereitung, Produktionsplanung und -steuerung, Prozeßsteuerung usw.) ist zwar bisher noch nicht gegeben, u.a. wird aber im Rahmen von europäischen Normungsprojekten intensiv daran gearbeitet: "CAD Interfaces" (CAD \*I, Esprit-Projekt 322), "CIM Open Systems Architecture" (CIM-OSA, Esprit-Projekt 2422), "Communication Networks for Manufacturing Applications" (CNMA, Esprit-Projekt 2617) u.v.a.

Die Übersichtlichkeit von CIM-Systemen konnte durch Hierarchiebildung (s. Bild 2.1) und Modularisierung ("Produktionszellen") erreicht werden.

### Flexible Produktionszellen

*Flexible Produktionszellen* ("allgemeine" *flexible Fertigungszellen*) sind entweder *flexible Fertigungszellen* (z.B. für die spanende Bearbeitung von Werkstücken mit dem Hauptaktionsträger "CNC-Werkzeugmaschine") oder *flexible Montagezellen* (z.B. um eine Baugruppe aus mehreren Komponenten zusammenzusetzen, mit dem Hauptaktionsträger "Roboter", s. Bild 2.2). "Flexible Produktionszelle" und "flexible Fertigungszelle" werden aus kommunikationstechnischer Sicht nicht differenziert. In der Literatur werden die Begriffe meist synonym verwendet.

Die Begriffsbestimmung von *flexiblen Produktionszellen* wird zumeist aus dem Zusam-

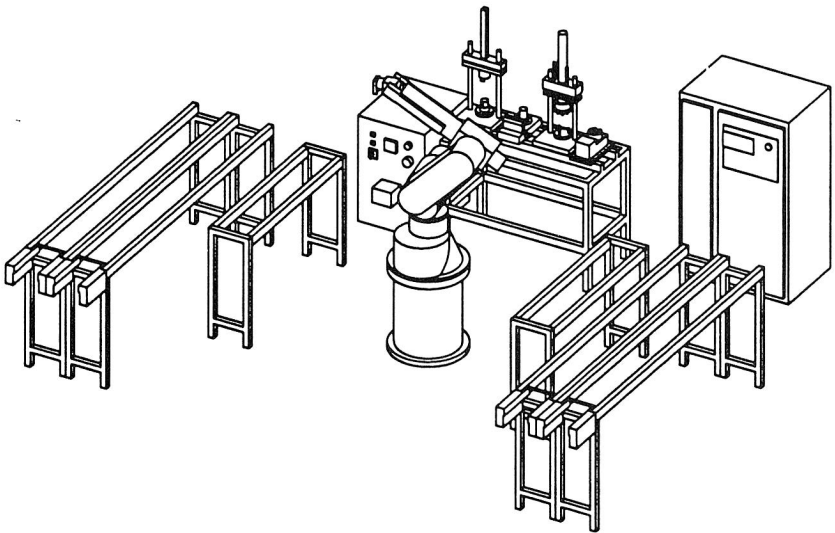


Bild 2.2: Beispiel für eine Montagezelle aus der Modellfabrik

menhang mit gesamten CIM-Systemen hergeleitet: *Flexible Produktionssysteme* folgen dem Prinzip der modularen Strukturierung und verteilen die Gesamtaufgabe auf mehrere autonome Einheiten:

"Cellular manufacturing is the physical division of the manufacturing facilities machinery into production cells" /9/.



Als Kriterium für die *Flexibilität* wird in /10/ angegeben:

"Der Begriff der "Fertigungszelle" läßt sich in Anlehnung an den Begriffe des Fertigungssystems als "Gesamtheit aller Komponenten definieren, die für die Bearbeitung einer Menge bestimmter Werkstücke auf einer Werkzeugmaschine erforderlich sind [...]. Unter dem Begriff "flexible Fertigungszelle" ist die Fertigungseinrichtung zu verstehen, die automatisch unterschiedliche Werkstücke bearbeiten kann".

Verschärft wird die Flexibilitätorderung heute insofern, als die Fertigung unterschiedlicher Produkte möglichst **ohne Umrüstung im Stillstand** erfolgen soll.

Der Aufbau einer flexiblen Produktionszelle wird wie folgt beschrieben:

"Bei einer *allgemeinen flexiblen Fertigungszelle* handelt es sich um eine rechnergeführte Arbeitsstation in verteilter oder konzentrierter Struktur und ihrer zugeordneten Peripherie mit Überwachungseinrichtungen und Einrichtungen zur Bereitstellung und Zuführung von Werkstücken und Fertigungsmitteln und zur Verkettung mit anderen Zellen". Für die Ausprägung als Montagezelle gilt, daß "jede Einrichtung durch eine eigene Steuerungskomponente repräsentiert wird. Die Komplexität der eingesetzten Steuerungen reicht von einfachen Funktionssteuerungen, [...], bis hin zu aufwendigen Steuerungen bei Industrierobotern, [...]" /11/.

Aus Sicht des Informatikers und Kommunikationstechnikers, der die Spezifika des technologischen Fertigungsablaufs vernachlässigen darf, erscheint eine flexible Produktionszelle als

"koordinierter Verbund gleichartiger oder unterschiedlicher Geräte, wie Roboter, CNC-Automaten oder speicherprogrammierbarer Steuerungen (SPS) zur Lösung einer bestimmten Produktionsaufgabe" /12/.

Es folgen Erläuterungen zu den wichtigsten Automatisierungsgeräten.

Speicherprogrammierbare Steuerungen (SPS) (Programmable Logic Control(ler), PLC) werden zur Steuerung und Überwachung von Fertigungsabläufen in der Weise eingesetzt, daß sie binäre Signale von externen Gebern aufnehmen, diese gemäß vorgegebenem Programm verknüpfen und die Ergebnisse an binär arbeitende Stellglieder ausgeben. Sie beinhalten hauptsächlich Funktionen zur Bitverarbeitung und ersetzen heute die historischen Relaissteuerungen.

Mit numerischen Steuerungen (Computerized Numerical Control, CNC) werden Werkzeugmaschinen, z.B. Fräsmaschinen, Drehmaschinen usw. gesteuert. "Computerized Numerical Control" hat "Numerical Control" weitgehend abgelöst. Die prinzipiellen Unterschiede sind, daß NC mit fest verdrahteten Modulen arbeitet, wohingegen CNC Mikroprozessoren verwendet /13/. In der Fachwelt werden beide Begriffe meistens synonym verwendet.

Mit Robotersteuerungen (Robot Control, RC) werden Industrieroboter gesteuert.

Anders als bei der SPS steht bei CNC und RC die geometrische Bewegungssteuerung mit den Aufgaben der Interpolation, Lageregelung usw. im Vordergrund. Robotersteuerungen unterscheiden sich von numerischen Steuerungen dadurch, daß

- die Anzahl der zu steuernden Achsen i.allg. höher ist (4-7) und
- andere Programmiermöglichkeiten ("Teach-In") und Interpolationsverfahren vorgesehen sind.

Bild 2.3 stellt die verschiedenen Gerätesteuern einander gegenüber.

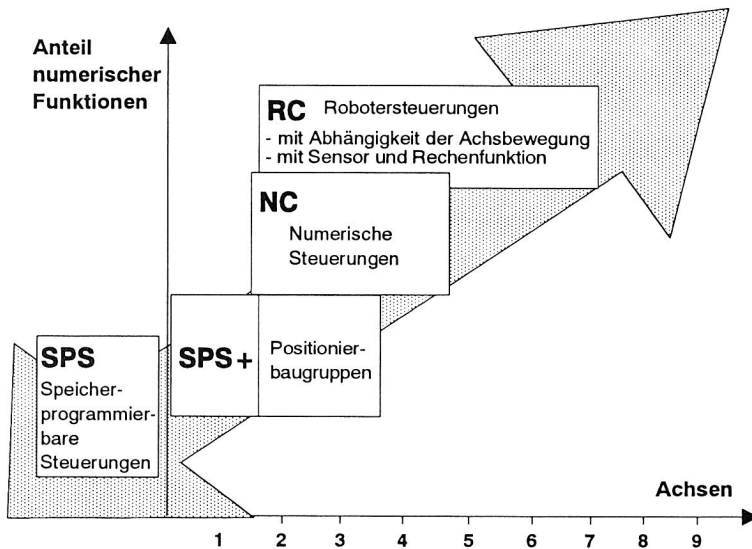


Bild 2.3: Gerätesteuern im Vergleich

Die am weitesten verbreitete "Schnittstelle"<sup>1</sup> zu numerischen Steuerungen und Robotersteuerungen heißt Direct Numerical Control (DNC). Als Eingabemedien für Gerätesteuern waren früher Lochstreifen, Magnetbänder oder Disketten vorgesehen. DNC bezeichnet die Leitungskopplung zwischen CNC bzw. RC und übergeordneten Fertigungssteuerungen (z.B. Prozeßrechner) für die "Online-Übertragung", das Starten, das Stoppen usw. von Programmen. /13/

Zur Erfüllung der Fertigungsaufgabe in Produktionszellen tragen neben den aufgeführten Klassen von Gerätesteuern auch einfache Automatisierungskomponenten wie Sensoren und Stellglieder bei.

Das Aufgabenfeld von flexiblen Produktionszellen ist in besonderem Maß von Echt-

<sup>1</sup> Unter "Schnittstelle" wird in diesem Zusammenhang immer eine "Kommunikationsschnittstelle" verstanden im Unterschied zu z.B. mechanischen Schnittstellen für Stellaufgaben im technischen Prozeß

Sensor/ Aktor	Anzahl Bits	Typische Antwortzeit	Preis (ca.)	Beispiele
Schneller, analoger Sensor	12	1 ms	US \$50	Strom, Kraft Spannung, Drehzahl
Langsamer, analoger Sensor	12	1 s	US \$50	Temperatur, Durchfluß
Digitale Eingänge (Ereignis)	1	1 ms	US \$1-5	Endschalter, Trigger
Digitale Eingänge (Zustand)	1	20-100 ms	US \$1-5	Identifizier
Aktoren (ON/OFF)	1	20 ms - 1 s		Relais, Ventil
Motoren	12	20 ms	US \$500	Servomotoren
Zähler	16	1 ms	US \$200	Timer, Drehgeber, Digitalmaßstab

Bild 2.4: Merkmale von Sensoren und Aktoren /14/

zeitverhalten (s. Abschnitt 2.2) geprägt. Die Forderung nach **garantierten** Antwortzeiten wird verschärft durch das Sensor-/Aktor-eigene Verhalten mit den extrem **kurzen** Reaktionszeiten. Bild 2.4 gibt u.a. einen Überblick über typische Antwortzeiten von Einfachgeräten.

## 2.2 Wichtige Begriffe aus der Technischen Informatik und Kommunikationstechnik

Die im folgenden erläuterten Begriffe sollen die technischen Gesichtspunkte der Kommunikation, wie sie für den praktischen Gebrauch in der Fabrikumgebung relevant sind, in den Vordergrund rücken. Grundlagen zur Theorie von Kommunikationssystemen, die man z.B. für die Spezifikation oder formale Verifikation von Protokollen anwendet, werden nicht betrachtet. Vielmehr wird ein Überblick über **physische** und **logische** Verbindungsstrukturen gegeben. Ferner werden wichtige Gesichtspunkte für die Einführung und Verbreitung von "offenen" Kommunikationssystemen erörtert.

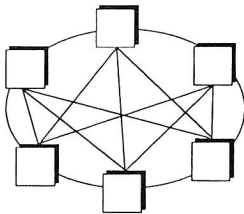
Neben dem Einsatz von Datenverarbeitung fordert der Integrationsaspekt von CIM insbesondere die "Kommunikation über Netze". In der Kommunikationstechnik wird unter einem *Netz* allgemein

"ein System von Leitungen und Vermittlungen zum Sammeln, Mischen und Verteilen von Informationsströmen" /15/

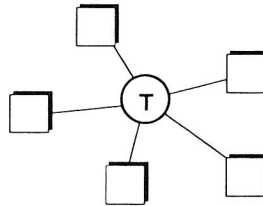
verstanden.

Dies schließt ein, daß als Hardware-Funktionseinheiten nicht nur **verarbeitende** (z.B. Rechner, Gerätesteuerungen), sondern auch **vermittelnde** (z.B. Router) Rechensysteme beteiligt sind.

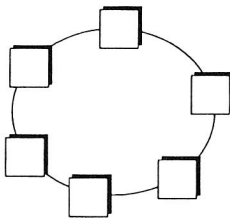
Die Möglichkeit, daß Rechensysteme über ein Kommunikationsnetz Daten austauschen, setzt die **physische** Verbindung der Komponenten voraus. Bild 2.5 zeigt die prinzipiellen Möglichkeiten für elementare Verbindungsstrukturen (Vollvermaschtes



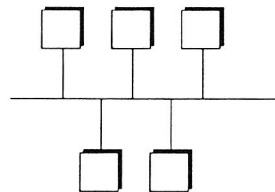
Vollständige Verbindung



Stern



Ring



Bus

Bild 2.5: Elementare Topologien

Netz, Stern, Ring, Bus). Daneben existieren alle möglichen Mischformen ( z.B. Bus mit zentralem Schalter).

Bei den physischen Nachrichtenkanälen kennt man (neben gemeinsamem Speicher, der für vorliegende Betrachtung ausscheidet) *leitungsgebundene* und *leitungslose* Übertragungssysteme:

- Zwei- und Mehrdrahtleitungen,
- Koaxialkabel,
- Lichtwellenleiter sowie
- Funk- und Satellitenverbindungen.

In der Fertigungsumgebung kommen überwiegend leitungsgebundene Übertragungsformen zum Einsatz. Eine Auswahl wird meist im Hinblick auf die gestellten Anforderungen (Kosten, störintensive Umgebung usw.) getroffen. Innovative Lösungen schlagen Infrarot- /16/ oder Mikrowellen-Übertragung /17/ an mobile Teilnehmer (z.B. fahrerloses Transportsystem) vor.

In der Literatur werden unterschiedliche Übertragungstechniken angegeben, die Begriffe sind aber nicht eindeutig trennbar:

- *Basisbandübertragung* mit der direkten Übertragung der 0/1-Signale.
- *Carrier- bzw. Trägerbandverfahren*:  
Die digitale Information wird durch Frequenzumtastung moduliert.
- *Breitbandübertragung*.  
Durch Frequenzmultiplexen entstehen mehrere unabhängige Übertragungskanäle.

Die Vorteile der Basisbandübertragung für den Einsatz in der Fabrikumgebung liegen in der einfacheren Technologie und damit den geringeren Kosten für Installation und Wartung. Breitbandsystemen haben aufgrund der Frequenzbandtechnik eine höhere Übertragungskapazität und können größere räumliche Distanzen überbrücken. Die für

die Fertigungsumgebung genormte MAP-Protokollarchitektur läßt (bisher) Carrierband- und Breitbandssysteme zu.

Bei den im folgenden behandelten **logischen** Verbindungsstrukturen werden Eigenschaften von Protokollen und Protokollarchitekturen behandelt. Zuvor jedoch werden die Begriffe "Schnittstelle", "Protokoll" und "Dienst"<sup>2</sup> kurz eingeführt:

"Die Verbindungsstelle zweier interagierender Teilsysteme nennt man *Schnittstelle*. Die Regeln für den Informationsaustausch und die Zusammenarbeit nennt man *Protokoll*. Die Kurzbezeichnung einer Leistung, die dem Benutzer an der Schnittstelle mit Hilfe eines Protokolls angeboten wird, nennt man *Dienst*" /15/.

Folgende Kommunikationsmechanismen werden besprochen:

- Multicast- und Broadcast,
- verbindungsorientierte und verbindungslose Kommunikation,
- bestätigte und unbestätigte Dienste.

Die für die Fertigungsautomatisierung besonders wichtige Eigenschaft der "Echtzeitfähigkeit" wird anschließend behandelt.

#### Multicast/Broadcast:

Tanenbaum knüpft die Möglichkeiten logischer Verbindungsstrukturen eng an die Gegebenheiten von physischen Verbindungsstrukturen. Er unterscheidet zwei Typen von "communication subnets" (Synonyme: "transport system", "transmission system"), die mit verschiedenen Kanälen arbeiten: "point-to-point channels" und "broadcast channels". Topologien für Punkt-zu-Punkt-Verbindungen sind Stern, "Loop", Baum etc. Busse, Satelliten- bzw. Funk-Verbindungen und Ringe sind Topologien für Broadcast-Systeme /18/.

---

<sup>2</sup> Bei den noch folgenden Ausführungen zum "ISO/OSI-Referenzmodell" wird näher auf Protokolle und Dienste eingegangen.

Generell lassen sich aber auch in Stern-, "Loop"- und Baumverbindungen Broadcast-mechanismen verwirklichen.

Die bereits erwähnten Multi- bzw. Broadcast-Mechanismen treten auch innerhalb von Rechensystemen mit kooperierenden Prozessen auf. Nach /19/ spricht man von einem *Broadcast*, wenn die Nachricht an alle im System vorhandenen Prozesse verschickt wird. Falls eine Nachricht nur an eine Teilmenge aller Prozesse geleitet wird, nennt man dies einen *Multicast*. In Anlehnung an diese Begriffsbildung wird eine 1:1-Kommunikationsform auch als *Unicast* bezeichnet. Broadcast und Unicast können als Spezialfälle eines Multicast aufgefaßt werden.

In der Fertigungsumgebung wird Multi- und Broadcast z.B. für Alarmmeldungen oder für die Synchronbewegung von Handhabungsgeräten benötigt.

#### Verbindungsorientierte/verbindungslose Kommunikation:

Bei der *verbindungsorientierten* Kommunikation ("connection oriented") kennt man drei Phasen:

- Verbindungsaufbau,
- Nutzdatenübertragung,
- Verbindungsabbau.

Bei *verbindungsloser* Kommunikation ("connectionless") wird jedes Datenpaket ("Datagramm") einzeln und unabhängig von anderen Datenpaketen transportiert. Sie kann unbestätigt (z.B. IEEE 802.2 Type 1) oder bestätigt (z.B. IEEE 802.2 Type 3) erfolgen.

Die Vorteile der verbindungsorientierten Kommunikation liegen laut /20/ darin, daß die Nutzdaten lediglich eine Verbindungskennung statt der vollständigen Adreßinformation mit sich führen und daß Maßnahmen zur Flußkontrolle, Sicherung usw. getroffen werden können.



Die Nachteile liegen einerseits in der aufwendigeren Implementierung und andererseits darin, daß die Abarbeitung betriebsmittelintensiv ist, denn die eine Verbindung beschreibenden Daten müssen verwaltet und ständig aktualisiert werden.

Für Anwendungen der Automatisierungstechnik wird meist vorgeschlagen, daß die Übertragung großer Datenmengen über zuverlässige Verbindungen - wie sie z.B. ISO-Transportprotokolle der Klasse 4 bieten - erfolgen soll. Für zeitkritische Kommunikation sollten Datagramm-Dienste verwendet werden, weil sie weniger aufwendig sind und kürzere Reaktionszeiten erwarten lassen /21/.

#### Bestätigte und unbestätigte Dienste

Die elementaren Interaktionen zwischen einem Benutzer und dem Dienstbringer nennt man *Dienstelemente*, die logisch als unteilbare Ereignisse betrachtet werden. Werden alle 4 Dienstelemente, wie sie Bild 2.6 zeigt, verwendet, so handelt es sich um einen *bestätigten* Dienst. Bei einem *unbestätigten* Dienst entfallen die Dienstelemente für die Antwort des Benutzers "*Response*" und "*Confirmation*".

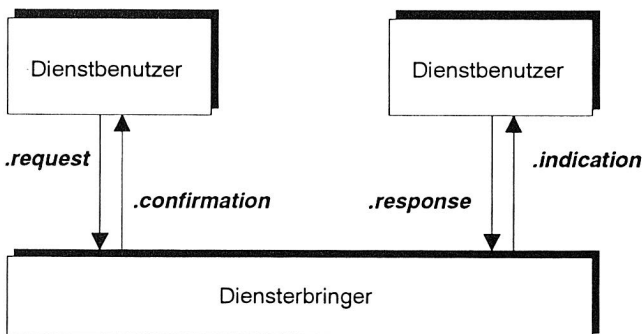


Bild 2.6: Dienstelemente

In der Fachliteratur findet man auch hier die 2-Teilung von Kommunikationsaufgaben hinsichtlich des Echtzeitverhaltens: unbestätigte Dienste für zeitkritische Kommunikation, bestätigte Dienste für zeitunkritische Kommunikation. Eine Sendewiederholung für zyklische, zeitkritische Informationen, wie z.B. Sensorwerte, ist nicht notwendig, da

der Sensorwert nur in Verbindung mit dem aktuellen Sendezeitpunkt Gültigkeit hat /22/.

Neben den aufgezeigten Kommunikationsmechanismen wird von Kommunikationsprotokollen (und Rechensystemen) in der Fertigung vor allem *Echtzeitverhalten* (s. auch /23/) gefordert:

"Unter Realzeitbetrieb eines Rechensystems versteht man im allgemeinen eine Betriebsweise, bei der die Aufträge im Rechensystem schritthaltend mit externen Vorgängen abgearbeitet werden, daß also die Verarbeitung simultan zur Datenentstehung durchgeführt wird. Dies bedeutet, daß die Verarbeitungsergebnisse innerhalb einer vorgegebenen Zeitspanne verfügbar sein müssen (garantierte Antwortzeiten), und daß mehrere Aufträge gleichzeitig zu bearbeiten sind (Mehrprogrammbetrieb). Die Größenordnung der Antwortzeiten ist problemabhängig [...]. Eine wichtige Kenngröße ist die maximale Zugriffszeit auf das Kommunikationsmedium, die entstehen kann" /24/.

Für zeitkritische Anwendungen werden deshalb *deterministische* Medienzugriffsverfahren gewählt. Bei Token-Verfahren hängt die maximale Zugriffszeit hauptsächlich von der Verweilzeit des Tokens im Teilnehmer und von der Anzahl der Teilnehmer ab. Nichtdeterministische (zufällige) Verfahren können keine maximale Wartezeit gewährleisten.

Echtzeitfähigkeit soll durch verschiedene Maßnahmen bei der Protokollgestaltung erreicht werden: verbindungslose Kommunikation, unbestätigte Dienste usw. Eine für die Echtzeitkommunikation vorgesehene Protokollarchitektur ("Enhanced Performance Architecture", s. Abschnitt 3) verzichtet auf Kommunikationsfunktionen, z.B. für Teilnetz-übergreifende Adressierung, um den Protokollaufwand zu vermindern und kürzere Antwortzeiten zu erreichen.

Zu den logischen Verbindungsstrukturen gehören ferner Vereinbarungen, wie sie mit dem ISO/OSI-Referenzmodell getroffen wurden.

Das ISO/OSI-Referenzmodell /25/ stellt ein Ordnungsschema zur Strukturierung von Kommunikationsaufgaben dar. Zugrundeliegendes Ziel war es, daß Rechensysteme verschiedener Hersteller und verschiedenen Typs "offen" miteinander kommunizieren können. Heutzutage wird das ISO-Referenzmodell fast immer dazu herangezogen, um die Funktionalität und die Dienstleistungen von Kommunikationssystemen zu bewerten und zu vergleichen. Das Modell gehorcht den Prinzipien der hierarchische Komposition und gliedert sich in zwei Teile:

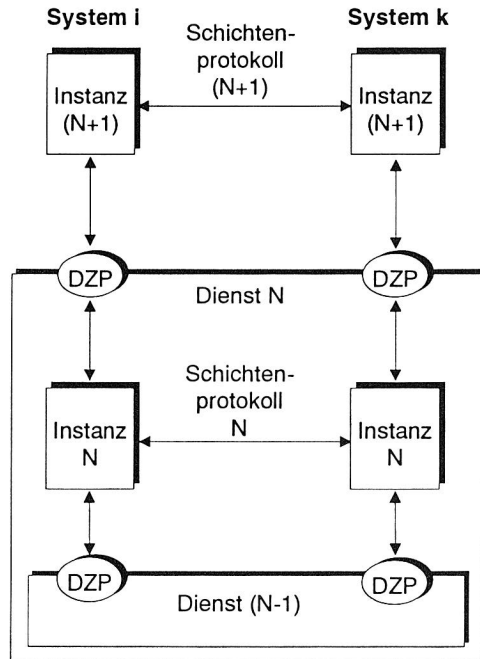
- Beschreibung eines Modells für geschichtete Protokolle und die dazu nötigen Begriffsdefinitionen,
- allgemeine Aufgabenbeschreibung der einzelnen Protokollschichten.

Kommunikationsaufgaben werden in sieben Schichten gruppiert:

- Ein *System* (im Sinne von ISO/OSI) besteht aus Rechensystemen, die über physische Verbindungen miteinander kommunizieren.
- *Anwendungsprozesse* stützen sich bei ihrer Kommunikation auf eine Hierarchie von *Kommunikationsdiensten*. Ein Dienst umfaßt alle Dienste der nächst niedrigeren Hierarchiestufe (s. Bild 2.7).
- *Funktionsschichten* ("*Layers*") entstehen wie folgt:  
Die Funktionalität, die zu einem niederen Kommunikationsdienst hinzugefügt werden muß, um den höheren Kommunikationsdienst zu erbringen, wird in den einzelnen Systemen durch *Instanzen* geleistet.
- Eine *Instanz* ist der abstrakte, funktionelle Träger der in einer Schicht in einem System erbrachten Dienste. Instanzen derselben Schicht in verschiedenen Systemen heißen *Partner-Instanzen* ("*Peer Entities*").
- Zur Erbringung eines Dienstes kommunizieren die Partner-Instanzen einer Schicht unter Inanspruchnahme der Dienstleistung der darunterliegenden Schicht.
- Die Kommunikationsregeln für Instanzen einer Schicht heißen *Schichten-Protokoll*. Die dabei ausgetauschten Daten heißen *Protokolldateneinheiten* ("*Protocol Data Units*"). Prinzipiell unterscheidet man Schichten-Protokolle zwischen

korrespondierenden Schichten unterschiedlicher Teilnehmer von *Dienstprotokollen* zwischen benachbarten Schichten eines Teilnehmers.

- Der Dienst für eine höhere Schicht wird von den Partner-Instanzen der darunterliegenden Schicht an *Dienstzugangspunkten* ("Service Access Points") im jeweiligen System erbracht. /26/



DZP: Dienstzugangspunkt

Bild 2.7: Zusammenhang zwischen Schichten, Diensten und Instanzen /25/

Den Zusammenhang zwischen Schichten, Diensten und Instanzen zeigt Bild 2.7, Bild 2.8 weist die Aufgaben der einzelnen Schichten aus.

Anwendungs-orientierte Kommunikationsdienste	7	Anwendungs-schicht	Direkte Anwenderunterstützung: Verbindungsaufbau zum Kommunikationspartner, Fernauslösung von Funktionen
	6	Darstellungs-schicht	Einheitliche Informationsdarstellung: Datenformate, inhaltliche Interpretation
	5	Sitzungs-schicht	Organisation und Steuerung des Dialoges zwischen den kommunizierenden Partnern
Transport-orientierte Kommunikationsdienste	4	Transport-schicht	Zuverlässiger und effizienter Datentransport zwischen den kommunizierenden Partnern
	3	Vermittlungs-schicht	Wegeauswahl und Vermittlung bis zum Ziel: Segmentieren, Füllen, Flußkontrolle
	2	Sicherungs-schicht	Einzelne zuverlässige Übertragungsabschnitte: Fehlererkennung u. -korrektur, Flußkontrolle
	1	Bitübertragungsschicht	Mechanische, elektrische und funktionelle Steuerung der Bitübertragung

Bild 2.8: Das ISO/OSI-Referenzmodell

Die im folgenden dargestellten "Lokalen Netze" und "Gateways" definieren sich sowohl über physische als auch logische Verbindungsstrukturen.

"Local Networks" ist ein sehr weit gefaßter Begriff und ist historisch entstanden aus dem Wunsch, einerseits an gemeinsamen Datenbeständen partizipieren und Daten austauschen und andererseits kostspielige Ressourcen ausnützen zu können.

"A local network is a communications network that provides interconnection of a variety of data communicating devices within a small area [...]. Data communicating devices are:

Computers, terminals, peripheral devices, sensors (temperatur, humidity, security alarm sensors), telephones, television transmitter and receivers, facsimile" /27/.

Unter dem Oberbegriff "local networks" werden die Klassen "local area networks", "high speed local networks" und zentral gesteuerte Nebenstellenanlagen ("Private

Branch Exchange", PBX) zusammengefaßt.

"The term *local area network (LAN)* ist typically used to refer to a general-purpose local network, which can serve a wide variety of devices over a large area. LANs support minis, mainframes, terminals, and peripherals [...]. A common type of LAN is a bus or tree using coaxial cable. Rings using twisted pair, coax, or even fiber are an alternative. The data transmission rates on LANs (1 to 20 Mbps) are high enough to satisfy most requirements and provide sufficient capacity to permit large numbers of devices to share the network" /27/.

Lokale Netze sind in Bus- oder Ringtopologie ausgelegt. In der Produktionshierarchie haben sie sich auf den höheren Ebenen etabliert: ETHERNET-LANs mit zufallsorientierter Buszuteilung (CSMA/CD: Carrier Sense Multiple Access with Collision Detection) sind weit verbreitet und für die Produktionsleitebene, Fertigungsleitebene und Zellenebene gut geeignet, da kein Echtzeitverhalten gefordert ist. Eine wesentlich höhere Übertragungsleistung (bis zu 100 MBit/s) erzielen Glasfasersysteme ("Fiber Distributed Data Interface", FDDI /28/) mit allerdings komplizierter Anschlußtechnik. Besondere Bedeutung für den Produktionsbereich haben Token-Bus-Netze mit MAP-Protokollarchitektur (s. Abschnitt 3) erlangt.

Lokale Netze gehören zum Stand der Technik und sind u.U. auf bestimmte Anwendungen hin ausgerichtet: z.B. Bürokommunikation, Fabrikkommunikation usw. Durch Netzübergänge soll die Kopplung solcher Einzelnetze mit unterschiedlichen Fähigkeiten oder die Erweiterung von Netzen erreicht werden. Sie erfolgt i.allg. über sog. *Gateways* (Synonym: Netzübergangspunkt, Protokollumsetzer). Dabei ist eine Abstimmung und Anpassung der evtl. unterschiedlichen Dienstleistungen notwendig, beginnend bei der Bitübertragungsschicht bis hin zur Anwendungsschicht.

Folgende Nomenklatur hat sich durchgesetzt:

- **Gateway:** Allgemein übliche Bezeichnung für einen Protokollumsetzer: Z.T. wird sie nur für einen Protokollumsetzer auf den Ebenen 4 bis 7 benutzt.

- *Router:* Umsetzer auf Schicht 3,
- *Bridge:* Umsetzer auf Schicht 2,
- *Repeater:* Umsetzer auf Schicht 1 (Leitungsverstärker). /15/

Tanenbaum definiert:

"The function of the gateway is to convert packets from one protocol to another"  
/18/.

Allerdings gehört zur Aufgabe eines Protokollumsetzers neben der Kodierungsumsetzung, die die Transformation der zugrundeliegenden Grammatiken und Kodierungsregeln beinhaltet, auch die Umsetzung des prozeduralen Protokollablaufs, der u.a. die Reihenfolge von Aufträgen und Quittungen bestimmt. Bild 2.9 zeigt die Verwendung von "Bridge" und "Router" am Beispiel einer MAP-Demonstrationsanlage.

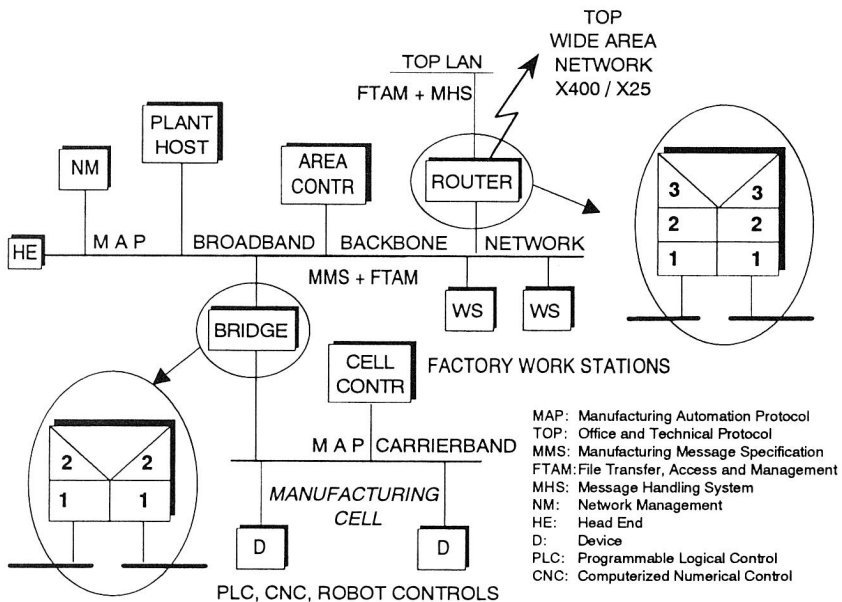


Bild 2.9: Bridges und Router in einer MAP-Demonstrationsanlage /30/

Im Anschluß an die Betrachtungen zu den physischen und logischen Verbindungsstrukturen folgt die Erläuterung von Begriffen, die verdeutlichen, daß die Einführung "offener" Kommunikationssysteme insbesondere in die Produktionstechnik forciert wird: "Migration", "Konformität", und "Interoperabilität".

"Migration zu offener Kommunikation" ist derzeit ein strapaziertes Schlagwort und subsummiert alle Zwischenschritte auf der "Wanderung" von spezifischen Netzen zu standardisierten Kommunikationssystemen. Rose unterscheidet "transition" und "migration" dadurch, daß bei der Migration auch Rückwärtsbewegungen zur vorher betriebenen Protokollarchitektur erlaubt sind /29/.

Namhafte Firmen besitzen derzeit "Migrationsstrategien", um die Konformität ihrer Netzwerkprodukte - unter Wahrung der vorhandenen Funktionalität - zum MAP-Standard herzustellen /31/.

Konformität bezeichnet die Eigenschaft, daß eine Netzwerkkomponente die Vorschriften von internationalen Kommunikationsstandards einhält. Um dies zuverlässig überprüfen zu können, werden sog. "Konformitätstests" ausgearbeitet, die einzelne Protokollimplementierungen daraufhin untersuchen, ob sie die in den Normen geforderten Dienste fehlerfrei erbringen. Nach bestandem Test werden die Produkte von international anerkannten Einrichtungen (z.B. Institut für Informations- und Datenverarbeitung in Karlsruhe) zertifiziert /32/. Derzeit wird an Konformitätstests für MAP (Manufacturing Automation Protocol), TOP (Technical and Office Protocol) und PROFIBUS (Process Field Bus) gearbeitet /33, 34/.

Während der Konformitätstest ausschließlich jeweils **eine** Implementierung auf Konformität mit den spezifizierten Funktionen einer ausgewählten ISO/OSI-Norm testet, wird mit dem "Interoperabilitätstest" das Zusammenwirken **mehrerer**, bereits auf Konformität getesteter Implementierungen geprüft, um Unverträglichkeiten zu erkennen, die aufgrund der Ausnutzung von Definitionslücken der Norm und in der Norm enthaltener Inkonsistenzen entstanden sind.

Andererseits wird unter Interoperabilität eine Eigenschaft auf ISO/OSI-Ebene 7



verstanden, wo die Abbildung von realen Geräteeigenschaften (z.B. CNC, RC, SPS) auf abstrakte Objekte erfolgt. Durch Interoperabilität soll erreicht werden, daß

"unabhängig von der Art der anzuschließenden Geräte kein spezielles Wissen über die Geräte vorhanden sein muß, um einen Nachrichtenaustausch durchzuführen. Das bedeutet, daß Kommunikationsprotokolle festgelegt sind und der Datenaustausch objektorientiert abläuft" /35, 36/.

Konformität und Interoperabilität setzen "Interkonnektivität" - Eigenschaften der ISO/OSI-Schichten 1 und 2 - voraus, wonach Geräte gemäß vereinbarter elektrischer und physikalischer Spezifikationen gekoppelt und Nachrichten zwischen ihnen ausgetauscht werden können. Oberhalb der Anwendungsebene beinhaltet "Interchangeability" zusätzlich Vereinbarungen über die Semantik der Nachricht (z.B. unter Verwendung von "Device Description Language" /37/).

Die angeführten Begriffe geben folgende Grundgedanken zum Ausdruck:

- Offene Kommunikation in der Fertigung wird durch die Einhaltung standardisierter Kommunikationsprotokolle erreicht.
- Dies gilt im besonderen Maß für die Schnittstelle zum Anwender.

Hinweis: Die Ausführungen vorliegender Arbeit sollen u.a. deutlich machen, daß die uneingeschränkte Konformität mit dem MAP-Anwendungsprotokoll ("Manufacturing Message Specification", s. Abschnitt 3) und die Verwendung der spezifischen Erweiterungen für verschiedene Gerätetypen ("Companion Standards") einheitlichen Informationsaustausch zwischen Automatisierungskomponenten verwirklicht.

### 3 Darstellung und Wertung derzeitiger Kommunikationssysteme in der rechnerintegrierten Produktion

Die Ausführungen dieses Abschnittes orientieren sich am Ebenenmodell der rechnerintegrierten Produktion. In Top-Down-Vorgehensweise wird mit **Kommunikationssystemen in der Produktions- und Fertigungsleitung** begonnen, **Feldbusse** und **Kommunikationsschnittstellen für Sensoren und Aktoren** bilden den Abschluß.

Wichtige Aspekte für die Kommunikation im Fertigungsbereich wie **Anwendungsorientierung** und **Echtzeitunterstützung** werden für sich betrachtet. (Bild 3.1 ordnet die einzelnen Abschnittsthemen dem CIM-Ebenenmodell und dem ISO/OSI-Ebenenmodell zu.)

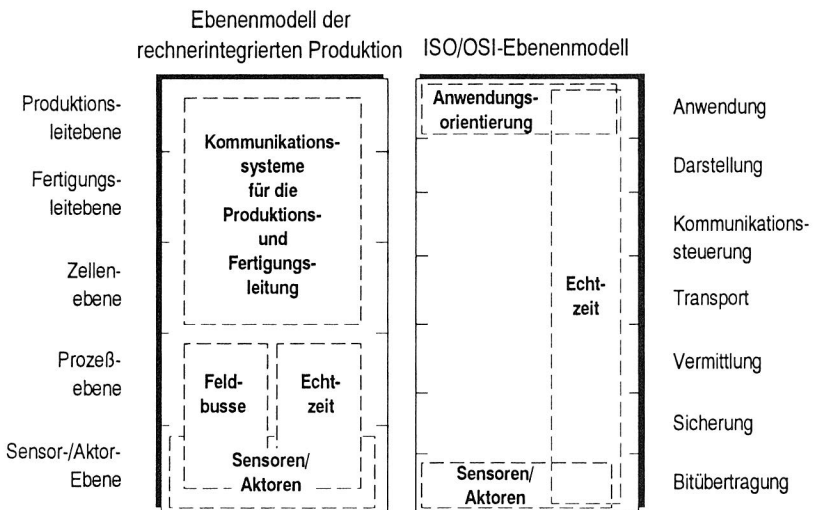


Bild 3.1: Einordnung der Abschnittsthemen

### 3.1 Kommunikationssysteme in der Produktions- und Fertigungsleitung

Kommunikationsvorgänge in und zwischen Produktionsleit-, Fertigungsleit- und Zellenebene lassen sich quantitativ folgendermaßen charakterisieren (s. Bild 3.2):

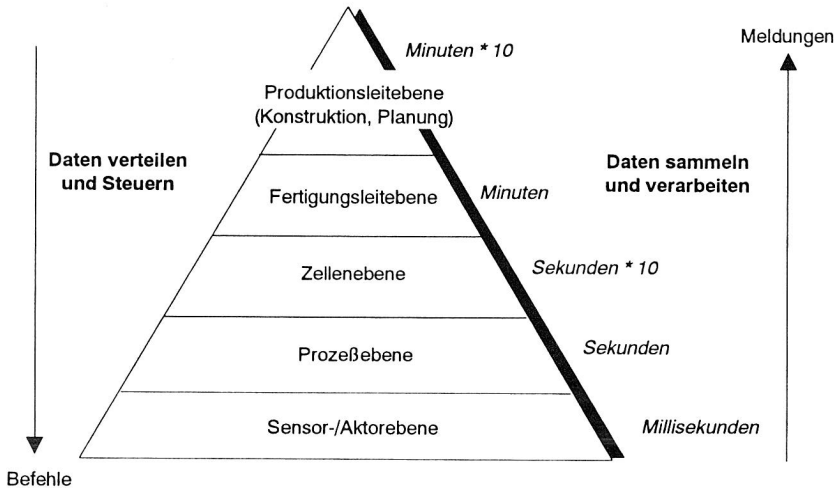


Bild 3.2: Datenaufkommen und Zeit-Restriktionen im hierarchischen CIM-Modell

- Die zu übermittelnden Datenmengen sind groß. Z.B. können in der Arbeitsvorbereitung erstellte Steuerungsprogramme für Roboter oder CNC-Maschinen bis zu einige hundert Kilobyte umfassen. Mit CAD-Systemen erzeugte Werkstück-Dateien sind je nach geometrischer Komplexität von einigen Kilobyte bis zu einigen Megabyte groß.
- Die Zeitanforderungen sind gering und liegen im Bereich von Tagen, Stunden oder Minuten. Z.B. müssen Arbeitspläne und Aufträge von der Produktionsplanung und -steuerung nur bei Änderungen am Fertigungsspektrum übermittelt werden.

An Dienstfunktionalität ist meist eine Filetransfer-Funktion ausreichend.

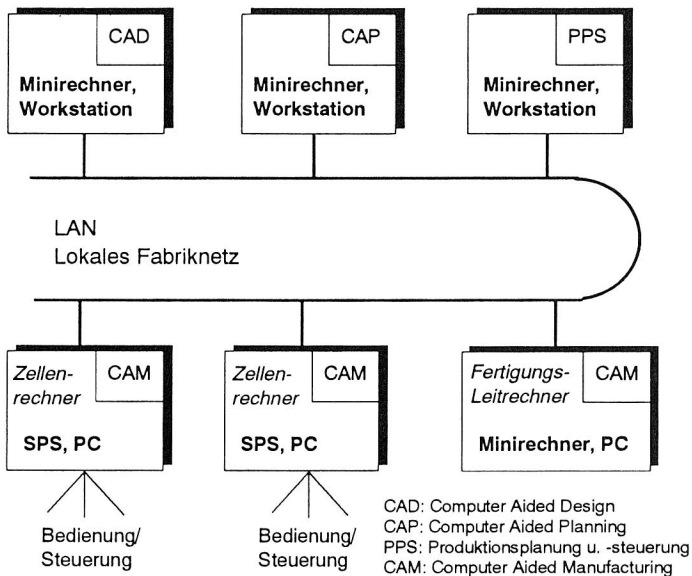


Bild 3.3: Rechnerausstattung in der Produktion

Bild 3.3 zeigt eine exemplarische Rechnerausstattung in einem Produktionsbetrieb. Die Zahl der angeschlossenen Komponenten hängt von der Größe des Betriebes ab und kann bis zu einige hundert erreichen. Als Übertragungssysteme werden **Lokale Netze** in verschiedenen Ausführungen eingesetzt:

Neben ETHERNET-LANs und Glasfasersystemen haben Token-Bus-Netze mit MAP-Protokollarchitektur besondere Bedeutung für den Produktionsbereich erlangt.

**Manufacturing Automation Protocol (MAP)** wurde anfang der 80er Jahre von der Fa. General Motors mit dem Ziel ins Leben gerufen, ein standardisiertes Kommunikationssystem für die automatisierte Fertigung zu schaffen /6, 38, 39/. Für die einzelnen Ebenen des ISO/OSI-Modells sind Normprotokolle festgelegt (s. Bild 3.4). MAP wurde inzwischen von verschiedenen Rechner- und Steuerungsherstellern implementiert und das Zusammenspiel ihrer Produkte auf einigen Messen demonstriert /30, 40, 41/, ...).

<b>Anwendungs-orientierte Kommunikationsdienste</b>	7	Anwendungs-schicht	SASE	File Transfer, Access and Management (FTAM) IS 8571, Manufacturing Message Specification (MMS) IS 9506
	6	Darstellungsschicht	CASE	Association Control Service Elements (ACSE) IS 8649/8650
	5	Sitzungsschicht		Presentation IS 8822/8823 Session IS 8236/8237
<b>Transport-orientierte Kommunikationsdienste</b>	4	Transportschicht		Transport IS 8073 Class 4
	3	Vermittlungsschicht		Network Service Connectionless Mode (PCNS) IS 8473
	2	Sicherungsschicht	LLC	Logical Link Control Class 1 IS 8802/2
	1	Bitübertragungsschicht	MAC	Token Bus IS 8802/4 Breitband, Carrierband

Bild 3.4: Die MAP-Protokollhierarchie

Die Anwendung Lokaler Netze und der MAP-Protokollarchitektur bleibt aus folgenden Gründen auf die Produktions- und Fertigungsleitung beschränkt

- Das Konzept Lokaler Netze geht von gleichberechtigten Kommunikationspartnern aus. Die Belange flexibler Produktionszellen, wo hinsichtlich der Kommunikationsfähigkeit sehr heterogene Teilnehmer angesiedelt sind, werden nicht berücksichtigt.
- Die Tauglichkeit innerhalb flexibler Produktionszellen wird wegen des erheblichen Protokollaufwandes durch die volle Ausprägung aller 7 ISO/OSI-Schichten ("Full MAP") und dem damit verbundenen schlechten Leistungsverhalten auf Anwendungsebene bestritten.
- Die MAP-Anschlußkosten sind für Geräte der Prozeß- und Sensor/Aktor-Ebene zu hoch.

Bild 3.5 zeigt zum Vergleich Geräteanordnung und **Vernetzung bei Prozeßleitsystemen** in der Verfahrenstechnik. Im Unterschied zum CIM-Mehr-Ebenenmodell dominiert hier die Trennung von "Warte" und "Feld" mit u.U. explosionsgefährdeter Umgebung. Leitstation(en) und Prozeßstation(en) kommunizieren über ein herstellerspezifisches Netz. Die Zuordnung von Prozeßgeräten und Prozeßein-/ausgängen wird durch Rangierverteiler (z.B. Klemmenleiste) und zusätzliche örtliche Verteiler vorgenommen.

Die Daten werden wie im prozeßnahen Bereich der Fertigung noch überwiegend in Form von statischen analogen oder binären Standardsignalen übertragen. Angestrebt wird ein einheitliches, **offenes** Kommunikationssystem für Prozeß- und Sensor/Aktor-Ebene: der Feldbus.

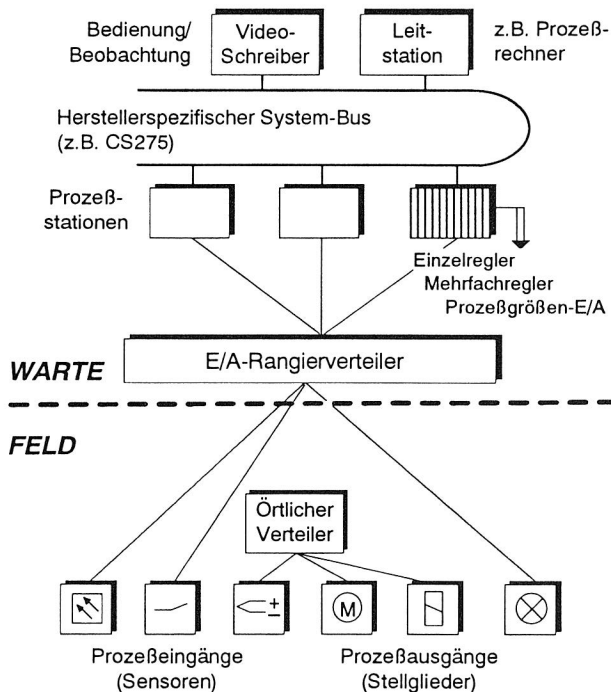


Bild 3.5: Struktur von Prozeßleitsystemen

## 3.2 Feldbusse

### 3.2.1 Allgemeine Eigenschaften

Aufgrund der Nähe zum technischen Prozeß sollen Feldbusse folgende Eigenschaften haben:

- **wirtschaftlich** mit serieller Datenübertragung und Zweidraht-Leitung als Medium, geringen Anschaltkosten für den Teilnehmer und evtl. Hilfsenergieübertragung über den Bus,
- **echtzeitfähig**, z.B. für Alarmer und Regelkreise und
- **störsicher**.

Problematisch ist die große **Diversität der Feldbusstationen**: Leistungsfähige Steuerungen (numerische, Roboter-) stehen einfachen Sensoren und Stellgliedern gegenüber.

Das Ersetzen der konventionellen sternförmigen Verdrahtungstechnik durch **Bustopologien** hat zur Folge, daß Feldbusse meist eine "Single Master"-Struktur aufweisen: Ein zentraler "Master" hat allein das Zugriffsrecht auf den Bus und kann bei Bedarf einem der "Slaves" die Sendeberechtigung erteilen.

Es werden folgende Maximalwerte für charakteristische Parameter angegeben:

- Distanzen bis zu 10 km werden überbrückt.
- Bis zu 1000 Stationen können angeschaltet sein.
- Kleine Paketlängen (typisch: 2 bis 16 Byte) ermöglichen in Verbindung mit optimierten Protokollen (und bei geringer Stationszahl) Reaktionszeiten unter 10 ms.
- Übertragungsraten bis zu 5 MBit/s sind üblich. /42/

Feldbusse werden abgegrenzt gegenüber

- Lokalen Netzen und
  - "Rechnerperipherie"-Bussen (z.B. SCSI-Bus, IEC-Bus) mit paralleler Datenübertragung, die Standardperipherie (z.B. Hintergrundspeicher) und komplexe Sondergeräte (z.B. Meß- und Prüfgeräte) mit PCs und Workstations verbinden.
- /42/

Normung	Industriestandards	Anwendung
PROFIBUS (DIN 19245)	BITBUS	Automatisierung
FIP	CS275	Prozeß- leitsysteme
DIN Meßbus (DIN 66348)	PDL	
Eureka "Fieldbus"	SINEC L1	
ISA SP 50	Contronic P	Automobil
ESPRIT CNMA/Fieldbus	ABUS	
PROWAY C (ISA SP 72)	CAN	
	Mil Standard 1553B	Militär
	SERCOS <sup>1)</sup>	Werkzeug- maschinen

IEC-BUS (IEEE 488)  
(PDV-Bus (DIN 19241))

1) LWL-Ring

Bild 3.6: Aktuelle Feldbuslandschaft

Bild 3.6 gibt einen Überblick über Normen und Normvorhaben sowie industrielle Standards mit ihren Anwendungen. Einige der genannten Feldbusse arbeiten mit elektrischen Anschaltungen gemäß EIA RS-485 (Electronic Industry Association Recommended Standard 485). Es handelt sich dabei um eine symmetrische Übertragung. Sie hat gegenüber der asymmetrischen, wo das Nutzsignal dem Absolutwert gegenüber dem Massepotential entspricht, den Vorteil der Störuneempfindlichkeit gegenüber Gleichtaktstörungen, da das Signal aus der Spannungsdifferenz von zwei Leitungspaaren gewonnen wird.



### 3.2.2 Der PROFIBUS als exemplarischer Vertreter

Das BMFT-Verbundprojekt PROFIBUS wurde 1987 mit dem Ziel gestartet, einen Vorschlag für eine internationale Feldbus-Norm zu erarbeiten.

Als Ergebnis wurde die PROFIBUS-Norm veröffentlicht: Teil 1 beinhaltet "Übertragungstechnik, Buszugriffs- und Übertragungsprotokoll, Dienstschnittstelle zur Anwendungsschicht, Management" /43/, Teil 2 beschreibt "Kommunikationsmodell, Dienste für die Anwendung, Protokoll, Syntax, Codierung, Schnittstelle zur Schicht 2, Management" /44/. Es wird eine auf die Schichten 1/2 und 7 reduzierte Struktur mit einer Ebene-7-"Sub-Schicht", dem "Lower Layer Interface" (LLI), angegeben, die aber Transportfunktionen ausübt. Demnach hat der PROFIBUS die in Bild 3.7 dargestellte Architektur.

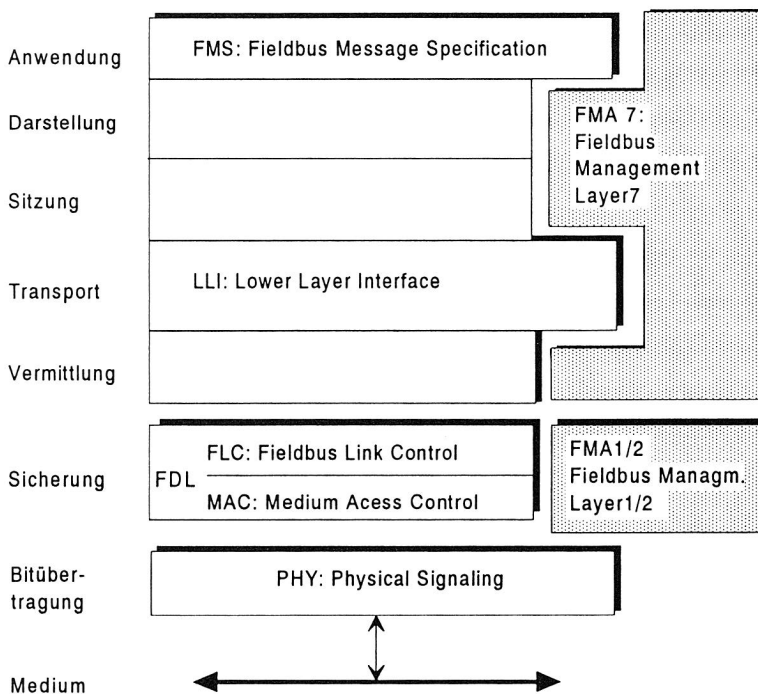


Bild 3.7: PROFIBUS-Schichtenstruktur

Für die Bitübertragungsschicht (PHY) wurden die in Bild 3.8 genannten Vereinbarungen getroffen. In einer ersten Version richtet sich die Übertragungstechnik nach dem USA-Standard EIA RS-485. Als geplante weitere Versionen sind eigensichere Ankopplungen (Ex i) für explosionsgefährdete Umgebungen in der Verfahrenstechnik mit zudem besonderer elektromagnetischer Verträglichkeit (EMV) geplant. Als Medium werden auch Lichtwellenleiter (LWL) in Betracht gezogen. Für Geräte mit niedriger Leistungsaufnahme ist die Hilfsenergieübertragung auf der Busleitung vorgesehen.

PHY + Medium	
o	Geschirmte, verdrehte 2-Drahtleitung
o	Maximale Leitungslänge pro Segment: 1200m; gesamt: 4800m
o	Maximale Teilnehmerzahl pro Segment: 32; gesamt: 122
o	Symmetrische Übertragung EIA RS-485
o	Übertragungsraten:
	9,6 , 19,2, 93,75 kBit/s bis 1200m,
	187,5 kBit/s bis 600 m,
	500 kBit/s bis 200 m
o	"UART-Characters" mit 8 Datenbit, 1 Paritätsbit, 1 Stopbit für asynchrone Übertragung

Bild 3.8: PROFIBUS: Medium und Bitübertragungsschicht

Beim PROFIBUS-Zugangsprotokoll (MAC) handelt es sich um eine "hybride" Methode, die das Token-Passing-Verfahren und die Master-Slave-Kommunikation kombiniert (s. Bild 3.9). Die Teilnehmer werden gemäß einer Zweiklassenstruktur ("aktiv"/"passiv") eingeteilt. Aktive Teilnehmer (Master) bilden einen logischen Ring und erlangen dezentral durch Token-Weitergabe Zugriff auf den Bus. Jedem aktiven Teilnehmer sind ein oder mehrere passive Teilnehmer (Slaves) zugeordnet, die nur nach Aufforderung senden dürfen.

Die Funktionalität der Sicherungsschicht (FLC) ist in Bild 3.10 dargestellt.

Mit Hilfe der Dienste "CRDR" und "CSRD" kann ein aktiver Teilnehmer **zyklisch** "seine" Slaves oder auch andere Master ansprechen. Dies tut er anhand einer vorgegebenen Reihenfolge, der sog. "Poll-Liste".

Das sog. "Lower Layer Interface" (LLI) ist laut PROFIBUS-Veröffentlichungen ein "Sub-

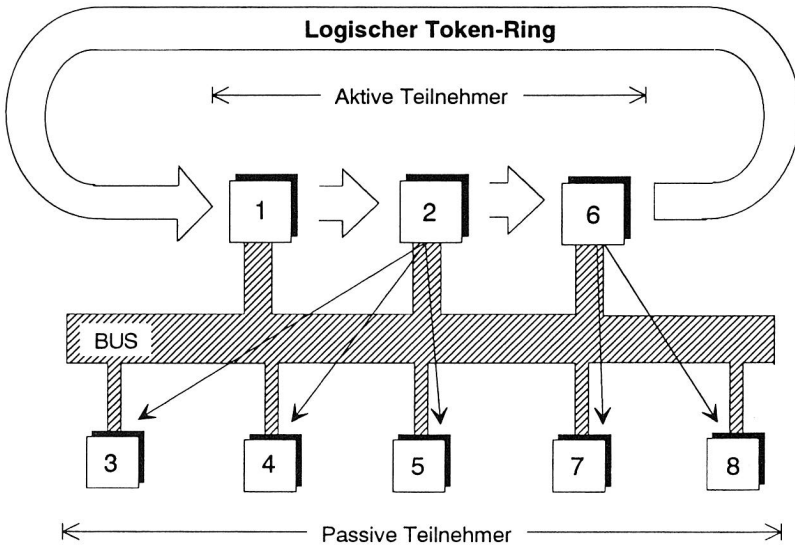


Bild 3.9: PROFIBUS-Zugriffsprotokoll

Layer" in der Schicht 7 und übernimmt die Abbildung der Anwendungsdienste auf die feldbusspezifische Schicht 2 /45/. Tatsächlich aber verbergen sich dahinter typische Transport-Funktionen wie Flußkontrolle und die Bereitstellung logischer Verbindungen: Für jeden Teilnehmer werden beim Systemstart unter Verwendung des Netzwerkmanagements Verbindungen zu den anderen Teilnehmern in der sog. "Kommunikations-Beziehungs-Liste" (KBL) festgelegt. Die jeweilige Verbindung wird bei einem Schicht-7-"Initiate"-Aufruf aktiviert bzw. "Abort"-Aufruf deaktiviert.

Das Anwendungsprotokoll "Fieldbus Message Specification" (FMS) ist an MMS angelehnt und beinhaltet die in Bild 3.11 dargestellten Dienstgruppen. Da es mit FMS nicht möglich ist, zur Laufzeit Objekte zu kreieren oder zu löschen, wurden spezielle Objektmanagement-Dienste hinzugefügt. Die Objekte eines Teilnehmers werden anhand einer Reihe von Verzeichnissen (Objekt-, Datentyp-) verwaltet. Jeder Teilnehmer beinhaltet sein lokales Objektverzeichnis und Kopien der Objektverzeichnisse

"Fieldbus Data Link" - Dienste	"Fieldbus Management" - Funktionen
<b>Send Data with Acknowledge (SDA)</b> Daten an einen Teilnehmer senden und unmittelbare Quittierung	Rücksetzen der Schichten 1 und 2
<b>Send Data with No Acknowledge (SDN)</b> Daten an einen, mehrere (Multicast) oder alle (Broadcast) Teilnehmer senden ohne Quittierung	Änderung lokaler Betriebsparameter
<b>Send and Request Data (SRD)</b> Daten von einem Teilnehmer anfordern und gleichzeitig Daten senden; unmittelbare Quittierung mit Antwortdaten	Anforderung des aktuellen Teilnehmerabbildes
<b>Cyclic Send and Request Data (CSR)</b> Zyklische Datenanforderung und -sendung mit Antwort nach Teilnehmer-Poll-Liste	Konfiguration und Aktivierung/Deaktivierung von Dienstzugangspunkten
	Melden unerwarteter Ereignisse

Bild 3.10: PROFIBUS-Sicherungsschicht

seiner Kommunikationspartner. Die Partner-Objektverzeichnisse können mit den Diensten aus dem "Objektverzeichnis Management" gelesen und aktualisiert werden. Die Einzelobjekte werden über "spezielle Kurzadressen" ("Indizes") angesprochen (s. Bild 3.12). PROFIBUS kennt statische und dynamische Objekte. Die "dynamischen" Objekte werden aus den statischen zusammengesetzt (z.B. Variablenlisten aus Einzelvariablen).

Folgende Kritikpunkte seien angemerkt:

- Die Netzwerkschicht in der PROFIBUS-Schichtenstruktur ist nicht besetzt. Über sog. "Repeater" läßt sich zwar die geographische Netzausdehnung und Teilnehmerzahl vergrößern, die Möglichkeit einer strukturierten Teilnetzbildung mit entsprechender Adressierung besteht allerdings nicht. Ebenso kann man keine unterschiedlichen Medien (und damit Übertragungstechniken) gemischt betreiben.
- Da das "Lower Layer Interface" typische Transportfunktionen übernimmt, sollte es auch formal der ISO/OSI-Schicht 4 zugeordnet werden.

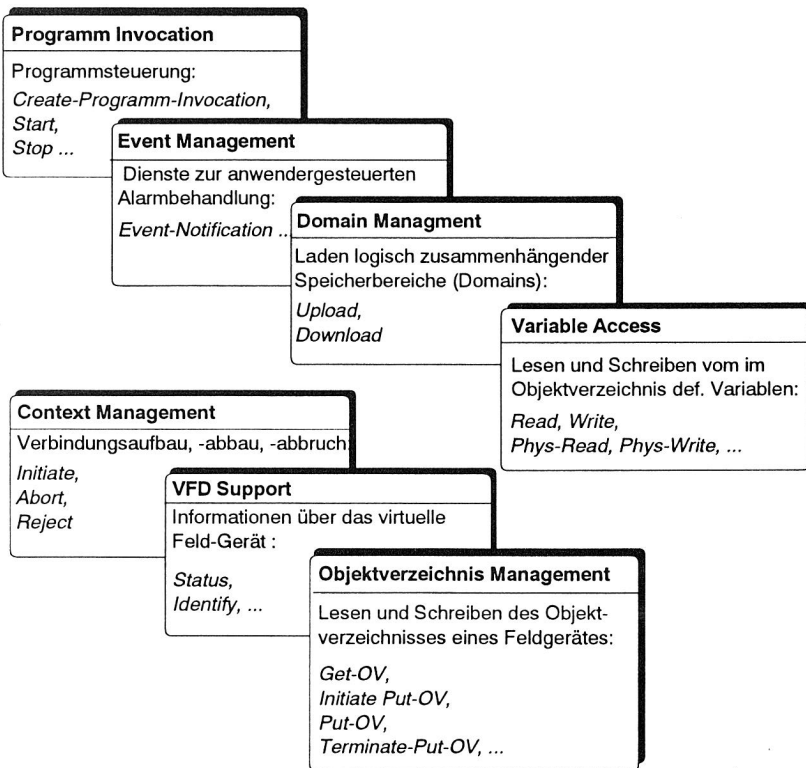


Bild 3.11: PROFIBUS-Anwendungsdienste

- Das Konzept der Kommunikationsbeziehungen läßt zur Laufzeit keine Korrespondenz zwischen beliebigen Stationen zu, da alle Verbindungen in einer vorangegangenen Projektierungsphase festgelegt werden.
- Wegen der **statischen** Objekte müssen alle für die jeweilige Kommunikationsanwendung relevanten Objekte vordefiniert sein (z.B. beim Download die "Ziel"-Domain im Server). Mit Hilfe des Zugriffs über Indizes und der zugehörigen Typinformation im Objektverzeichnis soll die Nutzinformation bei der Übermittlung verringert werden. Dies führt zur Mehrfach-Datenhaltung der Objektverzeichnisse und den damit verbundenen Datenkonsistenzproblemen. (Bei MMS wird die

Datentypverzeichnis										
Index	Objekt-code	Bedeutung								
1	DTyp	Integer 8								
2	DTyp	Integer 16								
3	DTyp	Integer 32								
⋮										
→ 7	DTyp	Integer 16, Grad Celsius								

Objektverzeichnis										
Index	Objekt-code	Index auf Datentyp	Länge	Pass-wort	Zugriffs-gruppen	Zugriffs-rechte	Int. Adresse	Symbol	Exten-sion	
110	Var	3	1	--	--	--	4046 H	Zähler 1	--	
111	Var	4	1	--	--	--	6020 H	Zähler 2	--	
112	Var	→ 7	3	--	--	--	6040 H	Temp Oel	--	
				--	--	--			--	
---	---	--	--	--	--	--	-----	-----	--	
				--	--	--			--	
120	Var	1	2	--	--	--	4046 H	Zähler 1	--	

Bild 3.12: Beispiele für PROFIBUS-Objektverzeichnisse

Typinformationen der Objekte in den Dienstaufrufen mitgeführt.)

- Trotz syntaktischer (Formulierung der PDUs in ASN.1) und semantischer (gleichlautende Objekte und Dienstgruppen) Ähnlichkeiten unterscheidet sich FMS von MMS in wesentlichen Punkten. Die Kompatibilität zur MAP-Anwendungsschnittstelle ist deshalb nicht gegeben.

### 3.3 Kommunikationsschnittstellen für Sensoren und Aktoren

Schnittstellen mit Einfachgeräten (Sensoren/Aktoren bzw. Meßwertaufnehmer/Stellglieder) legen mechanische, elektrische und funktionelle Eigenschaften fest, wie sie der ISO/OSI-Ebene 1 zuzuordnen sind (mit Ausnahme der Analogsignale, s.u.). Die Kommunikation ist auf den Austausch von Signalen beschränkt, die man klassifiziert als

- analog,
- binär,
- digital.

Beim Binärsignal handelt es sich zwar um ein spezielles Digitalsignal /46/, nämlich eines mit genau 2 diskreten Funktionswerten ("0", "1"), es wird hier aber getrennt aufgeführt, da eine Vielzahl von Einfachgeräten (sog. "schaltende Sensoren" z.B. zur Anwesenheitskontrolle) mit dieser Signalform arbeiten.

Für die Bildung **analoger** Signale kommen die elektrischen Größen Stromstärke und Spannung in Frage, Frequenz und Phase spielen bei der Übertragung keine Rolle. Dafür wurden Einheitssignale vereinbart, die z.B. Spannungswerte für Analogeingaben zwischen -10V und +10V und Stromstärken zwischen 4 und 20mA zulassen /47/. Beim Stromsignal ist dies so zu verstehen, daß einem Meßbereich 0 bis 100% ein Meßsignal zwischen 4 bis 20mA entspricht. Das Gleichspannungssignal ist für die Weiterverarbeitung durch Analog/Digitalumsetzer mit den dort üblichen Spannungseingängen besser geeignet /48/. Sensoren mit Analogsignalen sind z.B. Thermoelemente, Manometer, Ringwaagen. Zu analog steuerbaren Aktoren gehören Servo-Antriebe mit +/- 10V-Schnittstelle /49/ oder Thyristor-Leistungsteller für elektrische Antriebe.

Nennwerte und Arbeitsbereiche für **binäre** Signalein- und -ausgaben sind in DIN 19240 genormt. Bild 3.13 zeigt charakteristische Bereiche für Ein- und Ausgänge bei 24V Nennspannung und Signalsprache "Positive Logik". Stellglieder mit Binärsignalen sind z.B. Magnetventile, Schütze oder Meldeleuchten. Binäre Meßwertaufnehmer sind induktive und optische Näherungsschalter sowie Lichtschranken.

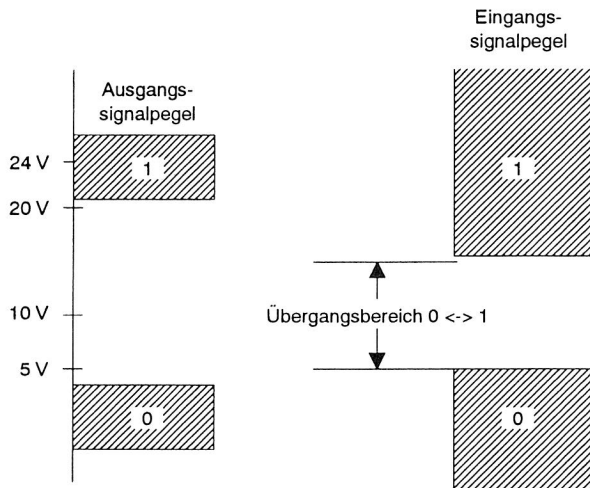


Bild 3.13: Bereiche von Ein- und Ausgangssignalpegeln binärer Schnittstellen /48/

**Digitale** Schnittstellen im obigen Sinn (mit 2<sup>n</sup> Funktionswerten) sind weniger verbreitet und kommen nur bei Zähler-Bausteinen vor. Prinzipiell gehören dazu auch alle Meßwertaufnehmer mit integriertem A/D-Umsetzer. Üblich ist die Übertragung digitaler Signale in Form von seriellen Schnittstellen, wie sie in Bild 3.14 aufgeführt sind. RS-232-C ("Recommended Standard Number 232, Revision C from Electronic Industry Association"), RS-423-A und RS-422-A unterscheiden sich in der Übertragungsrate, mit RS-232-C können nur relativ geringe Entfernungen überbrückt werden. RS-232-C ist eine Punkt-zu-Punkt-Verbindung, mit RS-423-A und RS-422-A können 1:10-Master-Slave-Beziehungen realisiert werden. Publierte vergleichende Darstellungen der EIA-"Normen" führen auch RS-485 auf. Diese Richtlinie gestattet die Anschaltung von bis zu 32 gleichberechtigten Teilnehmern. Aufgrund der "Busfähigkeit" ist RS-485 der Feldbus-Thematik (s. Abschnitt 3.2.1) zuzuordnen.

Die 20mA-Stromschleife /50/ arbeitet im Unterschied zu den RS-"Normen", wo den "0"/"1"-Funktionswerten Spannungspegel zugeordnet sind, mit Stromstärken.



Schnittstellen	V.24/V.28 RS-232-C	V.10/X.26 RS-423-A	V.11/X.27 RS-422-A
Übertragungsart	unsymmetrisch	unsymmetrisch	unsymmetrisch
Zahl der Treiber	1	1	1
Zahl der Empfänger	1	10	10
Übertragungsstrecke	15 m	1200 m	1200 m
Max. Übertragungsrate	20 kBit/s	100 kBit/s	10 MBit/s
<u>Sender:</u>			
zulässige Treiber- ausgangsspannung	$\pm 25 \text{ V}$	$\pm 6 \text{ V}$	- 0,25 ... 6 V
Treiber- ausgangssignal			
ohne Last	$\pm 15 \text{ V}$	$\pm 6 \text{ V}$	$\pm 5 \text{ V}$
mit Last	$\pm 5 \text{ V}$	$\pm 3,6 \text{ V}$	$\pm 2 \text{ V}$
Treiberlast	3 ... 7 k $\Omega$	= 450 $\Omega$	= 100 $\Omega$
<u>Empfänger:</u>			
Eingangsspannung	$\pm 15 \text{ V}$	$\pm 12 \text{ V}$	$\pm 7 \text{ V}$
Empfindlichkeit	$\pm 3 \text{ V}$	$\pm 200 \text{ mV}$	$\pm 200 \text{ mV}$
Eingangswiderstand	3 ... 7 k $\Omega$	4 k $\Omega$	12 k $\Omega$

Bild 3.14: Vergleich von Schnittstellen-"Normen" /51/

### 3.4 Anwendungsorientierung

#### 3.4.1 Manufacturing Message Specification

Manufacturing Message Specification (MMS) ist ein international genormtes Anwendungsprotokoll und Bestandteil von MAP /52/. MMS basiert auf dem *Client-Server*-Modell: Ein übergeordneter Auftraggeber (*Client*), z.B. ein Fertigungsleitrechner oder ein Zellenrechner, erteilt Aufträge bzw. stellt Anfragen an einen allgemein zugänglichen Auftragnehmer (*Server*), z.B. eine Robotersteuerung, die dieser ausführt bzw. beantwortet. Die Partner kommunizieren mittels einer *Manufacturing-Message-Protocol-Machine (MMPM)*. Ihre Aufgabe ist die Kodierung der Dienstaufrufe bzw. Dekodierung der Protokolldateneinheiten sowie die normkonforme Abwicklung des MMS-Protokolls. Dafür sorgen mehrere dedizierte "Protokollmaschinen" im Sinne der Automatentheorie.

MMS ermöglicht es, die realen Gegebenheiten der Fertigung in die ISO/OSI-Welt abzubilden. Mit Hilfe des *VMDs* (*Virtual-Manufacturing-Device*) und einer Reihe weiterer abstrakter Objekte kann die für die Kommunikation relevante Funktionalität eines realen Gerätes (z.B. Werkzeugmaschine, Roboter) dargestellt werden. Die MMS-Dienste sind gemäß den Objektklassen, auf denen sie operieren, zu den in Bild 3.15 dargestellten funktionellen Einheiten zusammengefaßt.

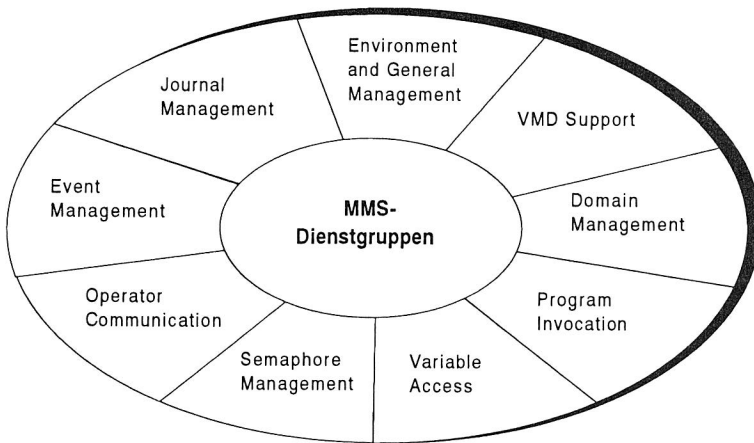


Bild 3.15: MMS-Dienstgruppen

*Environment-and-General-Management-Services* sind Dienste zum Auf- und Abbau von Kommunikationsverbindungen.

Die *VMD-Support-Services* dienen dem *Client* zu Überwachung des *VMDs*, z.B. für Abfragen der Geräteidentifikation oder innerhalb des *VMD* definierter Objekte.

*Domains* enthalten Informationen, die als Programm-Daten oder als Programm-Code interpretiert werden können. Mit Hilfe von *Domain-Download-Services* aus der Gruppe des *Domain-Managements* kann der *Client* *Domains* in ein *VMD* transferieren. Mit Diensten aus dem *Program-Invocation-Management* werden man unter Verwendung der *Domains* ablauffähige Prozesse kreiert und gesteuert: die *Program-Invocations*.

Mit *Variable-Access*-Diensten können Variable geschaffen, geschrieben und gelesen werden. Clients koordinieren mit *Semaphore-Management*-Diensten den Zugriff auf

gemeinsame Betriebsmittel des *Servers*, *VMDs* und *Domains*.

Mit *Operator-Communication* werden Ein- und Ausgaben eines Bedieners behandelt. Beim *Event-Management* werden *Event Conditions* mit *Event Actions* verknüpft. Der *Client* veranlasst beim Auftreten von bestimmten Ereignissen, z.B. Alarmen, den *Server* dazu, ihn darüber unverzüglich zu informieren und/oder selbst Aktionen auszulösen. Das *Journal-Management* befaßt sich mit dem Abspeichern und Abrufen von chronologisch geordneten Informationen.

Zusätzlich ist im Anhang von MMS ein *File-Management* vorgesehen, für Dateioperationen steht allerdings FTAM /53/ als eigenständiges Protokoll zur Verfügung.

Bei den meisten MMS-Diensten handelt es sich um bestätigte Dienste ("Confirmed Services"): Der Service-Request eines MMS-Anwenders (*Requester*) wird dem Partner von dessen MPPM als Service-Indication angezeigt. Dieser antwortet mit einer Service-Response, und der Requester erhält die Bestätigung durch eine Service-Confirmation.

Für die Spezifizierung ISO-Protokollen wurde die Sprache "Abstract Syntax Notation One" (ASN.1) /54, 55/ geschaffen, in der auch die MMS-Dienste bzw. MMS-Protokoll-dateneinheiten formuliert sind.

Bild 3.16 zeigt eine typische MMS-Anordnung. Das *VMD* ist Bestandteil des *Servers*, der im Unterschied zum *Client* zur Anwendungsschicht gehört, und modelliert die Gerätespezifika des Roboters bzw. der Robotersteuerung. Es verfügt über eine Reihe von *Domains* und *Variables*. Die *Domains* enthalten z.B. Bewegungsprogramme für die Handhabung von Werkstücken. *Variables* sind geeignet, um Gerätestatus-Informationen zu speichern, die der *Client* z.B. mit *Read* abrufen kann.

Um die Interoperabilität von MMS-Software unterschiedlicher Herkunft zu erreichen, wurden 7 Implementierungsklassen ("Implementation Classes MAP 1-7") vereinbart /39/. Sie definieren einen Mindest-Funktionsumfang, indem sie festlegen, welche Dienste und Variablentypen verwendet werden (s. auch "Konformitätsklassen" bei MMS-Companion-Standards in Abschnitt 3.4.2).

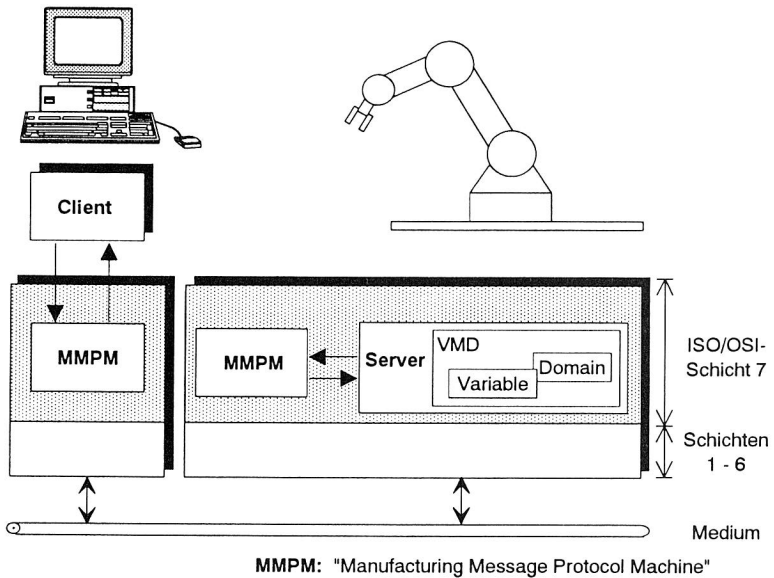


Bild 3.16: Typische MMS-Anordnung

Mit MMS ist eine einheitliche Schnittstelle für die Kommunikation von Automatisierungskomponenten geschaffen. Als internationaler Standard ist es für aktuelle und künftige Hard- und Softwareentwicklungen für Gerätesteuern zwingend vorgegeben. Mit Rücksicht auf einfache Gerätetypen sollte eine kostengünstige Realisierung angestrebt werden. Der Kommunikationsaufwand kann z.B. dadurch vermindert werden, daß aus den insgesamt 86 Diensten nur eine für das jeweilige Gerät sinnvolle Teilmenge implementiert wird. Die Implementierungsklassen erscheinen zu umfangreich, zumal bei deren Definition von leistungsfähigen Gerätetypen ausgegangen wurde. ("MAP 1" für NC-Maschinen, "MAP 2" und "MAP 3" für speicherprogrammierbare Steuerungen, "MAP 4" für Robotersteuerungen usw.) Die funktionale Mächtigkeit wird gleichfalls geringer, wenn man die Anzahl der dedizierten Protokollmaschinen herabsetzt.

### 3.4.2 MMS-Companion-Standards

MMS-Companion-Standards sind Erweiterungen von Manufacturing Message Specification Teil 1 und 2 ("MMS-Core") für die verschiedenen Gerätetypen wie Robotersteuerungen, numerische Steuerungen, speicherprogrammierbare Steuerungen und Prozeßleitsysteme (s. Bild 3.17).

Titel	Norm	Version	Fachgremium
Manufacturing Message Specification- Part 3: Companion Standard for Robotics (RCMS)	ISO/IEC IS 9506-3	Aug. 1991	nat.: NAM 96.2.4 internat.: ISO TC 184/SC 2/WG 6
Manufacturing Message Specification- Part 4: Companion Standard for Numerical Control (NCMS)	ISO/IEC DIS 9506-4	Juli 1991	nat.: NAM 96.1.3 internat.: ISO TC 184/SC 1/WG 3
Manufacturing Message Specification- Part 5: Companion Standard for Programmable Controller (PCMS)	IEC/ISO Draft 9506-5	Sept. 1990	nat.: DKE AK 936.0.7 internat.: IEC SC 65A/WG 6/TF 7
Manufacturing Message Specification- Part 6: Companion Standard for Process Control (PIMS)	ISO/IEC DIS 9506-6	Aug. 1992	nat.: DKE AK 933.3.1 internat.: IEC SC 65C/WG 1

Bild 3.17: MMS-Companion-Standards

Für die Gestaltung von Companion-Standards sind Richtlinien ("Requirements for Companion Standards") vorgegeben. Danach werden die Eigenschaften der realen Automatisierungskomponente (I) in einem mehrstufigem Abbildungsprozeß in das abstrakte MMS-Modell (V) übergeführt (s. Bild 3.18). In der ersten Stufe wird ein MMS-unabhängiges System-Modell (II) abgeleitet. Es besteht aus der Modellierung der Gerätespezifika und der Fertigungsumgebung (*Application-Model*) sowie aus Anwendungsfunktionen (*Application-Specific-Functions*). Für die Modellierung der Gerätespezifika werden die relevanten mechanischen Komponenten und steuerungstechnischen Module vorgestellt.

Als Anwendungsfunktionen werden z.B. im NC-Companion-Standard folgende

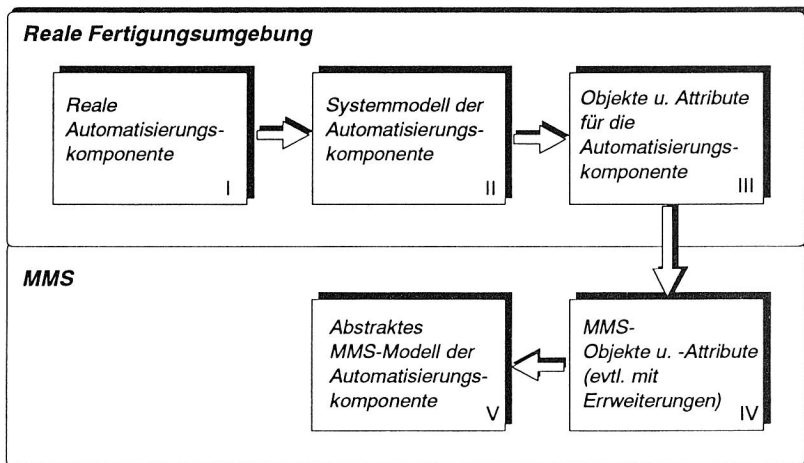


Bild 3.18: Mehrstufiger Abbildungsprozeß bei MMS-Companion-Standards angegeben:

- *Request-Data-Transfer,*
- *Alarm-Processing,*
- *NC-Information,*
- *Identify-Functions,*
- *Process-Management,*
- *Interact-With-The-Operator,*
- *Switching-Functions.*

Aus dem Systemmodell (II) können dann spezifische Objekte und Attribute (III) gewonnen werden.

Nach der Abstrahierung der Gegebenheiten des realen Systems erfolgt eine Abbildung der Objekte und Attribute in die MMS-Welt (IV). Die MMS-Objekte sind u.U. für die Abbildung der gerätespezifischen Eigenschaften um zusätzliche Attribute erweitert. Z.B. wird für das VMD-Objekt u.a. ein Attribut "*Robot-VMD-State*" mit den selbsterklärenden

Zuständen "*ROBOT-IDLE*", "*ROBOT-LOADED*" usw. als Wertbelegungen eingeführt:

*Object: VMD*

*All MMS defined Attributes*

*Attribute: ...*

*Attribute: Robot VMD State (ROBOT-IDLE, ROBOT-LOADED, ROBOT-READY, ROBOT-EXECUTING, ROBOT-PAUSED, MANUAL-INTERVENTION-REQUIRED)*

*Attribute: ...*

MMS-Core stellt Objektklassen (*Variables, Domains, Program Invocations, ...*) zur Verfügung, mit denen eine Gerätesteuerung dargestellt werden kann. Da innerhalb der Gerätetypen (NC, RC usw.) über eine Reihe von Objektausprägungen Einigkeit besteht, wird diesen ein "Standard"-Bezeichner zugewiesen. Z.B. verfügt jede NC-Steuerung über einen einstellbaren Korrekturfaktor für die Vorschubgeschwindigkeit ("Override"). Dieser wird beim NC-Companion-Standard mit Hilfe einer *Named-Variable* dargestellt, die *N\_FRO* heißt:

*Object: Named Variable*

*Key Attribute: Variable Name = domain-specific*

*{ domainID "N\_DEV\_...", itemID "N\_FRO" }*

*Attribute: MMS Deletable = FALSE*

*Attribute: Type Description = STRUCTURE {*

*components {*

*{ componentName "overridePerCent"*

*componentType boolean },*

*{ componentName "overrideValue"*

*componentType floating-point*

*{ format-width 32*

*exponent-width 8 } } }*

*Attribute: Access Type = locally defined*

*Access right = Read/Write*

("N\_FRO" selbst gehört zu einer Domain, deren Name mit "N\_DEV" beginnt.)

MMS-Dienste, die auf modifizierten MMS-Objekte zugreifen, sind entsprechend erweitert, bzw. für nicht auf MMS-Core abbildbare Funktionen werden neue Dienste angegeben. Der RC-Companion-Standard definiert beispielsweise einen *VMD-Stop-Service*, der die Robotersteuerung ausschaltet, weil der Eingriff eines Bedieners notwendig ist.

MMS-Objekte, -Attribute und -Dienste machen das abstrakte MMS-Modell der Automatisierungskomponente (V) aus.

Die MMS-unabhängige Klassifizierung von Anwendungsfunktionen (*Application-Specific-Functions* aus dem Systemmodell (II)) wird in Form von *Conformance-Classes* in den MMS-Kontext übernommen. Den Zusammenhang zwischen den Anwendungsfunktionen und den Konformitätsklassen zeigt Bild 3.19. Konformitätsklassen bestehen aus sog. *Conformance-Building-Blocks*, die elementare MMS-Funktionen ausdrücken. Man unterscheidet:

- a) den *Service-Conformance-Building-Block*, der aus einem MMS-Dienst besteht und
- b) den *Parameter-Conformance-Building-Block*, der die Komplexität von Variablen, z.B. Variablentyp, Schachtelungstiefe usw. festlegt.

Konformitätsklassen unterscheiden sich in den verschiedenen *Companion-Standards*. Für Roboter gibt es eine *Base-Conformance-Class* und *Higher-Conformance-Classes*. Die *Robot-Base-Conformance-Class* umfaßt die folgenden *Conformance-Building-Blocks*.



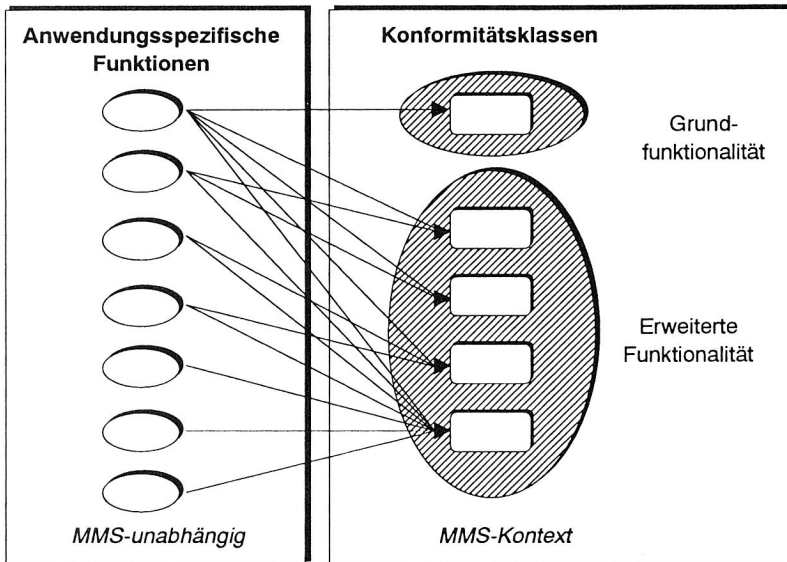


Bild 3.19: Zusammenhang zwischen Anwendungsfunktionen und Konformitätsklassen

*Service-Conformance-Building-Blocks:*

- Zur Verwaltung von MMS-Verbindungen:  
*Initiate, Conclude, Abort, Reject und Cancel.*
- Zur Identifizierung und für Statusabfragen an bzw. Statusmeldungen von der Robotersteuerung:  
*Identify, Status, Unsolicited-Status.*
- Für die Verwaltung von Programmen und Daten:  
*Get-Domain-Attributes, Delete-Domain* u.a. Dienste aus dem *Domain-Management*.
- Für die Verwaltung und Bedienung von Programmen:  
*Create-Program-Invocation, Delete-Program-Invocation, Start, Stop* usw.
- Für das Lesen und Schreiben von Variablen in der Robotersteuerung:  
*Read, Write* und *Information-Report*.

*Parameter-Conformance-Building-Blocks:*

*STR1, STR2, VNAM, NEST = 2.*

Eine Robotersteuerung gemäß der *RC-Companion-Base-Conformance-Class* muß an Variablentypen sog. *Named-Variables*, die über ihren Namen angesprochen werden (im Unterschied zur Adressvariable) (*VNAM*), Felder (*STR1*) und Strukturen (*STR2*) unterstützen, wobei die maximale Schachtelungstiefe 2 beträgt (*NEST = 2*).

MMS-Companion-Standards wurden (und werden) erarbeitet, um anwendungsspezifische Eigenschaften der verschiedenen Gerätetypen zu berücksichtigen, die MMS-Core nicht vorsieht. Innerhalb der einzelnen Standards werden **anwendungsspezifische Kommunikationsfunktionen** (*Application-Specific-Functions*) sowie **Konformitätsklassen** (*Conformance-Classes*) definiert. Sie klassifizieren somit Kommunikationssituationen auf Anwendungsebene (Schicht 7) und in der Anwendung. Die Normungsgremien sind bestrebt, alle Anforderungen mit den vorhandenen Companion-Standards abzudecken. Es sollen keine zu großen Abweichungen (neue Attribute, neue Dienste) von MMS-Core und der Companion-Standards untereinander entstehen. Dies ließe sich vermeiden, wenn Companion-Standards lediglich "Standard"-Objekte festlegen und Anleitungen bezüglich der Verwendung der Dienste geben würden. Der Gerätevielfalt in der automatisierten Stückgutproduktion ist man nämlich noch nicht gerecht geworden (z.B. Leiterplatten- und Halbleiterproduktion). Dem Bestreben, auch mit einfachen Aktoren und Sensoren über eine **offene** Anwendungsschnittstelle zu kommunizieren, würde ein Companion-Standard weiterhelfen.

### 3.4.3 Semiconductor-Equipment-Communication-Standard

Semiconductor-Equipment-Communication-Standard (SECS) ist ein vom "Semiconductor Equipment and Materials Institute, Inc." (SEMI) entwickelter Industriestandard für die Kommunikation in der Elektronikproduktion /56/. Er beinhaltet:

- SECS-I "Message Transfer" und
- SECS-II "Message Context".

SECS ist, wie man am Erscheinungstermin erkennt (SECS-I: 1980), nicht vom ISO/OSI-Modell ausgegangen, man ordnet aber inzwischen die Funktionalität von SECS-I den ISO/OSI-Schichten 1 bis 6 und die von SECS-II der Anwendungsschicht zu /57/.

Während sich SECS-I mit der physikalischen Übertragung (EIA RS-232-C) und dem Format von Nachrichten beschäftigt, trifft SECS-II semantische Festlegungen, die auf die Belange der Halbleiterfertigung bzw. globaler die der Elektronikproduktion abgestimmt sind. SECS-II beinhaltet folgende Bereiche:

- Format von Dateneinheiten,
- Übersicht über Standard-Dateneinheiten,
- Übersicht über *Streams* und *Functions* und
- Protokolle für den Dialog (*Conversation*).

SECS-II definiert den Inhalt einer Nachricht als eine Folge von Dateneinheiten (*Data-Items*). Jede Dateneinheit besteht dabei aus einem Formatbezeichner, einem Längenbezeichner und dem Datum selbst.

Ein *Stream* bezeichnet eine Kategorie ähnlicher Funktionen. Unter einer *Function* wird eine spezifische Aktivität verstanden: Z.B. gilt für jeden *Stream* die Konvention, daß die Funktion "0" die Bedeutung *Abort-Transaction* besitzt und anstelle einer erwarteten Antwort auf eine Anfrage gesendet werden kann. Funktionen werden durch entsprechende Nachrichten repräsentiert: *Primary-* (Anfrage, Befehl) bzw. *Secondary-*

*Message (Antwort)*. Bild 3.20 zeigt die vereinbarten *Streams*, z.B. für Statusabfragen (*Equipment-Status*), für Diagnosefunktionen (*Equipment-Control-and-Diagnostics*) usw.

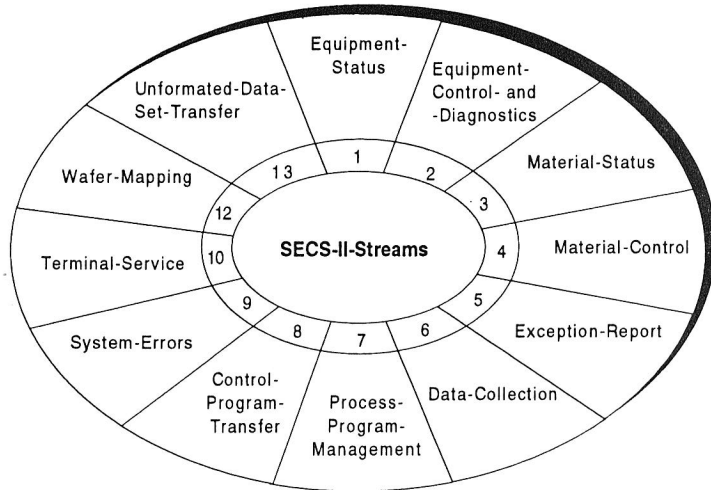


Bild 3.20: SECS-II-Streams

SECS-II definiert einige Dialoge zwischen Sender und Empfänger. Jeder Dialog beginnt dabei mit einer aus einem *Block* (Informationseinheit von SECS-I) bestehenden *Primary-Message*:

- *Single-Message/No-Reply*-Dialog,
- *Request/Data*-Dialog,
- *Send/Acknowledge*-Dialog,
- *Inquire/Grant/Send/ACK*-Dialog,
- *Material-Control*-Dialog,
- Benutzerdefinierte Dialoge.

Die Kommunikation mit SECS wirft folgende Probleme auf:

- SECS bietet keine Fabrikvernetzung wie MAP:  
SECS-I beruht auf Punkt-zu-Punkt-Verbindungen zwischen Leitrechner ("host") und Geräterechner ("equipment"). Im Vergleich mit den geringen Kosten von V.24-Anschaltungen kommen für Elektronikproduzenten MAP-Breitband- oder -Carrierband-Knoten nicht in Frage, solange die Anzahl der Geräteverbindungen pro Leitrechner eine Grenze von 300 nicht übersteigt. Für umfangreichere Installationen kann die MAP-Architektur übernommen werden, mit Ausnahme der Schicht 7, die durch SECS-II ersetzt wird. Den Übergang zwischen dem ACSE-Protokoll und SECS-II soll "SECS Message Services" (SMS) leisten. /58/
- SECS-II ist keine **offene** Anwendungsschnittstelle:  
Nach Meinung von SEMI ist SECS-II bestmöglich auf die Halbleiterfertigung abgestimmt und es besteht kein Bestreben zum Einsatz der MAP-Anwendungsdienste. Allerdings sind viele semantische Übereinstimmungen zu MMS zu erkennen:
  - SECS-Streams sind vergleichbar mit MMS-Dienstgruppen, z.B. *Equipment-Status* mit *VMD-Support*, *Exception-Report* mit *Event-Management* usw.
  - Einfachen SECS-Dialogen entspricht bei ISO/OSI die Verwendung von Dienstprimitiven (*Request*, *Response*). Der kompliziertere Dialog *Inquire/-Grant/Send/ACK* für die Übermittlung von Prozeßprogrammen wird bei MMS mit der Dienstfolge für den *Download* durchgeführt.

### 3.5 Echtzeitunterstützung

#### 3.5.1 MAP-Enhanced-Performance-Architecture und Mini-MAP

MAP-Enhanced-Performance-Architecture ist Bestandteil der MAP-3.0-Spezifikation /39/. Ziel war es, Full-MAP-Stationen so zu modifizieren, daß sie vorabbestimmte (möglichst kurze) Antwortzeiten garantieren und somit auch für zeitkritische Anwendungen geeignet sind. MAP/EPA-Stationen sind demnach hauptsächlich innerhalb flexibler Produktionszellen und in Prozeßleitsystemen für verfahrenstechnische Prozesse angesiedelt.

Die gestellten Anforderungen sollen dadurch erfüllt werden, daß die zeitkritischen

Anwendungen über eine Architektur kommunizieren, bei der Vermittlungs-, Transport-, Sitzungs- und Darstellungsschicht leer bleiben. Besteht in einer Station lediglich die reduzierte Schichtenarchitektur, so spricht man von einem "Mini-MAP"- oder "Real-Time-MAP"-Knoten (s. Bild 3.21). /59/

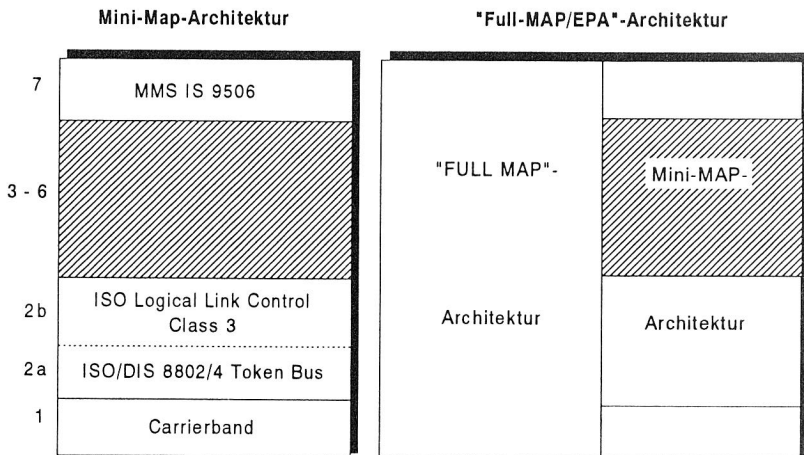


Bild 3.21: Full-MAP/Enhanced-Performance-Architecture und Mini-MAP

Die physikalische Schicht entspricht der Full-MAP-Spezifikation, Carrierbandanschaltung werden aber aufgrund der geringeren Kosten vorgezogen. Der Netzzugang erfolgt ebenso wie in MAP gemäß IEEE-802.4-Token-Passing-Protokoll.

Die Sicherungsschicht arbeitet mit verbindungslosen Diensten, d.h. ohne Verbindungsaufbauphase gemäß "IEEE 802.2 Class 3". Dazu gehören "Type 1"-Dienste ohne Bestätigung und bestätigte "Type 3"-Dienste. Mit Hilfe der unbestätigten Dienste und unter Angabe einer entsprechenden Adresse können im Unterschied zu Full-MAP Multi- bzw. Broadcast-Nachrichten an mehrere bzw. alle Teilnehmer verschickt werden.

"Type 3"-Dienste beinhalten solche zur Kommunikation mit Stationen, die nicht zum logischen Ring gehören ("immediate response"):

Diese Stationen können eine vorgefertigte Nachricht bei Anfrage einer im Ring befindlichen Station abliefern. Mit diesem Mechanismus kann ein Polling bei sog.

"passiven" Stationen durchgeführt werden. Auf der Ebene 7 wird deshalb neben MMS ein zusätzliches "Specific Service Element" (SASE) benötigt: Der "Obtain Reply"-Dienst ermöglicht dem Anwender die Abfrage einer "Nicht-Token-Station".

EPA führt neben den 7 von MAP eine zusätzliche Implementierungsklasse ein: Sie enthält nur die MMS-Dienste *Information-Report* (Melden eines Variablenwertes vom Server) und *Unsolicited-Status* (Zustandsmeldung des Servers) sowie den *Obtain-Reply*-Dienst.

Die Kommunikation gemäß EPA bzw. Mini-MAP hat folgende Nachteile gegenüber Full-MAP:

- Die Nachrichtenlänge ist durch die Vorgaben der Sicherungsschicht beschränkt.
- Mini-MAP-Stationen können nur innerhalb eines Teilnetzes kommunizieren. Zur Anbindung nach "außen" ist ein Gateway notwendig.
- Fehlererkennungs- und -korrekturmaßnahmen der Transportschicht fehlen.
- Abstrakte Syntax und Kodierung müssen a priori bekannt sein.
- Ohne ACSE-Service-Element können keine Anwendungsverbindungen aufgebaut werden, die sich auf einen gemeinsamen Anwendungskontext einigen.

### 3.5.2 Factory-Automation-Interconnection-System

Factory-Automation-Interconnection-System (FAIS) ist der japanische Vorschlag für die Echtzeitkommunikation in der automatisierten Fertigung /60/. Seit 1987 arbeitet das "International Robotics and Factory Automation Center" (IROFA, Tokio) mit finanzieller Unterstützung von Regierung und Industrie an der Architektur eines Kommunikationssystems, das innerhalb flexibler Produktionszellen (*Cell Level*) Verwendung finden soll. FAIS hat die in Bild 3.22 gezeigte Schichtenstruktur, die prinzipiell Mini-MAP entspricht. Sie unterscheidet sich davon in folgenden Punkten:

- Die Bitübertragung ist nicht in Koaxial-Breitbandtechnik, sondern mit Vernetzung über Lichtwellenleiter realisiert.

- Multi- und Broadcast-Nachrichten sind nicht vorgesehen, da nur LLC-Type-3-Dienste verwendet werden.
- Stationen, die sich nicht im logischen Ring befinden, sind nicht zugelassen. Demzufolge wird auf Anwendungsschicht kein *Obtain-Reply*-Dienst benötigt.
- ACSE-Anwendungsverbindungen werden nachgebildet mit Hilfe einer "*Association Control Machine*" (ACM).
- MAP-2 ist die einzige Implementierungsklasse. Sie beinhaltet folgende *Service-Conformance-Building-Blocks*:  
*Initiate, Conclude, Cancel, Read, Write, Status, Identify, GetNameList* und *GetVariableAccessAttributes*.  
 Die Variablen sind über Namen (*VNAM*) oder Adressen (*VADR*) ansprechbar, Felder oder Strukturen sind nicht zugelassen (*NEST = 0*).

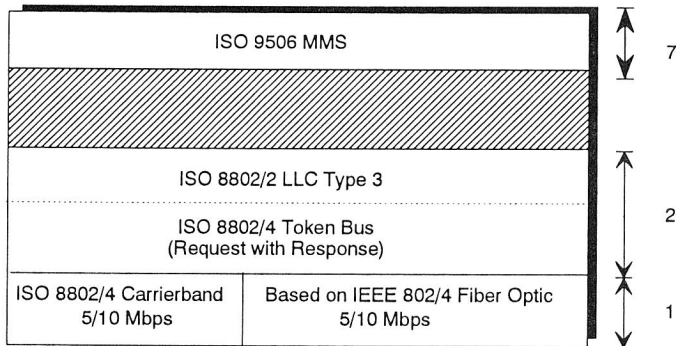


Bild 3.22: FAIS-Architektur

Aufgrund der starken Anlehnung an Mini-MAP gelten für das FAIS-Konzept die gleichen Kritikpunkte (s. Abschnitt 3.5.1). Die FAIS-Projektziele (Entwicklung von Werkzeugen für Konformitätstests, Demonstration beim "FAIS Networking Event" 1992 in Tokio) lassen die erste reale Umsetzung eines "Mini-MAP-Verwandten" erwarten /61/.



### 3.5.3 Time-Critical-Communication-Architecture

Im Unterschied zu den zuvor beschriebenen Lösungsvorschlägen für Echtzeitkommunikation werden bei "Time Critical Communication Architecture" (TCCA) **keine Protokollarchitekturen oder Protokolle** für Echtzeitanwendungen vorgeschlagen, sondern lediglich **Anforderungen an** bzw. **Empfehlungen für** Netzwerke mit gemischt zeitkritischer und nicht zeitkritischer Kommunikation bei der Stückgutproduktion aufgezeigt. Seit 1989 arbeitet an diesem Thema ein Ausschuß der internationalen Organisation für Standardisierung ("TCCA Rapporteurs Group of ISO/TC 184/SC 5/WG 2"), der von weiteren Gremien wie der "European MAP Users Group" (EMUG) und der "National Electrical Manufacturers Association" (NEMA) unterstützt wird.

Innerhalb der CIM-Hierarchie sieht sich TCCA eingeordnet zwischen "Factory Communication Networks" mit MAP-Architektur auf den höheren Ebenen und Sensor- oder Feldbussen auf den unteren Ebenen.

TCCA definiert wie folgt /62/:

"[...] communication is time-critical if the application process sending a message requires it to be received (or received and acted upon, or received and acted upon and confirmed) within a certain time after it submits its 'send' request to the system."

"Time-critical communication architecture" (TCCA) kommt demnach den Anforderungen zeitkritischer Kommunikation nach, "time-critical communication entity" ist eine Einheit, die an zeitkritischer Kommunikation teilhat.

Als wesentliches Merkmal wird angesehen, daß zeitkritische Anwendungen "*event driven*" sind, d.h. daß Kommunikationsvorgänge applikationsabhängig und nicht vorhersehbar sind.

TCCA führt das Konzept der **zeitlichen** und **räumliche Kohärenz** ein. Unter zeitlicher Kohärenz ("*temporal coherence*") wird die Eigenschaft einer Variablenliste verstanden, die angibt, daß ein Variablenwert innerhalb eines Zeitintervalls erzeugt oder übermittelt und/oder empfangen wurde. Dafür wird eine Realisierung durch absolute oder relative

Zeitstempel vorgeschlagen. Durch die Angabe solcher Zeitstempel ist es einer TCCE möglich, die bis zum Empfang einer Nachricht verstrichene Zeit zu errechnen. Somit kann der Benutzer die Leistungsfähigkeit für die Nachrichtenübermittlung beurteilen. Das Konzept der räumlichen Kohärenz ("*spatial coherence*") bezieht sich auf die Gültigkeit redundanter Daten in einem verteilten Steuerungssystem.

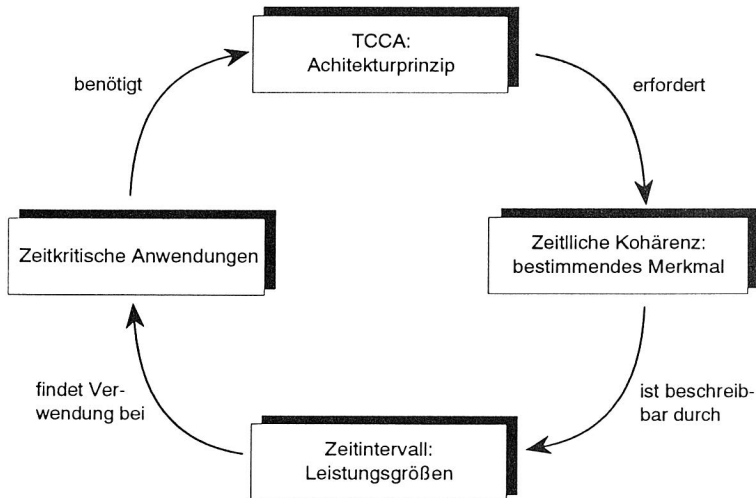


Bild 3.23: Stellung von TCCA

Allen Zeit-Betrachtungen wird bei TCCA die Existenz eines Zeitintervalls ("*time window*") zugrundegelegt, innerhalb dessen eine oder mehrere Aktionen mit vorgegebener Wahrscheinlichkeit abgeschlossen sein sollen. Es werden keine absoluten Angaben zur Länge des Intervalls gemacht, da diese anwendungsabhängig ist. Bild 3.23 zeigt den Zusammenhang zwischen TCCA, zeitlicher Kohärenz, Zeitintervall und zeitkritischen Anwendungen.

Folgende Gesichtspunkte werden für zeitkritische Kommunikationsarchitekturen für maßgebend erachtet:

### Integration:

TCCA orientiert sich am ISO/OSI-Modell, der "Full-7-Layer-OSI-Stack" wird nicht ausgeschlossen, allerdings erscheint eine Schichtenreduzierung geeigneter. Eine Untermenge von MMS-Diensten mit standardisierter Programmierschnittstelle soll verwendet werden. Bei zeitkritischen Nachrichten wird keine Segmentierung zugelassen und eine explizite Flußkontrolle vorgeschlagen. Ferner wird auf die Bildung von Konformitätsklassen bzgl. des Erfüllens der zeitkritischen Anforderungen hingewiesen.

### Differenzierung von PDUs:

Eine Beschränkung auf die zwei Typen **zeitkritische** und **nicht zeitkritische** Nachrichten wird favorisiert. Allerdings wird auch zyklischer-zeitkritischer, azyklischer-zeitkritischer und nicht-zeitkritischer Datentransfer erwähnt. Es wird eine Unterscheidung mit Prioritäten vorgeschlagen.

### Zeitaspekte:

Es wird die Abschätzbarkeit von Nachrichtenübertragungszeiten, Wahrscheinlichkeiten für die fehlerfreie Übertragung zeitkritischer Nachrichten und Rekonfigurationszeiten gefordert.

Dazu gehören die Betrachtungen bzgl. der zeitlichen und räumlichen Kohärenz.

### Fehlerbehandlung und Verfügbarkeit:

Die Fehlerbehandlung soll vom Benutzer (z.B. durch Nachrichtenkategorien "Bestätigung mit x-maliger Sendewiederholung") steuerbar sein.

Es werden Monitormechanismen zur Entdeckung latenter Fehler vorgesehen, damit die Beeinträchtigung des zeitkritischen Nachrichtenverkehrs vermieden werden kann.

Es wird auf allgemein gültige Maßnahmen der Verfügbarkeitssteigerung durch "perfektionistischen" Ansatz und/oder Fehlertoleranz hingewiesen.

### Topologie:

Teilnetzübergreifende Kommunikation soll unter Einhaltung der zeitlichen

Restriktionen möglich sein.

Netzwerkmanagement und übergeordnete Steuerung:

Aufgrund der hohen Verfügbarkeitsanforderungen soll das Netzwerkmanagement besondere Fehlererkennungsmechanismen haben. Zur Einhaltung der strengen zeitlichen Restriktionen wird eine übergreifende Kontrollinstanz für notwendig erachtet, die oberhalb der ISO/OSI-Schichten angesiedelt ist und die Identifizierung und Konfiguration von TCCEs vornimmt.

TCCA beinhaltet ferner Vorschläge für Metriken und Benchmarks, mit deren Hilfe Kommunikationssysteme auf ihre Eignung für zeitkritische Anwendungen hin untersucht werden können.

An TCCA wird folgendes kritisiert:

- Die Nachrichtenklassifizierung ist nicht eindeutig (2, 3 oder mehr Klassen). Die Implementierungsvorschläge für Prioritäten sind unvollständig (Puffer, virtuelle Kanäle, PDU-Attribute): Grundsätzlich besteht auch die Möglichkeit, spezielle Dienste für zeitkritische Kommunikation einzuführen.
- Hinweise auf eine Realisierung von TCCA erfolgen sporadisch. Nicht angegeben wird, welche ISO-Protokollarchitektur in Frage kommt und wie die einzelnen ISO-Schicht-Protokolle modifiziert werden müssen.
- Die Festlegung von Kommunikationseigenschaften **außerhalb** der Kommunikationsarchitektur (s. "Netzwerkmanagement") erscheint problematisch.
- Die Eignung von Kommunikationssystemen wird (gemäß Aufgabenstellung) "nur" hinsichtlich der Echtzeitfähigkeit bewertet.

### 3.5.4 Real-Time-Manufacturing-Message-Specification

Forschungsarbeiten an der "University of Wales" in Swansea (Großbritannien) beschäftigen sich mit der Erweiterung des MAP-Anwendungsprotokolls für die Kommunikation in verteilten Echtzeitsystemen. Die bisher erzielten Forschungsergebnisse /63, 64/ sollen nicht zu einem MMS-Companion-Standard führen, sondern sollen *MMS-Core um Eigenschaften der Echtzeitkommunikation* erweitern.

Für die Gestaltung von Echtzeitsystemen sollten generell folgende Prinzipien berücksichtigt werden:

- Autonomie der Kommunikationsteilnehmer,
- Gültigkeitszeiträume für die übermittelten Daten,
- globale Zeitinformationen
- garantierte Übertragungszeiten beim Nachrichtenaustausch zwischen **Anwendungsprozessen**,
- Datagramm-Dienste und Broadcast,
- zustandsgesteuerte ("*state driven*") im Gegensatz zu ereignisgesteuerten ("*event driven*") Prozessen.

Manufacturing-Message-Specification ist nach Meinung der Autoren für Echtzeitanwendung im Hinblick auf die verwendeten Kommunikationsmechanismen und Objekte nicht geeignet.

#### Kommunikationsmechanismen:

- MMS basiert auf verbindungsorientierten Kommunikationsdiensten. Aufgrund der aufwendigen Protokollabwicklung ist der Zeit- und Ressourcenverbrauch unzumutbar hoch.
- Broadcast-Mechanismen fehlen.

#### Objekte:

- Die "Client-Server-Beziehung" wird bestimmt von der Anordnung des *Virtual-*

*Manufacturing-Device* im *MMS-Server*. Diese Rollenverteilung nach "Master-Slave-Prinzip" widerspricht der Forderung nach autonomen Komponenten.

- MMS-Objekte haben keinen Zeitbezug.

Den im folgenden erläuterten MMS-Erweiterungen (Erweiterungen des *VMD*-Objekts, neue Objekte, neue Dienste) liegt eine fundamentale Annahme zugrunde:

Alle Kommunikationsteilnehmer verfügen über **globale Zeitinformation**. Hinsichtlich der Zeitsynchronisation wird auf bestehende Konzepte und realisierte Lösungen verwiesen /65/.

#### Erweiterungen des VMD-Objekts:

Die Definition in *MMS-Core* wird für das *Real-Time-Virtual-Manufacturing-Device* wie folgt erweitert.

*Object: RTVMD*

*Key Attribute: Executive Function*

*Attribute: Vendor Name*

. (Attribute analog zu *MMS-Core*)

*Attribute: List of Upload State Machines (USLM)*

*Attribute: Lists of Other VMD-specific Objects*

***Attribute: Real-Time Support (TRUE, FALSE)***

***Constraint: Real-time Support-Supported***

***Attribute: Global Time***

***Attribute: List of Configurations***

***Attribute: List of State Variables***

***Attribute: List of State Variable Lists***

*Attribute: Additional Detail*

Die globale Zeitinformation ist als Attributwert im *RTVMD* verfügbar. Als neue Objekte werden Zustandsvariable (*State-Variable*), Konfigurationsobjekte (*Configuration*) und Zustandsvariablen-Listen ("*State-Variable-List*") eingeführt.

Neue Objekte:

- **Zustandsvariable** unterscheiden sich von anderen Variablen dadurch, daß sie mit Zeitinformation versehen sind, woraus der Empfänger (einer *Read.Confirmation*-Protokolldateneinheit) auf den Zeitpunkt der Belegung und den Gültigkeitszeitraum des Variablenwertes schließen kann. Zustandswerte sollen über Broadcast-Dienste an alle Teilnehmer verschickt werden.
- Mit **Konfigurationsobjekten** sollen andere Teilnehmer in ihrer Aktivität gesteuert werden. Es können Zeitpunkte für das Starten und Beenden eines Programmes sowie zyklisches oder azyklisches Verhalten bei einem Teilnehmer bestimmt werden.
- **Zustandsvariablen-Listen** geben an, welche Zustandsvariablen vom Teilnehmer über das Netz erwartet werden.

Neue Dienste:

Für die Behandlung von Zustandsvariablen und Konfigurationsobjekten braucht man Dienste zum Schaffen, Löschen, Modifizieren und für Attribut-Anfragen. Zur Verteilung der Zustandsinformation wird ein Broadcast-Dienst vorgesehen.

Mit einem kommerziell verfügbaren MAP-Board wurden als erste Implementierungsergebnisse für das Lesen einer Zustandsvariable ca. 70 Millisekunden erzielt.

Real-Time-MMS (RTMMS) wird wie folgt beurteilt:

- Es handelt sich um den ersten bekannten Vorschlag für ein Anwendungsprotokoll mit Echtzeiteigenschaften. Bei den Protokolleigenschaften für Echtzeitkommunikation werden in der Fachwelt hauptsächlich deterministische Medienzugangsverfahren behandelt /66/.
- RTMMS ist eine "Echtzeit-Alternative" zu MMS-Core und prinzipiell unabhängig von den unterlagerten Protokollschichten.
- Als "Real-Time-OSI-Stack" wird eine Architektur analog zu Mini-MAP vorgeschlagen. RTMMS auf Anwendungsebene kann (im Unterschied zum verbindungsorientierten MMS-Core) für Broadcast-Aufrufe die unbestätigten

Schicht-2-Dienste ("LLC Type 1") nützen.

- Durch die Einbeziehung von Zeitinformation bietet RTMMS noch kein **Echtzeit**-, sondern nur **zeitbehaftetes Verhalten**. Die Attributierung der Objekte mit Zeitstempeln hat aber den Vorteil, daß die Gültigkeit und Relevanz von Variablenwerten im Zeitzusammenhang überprüft werden können.



## 4 Kommunikationskonzepte für die Anwendung in flexiblen Produktionszellen

### 4.1 Zielsetzung

Ziel des vorliegenden Abschnittes ist es, signifikante Eigenschaften für die Kommunikation innerhalb flexibler Produktionszellen herauszuarbeiten. Die erstellte Merkmalsammlung wird als "Klassifikationsschema" bezeichnet, da aufgrund der Erfüllung, Nicht-Erfüllung oder teilweisen Erfüllung der genannten Eigenschaften Klassen von Automatisierungsgeräten bzw. von Vertretern für zelleninterne Kommunikationssysteme gebildet werden können. Antretende "Kandidaten" müssen sich einer Befragung gemäß Abschnitt 4.2 stellen. Die Relevanz der gefundenen Fragen stützt sich einerseits auf die ausgewiesene Literatur und andererseits auf eigene, in der Lehrstuhl-Modellfabrik gewonnene Erfahrungen (s. Bild 4.1).

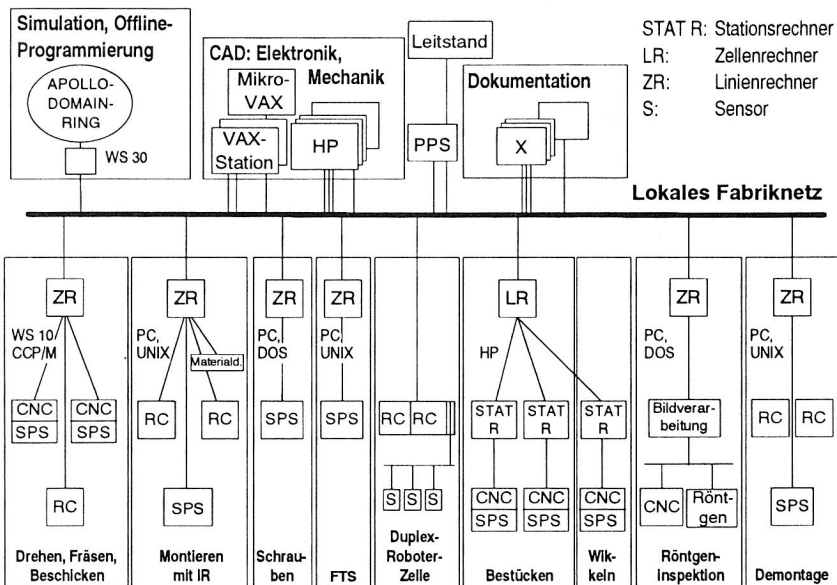


Bild 4.1: Anordnung flexibler Produktionszellen in der Modellfabrik

## **4.2 Fragen bzgl. der Kommunikationsfähigkeit von Automatisierungsgeräten**

Im folgenden werden zunächst alle Fragen bzw. Fragenkomplexe aufgelistet und im Anschluß daran unter dem jeweiligen Stichwort im einzelnen erläutert.

### **Offene Kommunikation:**

Kann das Automatisierungsgerät offen kommunizieren, d.h. werden internationale Standards verwendet?

### **Stationstypen:**

Sind unterschiedliche Stationstypen formal festgelegt? Welche Rolle spielen diese?  
Können einfache Automatisierungsgeräte integriert werden? Durch welche Maßnahmen wird der Kommunikationsaufwand bei diesen vermindert?

### **Datentransfermodi:**

Welche Datentransfermodi sind vorgesehen:

Gibt es einen zyklischen Abfragebetrieb?

Kann das Automatisierungsgerät auch spontane Meldungen absetzen? Können mit einer Nachricht mehrere/alle Teilnehmer erreicht werden?

### **Dynamische Konfigurationsänderung:**

Wie flexibel wird auf Konfigurationsänderungen reagiert? Sind neue Teilnehmer dynamisch integrierbar?

### **Teilnetz-Bildung:**

Können Geräte aus unterschiedlichen Teilnetzen kommunizieren?

### **Echtzeit:**

Kann in Echtzeit kommuniziert werden? Durch welche Methoden wird dies erreicht?

**Leistungsbewertung:**

Welche Durchsatzraten bzw. Übertragungszeiten können bei der Datenübertragung erzielt werden? Welche Leistungsangaben können insbesondere für Anwendungsdienste gemacht werden?

**Alarmbehandlung:**

Wie ist das Kommunikationsverhalten des Automatisierungsgerätes bei Alarmsituationen?

**Fehlertoleranz:**

Durch welche Fehlertoleranzmaßnahmen kann die Zuverlässigkeit der Kommunikation erhöht werden?

**Netzdiagnose:**

Welche Möglichkeiten gibt es, Kommunikationsfehler zu diagnostizieren?

**Projektierungshilfen:**

Welche Unterstützung hinsichtlich der Projektierung und Wartung stehen zur Verfügung?

**Anmerkung:**

Einige der oben aufgeführten Aspekte können den im ISO-**Netzmanagement** aufgenommenen Funktionalitätsbereichen /67/ zugeordnet werden: z.B. "Projektierungshilfen" und "dynamische Konfigurationsänderung" zum "*Configuration and Name Management*", "Fehlertoleranzmaßnahmen" und "Netzdiagnose" dem "*Fault Management*" und "Leistungsbewertung" dem "*Performance Management*". Für die vorliegende Betrachtung spielt es allerdings keine Rolle, inwieweit die genannten Eigenschaften durch das Netzmanagement oder die Instanzen der Kommunikationsprotokolle erfüllt werden, zumal in jüngster Zeit festgelegte Netzmanagement-Aufgaben, wie z.B. Erfassen und Melden von Fehlern, Sammeln von Statusinformationen, vorher bereits von den einzelnen Protokollschichten übernommen wurden.

#### 4.2.1 Offene Kommunikation

Frage: KANN DAS AUTOMATISIERUNGSGERÄT OFFEN KOMMUNIZIEREN, D.H. WERDEN INTERNATIONALE STANDARDS VERWENDET?

Bild 4.2 zeigt die typische Steuerungsstruktur von Montagezellen. Gemäß dem **zentralistischen** Prinzip ist die Zellensteuerung über serielle Punkt-zu-Punkt-Verbindungen mit den Geräten (RC, SPS, Identifikationssystem) verknüpft. Ein spezieller "Kommunikationsprozessor" muß die u.U. unterschiedlichen Übertragungstechniken (V.24, 20mA) beherrschen. Die dargestellte Problematik läßt sich am Beispiel der einer lehrstuhleigenen Drehzelle konkretisieren (s. Bild 4.3): Zwar ist es gelungen, die Drehbearbeitung durch den Verbund der Gerätesteueringen zu automatisieren, die Kommunikation zwischen den Komponenten wird aber aufgrund der verschiedenen, z.T. herstellerspezifischen Übertragungsprozeduren (DK 3964 R, LSV2) und dem Nebeneinander von serieller Übertragung und Binärsignalen recht kompliziert. Besondere Nachteile der gezeigten Kommunikationsstrukturen sind, daß

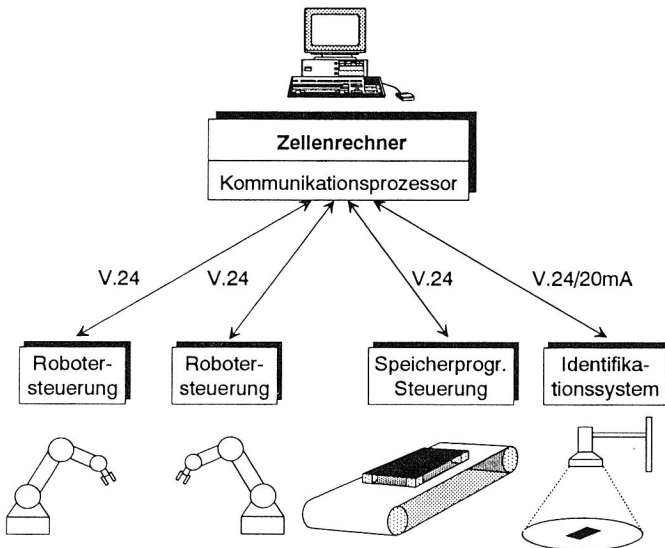


Bild 4.2: Steuerungsstruktur von Montagezellen

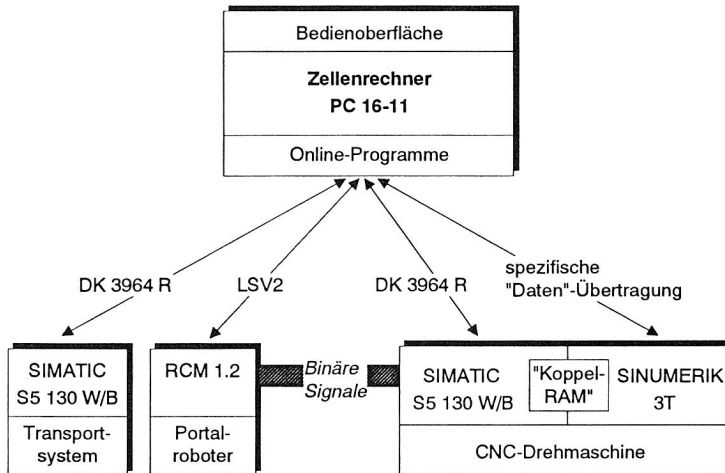


Bild 4.3: Schnittstellenproblematik am Beispiel einer Drehzelle

- der Zellenrechner über die notwendige Hard- und Software-Ausstattung für **alle** Kommunikationsanbindungen verfügen muß,
- die Gerätesteuern sich **nicht direkt** miteinander verständigen können (z.B. für Sensor-/Roboter-Anwendungen), und indirekte Kommunikation nur durch zusätzlichen (Software-)Aufwand im Zellenrechner ermöglicht werden kann.

Zu der dargestellten Situation hat einerseits die Typenvielfalt bei den Automatisierungsgeräten - bedingt durch die Hersteller -, und andererseits die Vielzahl der Schnittstellen pro Gerät geführt.

Im folgenden seien die für Robotersteuerungen typischen Kommunikationsschnittstellen angeführt.

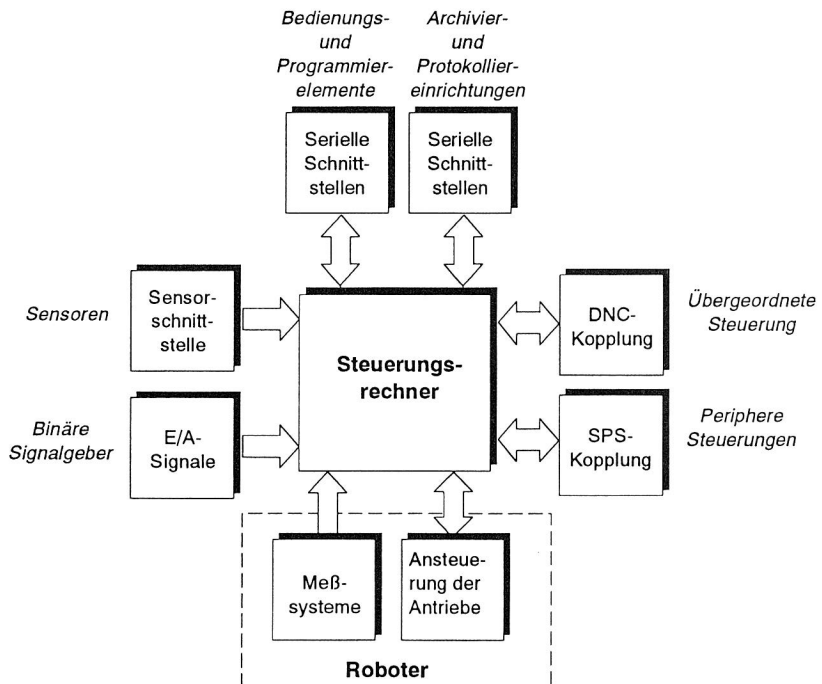


Bild 4.4: Kommunikationsumgebung einer Robotersteuerung /68/

- Als Schnittstellen zu Bedien-, Programmier- und sonstigen peripheren Einrichtungen kommen z.B. zum Einsatz:
  - 20-mA-Schnittstelle für ein Bedienungsterminal (z.B. Teletype, Bildschirmterminal usw.),
  - RS-422 für ein Handbediengerät,
  - V.24-Schnittstellen für Drucker und Floppy-Disk-Schnittstellen.
- Ferner besteht die Möglichkeit
  - zur Kopplung zu einer speicherprogrammierbaren Steuerung über Dual-Port-RAM,
  - zum Anschluß an ein Fertigungsleitsystem über einen herstellerspezifischen Anlagenbus.

- Die Sensordateneingabe kann erfolgen über
  - binäre Eingänge zur Abfrage und logischen Verknüpfung durch die Bewegungsprogramme,
  - analoge (+/- 10V-)Schnittstellen für analoge Sensorsignale, z.B. bei taktilen Sensoren,
  - V.24-Schnittstellen zur Beeinflussung des Roboter-Bewegungsablaufes. /68/

Bild 4.4 zeigt, welche unterschiedlichen Automatisierungseinheiten mit einer Robotersteuerung kommunizieren.

Um eine "Quasi-Offenheit zu erreichen, entstanden zumeist spezielle Kopplungen zu Fremdsystemen, wodurch der "Wildwuchs" allerdings noch vermehrt wurde.

Somit zeigt sich, daß "offene" Kommunikation im Sinne des einheitlichen, standardisierten Informationsaustausches bisher innerhalb flexibler Produktionszellen noch nicht gegeben ist. Auch wenn die neuen Feldbussysteme derzeit viel von sich reden machen, gehören sie noch nicht zur industriellen Praxis.

Kommunikationsanschlüssen für Automatisierungsgeräte sollten sich an der Schichtenstruktur des ISO/OSI-Referenzmodells orientieren. Prinzipiell können Schichten, deren Funktionalität nicht gebraucht wird, weggelassen werden, auf Netzwerk- und Transportfunktionen kann aber wegen der Möglichkeit der Teilnetzbildung, Segmentierung von Nachrichten usw. nicht verzichtet werden (s. auch Schlagwort "Teilnetzbildung"). Die Transparenz für den Benutzer, der u.U. gleiche Anwendungen mit unterschiedlichen Übertragungs- oder Transportsystemen realisieren will, wird durch standardisierte Kommunikation auf Anwendungsebene erreicht. Es ist deshalb notwendig, Automatisierungsgeräte mit MMS-Funktionen und evtl. zugehörigen Companion-Standards auszustatten (s. auch /69/).

#### 4.2.2 Stationstypen

Frage: SIND UNTERSCHIEDLICHE STATIONSTYPEN FORMAL FESTGELEGT? WELCHE ROLLE SPIELTEN DIESE?  
KÖNNEN EINFACHE AUTOMATISIERUNGSGERÄTE INTEGRIERT WERDEN? DURCH WELCHE MAßNAHMEN WIRD DER KOMMUNIKATIONSAUFWAND BEI DIESEN VERMINDERT?

Die in flexiblen Produktionszellen eingesetzten Geräte unterscheiden sich hinsichtlich ihrer Leistungsfähigkeit stark voneinander: Für umfangreiche Automatisierungsaufgaben (z.B. zentrale Zellensteuerung) werden häufig speicherprogrammierbare Steuerungen der gehobenen Leistungsklasse eingesetzt. Ihre Fähigkeiten entsprechen annähernd denen von Prozeßrechnern mit zahlreichen logischen, arithmetischen und sonstigen Operationen für die Informationsverarbeitung /70/. Demgegenüber dominieren zur Erfassung von Prozeßgrößen in Montageanlagen einfache Geräte: Sensoren mit schaltender Funktion. Aufgaben wie Anwesenheitskontrolle, Orientierungsbestimmung usw. werden dort meist durch Taster, induktive Näherungsschalter oder Lichtschranken gelöst /71, 72/.

Für die Gesamtleistungsfähigkeit eines Automatisierungsgerätes ist neben seiner Verarbeitungsleistung maßgebend, ob und in welchem Umfang es mit der Außenwelt kommunizieren kann. Beide Kenngrößen (Verarbeitungsleistung, Kommunikationsfähigkeit) sind voneinander unabhängig und beim jeweiligen Gerät unterschiedlich stark ausgeprägt: z.B. arbeiten Robotersteuerungen komplizierte Algorithmen für Koordinatentransformation oder Interpolation ab, kommunizieren aber über analoge oder binäre Ein-/Ausgänge mit Peripheriegeräten bzw. über serielle Schnittstellen mit dem Zellenrechner.

Der Grad der Kommunikationsfähigkeit bei den Gerätetypen sollte sich auf das Kommunikationskonzept auswirken, indem verschiedene Stationstypen formal festgelegt werden. Um die Vielfalt sinnvoll zu beschränken, erscheinen 2 Typen ausreichend: "kommunikationsreiche" und "kommunikationsarme" Stationen. Die



Umsetzung einer 2-Klassenstruktur kann z.B. beim Netzzugang ("aktive", "passive" Teilnehmer) oder auf Anwendungsebene ("Client"/"Server") erfolgen.

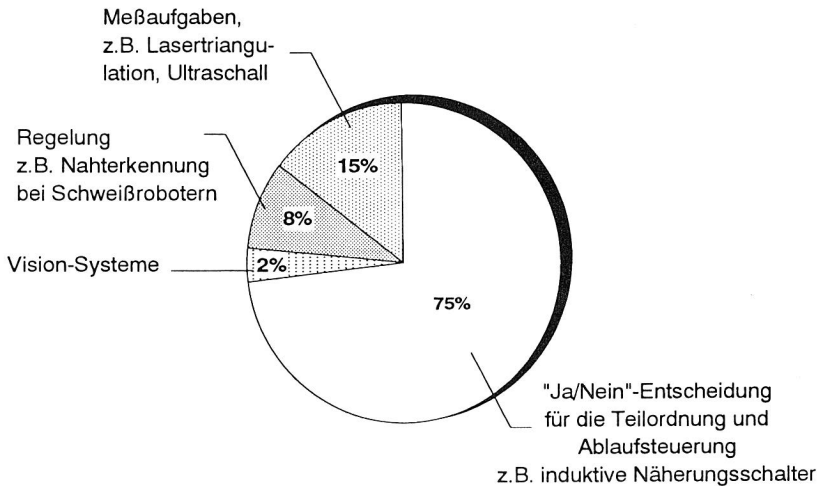


Bild 4.5: Einsatz von Sensoren für Roboteranwendungen /73/

Im Rahmen dieser Arbeit wird vor allem die Integration von einfachen Geräten für wichtig erachtet. Laut /74/ bewältigen Schalter und Näherungssensoren für das Bestimmen der Teileanzahl sowie analoge und digitale Weg-Meßsysteme für Längen- und Positionsmessung die Hälfte aller Meßaufgaben in der Produktionstechnik. In Zusammenarbeit mit Industrierobotern werden für die Aufrechterhaltung der Teileordnung und die Ablaufsteuerung zu 75% einfache Sensoren mit JA-/NEIN-Ausgang eingesetzt /73/ (s. Bild 4.5). Das Problem besteht beim derzeitigen Entwicklungsstand im ungünstigen Preisverhältnis, das die Kosten für Kommunikationsfunktionen denen für die eigentlichen Anwendungsfunktionen gegenüberstellt. Statistische Erhebungen aus /75/ zeigen, daß die Anzahl von eingesetzten Sensoren in Montagezellen recht hoch ist bei gleichzeitig geringem Kosteneinsatz (s. Bild 4.6). Z.B. kann man induktive Näherungsschalter (BEROs) ab einem Preis von DM 60.- erwerben, wohingegen eine MAP-Anschaltung für eine Gerätesteuerung (z.B. SIMATIC S5) heute ca. DM 8000.-

kostet. Es gilt also den Kommunikationsaufwand zu reduzieren. Dies geschieht durch folgende Maßnahmen:

- Weglassen der Funktionalität von ISO/OSI-Schichten (z.B. Sitzungsschicht, Darstellungsschicht).
- Beschränkung auf bestimmte Dienste innerhalb aller Ebenen. Besonders auf Anwendungsebene werden u.U. nur wenige Funktionen benötigt (z.B. MMS-Read und -Write zum Lesen bzw. Schreiben von Variablen).
- Einfache elektrische Anschaltung (EIA RS-485, "twisted pair").

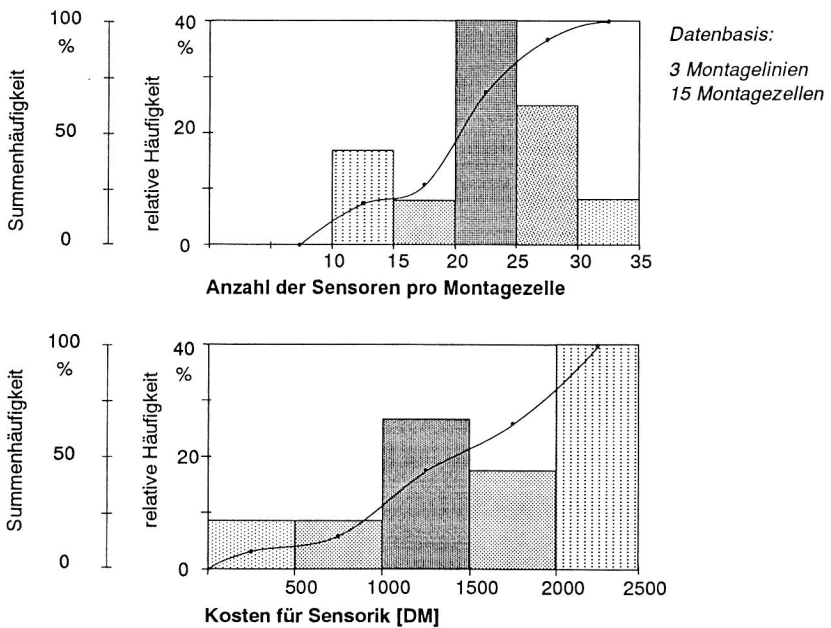


Bild 4.6: Einsatz von Sensoren in Montagezellen /75/

### 4.2.3 Datentransfermodi

Frage: WELCHE DATENTRANSFERMODI SIND VORGESEHEN?  
 GIBT ES EINEN ZYKLISCHEN ABFRAGEBETRIEB?  
 KANN DAS AUTOMATISIERUNGSGERÄT AUCH SPONTANE MELDUNGEN AB-  
 SETZEN? KÖNNEN MIT EINER NACHRICHT MEHRERE/ALLE TEILNEHMER  
 ERREICHT WERDEN?

Für die Automatisierung müssen besondere Formen des Datentransfers in Betracht gezogen werden, die sich vom Informationsaustausch zwischen gleichberechtigten Partnern unterscheiden. Sie sollten auf die Eigenheiten der Anwendung (z.B. kontinuierliche Messung, sporadischer Meldungsaustausch mit dem Zellenrechner u.a.) und auf das gerätespezifische Verhalten abgestimmt sein. Bild 4.7 zeigt eine Auswahl charakteristischer Datentransfermodi.

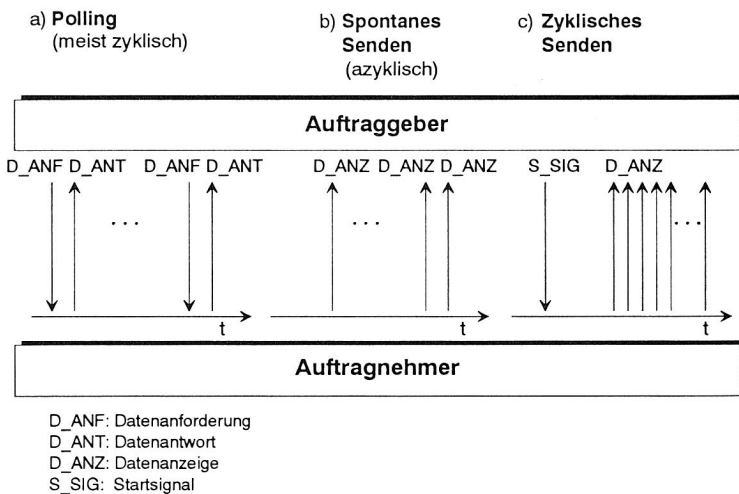


Bild 4.7: Charakteristische Datentransfermodi

- a) Durch **Polling** spricht man meistens einfache Geräte mit rudimentären Kommunikationsfunktionen an. Z.B. werden Kraftsensoren bei der Werkstückkonturaufnahme zyklisch auf die Überschreitung eines Schwellwertes hin abgefragt. (Typische Zykluszeiten liegen zwischen 1 und 10 Millisekunden).
- b) Im Unterschied dazu sollen Geräte auch **spontan** Nachrichten absetzen können ohne vorhergehende Stimulierung durch eine übergeordnete Instanz. Solche Nachrichten sind azyklisch und im Fall von Störungs- oder Alarmmeldungen zeitkritisch. Statusmeldungen, die den Fertigungsfortgang beschreiben (z.B. "FERTIG"-Signal) sind i.allg. nicht zeitkritisch.
- c) Mit **zyklischer Datenübertragung** lassen sich (quasi-)kontinuierliche Meßwertverläufe von Sensoren aufzeichnen. Z.B. müssen in der Halbleiterfertigung (gleichzeitig mehrere) Zustandsvariable (Temperatur, Luftdruck usw.) fortlaufend überwacht werden.

Ferner besteht der Bedarf, eine Nachricht gleichzeitig an mehrere ("Multicast") bzw. alle ("Broadcast") Teilnehmer zu versenden. Dieser Modus wird z.B. in Werkzeugmaschinen, wo mehrere Antriebe koordinierte Bewegungen ansteuern, gebraucht. Die Führungsgrößen bei der Positionsregelung für alle Antriebe können somit in einem Globaltelegramm übermittelt werden.

In einem anderen Fall kann mit Hilfe einer Broadcast-Nachricht eine übergeordnete Station (z.B. der Zellenrechner) im Störfall alle beteiligten Komponenten informieren und evtl. rücksetzen.

#### 4.2.4 Dynamische Konfigurationsänderung

Frage: WIE FLEXIBEL WIRD AUF KONFIGURATIONSÄNDERUNGEN REAGIERT? SIND NEUE TEILNEHMER DYNAMISCH INTEGRIERBAR?

Konfigurationsänderungen müssen in Produktionsanlagen und -zellen aus Flexibilitätsgründen immer möglich sein. Muß zu diesem Zweck die gesamte Anlage stillgelegt

werden, so ist dies mit erhöhten Produktionskosten verbunden. Somit ist es erforderlich, Stationen bei laufendem Betrieb ("dynamisch") integrieren zu können.

Beispielsweise kann sich bei der Qualitätsprüfung von Leiterplatten herausstellen, daß die 2D-Information des Bildverarbeitungssystems nicht ausreicht und durch Hinzunahme eines Abstandssensors zusätzliche Höhenwerte gewonnen werden müssen. Erweiterungen an der "Prüf-Anwendung" wie Abfragen des Abstandssensors, Auswertung der Höheninformation usw. müssen zwar vorgenommen werden, bleiben aber nur auf eine Station begrenzt.

Anmerkung:

Bei ISO/OSI wird das Konzept der dynamischen Anwendungsverbindungen in Ebene 7a durch das ACSE-Protokoll /76/ verwirklicht.

#### 4.2.5 Teilnetzbildung

Frage: KÖNNEN GERÄTE AUS UNTERSCHIEDLICHEN TEILNETZEN KOMMUNIZIEREN?

Auf die Wichtigkeit der Teilnetzbildung wird in der Literatur mehrfach hingewiesen /77, 78/. Folgende Sachverhalte unterstreichen diese Forderung:

- Umfangreiche Fertigungsanlagen können in ihrer **topologischen Ausdehnung** u.U. bis zu einem Quadratkilometer erreichen. Damit werden u.U. die maximal zulässigen Leitungslängen überschritten.
- Eine Strukturierung soll der Überlastung des Netzes bei größerer Teilnehmerzahl (bis zu einigen hundert Steuerungen) und damit **erhöhtem Verkehrsaufkommen** entgegenwirken.
- **Zeitkritische Anwendungen** erfordern, daß die Teilnehmerzahl reduziert wird. Die beteiligten Geräte sollten in einem Teilnetz zusammengefaßt werden.
- Die Bildung von Teilnetzen erhöht die **Verfügbarkeit** des Gesamtsystems. (Bei Ausfall eines Untersystems bleiben die restlichen funktionstüchtig.)

Bild 4.8 zeigt, welche Teilsysteme sich z.B. für eine Montagezelle mit mehreren Robotern und Sensor-geführten Endeffektoren ergeben, wenn eine funktionsorientierte Unterteilung vorgenommen wird, die die besonderen erhöhten Verkehrsbedürfnisse berücksichtigt: Der Robotergreifer für sich beinhaltet bereits eine Reihe von Komponenten (CCD-Kamera, taktile Sensoren, Kraft- und Momenten-Sensoren, Ultraschall-Abstandssensor, Motorsteuerung) zur Erfüllung der Teilaufgabe "Greifen", die sinnvoll zu einem Teilnetz zusammengefaßt werden. Ein geregelter Greifvorgang erfordert intensive Kommunikation zwischen den taktilen Sensoren und der Motorsteuerung: Die Zykluszeit für die serielle Informationsübertragung eines Feldes von 32x32 Matrixpunkten liegt derzeit bei ca. **8 ms** /79/. Teilnetzübergreifende Kommunikation mit der Robotersteuerung tritt auf, wenn auf Grund der Kamera- oder Abstandsinformation der gesamte Roboterarm verfahren wird.

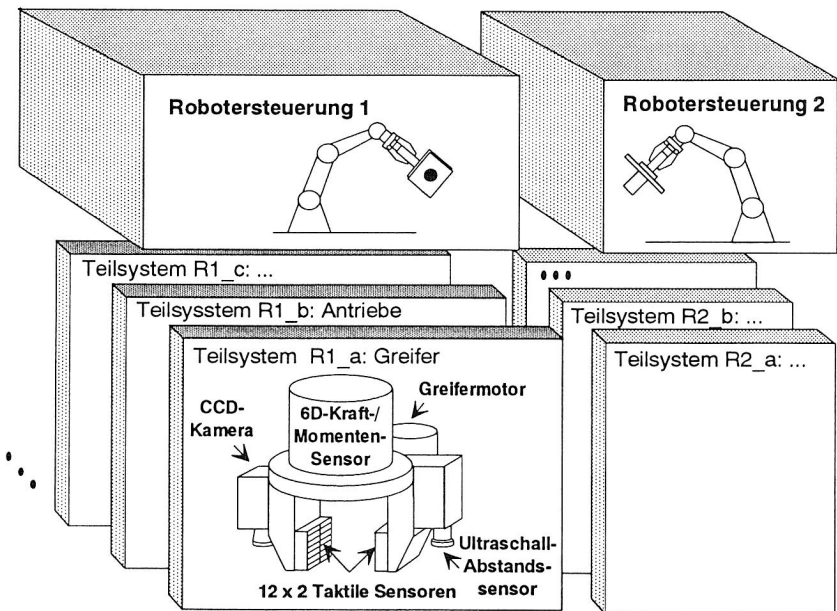


Bild 4.8: Teilsysteme in einer Roboterzelle

Die Vermittlungsschicht hat die Aufgabe, Teilnetze zu organisieren und transparente Netzwerkverbindungen zu schaffen. Sie darf deshalb nicht, wie bei aktuellen Feldbusarbeiten, gestrichen werden.

Durch sog. "Router" ist es zudem möglich, unterschiedliche Übertragungssysteme (ETHERNET, Token-Bus, FDDI usw.) zu koppeln.

#### **4.2.6      Echtzeit**

Frage:            KANN IN ECHTZEIT KOMMUNIZIERT WERDEN? DURCH WELCHE METHODEN WIRD DIES ERREICHT?

Bereits in Abschnitt 2 ("Zugrundeliegende Begriffe") wurde der Begriff "Echtzeit" näher erläutert. Ein Automatisierungsgerät muß demnach innerhalb einer vorgegebenen Zeitspanne ("Fenster") auf eine Nachricht antworten. Prinzipiell ist die Größe des Fensters abhängig von der jeweiligen Automatisierungsaufgabe. Der Begriff "Echtzeitkommunikation" ist zwar im allgemeinen eng an die Vorstellung einer möglichst schnellen Datenübermittlung geknüpft, besagt ab im Kern, daß für vorab festgelegte Kommunikationsvorgänge Ausführungszeiten zuverlässig garantiert werden. Typisches Beispiel ist die sensorgeführte Bahnsteuerung bei Industrierobotern: Für die Bahnführung müssen Sensordaten im Interpolations- bzw. Lageregeltakt zur Verfügung stehen. Für die "Bus-geregelte" Roboterbewegung bedeutet das, daß Antworten innerhalb der Zykluszeit, z.B. 5 ms, gewährleistet sind (s. Bild 4.9).

Um die Echtzeitkommunikation mit Automatisierungsgeräten zu erreichen, kommen prinzipiell folgende Methoden in Frage:

- **Deterministisches Mediumzugriffsverfahren:**  
Es muß gewährleistet sein, daß eine Station innerhalb einer vorgebbaren Zeitspanne die Sendeerlaubnis erhält. Dies gilt für die sog. "Token-Verfahren", zufallsorientierte Zugriffsverfahren (z.B. CSMA/CD: "Carrier Sense Multiple Access with Collision Detection") haben diese Eigenschaft nicht.
- **Prioritäten bei Nachrichten (PDUs) oder Verbindungen.**

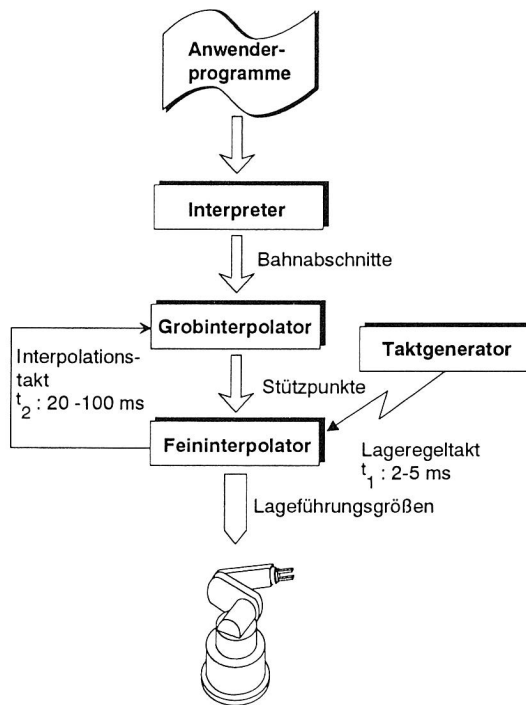


Bild 4.9: Struktur einer Industrierobotersteuerung

- Verbindungslose Dienste (= Datagramm-Dienste) im Unterschied zu verbindungsorientierten Diensten mit Verbindungsaufbau-, Datenübermittlungs- und Verbindungsabbauphase.
- Unbestätigte Dienste mit den Dienstprimitiven *Request* und *Indicaton* (ohne *Response* und *Confirmation*).

Für eine Differenzierung bereits auf Anwendungsebene wäre es wünschenswert, die Nachrichten (PDUs) mit Attributen wie "zeitkritisch"/"nicht zeitkritisch" (2 Prioritätsstufen) versehen zu können. Eine andere Möglichkeit, in (drei) Prioritätsstufen einzuteilen ist:



- zeitkritisch/zyklisch,
- zeitkritisch/azyklisch,
- nicht zeitkritisch.

Um die Segmentierung von Echtzeitinformationen auf den unteren Ebenen zu vermeiden, muß die Länge solcher Nachrichten bereits auf Anwendungsebene begrenzt werden.

Für die Echtzeitkommunikation werden i.allg. reduzierte Architekturen mit Bitübertragungs-, Sicherungs- und Anwendungsschicht vorgeschlagen. Mit dem Ziel, möglichst **schnell** kommunizieren zu können, werden dabei wichtige Funktionalitäten auf Kosten der Zuverlässigkeit (gesicherte Transportübertragung) und Flexibilität (Dynamik der Topologie, des Anwendungskontexts usw.) gestrichen.

#### 4.2.7 Leistungsbewertung

Frage: WELCHE LEISTUNGSANGABEN KÖNNEN GEMACHT WERDEN? KÖNNEN EVTL. KENNGRÖßEN ANGEGEBEN WERDEN?

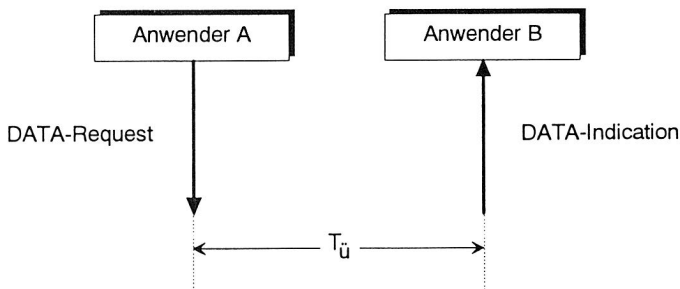


Bild 4.10: Übertragungsdauer

Zur Beurteilung des Kommunikationsverhaltens besonders im prozeßnahen Bereich mit

Echtzeitbedingungen gehören quantitative Angaben hinsichtlich der Kommunikationsgeschwindigkeit. Zu diesem Zweck müssen Metriken entwickelt werden, nach deren Vorgabe im Sinne eines Benchmarks Messungen an realen Systemen vorgenommen werden können. Unter Berücksichtigung netzwerktheoretischer Überlegungen /22, 80/ und leistungsbezogener Aspekte bei den *Quality-of-Service*-Parametern von ISO /29/ werden die Kenngrößen **Übertragungsdauer** und **Durchsatz** als maßgebend angesehen.

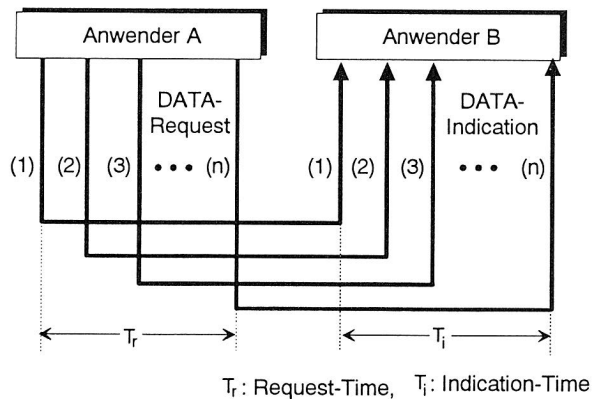


Bild 4.11: Durchsatz

Die **Übertragungsdauer** ( $T_u$ ) ist die Zeit zwischen einer Datenanforderung ("DATA-Request") und einer Datenanzeige ("DATA-Indication") (s. Bild 4.10).

Bei einer Folge von  $n$  Protokolldateneinheiten (PDUs) ist der **Durchsatz** die Anzahl der Daten-Oktette, die mit den letzten  $n-1$  PDUs übermittelt werden, dividiert durch die Zeit, die zwischen dem ersten und dem letzten DATA-Request bzw. -Indication ( $T_r$  bzw.  $T_i$  verstreicht (s. Bild 4.11). (Hinweis: Zwischen dem ersten und dem  $n$ -ten DATA-Request bzw. DATA-Indication liegen nicht " $n$ " sondern " $n-1$ " Übertragungen.)

Die genannten Größen können auf die verschiedenen Ebenen des Ebenenmodells angewendet werden und beziehen sich auf eine unquitierte Datenübermittlung. Für ein Protokoll mit Quittierungsmechanismus ist die **Anzahl bestätigter Dienste pro Zeiteinheit** ein aussagekräftiges Maß. Die Abwicklung eines Dienstes vollzieht sich dabei mittels der Dienstprimitive *Request*, *Indication*, *Response* und *Confirmation* und

benötigt prinzipiell **zwei** Übertragungen. Bild 4.12 zeigt den Ablauf für den *Read*-Dienst aus der MMS-Dienstgruppe *Variable-Access*.

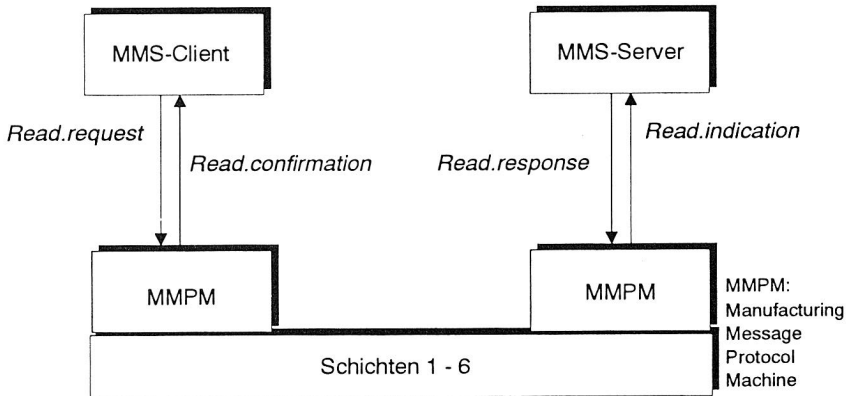


Bild 4.12: Bestätigter Anwendungsdienst

#### 4.2.8 Alarmbehandlung

Frage: WIE IST DAS KOMMUNIKATIONSVERHALTEN DES AUTOMATISIERUNGSGERÄTES BEI ALARMSITUATIONEN?

Da in der Automatisierungstechnik die Begriffe "Alarm" und "Fehler" oft synonym verwendet werden, seien zunächst folgende Definitionsmöglichkeiten angeführt:

Ein Alarm signalisiert ein schwerwiegendes Problem oder einen unvorhergesehenen Systemzustand, der nur durch externen Eingriff (z.B. Bediener) behoben werden kann /81/.

Aus Sicht numerisch gesteuerter Werkzeugmaschinen heißt es:

"Alarmmeldungen (Alarme) haben die Aufgabe, Störungen der Arbeitsmaschine an den Fertigungsrechner zu melden. Diese Störungen können einerseits zum Stillstand der Arbeitsmaschine führen [...]. Andererseits werden durch Alarmmeldungen aber auch

solche Alarme übertragen, die auf Betriebsereignisse zurückzuführen sind, die den Bearbeitungsprozeß nicht unterbrechen /82/.

Typische Alarmsituationen sind:

- Kollision von im selben Arbeitsraum kooperierenden Robotern,
- Meßbereichsüberschreitung bei Sensoren,
- Anfahren von Achsendschaltern bei Werkzeugmaschinen.

Bei derzeitig eingesetzten Roboter- und Maschinensteuerungen werden Alarme wie folgt behandelt:

In der Mehrzahl der Fälle werden Alarme als Alarmnummern (ganzzählig) in einem Alarmspeicher vermerkt. Bei Anfrage durch die übergeordnete Instanz ("Fertigungsrechner") werden alle anstehenden Alarme mit der Maschinenkennzahl übermittelt ("Sammelmeldung"). Bei einigen Geräten ist vorgesehen, daß die Maschinensteuerung von sich aus Alarme sowie Einzelmeldungen versenden kann.

Die Beherrschung von Alarmen aus kommunikationstechnischer Sicht verdichtet sich zu der Frage:

Wie kann ein Automatisierungsgerät schnell und ggf. unter Umgehung der Regeln für den Normalbetrieb ein unvorhergesehenes Ereignis gezielt melden?

Zu diesem Zweck erscheint insbesondere die Möglichkeit der **aktiven** Alarmgabe im Unterschied zu Pollingmechanismen als unumgänglich. Noch unterhalb der ISO/OSI-Ebeneneinteilung kann dieses Problem elektrisch/mechanisch durch Hinzunahme eines eigenen Aderpaares ("Klingelleitung") gelöst werden (z.B. SRQ-Leitung beim IEC-Bus).

Alarmbehandlung ist eine Form von Echtzeitkommunikation, denn die Dauer der Alarmübermittlung unterliegt Zeitrestriktionen, und somit greifen alle darunter aufgeführten Mechanismen wie Bildung von Prioritätsklassen, wie z.B.:

Priorität 1: Alarme, bei denen die laufende Bearbeitung unterbrochen werden muß,

Priorität 2: Alarme, bei denen die laufende Bearbeitung fortgesetzt werden kann (z.B. "Bedienfeldtemperatur zu hoch").

Aus Anwendersicht ist eine erweiterte Semantik bzgl. der Alarmmeldungen, also detaillierte Information über die Alarmsituation (Alarmbezeichnung, Zeitpunkt des Auftretens usw.) wichtig. Dies unterstützen spezielle Alarmobjekte und -dienste: Das *Event-Management* von MMS ermöglicht es, beliebige Ereignisse und daraufhin auszuführende Aktionen zu definieren. Die in der Fertigungsanlage auftretenden Alarmsituationen sind mit Hilfe der MMS-*Event*-Objekte und -*Event*-Dienste aus Anwendersicht für die Kommunikation beschreibbar.

An Objekten stehen zur Verfügung:

*Event-Condition*: Es beschreibt ein Ereignis, auf das gewartet wird und aufgrund dessen bestimmte Aktionen ausgelöst werden können.

*Event-Enrollment*: Es gibt an, ob beim Eintreten des Ereignisses der *Client* benachrichtigt werden soll. Ferner kann es auf ein *Event-Action*-Objekt verweisen.

*Event-Action*: Es gibt den MMS-Dienst an, der nach Eintreffen des Ereignisses ausgeführt werden soll.

Bild 4.13 zeigt die Alarmbearbeitung für eine Meßbereichsüberschreitung: Im MMS-*Server* für einen Druck- oder Temperatursensor ist als *Event*- (bzw. "Alarm"-) *Condition* eine boolsche Variable "Meßbereichsüberschreitung" definiert. Das Überschreiten des Grenzwertes hat die Änderung des Variablenwertes "FALSCH -> WAHR" zur Folge. Ob im *Server* vor der *Event-Notification* eine Aktion (z.B. Lesen des tatsächlichen Meßwertes) ((3),(4)) ausgelöst werden soll, hängt von den Festlegungen im *Event-Enrollment*-Objekt ab. Andernfalls kann der *Client* unmittelbar alarmiert werden ((2)).

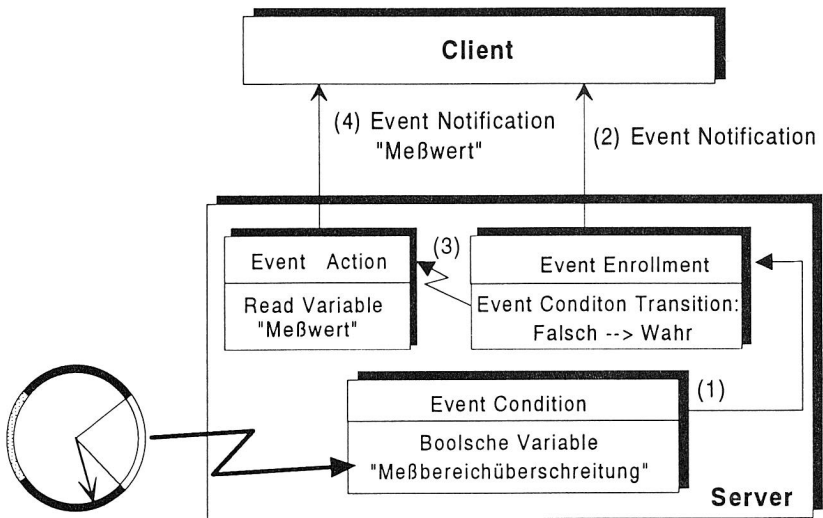


Bild 4.13: Alarmbehandlung bei Meßbereichsüberschreitung

#### 4.2.9 Fehlertoleranz

Frage: DURCH WELCHE FEHLERTOLERANZMAßNAHMEN WERDEN DIE BESONDEREN ZUVERLÄSSIGKEITSANFORDERUNGEN DER FERTIGUNG ERFÜLLT?

Aufgrund des erheblichen Kostendrucks bei hochtechnisierten Systemen sind die Betreiber gezwungen, die Verfügbarkeit ("availability") und Zuverlässigkeit ("reliability") der Anlagen zu verbessern, indem sie einerseits versuchen, das Auftreten von Fehlern zu vermeiden ("perfektionistischer Ansatz"), oder andererseits Fehlertoleranzmaßnahmen zur Schadensbegrenzung ergreifen /83/. Z.B kann für eine Produktionsanlage akzeptiert werden, wenn im Störfall die Geschwindigkeit eines Transportbandes und damit die produzierte Stückzahl zwar erniedrigt, aber ein Gesamtausfall verhindert wird. Man kennt verschiedene Klassifizierungsschemen, nach denen Fehler bzw. Ausfälle z.B. eingeteilt werden in

- "permanente" und "transiente" Fehler oder
- "Bedienfehler", "Entwurfsfehler", "Hardwareausfälle" und "Umgebungsstörungen" /83/ oder
- "node failure", "communication failure", "software failure" und "timing failure" /77/.

Alle wesentlichen Fehlertoleranztechniken basieren darauf, daß **nützliche Redundanz** eingeführt wird in Form von Hardwareredundanz (zusätzliche Bauteile), Softwareredundanz (zusätzliche Programme) und/oder Zeitredundanz (Wiederholung von Operationen) /84/.

Hauptsächlich bei der SPS-Technik haben sich redundante Auslegungen für die Zentraleinheiten, Ein-/Ausgabebaugruppen und Übertragungsleitung auf getrennten Leitungstrassen durchgesetzt. /85/

In vernetzten Automatisierungssystemen sind demnach nicht nur bei den Einzelkomponenten (z.B. Maschinensteuerungen, Transportsteuerungen, Zellen- und Leitrechnern), sondern auch bei dem diese verbindenden Kommunikationssystem Fehlertoleranzmechanismen einzusetzen, wenngleich diese mit Ausnahme von Fehlern in der Übertragungsleitung wiederum von den Einzelkomponenten geleistet werden muß.

Beispiele für permanente Fehler beim Medium sind:

- Kurzschluß zwischen Busleitungen,
- Bruch einer Busleitung,
- defekter Leitungsabschlußwiderstand.

Auf die besonderen Anforderungen in der störempfindlichen Fabrikumgebung wird stets hingewiesen. Messungen von /86/ hinsichtlich der auftretenden elektromagnetischen Interferenzen ("EMI") in Produktionsbetrieben des Schwermaschinenbaus mit Schweißmaschinen, Pressen, Fräsmaschinen usw. ergaben aber, daß für ein Lokales Netz gemäß IEEE-Spezifikation 802.3 die festgestellten Pegel der Spitzenwerte unter 10% der Angaben für Unempfindlichkeit liegen. Die Ergebnisse erlangen aber für "Low Cost"-Netze im Sensor-/Aktorbereich mit z.B. verdrehten Aderpaaren keine All-

gemeingütigkeit.

Im Rahmen des ISO/OSI-Modells werden Maßnahmen der Fehlererkennung und -korrektur insbesondere in der Sicherungsschicht, Transportschicht und Anwendungsschicht durchgeführt. Bild 4.14 führt einige Beispiele auf. Die Vermittlungsschicht

7b	Bestätigte Dienste/ Spezielle Fehlerdienste
7a	Gesicherte Anwendungsverbindungen
4	Quitierte Datenvermittlung Folgenummern Prüfsummen Wiederholtes Senden mit Timeout
2b	Paritätsbildung, Blocknummernprüfung, Polynomiale Codes ("CRC")

Bild 4.14: Beispiele für Fehlerbehandlung im ISO/OSI-Modell

ermöglicht eine Modularisierung in Form von Teilnetzen und somit eine Fehlereingrenzung im Störfall. In /77/ wird ein Teilnetz ("segment") als elementares Element ("basic system") für Fehlertoleranzstrategien bei Automatisierungssystemen angesehen.

#### 4.2.10 Netzdiagnose

Frage: WELCHE MÖGLICHKEITEN GIBT ES, KOMMUNIKATIONSFEHLER ZU DIAGNOSTIZIEREN?

Diagnose- und Überwachungseinrichtungen gehören zu den Forschungsschwerpunkten bei der Automatisierungstechnik. Ziel der Diagnose ist die schnellere und bessere Infor-



mation über Störfälle und ihre Ursachen. Der wirtschaftliche Effekt ergibt sich aus einer größeren störfreien Laufzeit und kürzeren Ausfallzeit bei den verbleibenden Störungen /87/. Neben der Diagnose für Einzelkomponenten von Fertigungssystemen bzw. -zellen ist es erforderlich, auch Fehler beim Kommunikationssystem (bzw. beim Kommunikationsanteil der Komponente) (möglichst frühzeitig) zu erkennen, zu lokalisieren und evtl. auch zu beheben.

Netzdiagnose gehört zum Aufgabenbereich des Netzwerkmanagements, das dafür allgemeine Managementfunktionen beinhaltet wie:

- Gewährleistung eines sicheren Betriebs durch entsprechende Überwachungs- und Steuerungsfunktionen sowie durch vorausschauende Wartung,
- Unterstützung bei Fehlersuche, -diagnose und -korrektur. /88/

Das Netzmanagement bei ISO beinhaltet ein *Fault-Management* /89/ für das Entdecken von Fehlern, die Diagnose der Ursachen und deren Behebung. Als Fehler gilt nicht nur das abnorme Verhalten einer Komponente, sondern auch ein Fehlverhalten, z.B. zu häufiges Ausführen an sich fehlerfreier Aktionen. Das Netzwerkmanagement bei MAP wurde in Anlehnung an ISO festgelegt. Für die einzelnen Teilbereiche (u.a. *Fault-Management*) sind zwar noch keine speziellen Dienstdefinitionen, aber *Managed-Objects* der Ebenen 2 bis 7 definiert, mit denen Management-Anwendungen bei CNMA implementiert wurden.

Zum Einbringen von Informationen über eine konkrete Umgebung in das Netzwerkmanagement wurde von /88/ das Sprachkonzept *FAN (Frame-Analysing-Notation)* entwickelt. Es handelt sich dabei in der ersten Stufe (*FAN.1*) um eine Beschreibungssprache für die protokollorientierte Auswertung von Paketen und war Grundlage für den "LAN PROTOCOL TESTER B5100". Das für IEEE-802.3-Netze einsetzbare Werkzeug zeichnet Datenpakete gemäß vorgegebener Ereignisse auf, die auf mittels *FAN* formulierter Protokolldateneinheiten hin untersucht werden können.

Das in /91/ vorgestellte Monitorsystem für Lokale Netze integriert den Aspekt der Diagnose durch Anbindung an ein Expertensystem.

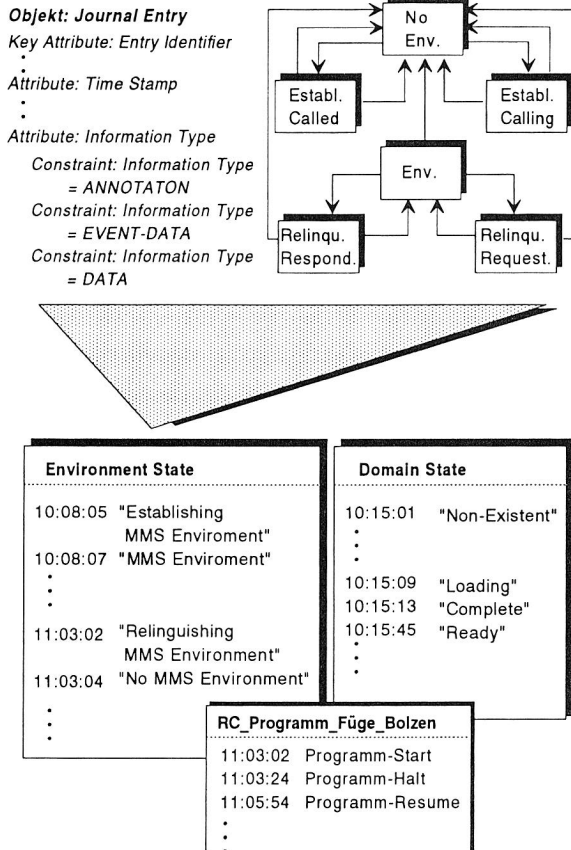


Bild 4.15: Aufzeichnung von Kommunikationsvorgängen mit dem MMS-Objekt "Journal"

Prinzipiell sind alle Diagnosestrategien darauf angelegt, aus Aufzeichnungen über den Fertigungs- bzw. Kommunikationsverlauf vor dem Fehlerfall Rückschlüsse auf mögliche Ursachen ziehen zu können. Bild 4.15 zeigt, welche Mechanismen MMS bereitstellt, um z.B. die - für höhere Protokolle aufschlußreichen - Zustandsübergänge von abstrakten Protokollmaschinen zu protokollieren. In das MMS-Objekt *Journal* können Einträge ("*Journal Entry*") gemacht werden, die u.a. mit einer Bezeichnung ("*Entry Identifier*"), einem Zeitstempel ("*Time Stamp*") und Angaben zum Typ der Eintragung versehen

sind:

- *ANNOTATION* beinhaltet Textangaben,
- *EVENT-DATA* nimmt Bezug auf ein zuvor bestimmtes Ereignis, Variablenwerte können mit übergeben werden,
- *DATA* liefert die Werte von zuvor festgelegten Variablen.

Zustandsübergänge beim abstrakten Automaten "Environment State", der MMS-Verbindungen überwacht, können in ein "Logbuch" notiert werden. Gleiches gilt für abstrakte Automaten bei MMS-Objekten wie *Domains* ("Domain State"), *Program-Invocations* ("Program Invocation State") usw. Auch der Verlauf des Fertigungsprozesses kann festgehalten werden, indem die Aufrufe an die aktuellen Bearbeitungsprogramme gespeichert werden. Man erhält allerdings keine absoluten Zeitangaben, da lediglich der Zeitpunkt für den jeweiligen MMS-Dienstaufruf (*Start*, *Stop*), nicht aber für den tatsächlichen Programmstart vorliegt.

Insgesamt gesehen stehen sich konzeptuelle Arbeiten für Aufgaben und Funktionen des Netzwerkmanagements und praktische Anwendungen von LAN-Monitoringsystemen /90, 92, 93/ gegenüber. Die Netzdiagnose bleibt also weiterhin Gegenstand der Forschung.

#### **4.2.11 Projektierungshilfen**

Frage 11: Welche Unterstützung hinsichtlich der Projektierung und Wartung stehen zur Verfügung?

Für eine benutzerfreundliche Handhabung des Kommunikationssystems sollten Werkzeuge für die Projektierung und Wartung zur Verfügung stehen. Das setzt voraus, daß keine statische Kommunikationsarchitektur vorliegt, sondern Parameter für die Generierung von Profilen offen gelassen sind. Ein *Profil* bestimmt u.a., ob der Teilnehmer als Master oder als Slave ausgeführt wird, welche Übertragungsphysik und Baud-Rate zur Anwendung kommen und welche Dienste und Datentypen unterstützt

werden. In /94/ werden als Beispiele für Anwendungsprofile Überwachungsaufgaben in Fertigungszellen oder Qualitätssicherungsaufgaben angegeben.

Innerhalb von ISO werden Projektierungsaufgaben dem *Configuration-and-Name-Management* zugeordnet. Unter Konfigurationsmanagement versteht man laut /88/ die Definition, die Speicherung und die Verwaltung von Konfigurationsinformationen. Als Konfiguration wird bei ISO nicht nur die Topologie des Netzwerks bezeichnet, sondern auch Informationen über die Netzkomponenten und deren aktuellen Zustand. Beispiele für Funktionen im Rahmen des Konfigurationsmanagements sind das Starten von Netzknoten, die Aktivierung und Deaktivierung von OSI-Ressourcen sowie deren Initialisierung. Das Namens-Management realisiert die Zuordnung von symbolischen Namen zu Adressen sowie Verwaltungsoperationen über diese Objekte.

Neben der Erstellung von Busparametern sollte ein Netzwerkkonfigurator auch Plausibilitätsprüfungen und u.U. eine Optimierung der globalen Parameter vornehmen. Verwaltung und Dokumentation der Konfigurationsdaten leisten vor allem bei Wartungsarbeiten eine große Hilfe.

## **4.3 Exemplarische Anwendung des Klassifikationsschemas**

### **4.3.1 Allgemeine Vorgehensweise**

Der vorliegende Kriterienkatalog dient dazu, Anwarter für zelleninterne Kommunikationssysteme anhand der vorgegebenen Merkmale zu differenzieren. Mit Hilfe des Klassifikationsschemas soll insbesondere auch eine Beurteilung ohne praktisches Arbeiten, also allein anhand der zur Verfügung stehenden Primär- (Normen, Datenblätter usw.) und Sekundärliteratur möglich sein.

Exemplarisch wurden die folgenden drei aktuellen Vertreter im Feldbusbereich herangezogen<sup>3</sup> :

- PROFIBUS /43, 44/,
- PDV-Bus /95/ und
- DIN-Meßbus /96/.

Nach einführenden Bemerkungen zur Entstehungsgeschichte und Einordnung des Feldbusses werden die vorhandenen Kommunikationsmechanismen, die zur Erfüllung der genannten Kriterien geeignet sind, skizziert. Um die drei Kandidaten grob vergleichen zu können, wird ein dreiwertiges Notenschema eingesetzt: "Kriterium erfüllt", "Kriterium teilweise erfüllt" und "Kriterium nicht erfüllt". Abschließend werden die Ergebnisse der Einzeluntersuchungen einander gegenübergestellt.

---

<sup>3</sup> Die Funktionalität von PDV-Bus und DIN-Meßbus läßt sich innerhalb des ISO/OSI-Modells den Schichten 1 und 2 zuordnen. Der PROFIBUS verfügt zusätzlich über Anwendungsfunktionen.

### 4.3.2 Anwendung auf den PROFIBUS

Eine ausführlich Darstellung des PROFIBUS erfolgte bereits in Abschnitt 3.

#### Offene Kommunikation

- An internationalen Normen wird für die Übertragungstechnik EIA RS-485 (entspricht ISO DP 8482) verwendet. *Das Kriterium gilt als teilweise erfüllt.*
- Der PROFIBUS selbst liegt mittlerweile als DIN 19245 Teil 1 und Teil 2 vor, hat aber bisher international noch keine Bedeutung erlangt.
- MAP- und MMS-Kompatibilität ist nicht gegeben. Im Unterschied zur MAP-Anwendungsschicht (MMS und ACSE) müssen beim PROFIBUS Anwendungsverbindungen konfiguriert werden. Ferner ist es nicht wie bei MMS möglich, Objekte beim Kommunikationspartner zu schaffen und zu löschen.

#### Stationstypen

Es werden zwei Stationstypen unterschieden:

*Das Kriterium gilt als erfüllt.*

- Master (*aktiv*) befinden sich im logischen Token-Ring.
- Slaves (*passiv*) dürfen nur nach Aufforderung durch einen Master senden.

(Unter Verwendung eines 8051-Mikrocontrollers wurde eine einfache Sensor-Station als PROFIBUS-Slave realisiert.)

#### Datentransfermodi

Es gibt:

*Das Kriterium gilt als erfüllt.*

- bestätigte und unbestätigte Dienste (auf Schicht 2 und Schicht 7),
- Broad- und Multicast,
- zyklische Dienste in der Schicht 2 zur Kommunikation mit den Slaves (mit Hilfe der *Poll-Liste*).

### Dynamische Konfigurationsänderung

- Kommunikationsbeziehungen zwischen Anwendungsprozessen werden in der *Kommunikationsbeziehungsliste* projiziert. Die darin enthaltenen *Kommunikationsreferenzen* sind gerätespezifisch und werden vom Netzbetreiber festgelegt.
- Mit der *Gap-Aktualisierung* (Aktualisierung der Teilnehmerliste in jeder Station) ist es zwar möglich, Teilnehmer in den logischen Token-Ring aufzunehmen oder davon zu entfernen, aber das Konzept wird bei den Anwendungsverbindungen nicht fortgeführt.

*Das Kriterium gilt als nicht erfüllt.*

### Teilnetz-Bildung

Netzwerkfunktionen fehlen in der PROFIBUS-Schichtenarchitektur.

*Das Kriterium gilt als nicht erfüllt.*

### Echtzeit

- Das Netzzugangsverfahren ist deterministisch.
- Es gibt 2 Dienstprioritäten in der Schicht 2: ("high", "low"). Folgende Grundeinstellung für niederpriore Dienste ist vorgegeben, kann aber auch abgeändert werden:
  - 1) Bearbeitung der *Poll-Liste* (zyklische Dienste),
  - 2) Bearbeitung von niederprioren, azyklischen Diensten,
  - 3) Teilnehmer-Erfassung,
  - 4) Gap-Aktualisierung.
- Die Priorität von Slaves kann durch Mehrfacheintragung in der Poll-Liste erhöht werden.
- Durch die Einstellung der *Token-Soll-Umlaufzeit* ( $T_{TR}$ ) kann die maximale Wartezeit auf das Senden einer Nachricht zwar beeinflusst werden, allerdings ist jedem Teilnehmer auch bei schon abgelaufener  $T_{TR}$  die Abwicklung eines hochprioren Nachrichtenzyklus erlaubt.

*Das Kriterium gilt als teilweise erfüllt.*

- Die vorgestellten Mechanismen zielen zwar auf Echtzeitverhalten ab, die Einhaltung von Antwortzeiten ist aber abhängig von zusätzlichen Voraussetzungen (Teilnehmerkonfiguration, Profil der Kommunikationsanwendung).

#### Leistungsbewertung

Bei der analytischen Bewertung des PROFIBUS-Medienzugangsverfahrens werden in /97/ Angaben zu "Nachrichtenübertragungszeiten" und "Paketankunftszeiten" gemacht. (Veröffentlichungen über Meßergebnisse stehen derzeit für die vorliegende Betrachtung insbesondere für FMS nicht zur Verfügung.)

*Das Kriterium gilt als erfüllt.*

#### Alarmbehandlung

- Alarmmeldungen können mit hoher Priorität übermittelt werden (s. Kriterium "Echtzeit").
- Es können die Dienste des FMS-Event-Managements genutzt werden.

*Das Kriterium gilt als erfüllt.*

#### Fehlertoleranz

- Die Eigenschaften von EIA RS-485 gewährleisten eine hohe Störsicherheit
- Optionale Redundanzen sind vorgesehen:
  - Zweites Medium,
  - zweite Leitstation bei Single-Master-Systemen.
- Die verwendeten UART-Zeichen (gemäß DIN 66022/66203) arbeiten mit einem Paritätsbit.
- Über alle empfangenen Zeichen wird eine Quersumme gebildet und als Prüfbyte (FCS: *Frame-Check-Sequence*) übermittelt.

*Das Kriterium gilt als erfüllt.*



- Es gibt verschiedene Zeitzähler in der Schicht 2:
  - *Idle-Timer*: zur Überwachung des Ruhezustands auf der Leitung; im Fehlerfall werden die Zählerstände neu geladen.
  - *Slot-Timer*: zur Überwachung der Quittierung eines Aufrufs mit Wiederholung im Fehlerfall.
  - *Time-Out-Timer*: zur Überwachung der Busaktivität mit Reinitialisierung des Busses bei Ablauf des Zählers.
  - *GAP-Update-Timer*: zur Überwachung der GAP-Liste mit Reinitialisierung der GAP-Liste nach Ablauf des Zählers.

### Netzdiagnose

- Folgende Funktionen stehen im Rahmen des Netzmanagements zur Verfügung: *Das Kriterium gilt als erfüllt.*
  - Lesen von Betriebsparametern: Übertragungsparameter, Teilnehmeradresse, Zeiten.
  - Teilnehmerverwaltung: Auflistung aller aktiven und passiven Teilnehmer am Bus.
  - Statusabfragen von Dienstzugangspunkten.
- Zeitüberwachung in der Schicht 2:
  - *Syn-Intervall-Timer*: zur Überwachung des Busses hinsichtlich eines "ständig-0"/"ständig-1"-Fehlers; im Fehlerfall erfolgt die Meldung an den Benutzer des Netzmanagements.
  - *Token-Rotation-Time*: zur Ermittlung der *Real-Rotation-Time*.
- Für die Beobachtung und Überwachung von Datenpaketen (auf Schicht 2 und Schicht 7) steht ein Protokollanalysator zur Verfügung /98/.

### Projektierungshilfen

- Folgende Funktionen stehen im Rahmen des Netzmanagements zur Verfügung: *Das Kriterium gilt als erfüllt.*
  - Setzen von Betriebsparametern: Übertragungsparameter, Teilnehmeradressen, projektierbare Zeiten ( $T_{TR}$ : Token-Soll-Umlaufzeit).
  - Konfiguration, Aktivieren/Deaktivieren von Dienstzugangspunkten.
  - Rücksetzen der Schichten 1 und 2.
- Mit Hilfe von Konfigurationswerkzeugen können
  - Kommunikationsbeziehungen zwischen den Teilnehmern beschrieben,
  - Kommunikationsobjekte der Teilnehmer definiert und
  - Betriebsparameter gesetzt werden. /99/

#### **4.3.3 Anwendung auf den DIN-PDV-Bus**

Der PDV-Bus wurde Ende der 70er Jahre unter der Federführung des Kernforschungszentrums Karlsruhe entwickelt. In der PDV-Bus-Norm sind keine Festlegungen bzgl. des Übertragungsmediums getroffen. Sie beinhaltet:

- Teil 1: Serielle Digitale Schnittstelle,
- Teil 2: Elemente des Übertragungsprotokolls und Nachrichtenstruktur,
- Teil 3: Buszuteilung in zentralgesteuerten Systemen.

Da die Produktion des unterstützenden Bausteines MEB3000 (Fa. Valvo) in jüngster Zeit eingestellt wurde, wird der PDV-Bus künftig keine Verwendung mehr finden.

Offene Kommunikation

- Wie die vorhergehenden Feldbusse liegt der PDV-Bus als deutsche Norm vor, hat aber international keine Bedeutung.
- Aufgrund seines Entstehungszeitpunkts ist MAP-Kompatibilität nicht gegeben.

*Das Kriterium gilt als teilweise erfüllt.*

Stationstypen

- Es werden folgende Stationstypen unterschieden:
  - Leitstation,
  - Unterstation mit und ohne Querverkehr,
  - Mithör- und Monitorstation.
- Unter Verwendung der Busoperation *Übergabe-Leitfunktion* (automatisch oder spontan) kann ein Multimastersystem betrieben werden.
- Mit Hilfe des Bausteins MEB3000 konnten Einfachstationen implementiert werden.

*Das Kriterium gilt als erfüllt.*

Datentransfermodi

- Die Kommunikation erfolgt ("bestätigt") in Aufruf-/Antwortzyklen.
- Es sind zyklische Busoperationen und nicht-zyklische Busoperationsfolgen durchführbar. Zyklische Dienste sind aber nicht expliziter Bestandteil der Norm.
- Über eine *Sammeladresse* sind bei der *Kurzabfrage* **alle** Busteilnehmer erreichbar ("Broadcast", "unbestätigt").
- "Querverkehr" (zwischen zwei Unterstationen) ist möglich.

*Das Kriterium gilt als teilweise erfüllt.*

Dynamische Konfigurationsänderung

Das PDV-Bus-Protokoll erkennt eine Änderung der Teilnehmer-situation nicht von sich aus. Allerdings stehen Funktionen, mit denen Stationen im Kommunikationssystem an- und abgemeldet werden, können zur Verfügung. (Im Rahmen der Anwen-

*Das Kriterium gilt als nicht erfüllt.*

dungsprogrammierung müssen Stationen bei Besitz der Leitfunktion zyklisch einen logischen Adreßraum durchsuchen.)

### Teilnetz-Bildung

Teilnetzübergreifende Adressierung ist nicht vorhanden und muß vom Benutzer durchgeführt werden.

*Das Kriterium gilt als nicht erfüllt.*

### Echtzeit

- Das Netzzugangsverfahren ist deterministisch.
- Gemäß dem Token-Prinzip existieren Mechanismen, mit denen Stationen die Sendeberechtigung zyklisch erhalten können: Die "diensthabende" Leitstation kann mit der *Statusabfrage* feststellen, ob eine andere Station die Leitfunktion übernehmen will und erteilt diese mit dem Befehl *Übergabe-Leitfunktion*.
- Wie für den PROFIBUS gilt, daß Antwortzeiten nur unter bestimmten Voraussetzungen (Teilnehmerkonfiguration, Kommunikationsanwendung) garantiert werden können.

*Das Kriterium gilt als teilweise erfüllt.*

### Leistungsbewertung

- Aufgrund der Freiheit bei der Wahl des Übertragungsmediums können keine Leistungsangaben für Übertragungszeiten und Durchsatz gemacht werden.
- Veröffentlicht wurden Leistungsdaten von ausgewählten Anwendungsumgebungen /100, 101/.

*Das Kriterium gilt als teilweise erfüllt.*

### Alarmbehandlung

- Durch Polling von bereitgestellten Antwortaufträgen kann eine "passive" Alarmgabe durchgeführt werden.
- Alarmnachrichten können als *Kurznachrichten* abgesetzt.

*Das Kriterium gilt als erfüllt.*

Fehlertoleranz

- Im Nachrichtenaufbau ist jedes 3. Byte als Sicherungsfeld (CRC) vorgesehen. *Das Kriterium gilt als erfüllt.*
- Antwortaufträge werden im Fehlerfall (NMAX-mal) wiederholt.
- Es gibt folgende Überwachungszeiten:
  - TAB: max. Zeit zwischen Aufruf und Antwort bei einer sog. *Kurzabfrage* oder *Statusabfrage*,
  - TAN: max. Zeit zwischen Aufruf und Antwort im normalen Nachrichtenzyklus,
  - TQV: max. Zeit für die korrekte Zuteilung des Querverkehrs,
  - TLN: zur Überwachung der Aktivität der Leitstation.

Netzdiagnose

Mithör- und Monitorstationen stellen ausschließlich Empfangsaufträge zur Verfügung. Aufbauend auf diesem Stationstyp können Protokollanalysatoren /102/ verwirklicht werden. *Das Kriterium gilt als teilweise erfüllt.*

Projektierungshilfen

Für jede Station wird in einer Initialisierungsphase die Stationsadresse, die maximale Anzahl von Wiederholungsaufrufen und verschiedene Überwachungszeiten projiziert. Spezielle Werkzeuge stehen nicht zur Verfügung. *Das Kriterium gilt als teilweise erfüllt.*

#### 4.3.4 Anwendung auf den DIN-Meßbus

Der DIN-Meßbus wurde von Meßgeräteherstellern und der Physikalisch-Technischen Bundesanstalt (PTB) im Normenausschuß Länge und Gestalt (NLG) bearbeitet, um ein einfaches Datenübertragungssystem im Bereich der Qualitätssicherung und der Betriebs- bzw. Maschinendatenerfassung (BDE/MDE) festzulegen. Auf eine möglichst störungsfreie, genaue Datenübertragung für Meßanwendungen ("Eichfähigkeit") wurde besonders Wert gelegt.

##### Offene Kommunikation

- Als internationale Normen werden für die Übertragungstechnik EIA RS-485 und die ASCII-Zeichendarstellung (ISO 646) verwendet. *Das Kriterium gilt als teilweise erfüllt.*
- Der DIN-Meßbus hat den Status einer deutschen Norm und international keine Bedeutung.
- Ein Anwendungsprotokoll ist als weiterer Teil geplant, aber MAP- und MMS-Kompatibilität ist wie beim PROFIBUS nicht gegeben.

##### Stationstypen

- Es sind 2 Stationstypen (*Leit- und Teilnehmerstationen*) vorgesehen. *Das Kriterium gilt als teilweise erfüllt.*
- Es handelt sich um ein "Single-Master"-System.

##### Datentransfermodi

Daten können wie folgt ausgetauscht werden:

- Der Datenaustausch ist verbindungsorientiert mit Aufforderungs-, Datenübermittlungs- und Abschlußphase. Einzelnachrichten ("Datagramme") sind nicht vorgesehen. *Das Kriterium gilt als teilweise erfüllt.*

- Polling: Im Rahmen einer Aufforderung kann ein Sendeaufbau durch die Leitstation an eine Teilnehmerstation erfolgen. Zyklische Dienste sind aber nicht expliziter Bestandteil der Norm.
- Es findet *Querverkehr* zwischen zwei Teilnehmerstationen über die Leitstation statt.
- Broadcast: Mit einer speziellen Empfangsadresse wird ein *Gruppenempfangsschnellaufbau* an **alle** Teilnehmerstationen ("unbestätigt") durchgeführt.

#### Dynamische Konfigurationsänderung

Mechanismen zur dynamischen Adressierung sind nicht vorgesehen. *Das Kriterium gilt als nicht erfüllt.*

#### Teilnetz-Bildung

Eine geographische Ausdehnung über Repeater ist möglich, Netzwerkfunktionen sind nicht vorhanden. *Das Kriterium gilt als nicht erfüllt.*

#### Echtzeit

- Prioritäten für die Abfrage von Teilnehmern können eingestellt werden. *Das Kriterium gilt als teilweise erfüllt.*
- Die *Betriebsüberwachung* legt fest, daß die Leitstation nach Abgabe des Senderechtes ("Querverkehr") dieses innerhalb der Zeitspanne  $T_c$  wieder erhält.
- Bzgl. der Antwortzeitgarantien gelten die gleichen Bemerkungen wie bei den zuvor genannten Feldbusvertretern.

#### Leistungsbewertung

Veröffentlichungen über Leistungsergebnisse stehen derzeit für die vorliegende Betrachtung nicht in ausreichendem Maß zur Verfügung. *Das Kriterium gilt als teilweise erfüllt.*

Für die Übertragungsgeschwindigkeit gelten die Vorschriften gemäß EIA RS-485 (meist 9600 Bit/s, max. bis zu 1 MBit/s).

### Alarmbehandlung

Durch die Einstellung von Prioritätsstufen kann die Leitstation "Alarmgeräte" bevorzugt bedienen.

*Das Kriterium gilt als erfüllt.*

### Fehlertoleranz

■Die Eigenschaften von EIA-RS-485 gewährleisten eine hohe Störsicherheit

*Das Kriterium gilt als erfüllt.*

■Paritätsbildung und Blockprüfung werden vorgenommen.

■Der Zeitabstand zwischen zwei übermittelten Zeichen darf einen Maximalwert nicht überschreiten.

■Die einzelnen Nachrichten in der Aufforderungs-, Datenübermittlungs- und Abschlußphase werden jeweils bestätigt.

■In der Datenübermittlungsphase erfolgt eine Antwortzeitüberwachung: Wenn ein Teilnehmer in der Zeit  $T_A$  nicht antwortet, wird er als defekt oder nicht vorhanden eingestuft.

### Netzdiagnose

Eine Leitstation oder Teilnehmerstation kann mit Hilfe eines Softwarepaketes Überwachungsaufgaben übernehmen /103/ und folgende Betriebszustände anzeigen:

*Das Kriterium gilt als erfüllt.*

■Überblick über die gesamte Buskonfiguration,

■Angaben über den Zustand der einzelnen Busteilnehmer (aktiv, zugeschaltet, gestört, Zeitpunkt der Kommunikation mit der Leitstation).

### Projektierungshilfen

Reihenfolge, Priorität und Pausen bei der Auslösung von Sensordaufrufen der Leitstation an die Teilnehmerstation sowie Überwachungszeiten können prinzipiell eingestellt werden.


































*Das Kriterium gilt als teilweise erfüllt.*




Veröffentlichungen über vorhandene Projektierungswerkzeuge stehen derzeit für die vorliegende Betrachtung nicht in ausreichendem Maß zur Verfügung.

4.3.5 Zusammenstellung der Ergebnisse


Im folgenden werden die Ergebnisse der Klassifizierung für die untersuchten Feldbusse tabellarisch einander gegenübergestellt.

Kriterium \ Feldbus	PROFIBUS	PDV-Bus	DIN-Meßbus
Offene Kommunikation			
Stationstypen			
Datentransfermodi			
Dynamische Konfigurationsänderung			
Teilnetzbildung			
Echtzeit			
Leistungsbewertung			
Alarmbehandlung			
Fehlertoleranz			
Netzdiagnose			
Projektierungshilfen			

Kriterium erfüllt



Kriterium teilweise erfüllt



Kriterium nicht erfüllt




Bild 4.16: Gegenüberstellung der klassifizierten Feldbusse

Die gewählten Feldbusvertreter unterscheiden sich in einigen Kriterien:

■ Stationstypen

Beim DIN-Meßbus handelt es sich um ein "Single-Master"-System. PROFIBUS und PDV-Bus lassen mehrere "aktive" Komponenten zu.

■ Datentransfermodi

Der PROFIBUS bietet mit den zyklischen Abfragemöglichkeiten die funktional vielseitigsten Mechanismen.

■ Leistungsbewertung

Für den PROFIBUS wurden bisher Ergebnisse auf der Basis analytischer Betrachtungen, für den PDV-Bus in Form von Messungen erzielt.

■ Netzdiagnose

Bei PROFIBUS und DIN-Meßbus stehen Diagnosewerkzeuge zur Verfügung, beim PDV-Bus müssen Diagnoseaufgaben vom Anwender implementiert werden.

■ Projektierungshilfen

Im Unterschied zu PDV-Bus und DIN-Meßbus bietet der PROFIBUS praktikable Projektierungswerkzeuge.

## 5 MAP-konforme Kommunikationsarchitekturen für das Gerätespektrum flexibler Produktionszellen

### 5.1 Gerätespektrum in flexiblen Produktionszellen

Der Abschnitt 4 befaßte sich mit der Kommunikationsfähigkeit von Automatisierungsgeräten und schlüsselte sie in einzelne, für den Einsatz in der Fertigungsumgebung wichtige Eigenschaften auf. Neben der **Kommunikationsfähigkeit** wird die Gesamtleistungsfähigkeit eines Automatisierungsgerätes bestimmt durch die eigentliche Anwendungsfunktionalität einhergehend mit der entsprechenden **Verarbeitungsleistung**. Vergleichbare Untersuchungen bewerteten ausschließlich die Rechenleistung von Automatisierungsgeräten und ziehen dafür Kriterien wie "Intelligenz" und "Speicherkapazität" heran /104/. Die innerhalb flexibler Produktionszellen eingesetzten Geräte unterscheiden sich hinsichtlich der beiden Kenngrößen stark voneinander. Deshalb wird eine Klassifizierung vorgenommen, wobei für jede Kenngröße 2 Werte zugelässig sind: **hohe** und **geringe** Verarbeitungsleistung bzw. Kommunikationsfähigkeit. Somit entstehen 3 für die Automatisierung sinnvolle Stationsklassen (s. Bild 5.1)

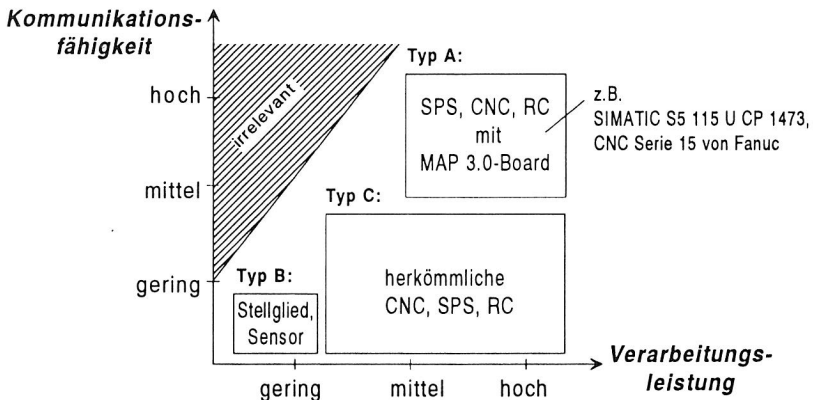


Bild 5.1: Einordnung von Gerätetypen

"Typ-A"-Geräte sind leistungsfähige Steuerungen (z.B. RC, NC) für komplexe technische Anwendungen (z.B. bahngesteuerte Bewegungen) mit Mikro- oder sogar Multiprozessortechnik oder Rechner mit dispositiven Aufgaben der Zellensteuerung (z.B. Personalcomputer oder Workstations als Zellenrechner). Sie verfügen über komfortable, vielseitige Kommunikationsdienste, wie sie MAP-Anschaltungen bieten.

Im krassen Gegensatz dazu stehen "Typ-B"-Geräte. Im Rahmen des Produktionsablaufes übernehmen sie oft lediglich "Ja"/"Nein"-Entscheidungen, die mit einfachen elektronischen Bauteilen (z.B. Fotodiode) realisiert sind.

Die Kommunikation beschränkt sich auf den Austausch von binären oder auch analogen Signalen. Zu diesem Typ gehören Sensoren, wie z.B. Taster, oder Aktoren, wie z.B. Ventile.

"Typ-C"-Geräte stehen denen vom Typ A hinsichtlich der Verarbeitungsleistung in nichts nach. Allerdings kommunizieren sie vergleichsweise "primitiv" und unflexibel. Diesem Profil entsprechen z.B. Robotersteuerungen mit herstellerspezifischen bzw. nicht-standardisierten Schnittstellen.

Es werden drei Kommunikationsarchitekturen für MAP-konforme Kommunikation mit Automatisierungsgeäten der genannten Klassen (Typ A, B, C) vorgestellt. In den anwendungsorientierten Schichten liegt den Architekturen MMS, in den transportorientierten Schichten das Kommunikationskonzept LABORINTH zugrunde, dessen Eigenschaften im folgenden erläutert werden.<sup>4</sup>

•

---

<sup>4</sup> Die im vorliegenden Abschnitt präsentierten Forschungsergebnisse wurden in Kooperation mit der Informatik Forschungsgruppe E am Lehrstuhl und Laboratorium für Technische Elektronik der Universität Erlangen-Nürnberg erzielt (/106/, /107/, /108/).

## 5.2 LABORINTH als konzeptueller Rahmen für prozeßnahe Kommunikationssysteme

Das verteilte Kommunikationskonzept LABORINTH entstand im Rahmen der Forschungsarbeiten am Lehrstuhl und Laboratorium für Technische Elektronik, Informatik Forschungsgruppe E der Universität Erlangen-Nürnberg /108/ und fand erste Anwendung im Bereich der Labordatenverarbeitung (Pilotinstallation im Zentrallabor des Universitätskrankenhauses Erlangen).

LABORINTH folgt dem Prinzip, daß ein **Kommunikationskonzept** sowohl Aussagen über die Eigenschaften eines **Kommunikationssystems** als auch über die seiner **Benutzer** treffen muß. Es umfaßt

- ein Kommunikationssystem für heterogene Verbunde aus Feld- und Peripheriebussen ("Small Area Networks") und
- Prozeßschablonen mit vorgegebenem Kommunikationsverhalten, die mit anwendungsspezifischen Funktionen zu füllen sind.

Dieser Abschnitt erläutert zunächst die Hard- und Softwarebausteine des LABORINTH-Konzepts mit ihren Kommunikationsbeziehungen. Dann folgen Beschreibungen der Schichtenstruktur des Kommunikationssystems und der an den Schnittstellen angebotenen Dienste. Angaben zu Implementierungen schließen den Überblick ab.

### 5.2.1 Die Hard- und Softwarebausteine des LABORINTH-Konzepts und ihre Kommunikationsbeziehungen

Bild 5.2 führt die Symbole für Bausteine und Beziehungen ein: Topologiebestimmend in einem LABORINTH-Netzwerkverbund ist das *Untersystem*.

*Untersysteme* umfassen Einzelnetze mit einheitlichem physischen Übertragungssystem wie z.B. PDV-Bus /95/, IEC-Bus /109/ und BITBUS /110/ und tragen eine eindeutige Untersystemnummer.

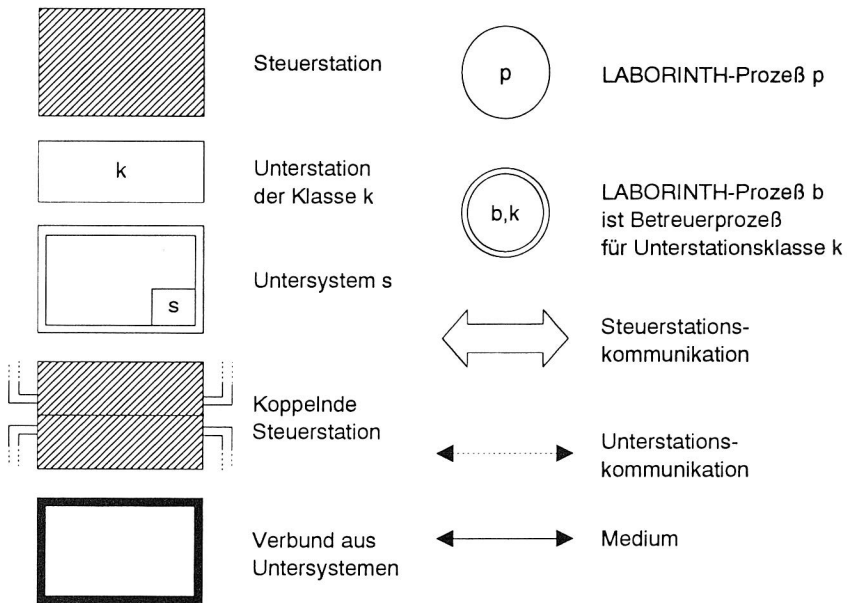


Bild 5.2: Bausteine des LABORINTH-Konzepts

Die Abbildbarkeit realer Stationen auf die abstrakten LABORINTH-Stationstypen *Steuerstation* und *Unterstation* ist Voraussetzung für die Integration in das LABORINTH-Konzept. Steuerstationen sind "reich" und universell ausgestattet mit Kapazität für Anwendungs- und Kommunikationsfunktionen, die Ausrüstung der Unterstationen hingegen ist "arm" hinsichtlich der Kommunikationsfähigkeit und insgesamt zweckgebunden.

*Koppelnde Steuerstationen*, d.h. Paare aus Steuerstationen, die unterschiedlichen *Untersystemen* angehören, verwirklichen im LABORINTH-Konzept die Übergänge zwischen benachbarten *Untersystemen*. *Untersysteme* müssen mittels koppelnder Steuerstationen so miteinander verbunden sein, daß eine zusammenhängende zyklensfreie Baumstruktur entsteht.

Hauptakteure in einem LABORINTH-Verbund sollten die Vertreter des abstrakten

Stationstyps *Unterstation* sein. Diese Geräte verwirklichen die Schnittstelle zwischen Prozeßrechnersystem und Prozeß; *Unterstationen* mit (aus der Sicht des Kommunikationssystems) identischem Verhalten können zu *Unterstationsklassen* zusammengefaßt werden.

Die Benutzerseite des vorgestellten Kommunikationskonzepts wird durch LABORINTH-*Prozesse* modelliert. Die Kommunikation zwischen den steuerstationsresidenten LABORINTH-*Prozessen* wird als *Steuerstationskommunikation* bezeichnet. Sie überspannt den gesamten LABORINTH-Verbund. Die *Unterstationskommunikation* hingegen ist auf *Untersysteme* beschränkt, um Echtzeit-Anforderungen erfüllen zu können: Sie findet statt zwischen einer Teilmenge der LABORINTH-*Prozessen*, den *Betreuerprozessen*, und individuellen Unterstationen bzw. Mitgliedern einer Unterstationsklasse. LABORINTH-*Betreuerprozesse* sind "Gerätetreiber" für Verteilte Systeme.

Die *Betreuerprozesse* kennen die individuellen Eigenschaften "ihrer" *Unterstation* bzw. "ihrer" *Unterstationsklasse* und verbergen sie vor dem Prozeßrechnersystem. So holt z.B. ein *Betreuerprozeß* für einfache Sensor-Unterstationen mit Hilfe des Kommunikationssystems zyklisch die gewonnenen Meßdaten ein und führt die Vorverarbeitung durch. Bild 5.3 zeigt mögliche Zusammenhänge innerhalb eines Untersystems.

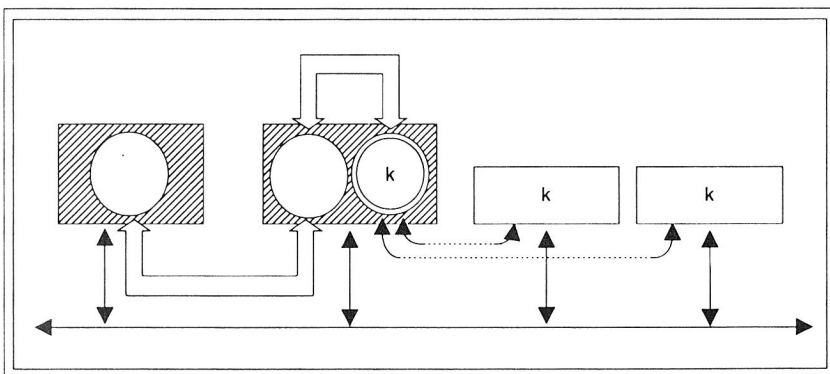


Bild 5.3: Beispiel eines Untersystems

### 5.2.2 Die Schichtenstruktur des LABORINTH-Kommunikationssystems

Das Kommunikationssystem des LABORINTH-Konzepts verfolgt dieselben Ziele wie die Bestrebungen, die in jüngerer Zeit unter Begriffen wie "Mini-MAP", "Real-Time-MAP" und "Enhanced Performance Architecture(EPA)" veröffentlicht wurden /111/: Verringerung der Schichtenzahl in ISO-OSI-orientierten Kommunikationsarchitekturen, um Echtzeitverhalten und die Einbindung einfach aufgebauter Stationen zu ermöglichen.

Das LABORINTH-Kommunikationssystem ist aufgebaut aus einer

- vom Übertragungssystem **abhängigen** LABORINTH-LINK-Schicht (z.B. für PDV-Bus, BITBUS, IEC-Bus) und einer
- vom Übertragungssystem **unabhängigen** LABORINTH-TRANSPORT-Schicht, die auch netzwerkorientierte Funktionen im Sinne der ISO-OSI-Vermittlungsschicht enthält.

Im folgenden werden die Dienste der LABORINTH-TRANSPORT- und LABORINTH-LINK-Schicht angegeben.

An der LABORINTH-TRANSPORT-Schnittstelle werden den LABORINTH-*Prozessen* folgende Dienstgruppen angeboten:

- An- und Abmelden von LABORINTH-*Prozessen*,
- Verbindungsloses Senden von Datagrammen zwischen LABORINTH-*Prozessen*,
- Auf- und Abbau dynamischer *Betreuungsbeziehungen* zwischen *Betreuerprozeß* und *Unterstationstation*,
- "Master-Slave"-Kommunikation zwischen *Betreuerprozeß* und *Unterstation*,
- Vermittlung von *Bedienungsrufen*.

Den Funktionen der LABORINTH-TRANSPORT-Schicht werden an der LABORINTH-LINK-Schnittstelle folgende Dienstleistungen angeboten:



- Nachrichtentransport zwischen *Steuerstationen*,
- "Master-Slave"-Kommunikation zwischen *Steuerstation* und *Unterstation*,
- Erzeugen eigener und Melden fremder *Bedienungsrufe*.

Die LABORINTH-LINK-Funktionen wurden für drei, wegen ihrer Verbreitung als Norm bzw. "Industriestandard" ausgewählte Übertragungssysteme realisiert:

- PDV-Bus,
- BITBUS und
- IEC-Bus.

Die im Abschnitt 5.5 dargebotenen Ergebnisse basieren auf der PDV-Bus-Version des LABORINTH-Kommunikationssystems. Für den Einsatz des PDV-Bus in einer prozeß-nahen Fertigungsumgebung sprechen

- das robuste Übertragungsprotokoll,
- die Freiheiten, die die PDV-Bus-Norm dem Anwender bei der Wahl der Übertragungstechnik (z.B. für störintensive Umgebungen) läßt, und
- die (mit einem verfügbaren LSI-Baustein gegebene) Möglichkeit zum Aufbau funktionell einfacher und kompakter Stationen (z.B. für Sensoranwendungen).

Das LABORINTH-Kommunikationskonzept stellt mit der Organisation

- der Koexistenz zweier grundverschiedener Stationsklassen im selben heterogenen Feldbus-Verbund und
- der resultierenden besonderen Kommunikationsformen

eine Lösung für die Aufgabe "Kommunikation im Feldbereich" dar.

## 5.3 Repräsentative Kommunikationsarchitekturen

### 5.3.1 Prinzipien des Architekturkonzepts

Verfolgt man - im Gegensatz zu den PROFIBUS-Bestrebungen - das Ziel, MMS-Dienste **uneingeschränkt normkonform** an der Schnittstelle zum technischen Prozeß zu verwirklichen, dann bietet das LABORINTH-Konzept (u.a.) die im folgenden beschriebenen Lösungen. Ihnen ist gemeinsam, daß sowohl die MMS-*Clients* als auch deren *MMPMs* als LABORINTH-*Prozesse* realisiert sind. Auf der *Server*-Seite hingegen herrscht Vielfalt: Ein MMS-*Server* und seine *MMPM* können LABORINTH-*Prozesse* oder unterstationsspezifische Instanzen unter der Regie eines *Betreuerprozesses* sein. Als Kriterium gilt, daß eine geforderte Teilmenge der MMS-Dienste uneingeschränkt und mit vertretbarem Aufwand auf einer Station angeboten werden kann. Die präsentierten Architekturvarianten beziehen allesamt die Möglichkeit ein, MMS-*Client* und MMS-*Server* in unterschiedlichen Teilnetzen, d.h. LABORINTH-*Untersystemen*, anzusiedeln.

Im Unterschied zu "Full MAP" werden im folgenden die MMS-Dienstprimitive nicht auf die ACSE-Dienstprimitive, sondern direkt auf die LABORINTH-Transportfunktionen abgebildet. Insofern ist noch keine strenge Normkonformität auf MAP-Ebene 7 gegeben. Eine effiziente Nachbildung von Anwendungsverbindungen kann im Rahmen des LABORINTH-Konzepts jedoch dadurch erreicht werden, daß für die Abwicklung des ACSE-Protokolls eine entsprechende Protokollmaschine "ACM" (*Association-Control-Machine*) als LABORINTH-*Prozeß* realisiert wird.

Die drei Architekturvarianten sind geeignet, die zuvor dargestellten, unterschiedlichen Gerätetypen in flexiblen Produktionszellen in ein gesamtes, MAP-konformes Kommunikationssystem einzubinden. Die Architekturvariante "MAP-/EPA-Station" wird der Geräteklasse A, die Architekturvariante "Einfachstation" der Geräteklasse B und die Architekturvariante "Station mit hoher Rechenleistung und geringer Kommunikationsfähigkeit" der Geräteklasse C zugeordnet.

### 5.3.2 Überblick über die Architekturvarianten

Die Architekturvariante A (s. Bild 5.4) entspricht der MAP/EPA-Konzeption: *Client* und *Server* befinden sich zusammen mit ihren *MMPMs* als *LABORINTH-Prozesse* auf Stationen mit ausreichender Rechen- und Kommunikationskapazität, d.h. auf *LABORINTH-Steuerstationen*.

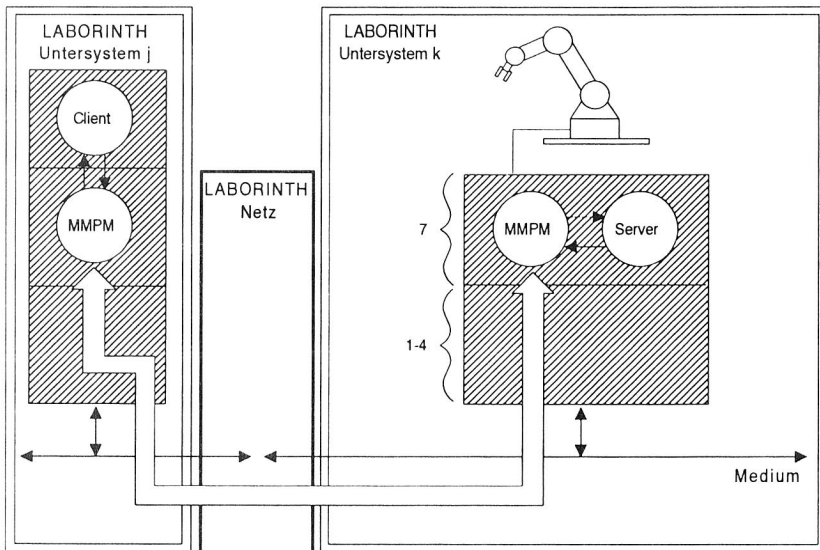


Bild 5.4: Architekturvariante A (MAP-/EPA-Station)

Im krassen Gegensatz dazu stehen die Anforderungen bei Einfachgeräten, wie z.B. bei A/D-Umsetzern für Sensoranwendungen, die in das *Client-Server-Modell* integriert werden sollen. Architekturvariante B löst das Problem (s. Bild 5.5), indem der *Server* als *LABORINTH-Betreuerprozeß* ausgelegt ist. Er steuert und überwacht "seine" *Unterstationen* und kommuniziert über eine ebenfalls als *LABORINTH-Prozeß* ausgeführte *MMPM* mit beliebigen *Clients* MMS-konform. Die *Unterstationen* müssen somit nicht MMS-kompatibel sein; für den Anschluß an die MAP-Welt sorgt der *Betreuerprozeß*.

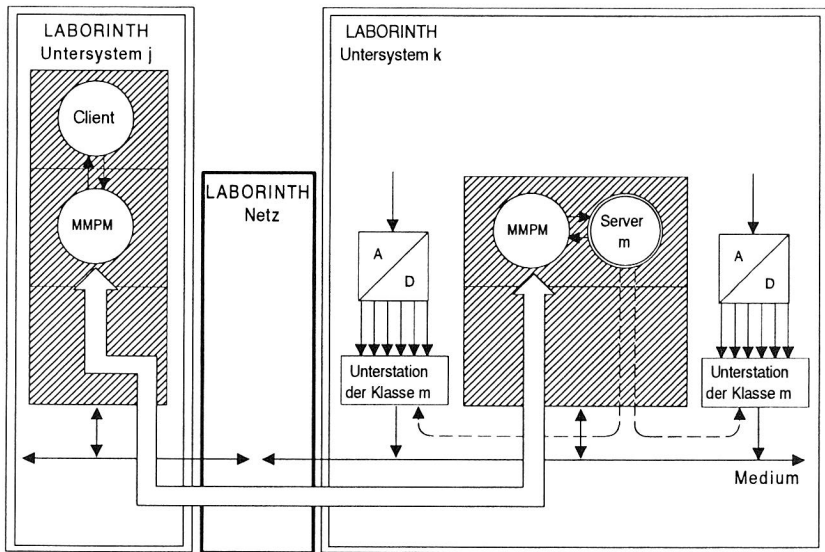


Bild 5.5: Architekturvariante B (Einfachstation)

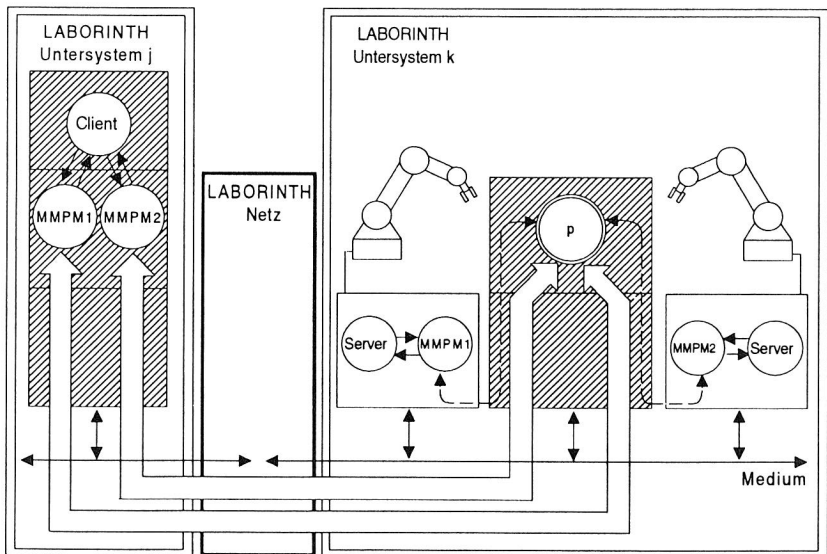


Bild 5.6: Architekturvariante C (hohe Rechenleistung, geringe Komm.-Fähigkeit)

Für "gemischt" leistungsfähige Geräte (hohe Rechenleistung, geringe Kommunikationsfähigkeit) ist die Architekturvariante C (s. Bild 5.6) geeignet: *Server* mit ihren *MMPMs* residieren auf Stationen mit hoher Rechen- und unzureichender bzw. inkompatibler Kommunikationskapazität. Mehrere *Server*-Instanzen auf *Unterstationen* samt ihren *MMPMs* sind beteiligt; auf der *Client-Steuerstation* muß für jede *Server-MMPM* eine *Client-MMPM* vorhanden sein. Der *Betreuerprozeß* bedient die "Typ-C"-spezifische Kommunikationsschnittstelle und vermittelt die zwischen den *MMPMs* ausgetauschten MMS-PDUs.

#### 5.4 Beispielhafte Implementierung ausgewählter Architekturvarianten

Im Rahmen einer Pilotimplementierung wurden zwei der konzipierten Architekturvarianten durch die physische Modellierung einer Roboterzelle validiert. Roboter sind die Hauptaktionsträger innerhalb flexibler Produktionszellen: Bei der Teilefertigung erfüllen sie Aufgaben wie die Werkzeug- und Werkstückhandhabung zur Versorgung der NC-Arbeitsmaschine, in Montagezellen führen sie die eigentliche Montageaufgabe (z.B. Fügeoperationen) durch. Derzeit eingesetzte Robotersteuerungen entsprechen dem Gerätetyp C mit leistungsfähiger Hardware für rechenintensive Anwendungsfunktionen (Transformation, Interpolation) bei gleichzeitig niedrigem Kommunikationsaufwand mit herstellerspezifischen, seriellen Übertragungssystemen. Somit war die exemplarische Realisierung der Architekturvariante C von besonderem Interesse. Die Zellensteuerung wurde als MAP/EPA-Station implementiert, repräsentiert aber im Unterschied zur Architekturvariante A den MMS-*Client*. Bild 5.7 zeigt die Experimentalumgebung mit zwei kartesischen Robotern.

Die technischen Daten der Hard- und Softwarekomponenten lauten:

- Einfache und *koppelnde Steuerstationen* sind 5-MHz-8088-Mikrorechner auf Basis des SIEMENS-SMP-Bus.
- Als lokales Betriebssystem läuft auf allen *Steuerstationen* das INTEL-Mehrprozeß-Betriebssystem RMX88.
- Die LABORINTH-TRANSPORT- und LINK-Schichten der *Steuerstationen* sind

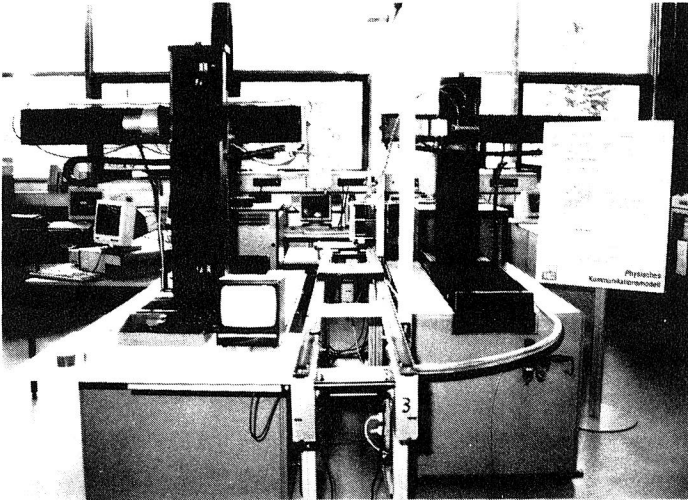


Bild 5.7: Experimentalumgebung

ausschließlich in der höheren Programmiersprache PLM86 geschrieben.

- Die *Unterstationen* der Pilot-Implementierung sind SIEMENS-MMC-216-Rechner mit 5-MHz-8086-Prozessor und dem Betriebssystemkern BD216.
- Als Übertragungssystem wird eine PDV-Bus-Version mit einer Übertragungsrate von 312 kBit/s eingesetzt.

Folgende MMS-Komponenten sind beteiligt:

*Client:* Er stellt Dienstanforderungen an die *Server* gemäß dem MMS-Protokoll.

*Server:* Nach MMS-Spezifikation ist der *Server-Prozeß* integraler Bestandteil der Anwendungsebene und muß über eine eigene Protokollmaschine verfügen. Demzufolge existiert pro *Server* ein *MMPM-Prozeß* (MMPM1, MMPM2). MMS läßt nur **eine** Kommunikationsbeziehung zwischen zwei *MMPMs* zu, so daß der *Client* zwei *MMPMs* anspricht. Jeder *Server* gibt Steuerbefehle auf Grund der empfangenen MMS-Dienstanforderungen an seinen Roboter aus.

**MMPM:** Die Protokollmaschine beinhaltet folgende *Conformance-Building-Blocks*:

*Service-CBBs:*

- *Initiate, Conclude, Abort, Reject,*
- *Read, Write,*
- *Initiate-Download-Sequence, Download-Segment, Terminate-Download-Sequence,*
- *Start, Stop.*

*Parameter-CBBs:*

- *VNAM, NEST = 0.*

Es folgen Angaben zum Umfang der wichtigsten Softwarekomponenten:

- Die LABORINTH-LINK-Funktionen der PDV-Bus-Version sind in 900, die LABORINTH-TRANSPORT-Funktionen in 1000 PLM86-Anweisungen niedergelegt. Der LINK-Code umfaßt 7,5 KByte, der TRANSPORT-Code 13 KByte.
- Die einfachsten in eine LABORINTH-Umgebung integrierbaren PDV-Bus-*Unterstationen* enthalten nur eine in Hardware realisierte Ablaufsteuerung; für *Unterstationen*, die dynamischen Betreuungsbeziehungen unterliegen und Bedienungsrufe erzeugen können, umfassen die Kommunikationsfunktionen weniger als 250 PLM86-Anweisungen.
- Ein LABORINTH-*Betreuerprozeß* für eine bestimmte *Unterstationsklasse* kann im allgemeinen mit weniger als 250 PLM86-Anweisungen formuliert werden.
- Der Softwareaufwand für *Client* und *Server* hängt prinzipiell von den Aufgaben ab, die der Anwender stellt. Der Dienstumfang der Pilotimplementierung umfaßt die oben genannten *Conformance-Building-Blocks*.
- Auf *Steuerstationen* wird die *MMPM* als LABORINTH-*Prozeß* realisiert; auf *Unterstationen* muß eine Anpassung an die Konventionen der Umgebung erfolgen. Im vorliegenden Fall sind die Programmierungsumgebungen identisch: Für die aufgeführten Dienste ergeben sich ca. 900 PLM86-Anweisungen für die Realisierung der Zustandsmaschinen; hinzu kommen ca. 5000 per ASN.1-Compiler (s.u.) erzeugte C-Anweisungen für Kodierung und Dekodierung der PDUs.

Für die Spezifizierung von Kommunikationsdiensten wurde international "Abstract Syntax Notation One" (ASN.1) mit zugehörigen Kodierungsvorschriften ("Basic Encoding Rules for ASN.1") /54/ festgelegt. Bei der Pilotimplementierung wurden für die in ASN.1 formulierten MMS-Protokolladateneinheiten mit einem ASN.1-Compiler /55/ C-Datenstrukturen mit zugehörigen Kodierungs- und Dekodierungsroutinen erzeugt.

MMS-Software-Komponenten				LABORINTH-Software-Komponenten			
	Client	Server	MMPM		Betreuer	Transport	Link
HW	LABORINTH- Steuerstation (uP 8808, SMP, RMX88)  als AT- Einsteckkarte	LABORINTH- Unterstation (uP 8808, SMP, RMX88)	LABORINTH- Steuerstation (uP 8808, SMP, RMX88)  und  LABORINTH- Unterstation (MMC216, uP8086, Basisdienste 216)	HW	LABORINTH- Steuerstation  (uP 8808, SMP, RMX88)		
SW	PL/M86: 2200 LOC	PL/M86: 700 LOC	PL/M86 900 LOC C: 5000 LOC	SW	PL/M86: 250 LOC	PI/M86: 1000 LOC	PL/M86: 1000 LOC

LOC: Lines of Code

Bild 5.8: Umfang der Softwarekomponenten

Bild 5.8 enthält Implementierungsdaten der beschriebenen MMS-Komponenten und des Kommunikationssystems LABORINTH.

## 5.5 Leistungsbetrachtung

Zur Beurteilung der im folgenden dargebotenen Leistungsdaten ist anzumerken, daß die Pilot-Implementierung keine optimierten oder Avantgarde-Komponenten verwendet (und insofern eine "worst case"-Lösung darstellt, für die leistungsverbessernde Maßnahmen verfügbar sind). Den Angaben liegen Messungen an den physischen Objekten und daraus abgeleitete Modelle für deren Kommunikationsverhalten zugrunde:



- Sowohl auf den *Steuer-* als auch auf den *Unterstationen* teilen sich Anwendungs- und Kommunikationsfunktionen einen einzigen Prozessor mit - gemessen am "Stand der Technik" - geringer Leistung.
- Alle Softwarebausteine sind in einer höheren Programmiersprache formuliert; bei leistungsbestimmenden Komponenten wie der Kodierung und Dekodierung von MMS-PDUs liegt aus ASN.1-Spezifikationen automatisch erzeugter C-Quelltext vor.
- Die eingesetzten Betriebssysteme erfahren keine Unterstützung durch die Rechnerhardware: Prozeßumschaltzeiten in RMX88 hängen von der Anzahl der Prozesse und Nachrichtenkanäle ab und können bis zu 1 ms betragen. Zwiespältigen Einfluß auf das Verhalten eines Echtzeit-Betriebssystems hat der zugrundeliegende Zeittakt (RMX88: "System Clock Rate"): Hohe Taktraten erhöhen den Betriebssystemanteil und vermindern den Nutzanteil der zur Verfügung stehenden Prozessorleistung, verringern jedoch die Reaktionszeiten, z.B. durch kürzere Abfrageintervalle auf dem PDV-Bus. Im vorliegenden Fall wird mit einer Taktrate von 10 ms ein "Arbeitspunkt" des RMX88-Betriebssystems gewählt, der bei sehr hohem anteiligen Verwaltungsaufwand angemessene Reaktionszeiten auf Ereignisse im LABORINTH-*Untersystem* gewährleistet.
- In diesem Zusammenhang muß darauf hingewiesen werden, daß das LABORINTH-Konzept auch einfache Sensorbus-Systeme mit "Single Master"-Struktur, die heute noch mit dem Begriff "Feldbus" gleichgesetzt werden, einbezieht: Bedienungsrufe und Token-Transfer finden nicht mehr statt; zyklische Abfragen durch die LINK-Funktionen sind damit überflüssig. Der Zugriff auf die Sensor-*Unterstationen* wird allein von den *Betreuerprozessen* gesteuert.
- Die maximal erreichbare Nutzdaten-Übertragungsrate der gewählten 312-kBit/s-PDV-BusVersion mit hochintegriertem Steuerbaustein liegt (bei einer Nachrichtlänge von 32 Byte) unter 185 kBit/s.

### 5.5.1 Leistungsdaten der LABORINTH-Funktionen

Es folgen Angaben zu den **LINK-Funktionen** für PDV-Bus-*Untersysteme*:

- Die Bus-Übertragungsdauer für eine Zeichenkette der Maximallänge 256 beträgt ca. 11 ms. (Für die Übertragung zusammenhängender längerer Zeichenketten, wie sie im MMS-Protokoll durchaus vorkommen können, müssen in der Anwendungsschicht, d.h. angegliedert an die *MMPMs*, geeignete Blocking/Deblocking-Maßnahmen vorgesehen werden.)
- Für die Token-Weitergabe gilt: Liegt kein Auftrag an der LINK-Schnittstelle vor, wird nach Ablauf von 10 ms eine Abfrage durchgeführt. Nach Abwicklung eines LINK-Auftrags wird immer eine Abfrage durchgeführt. Damit ist sichergestellt, daß spätestens 12 ms nach einer Token-Anforderung die Weitergabe erfolgt (vorausgesetzt, das PDV-Bus-Protokoll sieht genau einen Übertragungsversuch vor). Für jeden im PDV-Bus-Protokoll vorgesehenen weiteren Übertragungsversuch erhöht sich diese Zeitgrenze um ca. 12 ms. Die Obergrenze für die Umlaufzeit des Token hängt ausschließlich von der Anzahl der *Steuerstationen* im *Untersystem* und den Werten der o.a. Parameter ab. Ausführungszeiten für die Operationen der *Steuer-* und *Unterstationskommunikation* ergeben sich aus der Last an der LABORINTH-LINK-Schnittstelle.
- Die Gesamtheit der PDV-Bus-LINK-Funktionen eines *Untersystems* kann bei Zeichenketten der Länge 32 ca. 100 Übertragungen/s durchführen; bei einer Länge von 256 Zeichen sind ca. 60 Übertragungen/s möglich.

Für die Gesamtheit der **TRANSPORT-Funktionen** in einem solchen *Untersystem* ergeben sich unter den genannten Rahmenbedingungen für Nachrichten der Längen 32 bzw. 256 Übertragungsraten von 65/s bzw. 50/s. (Bei niedrigerer RMX88-"System Clock Rate" und Verzicht auf zyklische Abfragen stünden Übertragungsraten von über 200/s an der TRANSPORT-Schnittstelle zur Verfügung, d.h. in "Single Master"-Systemen könnten *N Unterstationen* alle 5*N* ms "abgetastet" werden.)

### 5.5.2 Leistungsdaten ausgewählter MMS-Funktionen

Die Leistung der LABORINTH-LINK- und TRANSPORT-Schichten begrenzt den Durchsatz der überlagerten MMS-Funktionen. Optimierungsziel ist die Zahl der im Dienste einer Anwendungsaufgabe ausgeführten *Confirmed-Services* pro Zeiteinheit.

Die folgenden Betrachtungen beziehen sich auf Dienste aus der Dienstgruppe *Variable-Access-Services*, bei denen (anders als bei den *Domain-Management-Services*) das Echtzeitverhalten im Vordergrund steht (s. auch /69/). Der Zeitablauf bei der Erbringung eines MMS-*Read*-Dienstes wird für die in Bild 5.6 skizzierte Architekturvariante analysiert. Dabei wird angenommen, daß

- alle beteiligten Stationen demselben *Untersystem* angehören, und daß
- zu jedem Zeitpunkt maximal eine Dienstleistung zwischen *Client* und *Server* in Bearbeitung ist. (Dienst-"Pipelining" (s. Abschnitt 2) würde zu höherem Durchsatz führen.)

In groben Zügen und in LABORINTH-Terminologie stellt sich das Szenario wie folgt dar:

- Ein *Client* auf *Steuerstation C* wünscht von einem auf der *Unterstation U* realisierten *Server* den Wert einer Variablen AKT\_Y\_POS zu erhalten.
- Der *Client* benutzt das *Request*-Dienstprimitiv für den Dienst *Read* unter Angabe des Variablennamens AKT\_Y\_POS.
- Von der *Steuerstation C* wird eine *Read-Request-PDU* zur *Steuerstation B*, die den *Betreuerprozeß* für die *Unterstation U* beherbergt, übertragen.
- Die *Steuerstation B* schreibt die *Read-Request-PDU* unverändert in die *Unterstation U*.
- In der *Unterstation U* wird die *Read-Request-PDU* dekodiert und per *Indication*-Dienstprimitiv dem *Server* angezeigt.
- Der *Server* bearbeitet den Auftrag, d.h. er stellt den Wert der Variablen AKT\_Y\_POS per *Response*-Dienstprimitiv bereit. Daraufhin wird eine *Read-*

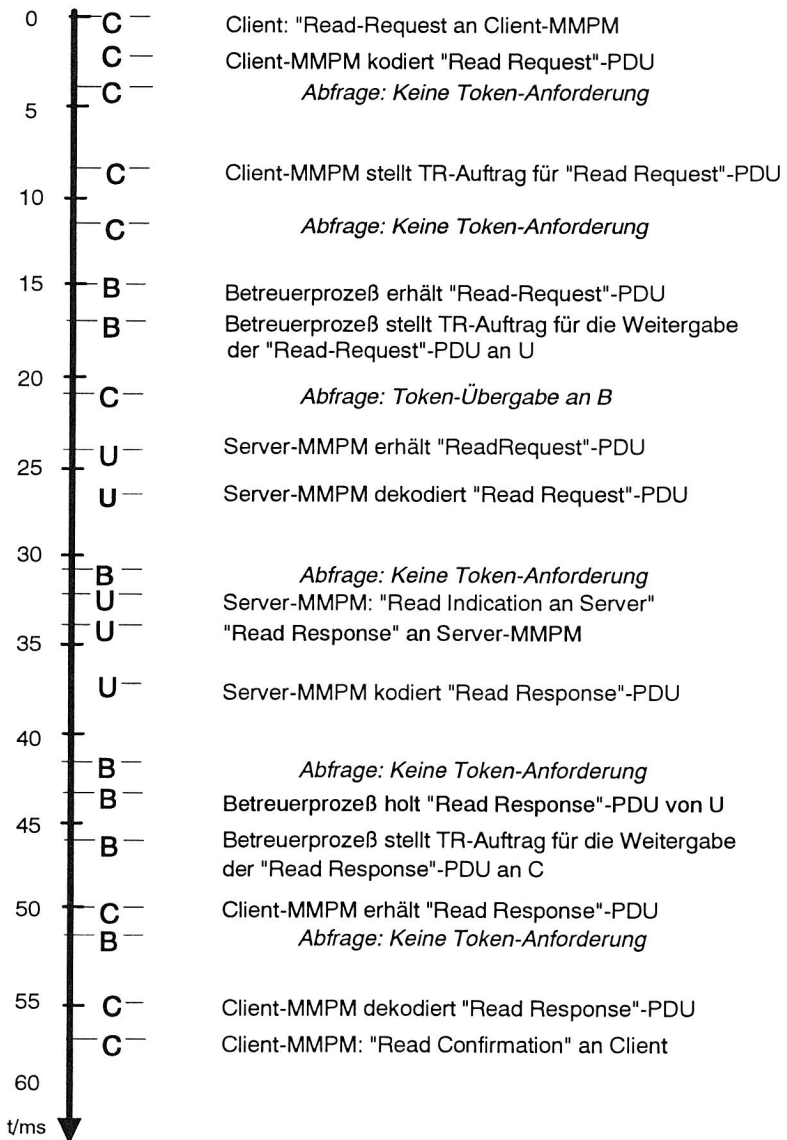


Bild 5.9: Zeitlicher Ablauf für die Erbringung eines MMS-Read-Dienstes

*Response*-PDU aufgebaut und von der *Unterstation* U zur Abholung angeboten.

- Der zuständige *Betreuerprozeß* auf der *Steuerstation* B holt die *Read-Response*-PDU von U und gibt sie unverändert an die *Client-Steuerstation* C weiter.
- Auf C wird die *Read-Response*-PDU dekodiert; der Wert der Variablen AKT\_Y\_POS wird per *Confirmation*-Dienstprimitiv an den *Client* übergeben.

Bild 5.9 zeigt den zeitlichen Ablauf über einem 5-ms-Raster. Es finden 4 Übertragungen statt; die PDV-Bus-Leitfunktion wird dabei genau einmal weitergegeben. Für die Ausführung der aus ASN.1-Anweisungen automatisch erzeugten Kodierungs- und Dekodierungsprozeduren für MMS-PDUs wurden jeweils ca. 8 ms gemessen. Auf den beteiligten Stationen waren keine weiteren leistungsmindernden Softwarekomponenten vorhanden.

Wichtigstes, durch Messungen am physischen Objekt bestätigtes Ergebnis der Zeitanalyse ist die Zahl von ca. 17 *Confirmed-Services/s*. Alle 55 ms kann demnach in der beschriebenen Architekturvariante eine zeitkritische MMS-Dienstleistung erbracht werden.

Die beiden anderen in den Bildern 5.4 und 5.5 dargestellten Architekturen lassen höhere Durchsatzraten erwarten, weil im Gegensatz zur hier behandelten Variante mit protokollumsetzendem *Betreuerprozeß* statt 4 nur 2 Zugriffe auf das physische Medium nötig sind.

### 5.5.3 Ausblicke auf Optimierungsmöglichkeiten

Leistungsverbessernde Maßnahmen, die die wichtigsten Merkmale der beschriebenen Lösung betreffen, liegen auf der Hand:

- Kommunikationsorientierte Architekturen für *Steuerstationen*, die Anwendungs- und Kommunikationsaufgaben parallel bearbeiten /112/.
- Übertragungssysteme mit höherer Übertragungsrate.

- Leistungsfähigere Prozessoren, die Echtzeit-Betriebssysteme unterstützen.
- Effizientere ASN.1-Compiler. (An der automatischen Übersetzung von ASN.1-Anweisungen sollte aus Konformitätsgründen auch im Feldbus-Bereich festgehalten werden.)
- Dienst-"Pipelining".

Raten von ca. 100 *Confirmed Services/s* für repräsentativ besetzte LABORINTH-Untersysteme sollten unter solchen Bedingungen erreichbar sein.

## 5.6 Zusammenfassung

Ziel dieses Abschnitts war es einerseits, MAP-Kommunikation generell innerhalb flexibler Produktionszellen zu etablieren. Durch Klassifizierung konnte die Gerätevielfalt auf 3 charakteristische Typen eingeschränkt und eine jeweils geeignete Kommunikationsarchitektur konzipiert werden.

Andererseits wurde gezeigt, wie insbesondere Einfachgeräte durch Auslagerung höherer Kommunikationsfunktionen einbezogen werden können. "Zwischengeschaltete" Stationen mit Betreuungsinstanzen kennen und verbergen das individuelle Kommunikationsverhalten der Sensoren und Aktoren. Dies gilt gleichermaßen für Einfachstationen mit transportorientierten Protokollimplementationen, z.B. mit PDV-Bus-Baustein, oder solche ganz ohne Kommunikationsfunktionen im ISO/OSI-Sinn, z.B. mit analoger Stromschnittstelle.

Allen Architekturen liegt eine reduzierte Schichtenstruktur im Sinne von Mini-MAP und EPA zugrunde, die auf die Verringerung des Kommunikationsaufwandes und Echtzeitverhalten abzielt. Folgende Eigenschaften werden aber für die Kommunikation in flexiblen Produktionszellen für notwendig erachtet und gemäß "Full MAP" beibehalten:

- Durch die explizite Aufnahme von Netzwerkfunktionen ist es möglich, Teilnetze zu adressieren und somit Verbunde aus einzelnen Feldbussen zu betreiben.
- Die Instanzen des Kommunikationskonzepts LABORINTH ermöglichen teilnetz-

übergreifende Prozeßkommunikation und sind somit ausgerichtet auf die Eigenschaften heutiger Gerätesteuern mit "Multitasking-fähigen" Echtzeitbetriebssystemen.

- Das Schaffen flexibler Anwendungsverbindungen und Vereinbarungen bzgl. gemeinsamer Kommunikationsparameter ("Anwendungskontext") wird - im Unterschied zu MAP/EPA - für wichtig erachtet. Bei den aufgezeigten Architekturvarianten ist die Abbildung der MMS-Dienstprimitive auf die ACSE-Dienstprimitive bisher ausgelassen. Im Rahmen des LABORINTH-Konzepts können aber Anwendungsverbindungen dadurch effizient nachgebildet werden, daß für die Abwicklung des ACSE-Protokolls eine entsprechende Protokollmaschine (ACM) als LABORINTH-*Prozeß* realisiert wird.
- Die MAP-Kompatibilität auf Anwendungsebene wird in erster Linie dadurch erreicht, daß MMS-Dienste - im Gegensatz zu PROFIBUS - **uneingeschränkt normkonform** verwirklicht sind. Die Verwendung transportorientierter Protokolle, die von der MAP-Spezifikation abweichen, ist legitim, da MMS als "Sprache für die Fertigung" Eigenständigkeit erlangt hat /113, 114, 115/. In zahlreichen Veröffentlichungen /116, 117/ und Projektberichten /41, 118/ wird auf die Notwendigkeit von MMS-Implementierungen für unterschiedliche Übertragungssysteme (ETHERNET, FDDI) und Transportsysteme (TCP/IP) hingewiesen.

Anhand einer beispielhaften Implementierung zweier Architekturvarianten ("MAP/EPA-Station" und "Station mit hoher Rechenleistung und geringer Kommunikationsfähigkeit") wurde die Tauglichkeit des Konzepts für die in automatisierte Montagezellen verwendeten Gerätetypen (Roboter- und Zellensteuerung) nachgewiesen. Die Arbeiten am Gerätetyp bzw. der Architekturvariante B werden zunächst konzeptuell fortgeführt durch die Darlegung eines reduzierten, Sensor-/Aktor-spezifischen Funktionsumfangs im Sinne eines MMS-Companion-Standards (s. Abschnitt 6).

## 6 Companion-Standard-Entwurf für Sensoren und Aktoren

In Abschnitt 5 wurde aufgezeigt, wie Einfachgeräte durch "Auslagerung" von MMS an der MAP-Kommunikation teilnehmen können. Das Ziel der "MMS-fähigen" Einfachgeräte wird weiterverfolgt, indem ein Companion-Standard für Sensoren und Aktoren skizziert wird. Der allseits geforderten Reduzierung der MAP-Funktionen /14, 69/ wird dadurch entgegengekommen, daß die notwendigen MMS-Modelle (MMS-Objekte und -Funktionen) in ihrer Anzahl und Funktionalität stark eingeschränkt werden.

Struktur und Inhalt des Entwurfs richten sich nach den Empfehlungen für die Verfassung von Companion-Standards /52, 119/. Durch Beschränkung auf die wesentlichen Aspekte wurde er bewußt kurz gehalten.

Zum besseren Verständnis der im angestrebten Companion-Standard gewählten Objektstrukturen wird zunächst als exemplarisches Einfachgerät ein Reflexionslichttaster in seiner Funktionsweise und Anwendung erläutert.

### 6.1 Exemplarisches Einfachgerät

Reflexionslichttaster gehören zu der Gruppe der optischen Näherungsschalter, die berührungslos die An- bzw. Abwesenheit von Gegenständen erfassen. Lichtsender und -empfänger befinden sich in einem Gehäuse. Der Sender enthält eine Leuchtdiode, die über eine Verstärkerstufe von einem Taktgenerator angesteuert wird. Dessen Signal taktet gleichzeitig den Vorverstärker des Empfängers, dem die Auswert-

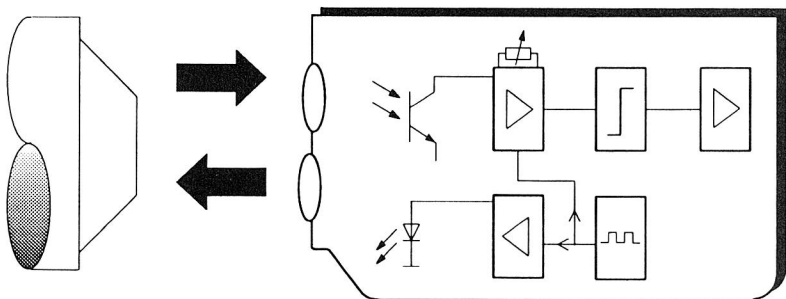


Bild 6.1: Funktionsweise eines Reflexionslichttasters



und Endstufe folgen. Der vom Sender kommende Lichtstrahl wird von dem abzutastenden Gegenstand zum Empfänger reflektiert. Die Empfängerschaltung wertet die Intensität des reflektierten Lichts aus. Mit einem Potentiometer stellt man den Schwellwert für die Zustände "Reflexion" und "keine Reflexion", die mit An- bzw. Abwesenheit eines Gegenstandes interpretiert werden, ein. Die beiden Reflexionszustände werden am Ausgang der elektronischen Schaltung (s. Bild 6.1) durch zwei Potentiale angezeigt. Bei der "Hellschaltung" bedeutet Betriebsspannung (z.B. wählbar zwischen 10V und 60V) am Ausgang "keine Reflexion", Nullpotential bedeutet "Reflexion". Die "Dunkelschaltung" ordnet Betriebsspannung und Nullpotential umgekehrt zu. Einige Reflexionstaster können wie folgt "programmiert" werden:

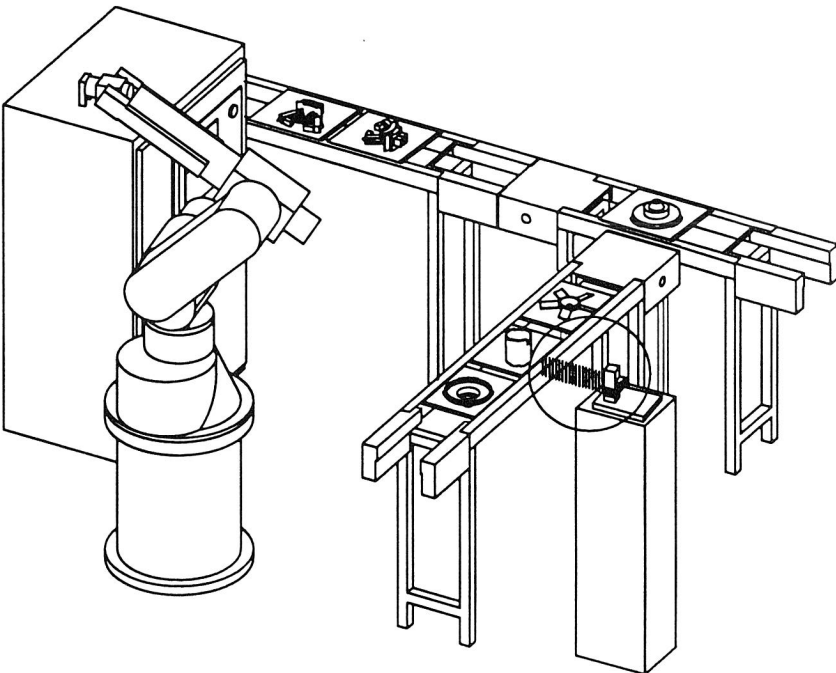


Bild 6.2: Einsatz von Reflexionslichttastern zur Anwesenheitskontrolle

- Die Arbeitsweise als "Hell-" oder "Dunkelschaltung" kann bei Schaltungsanschluß festgelegt werden.
- Die Empfindlichkeit des Empfängers läßt sich mit Hilfe eines Potentionmeters einstellen.

Bild 6.2 zeigt, wie Reflexionslichttaster die Anwesenheit von Werkstücken in einer Montagezelle erkennen.

## 6.2 Ausführung

Die inhaltliche Gliederung des vorliegenden Companion-Standard-Entwurfs orientiert sich an dem vorgegebenen Schema (s. Bild 6.3). Wegen der gebotenen Kürze werden nicht alle im Schema aufgeführten Unterkapitel in der verlangten formalen Ausführlichkeit wiedergegeben (z.B. "References", "Definitions", "Symbols and Abbreviations").

Für die Formulierung der benötigten Objekte wird die von MMS-Core vorgegebene Schreibweise herangezogen. MMS-Objektklassen werden damit wie folgt dargestellt:

*Object:*    (name of class)

*Key Attribute:* (name of attribute type (values))

    .  
    .

*Key Attribute:* (name of attribute type (values))

*Attribute:* (name of attribute type (values))

    .  
    .  
    .

*Attribute:* (name of attribute type (values))

*Constraint:* (constraint expression)

*Attribute:* (name of attribute type (values))

    .  
    .  
    .

*Attribute:* (name of attribute type (values))

Title Page	
Foreword	
Table of Contents	
Introduction	
<b>1 Scope</b>	
<b>2 Normative Reference</b>	
<b>3 Definitions</b>	
<b>4 Symbols and Abbreviations</b>	
<b>5 Application Description</b>	
5.1 Application-specific Model	
5.2 Application-specific Functions	
<b>6 Application-specific Context Mapping</b>	
6.1 Mapping of the Application-specific Model to the VMD-Object	
6.2 Definitions of the Application-specific Objects which Map to Domains	
6.3 Def. of the Application-specific Objects which Map to Program Invocations	
6.4 Def. of Application-specific Objects which Map to Other MMS Abstract Objects	
6.5 Def. of New MMS Abstract Objects to support other Application-specific Objects	
<b>7 Services</b>	
7.1 Use of the MMS Services	
7.1.1 Additions to the MMS Service Procedure	
7.1.2 Additions to the MMS Protocol Definition	
7.2 Definition and Use of Application-specific Services	
7.3 The Initiate Service and Protocol	
<b>8 Standardized Application-specific Objects</b>	
8.1 Domain Objects	
8.2 Program Invocation Objects	
8.3 Named Variable Objects	
8.4 Scattered Access Objects	
8.5 Named Variable List Objects	
8.6 Named Type Objects	
8.7 Semaphore Objects	
8.8 Operator Station Objects	
8.9 Event Condition Objects	
8.10 Event Action Objects	
8.11 Event Enrollment Objects	
8.12 Journal Objects	
8.13 Application-specific Objects	
<b>9 Conformance</b>	
9.1 Conformance Class Descriptions	
9.1.1 Service Conformance Building Blocks	
9.1.2 Parameter Conformance Building Blocks	
9.1.3 Other Initiate Specified Parameters	
9.2 Restrictions to MMS Optional Parameters	
9.3 Conformance to Standardized Objects	
9.4 Additions to the MMS PICS	
<b>A Application-specific Examples (Informative)</b>	

Bild 6.3: Gliederungsschema für MMS-Companion-Standards

Die verwendeten MMS-Dienste (z.B. *Read*, *Status* usw.) bzw. die zugehörigen "Protocol Data Units" (PDUs) sind in der Sprache "Abstract Syntax Notation One" (ASN.1) niedergeschrieben, die für die Abfassung von ISO/OSI-Protokolle genormt wurde.

### 6.2.1 Einleitende Bemerkungen zum Standpunkt und zur Einordnung

Der vorgestellte Companion-Standard-Entwurf soll Manufacturing-Message-Specification Teil 1 und 2 um die Anwendungsspezifika von Sensoren und Aktoren (Einfachgeräten) erweitern. Den geringen Kommunikationsfähigkeiten solcher Geräte wird entgegengekommen, indem nur wenige, notwendige MMS-Modelle (MMS-Objekte und -Dienste) verwendet werden.

### 6.2.2 Referenzen

Auf folgende internationale Normen wird Bezug genommen:

ISO IS 9506, ISO IS 8824.

### 6.2.3 Definitionen

Die folgenden Begriffe sind weder einheitlich festgelegt noch genormt, deshalb werden gebräuchlich gewordene Auslegungen zitiert.

Ein **Sensorelement**, oft als **Basissensor** oder **Meßwertaufnehmer** bezeichnet, ist ein aktives oder passives Bauelement, in dem die Umwandlung physikalischer oder chemischer Kenngrößen in ein zur Weiterverarbeitung geeignetes Signal stattfindet /94, 120/.

Ein **Sensor**

ist ein betriebsfähig eingebautes ("verpacktes") Sensorelement /94, 120/.

### Ein **Sensorsystem**

ist ein Sensor mit integrierter Meßsignalaufbereitung und u.U. auch mit Meßsignalverarbeitung /94, 120/. (An anderer Stelle /121/ wird die Signalaufbereitung und -verarbeitung durch **Meßumformer** auch dem **Sensor** zugeordnet.)

### Ein "intelligentes" Sensorsystem

ist ein Sensorsystem, das zusätzlich - ggfs. programmierbar - Umrechnungsfunktionen wahrnehmen kann /94, 120/.

Prinzipiell sind Begriffe wie "Aufnehmer", "Fühler", "Meßeinrichtung" /122/ heute durch die oben genannten ersetzt. Im Rahmen des vorliegenden Norm-Entwurfs werden die aufgeführten Begriffe **nicht** unterschieden, so werden "Sensor" und "Meßwertaufnehmer" synonym verwendet.

### Ein **Aktor** bzw. **Aktuator**<sup>5</sup>

ist ein anwendungsgebundenes, durch ein elektrisches Signal ansteuerbares System zur Beeinflussung mechanischer Größen wie Weg oder Kraft /123/.

### Ein **Stellglied**

stellt das Antriebselement innerhalb eines Aktors dar. Es verursacht die eigentliche Wirkgröße. Das Stellglied muß nicht grundsätzlich elektrisch angetrieben werden, sondern stellt innerhalb des Aktors das letzte Glied in der aktiven Kette zur Erzeugung der Wirkgröße dar. /123/

Auch die Begriffe "Aktor" und "Stellglied" werden in diesem Zusammenhang gleichwertig eingesetzt.

Weder "Sensoren" noch "Aktoren" im obigen Sinn verfügen über ausreichende

---

<sup>5</sup>In der Fachliteratur werden **Aktor** /70, 124, 125/ und **Aktuator** /126, 127/ im gleichen Zusammenhang ohne Hinweise auf eine semantische Unterscheidung verwendet.

Verarbeitungs- und Kommunikationsleistung für die Abarbeitung eines MAP-Anwendungsprotokolls. Dieser Normvorschlag wurde im Hinblick auf künftige Technologien entworfen: Der Umstand, daß man heutzutage Sensoreffekte mit Halbleitermaterialien erzielt, ermöglicht, daß Sensorfunktionen und Signalverarbeitung auf einem Systemträger realisiert werden können. Dies hat bereits zu einer Reihe von "1-Chip-Sensoren", z.B. zur Messung von Druck, Temperatur, Magnetfeldern usw. geführt /128, 129/. Die Arbeiten an den "intelligenten" Meßumformern /125, 130/ lassen erwarten, daß künftig auch komplexe Kommunikationsfunktionen integriert werden.

#### **6.2.4 Symbole und Abkürzungen**

ASN.1	Abstract Syntax Notation 1
CBB	Conformance Building Block
BKK	Basis-Konformitätsklasse
CS	Companion Standard
VMD	Virtual Manufacturing Device

#### **6.2.5 Beschreibung der Anwendung**

Der vorliegende Abschnitt charakterisiert die Eigenschaften von Sensoren und Aktoren und ordnet sie in die Fertigungsumgebung ein.

##### **6.2.5.1 Anwendungsspezifisches Modell**

Sensoren und Aktoren bilden das letzte Glied beim durchgängigen Informationsfluß in der automatisierten Fabrik. Mit Aktoren kann der technische Prozeß beeinflusst, mit Sensoren beobachtet und überwacht werden. Als Bestandteile von flexiblen Fertigungszellen sind Sensoren und Aktoren leistungsfähigen Gerätesteuern (RC, NC, SPS) untergeordnet (s. Bild 6.4). Sie können nur als MMS-Server ausgelegt sein, da sie selbst keine Geräte steuern.

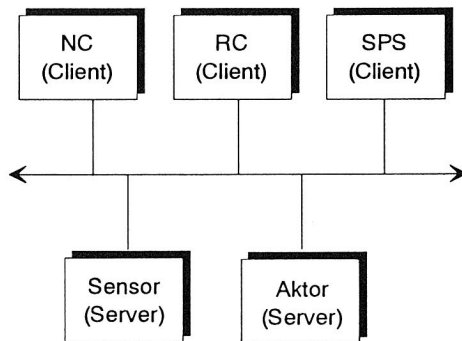


Bild 6.4: Sensoren und Aktoren im Client-Server-Modell

Sensoren und Aktoren (Einfachgeräte) werden hier nicht getrennt behandelt, da sie sich aus informationstechnischer Sicht nur in der Richtung des Informationsflusses unterscheiden: Sensoren **liefern** (Meß-)Werte, Aktoren **benötigen** (Stell-)Werte (s. Bild 6.5).

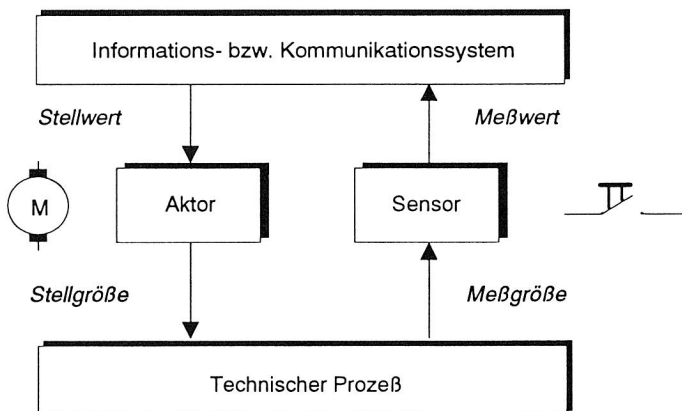


Bild 6.5: Informationsfluß /5/

Einfachgeräte werden durch folgende Größen charakterisiert (s. Bild 6.6):

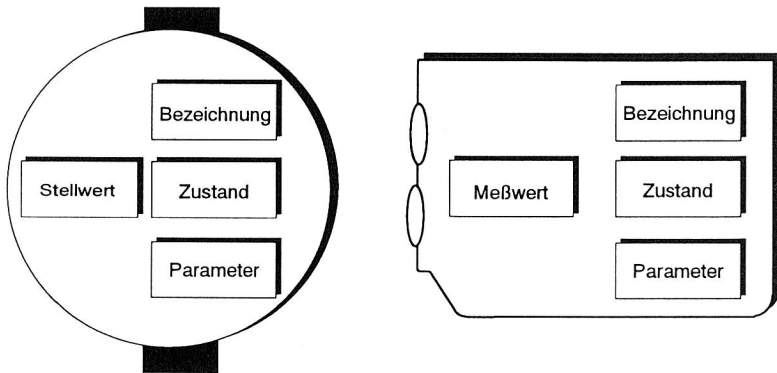


Bild 6.6: Sensor/Aktor-Modell

a) Sie haben eine eindeutige **Bezeichnung**.

Beispiel:

"OT 5001/OTT-FPKG" kennzeichnet einen Reflexionslichttaster einer bestimmten Bauform mit Tastweite 200 mm, Anschlußleitung 2m/0,34mm<sup>2</sup> usw.

b) Der **Zustand** beschreibt ihre derzeitige Betriebssituation.

Beispiel:

Ein Motor ist im einfachsten Fall entweder **ein-** oder **ausgeschaltet**.

c) Sie sind durch **Parameter** in ihrer Funktionsweise steuerbar.

Beispiele:

- Die Polumschaltung ändert bei Drehstrommotoren die Drehrichtung des Ankers. Mit Hilfe des entsprechenden Parameters kann Motor-Rechts- bzw. -Linkslauf eingestellt werden.
- Bei Reflexionslichttastern kann die Umschaltung von "Hell-" auf "Dunkelschaltung" bzw. umgekehrt über Parameter erfolgen.
- Umrechnungsfaktoren ( $\pi$  bei Winkelsensoren) oder Vervielfachungsfaktoren ( $x1$ ;  $x2$ ;  $x4$ ) können als Parameter übergeben werden.
- Schwellwerte, deren Überschreiten von Sensoren gemeldet wird, können vorgeben werden.



### 6.2.5.2 Anwendungsspezifische Funktionen

Mit anwendungsspezifischen Funktionen werden typische Kommunikationsvorgänge bei Sensoren und Aktoren strukturiert und klassifiziert. Die aufgezeigten Funktionen wurden ausgewählt, weil sie einerseits von praktischer Relevanz sind (s. Beispiele) und andererseits auf deren Bedeutung in der Literatur /131/ mehrfach hingewiesen wird.

#### Meßwerte von Sensoren:

Die originäre Aufgabe eines Sensors ist es, Meßwerte zu liefern. Dies kann auf vorhergehende Aufforderung (**stimuliert**) oder **spontan** erfolgen. Einfachgeräte agieren ("per definitionem") selten von sich aus und antworten meist auf gestellte Anfragen. Allerdings existieren auch Geräte, die von sich aus aktiv werden. Weiterhin unterscheiden sich Sensoren durch **sporadische** und **zyklische** Datenausgabe. Es werden also folgende Datentransfermodi vorgesehen:

#### a) **stimuliert/sporadisch**

Beispiel: Mit Hilfe von Bildverarbeitungssystemen erfaßte Szenen bzw. exzerpierte Kontur-, Flächen- oder Schwerpunktinformationen werden von der Robotersteuerung für die Bewegungsplanung bei Bedarf angefordert.

#### b) **stimuliert/zyklisch**

Beispiele: ■ Zur kontinuierlichen Signalarückführung für Industrieroboter werden Abstands- oder Kraft-/Momentensensoren zyklisch abgefragt.  
■ Bei Infrarotlötanlagen wird die Temperatur auf der Leiterplatte durch die Abstandsänderung des an der numerisch steuerbaren Achse befindlichen Flächenstrahlers geregelt.

#### c) **spontan/sporadisch**

Beispiel: Reflexionstaster und Lichtschranken signalisieren, sobald sie ein Werkstück "erkennen".

#### d) **spontan/zyklisch**

Die Anwendungsbeispiele entsprechen denen für "stimuliert/zyklisch" mit dem Unterschied, daß der Sensor seine Meßergebnisse aktiv meldet.

Anmerkung: Das sog. "Polling" kann gemäß der angeführten Klassifizierung dem "stimulierten" Modus zugeordnet werden, speziell wird darunter meist ein periodisches bzw. "zyklisches" Verhalten verstanden.

#### **Stellwerte für Aktoren:**

Aktoren erhalten Aktionsanweisungen in Form von Stellwerten. Je nach Aktortyp werden diese **sporadisch** (z.B. "Motor EIN") oder **zyklisch** (z.B. Reglerbausteine bei numerisch gesteuerten Achsen) ausgegeben.

#### **Befehle:**

Eine Steuerung der Funktion in Form einer Prozeßverwaltung ist bei Sensoren und Aktoren nicht gegeben. Dennoch muß es möglich sein, diese im Betrieb mit einem STOP-Befehl ferngesteuert zu unterbrechen sowie neu zu starten: z.B. sollen Interferenzen bei Multisensoranwendungen dadurch vermieden werden, daß ein Sensor (z.B. Induktiv-Sensor) zwischenzeitlich abgeschaltet wird. Der Wiederanlauf wird durch das entsprechende "START" initiiert. Einfachgeräte, die ihre Anwendungsaufgabe nicht von sich aus aufnehmen, werden gleichfalls durch den START-Befehl dazu aufgefordert.

#### **Lebenszeichen:**

Nach dem Einschalten setzt das Einfachgerät seinen Partner von seiner Funktionsbereitschaft in Kenntnis.

#### **Identifizierung:**

Die übergeordnete Steuerung identifiziert das Einfachgerät und informiert sich über dessen Typ, Leistungsfähigkeit usw.

**Statusabfragen:**

Die übergeordnete Steuerung setzt sich über den aktuellen Gerätezustand in Kenntnis, z.B. bei Fehlersituationen und zu Diagnosezwecken.

**Fehlerbehandlung:**

Das Einfachgerät meldet das Auftreten einer außergewöhnlichen Situation der übergeordneten Steuerung.

**Parametrisierung:**

Die übergeordnete Steuerung legt vor Aufnahme der Betriebsfunktion beim Einfachgerät Parameter fest, die dieses bei der Ausführung steuern. Sensoren benötigen z.B. Offset-Werte, Verstärkungsfaktoren, Grenzwerte usw.

**6.2.6 Abbildung der Anwendungsspezifika**

Die dargestellten Anwendungsbedürfnisse sollen nun auf die MMS-Modellwelt abgebildet werden, damit das Einfachgerät mit seinen Eigenschaften von der Außenwelt mit standardisierten Anwendungsfunktionen erreichbar wird. Da der MMS-Funktionsumfang für Einfachgeräte bewußt klein gehalten wird, kommen für die "Maximal-Ausrüstung" nur folgende MMS-Objektmodelle zum Einsatz:

- *Virtual-Manufacturing-Device* zur Abstrahierung von Einfachgeräten,
- *Named-Variable*-Objekte zum Lesen von Meßwerten und Setzen von Stellwerten,
- *Event-Management*-Objekte für zyklische Kommunikation.

MMS selbst enthält keine expliziten **zyklischen** Dienste. Die "spontane" bzw. "stimulierte" zyklische Kommunikation (s. "Meßwerte von Sensoren") mit Sensoren und Aktoren kann aber durch geeignete Instrumentierung von *Event-Management*-Objekten erreicht werden.

*Domains* oder *Program-Invocations* wie bei den RC-, NC- und SPS-Companion-

Standards werden nicht benötigt. Ebenso sind keine *Semaphore*-, *Journal*- und *Operator-Station*-Objekte vorgesehen.

#### 6.2.6.1 Definition des anwendungsspezifischen VMD-Objekts

Das VMD-Modell aus MMS-Core wird für Sensoren und Aktoren unverändert übernommen:

*Object: VMD*

*Key Attribute: Executive Function*

*Attribute: Vendor Name*

*Attribute: Model Name*

*Attribute: Revision*

*Attribute: List of Abstract Syntaxes Supported*

*Attribute: Logical Status (STATE-CHANGES-ALLOWED, NO-STATE-CHANGES-ALLOWED, LIMITED-SERVICES-PERMITTED, SUPPORT-SERVICES-ALLOWED)*

*Attribute: List of Capabilities*

*Attribute: Physical Status (OPERATIONAL, PARTIALLY-OPERATIONAL, INOPERABLE, NEEDS-COMMISSIONING)*

*Attribute: List of Program Invocations \*)*

*Attribute: List of Domains \*)*

*Attribute: List of Transaction Objects*

*Attribute: List of Upload State Machines (USLM) \*)*

*Attribute: Lists of Other VMD-specific Objects*

Aufgrund der Einfachgeräte-Struktur, die keine *Program-Invocations* und *Domains* benötigt, können die mit \*) entfallen bzw. bleiben leer. Bezeichnung und Zustand des Sensor/Aktor-Modells werden aus VMD-eigenen Attributen (*Model-Name* bzw. *Physical-Status* und *Logical-Status*) gewonnen.

### 6.2.6.2 Anwendungsspezifische Abbildung auf Variable-Objekte

Prinzipiell sind in MMS-Core fünf Basistypen von *Variablen* spezifiziert:

- das *Unnamed-Variable*-Objekt mit Zugriff auf die Variable über ihre Adresse,
- das *Named-Variable*-Objekt mit Zugriff auf die Variable über ihren Namen,
- das *Scattered-Access*-Objekt für zusammengesetzte Variablen,
- das *Named-Variable-List*-Objekt für zusammengesetzte Variablen mit Direktzugriff auf einzelne Komponenten,
- das *Named-Type*-Objekt für die Spezifikation neuer Variablentypen.

Aufgrund ihrer Komplexität scheiden *Scattered-Access*-Objekt, *Named-Variable-List*-Objekt und *Named-Type*-Objekt für den Gebrauch in Sensor-/Aktor-Umgebungen aus. Die Belegung von Variablen unter Angabe einer gerätespezifischen Adresse, wie es das *Unnamed-Variable*-Objekt vorsieht, wird in der Steuerungstechnik zwar häufig praktiziert, sollte aber zugunsten des Zugriffs mit anwendungsorientierten Namen vermieden werden. Deshalb werden Parameter, Meßwerte beim Sensor und Stellwerte beim Aktor auf *Named-Variable*-Objekte abgebildet. Diese sind prinzipiell folgendermaßen strukturiert:

*Object:* *Named Variable*

*Key Attribute:* *Variable Name*

*Attribute:* *MMS Deletable (TRUE, FALSE)*

*Attribute:* *Type Description*

*Attribute:* *Access Method (PUBLIC, ...)*

*Constraint:* *Access Method = PUBLIC*

*Attribute:* *Address*

Die MMS-unabhängige Modellierung von Sensoren und Aktoren (s. Bild 6.6) kann auf das in Bild 6.7 gezeigte Sensor/Aktor-VMD abgebildet werden.

Für die Versorgung des Sensors bzw. Aktors mit Parametern wird ein *Named-Variable*-Objekt *SA\_Parameter* eingeführt. Meßwerte des Sensors können dem *Named-*

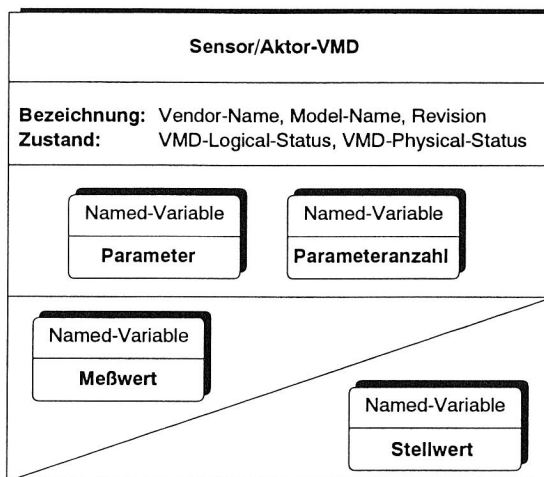


Bild 6.7: Sensor/Aktor-VMD-Modell

*Variable*-Objekt *SA\_Meßwert* entnommen werden. Stellwerte werden in das *Named-Variable*-Objekt *SA\_Stellwert* eingetragen.

Die dargestellten Objekte sind von elementarer Bedeutung im Sensor-/Aktorbereich. Treten darüberhinaus noch spezielle Anforderungen, z.B. die Übertragung von Linearisierungstabellen auf, so können diese gleichermaßen auf *Named-Variable*-Objekte abgebildet werden.

#### 6.2.6.3 Anwendungsspezifische Abbildung auf Event-Management-Objekte

Für das *Event-Management* stehen folgende Objekte zur Verfügung (s. Bild 6.8):

- das *Event-Condition*-Objekt legt die Bedingung für das Eintreten eines Ereignisses fest,
- das *Event-Action*-Objekt, wie auf das Eintreten eines Ereignisses reagiert wird,
- das *Event-Enrollment*-Objekt setzt *Event-Condition*- und *Event-Action*-Objekte miteinander in Beziehung und regelt die Benachrichtigung des *Client* beim Eintreten eines Ereignisses.

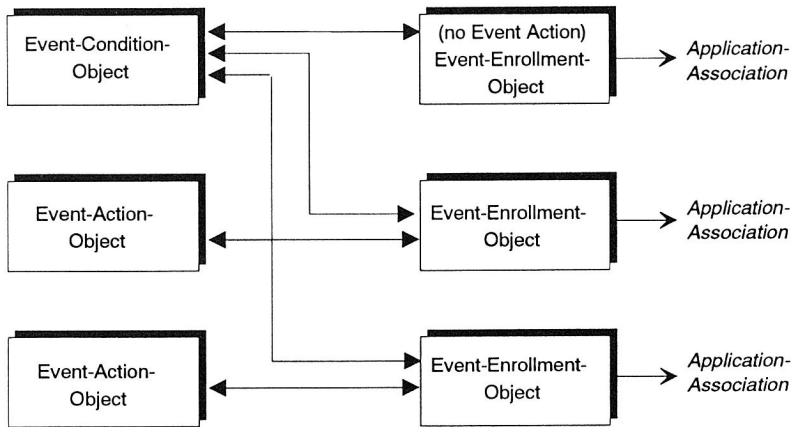


Bild 6.8: Beziehungen zwischen den *Event-Management*-Objekten

Es folgt die formale Darstellung der Objekte und eine kurze Erläuterung zu ihren wichtigsten Attributen.

*Object: Event Condition*

*Key Attribute: Event Condition Name*

*Attribute: MMS Deletable (TRUE, FALSE)*

*Attribute: Event Condition Class (NETWORK-TRIGGERED, MONITORED)*

*Attribute: State (DISABLED, IDLE, ACTIVE)*

*Attribute: Priority*

*Attribute: Severity*

*Attribute: List Of Event Enrollment Reference*

*Constraint: Event Condition Class = MONITORED*

*Attribute: Enabled (TRUE, FALSE)*

*Attribute: Alarm Summary Reports (TRUE, FALSE)*

*Attribute: Monitored Variable Reference*

*Attribute: Evaluation Interval*

*Attribute: Time Of Last Transition to Active*

*Attribute: Time Of Last Transition to Idle*

Das *Event-Condition*-Objekt wird identifiziert über seinen Namen. Ob das Objekt durch MMS löscherbar ist, kann eingestellt werden ("*MMS Deletable*"). Eine grundsätzliche Unterscheidung zum Typ des Ereignisses wird durch "*Event-Condition-Class*" festgelegt: Das Ereignis kann vom Kommunikationssystem oder vom Server selbst ausgelöst werden. Letzteres ist abhängig vom Wert einer referierten booleschen Variablen ("*Monitored Variable Reference*"). Der Zustand ("*State*") des Ereignisses hat bei "*NETWORK-TRIGGERED*"-Ereignissen immer den Wert "*DISABLED*", andernfalls kann mit diesem Attribut die Bezugnahme auf die Variable aufgehoben werden. "*IDLE*" bedeutet den Variablenwert "*FALSE*", "*ACTIVE*" bedeutet den Variablenwert "*TRUE*". "*Priority*" und "*Severity*" machen Angaben über die Wichtigkeit des Ereignisses innerhalb des VMDs bzw. hinsichtlich seiner Auswirkungen innerhalb eines Prozesses. "*List Of Event Enrollment Reference*" nimmt Bezug auf die betroffenen *Enrollment*-Objekte.

Für ein Ereignis der Klasse "*MONITORED*" sind weitere Attribute festgelegt: Mit "*Enabled*" kann die Verknüpfung mit dem *Event-Enrollment*-Objekt freigegeben bzw. gesperrt werden. Das Attribut "*Alarm Summary Reports*" entscheidet, ob das Ereignis in Alarmübersichten aufgenommen wird. Das "*Evaluation Interval*" gibt an, in welchem Zeitabstand (in Millisekunden) der Zustand des *Event-Condition*-Objekts abgefragt wird. "*Time Of Last Transition To Active*" ist der Zeitpunkt des letzten Zustandsüberganges nach "*ACTIVE*", "*Time Of Last Transition To Idle*" analog dazu bezogen auf "*IDLE*".

*Object: Event Action*

*Key Attribute: Event Action Name*

*Attribute: MMS Deletable*

*Attribute: Confirmed Service Request*

*Attribute: List Of Modifier*

*Attribute: List Of Event Enrollment Reference*

Die Bedeutung der Attribute "*Event Action Name*" und "*MMS Deletable*" ist analog zu denen des *Event-Condition*-Objekts. Mit "*Confirmed Service Request*" kann ein Dienst angegeben werden, der beim Eintritt des Ereignisses ausgeführt werden soll. "*List Of Modifier*" nimmt generell Bezug auf MMS-Dienste, die von dem Ereignis abhängig



gemacht werden. "*List Of Event Enrollment Reference*" stellt die Verbindung zu dem zugehörigen *Enrollment-Objekt* und damit dem *Event-Condition-Objekt* her.

*Object: Event Enrollment*

*Key Attribute: Event Enrollment Name*

*Attribute: MMS Deletable (TRUE, FALSE)*

*Attribute: Enrollment Class (MODIFIER, NOTIFICATION)*

*Attribute: Event Condition Reference*

*Attribute: Event Condition Transitions (DISABLED-TO-ACTIVE, DISABLED-TO-IDLE, IDLE-TO-ACTIVE, IDLE-TO-DISABLED, ACTIVE-TO-IDLE, ACTIVE-TO-DISABLED, ANY-TO-DELETED)*

*Attribute: Application Association Local Tag*

*Constraint: Enrollment Class = MODIFIER*

*Attribute: Invoke ID*

*Attribute: Remaining Acceptable Delay*

*Constraint: Enrollment Class = NOTIFICATION*

*Attribute: Notification Lost (TRUE, FALSE)*

*Attribute: Event Action Reference*

*Attribute: Acknowledgement Event Condition Reference*

*Attribute: Duration (CURRENT, PERMANENT)*

*Attribute: Client Application*

*Attribute: Alarm Acknowledgement Rule (NONE, SIMPLE, ACK-ACTIVE, ACK-ALL)*

*Attribute: Time Active Acknowledged*

*Attribute: Time Idle Acknowledged*

*Attribute: State (DISABLED, IDLE, ACTIVE, IDLE-NO-ACK-I, IDLE-NO-ACK-A, ACTIVE-NO-ACK-A, ACTIVE-ACKED, IDLE-ACKED)*

"*Enrollment Class*" kennzeichnet den Typ des Objekts: *Event-Enrollment-Objekte* der Klasse "*MODIFIER*" beeinflussen die Ausführung von MMS-Diensten generell (s. auch "*List Of Modifier*" beim *Event-Condition-Objekt*). Bei Objekten der Klasse "*NOTIFICATION*" wird eine Benachrichtigung an den *Client* geschickt. "*Notification Lost*" gibt an,

ob Benachrichtigungen, die an den *Client* geschickt werden sollten, verloren gingen. "*Event Action Reference*" stellt den Bezug zum *Event-Action*-Objekt her. "*Acknowledgement Event Condition Reference*" dient der Quittierung einer Ereignisbedingung, die vom Kommunikationssystem ausgelöst wird ("*NETWORK-TRIGGERED Event-Condition-Object*"). "*Duration*" gibt an, ob die Lebensdauer des *Event-Enrollment*-Objekts von der Anwendungsverbindung ("*CURRENT*") oder vom *VMD* ("*PERMANENT*") abhängig ist. "*Client Application*" weist auf den Anwendungsprozeß des *Clients* hin. Dieser muß Ereignisbenachrichtigung in Abhängigkeit von den Festlegungen in "*Alarm Acknowledgement Rule*" quittieren. "*Time Active Acknowledged*" ist die Zeit der letzten Quittierung beim Zustandsüberganges des *Event-Condition*-Objekts nach "*ACTIVE*", "*Time Idle Acknowledged*" ist analog dazu bei Zustandsübergang nach "*IDLE*".

"*Application Association Local Tag*" ist ein vom Typ des *Event-Enrollment*-Objekts unabhängiges Attribut und bezeichnet die Anwendungsverbindung, über die die Benachrichtigung geschickt werden soll.

*Event-Management*-Objekte werden in Sensor-/Aktor-Umgebungen für **zyklische** Kommunikationsvorgänge benötigt. Im folgenden wird eine exemplarische Lösungsmöglichkeit dafür vorgestellt, welche Objektausprägungen sich den "**spontan/-zyklischen**" Datentransfermodus für Stationen mit beschränkter Eigeninitiative eignen.

Bild 6.9 zeigt die Ausprägungen der benötigten Objekte für "spontan/zyklische" Kommunikation. Um möglichst wenige Dienste verwenden zu müssen, wurde auf Dienste zum Schaffen und Löschen von *Event-Management*-Objekten verzichtet ("*DefineEventCondition Service*", "*DefineEventEnrollment Service*", "*DefineEventAction Service*", "*DeleteEventCondition Service*" ...). Somit sind alle *Event-Management*-Objekte **nicht** von Diensten löscherbar ("*MMS Deletable* = *FALSE*"), die Lebensdauer des *Event-Enrollment*-Objekts entspricht der des gesamten *VMDs* ("*Duration* = *PERMANENT*"). Das *Event-Enrollment*-Objekt ist vom Typ "*NOTIFICATION*", denn bei Eintritt des Ereignisses erfolgt eine Meldung an die "Sensor-Anwendung" des *Clients*. Darin enthalten ist die Mitteilung des Meßwerts vom Sensor, denn das *Event-Action*-

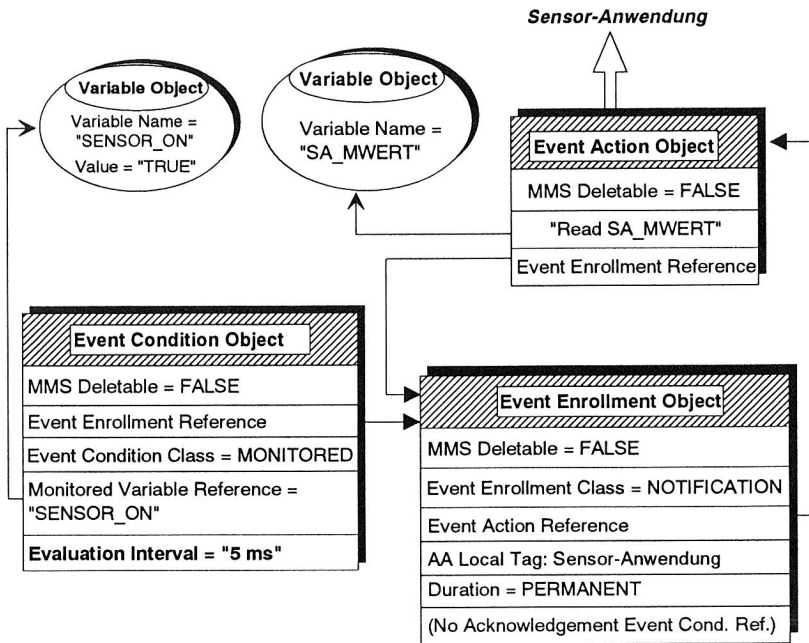


Bild 6.9: Verwendung von *Event-Management*-Objekten bei spontan/zyklischer Datenübertragung

Objekt beinhaltet den Dienst "Read" bezogen auf die Variable "SA\_MWERT".

Da der Sensor ein Ereignis von sich aus melden kann, ist das *Event-Condition*-Objekt im zugehörigen *Server* vom Typ "MONITORED". Die Ereignisbedingung ist abhängig vom Wert der boolschen Variablen "SENSOR\_ON". Die Eigenheit des hier dargestellten zyklischen Mechanismus besteht darin, daß diese Variable mit einem konstanten Wert belegt ist ("TRUE"). Damit gibt die Dauer des Abfrageintervalles ("*Evaluation Intervall* = 5 ms") den Zyklustakt an, bei dem der Sensorwert gelesen und dem *Client* übermittelt wird.

Der Modus für "spontanen/sporadischen" Datentransfer kann prinzipiell gleichermaßen mittels der *Event-Management*-Objekte und entsprechender Ausprägung ihrer Attribute durchgeführt werden. Allerdings wird für eine Eingrenzung der pro Gerätetyp notwendigen Dienstmenge nur die Verwendung des *Information-Report*-Dienstes **ohne**

*Event-Management* vorgeschlagen (s. Abschnitt 6.2.9 "Konformität").

Bei "*stimuliertem/zyklischem*" Datentransfer erfolgt die **zyklische** Abwicklung durch den *Client* - evtl. unter Verwendung des *Event-Managements* oder lokaler, nicht MMS-eigener Entscheidungen. Weil der Sensor/Aktor als *Server* nur auf die erhaltenen (zyklischen) Dienstanforderungen antwortet, müssen keine diesbezüglichen Mechanismen vorgesehen werden.

## 6.2.7 Verwendung der MMS-Dienste

Erweiterungen an MMS-Core-Diensten sind nicht notwendig, da die verwendeten Objekte aus MMS-Core unverändert übernommen wurden.

### 6.2.7.1 Definition und Verwendung von anwendungsspezifischen Diensten

Das Einfachgerät beherrscht keine Prozeßverwaltung mit Abbildung auf *Program-Invocations*. Zum Abbrechen des Betriebs wird ein spezieller Stop-Dienst, zur Wiederaufnahme ein Start-Dienst eingeführt. Die abstrakte Syntax für *Request*- und *Response*-Dienstprimitiv lautet:

```
AdditionalService-Request ::= CHOICE {
    vMDStop    [0] IMPLICIT VMDStop-Request,
    vMDStart   [1] IMPLICIT VMDStart-Request
}
```

```
AdditionalService-Response ::= CHOICE {
    vMDStop    [0] IMPLICIT VMDStop-Response,
    vMDStart   [1] IMPLICIT VMDStart-Response
}
```

Zusätzliche Fehlerdienste (*Additional-Service-Error*) oder unbestätigte Dienste (*Additional-Unconfirmed-Service*), wie sie Companion-Standards grundsätzlich erlauben, werden nicht benötigt.

Parameter-Name	Request	Indication	Response	Confirmation
Argument	M	M (=)		
Result (+)			S	S (=)
Result (-)			S	S (=)
Error Type			M	M (=)

"M" : Muß-Parameter  
"S" : Auswahl-Parameter  
"=" : Indication-/Confirmation-Parameter syntaktisch  
gleich mit Request-/Response-Parameter

Bild 6.10: *VMD-Stop*- und *VMD-Start*-Dienstprimitive

Die Argumente der *VMD-Stop*- und *VMD-Start*-Dienstprimitive zeigt Bild 6.10. *Request* und *Indication* haben kein Argument. *Response* und *Confirmation* teilen entweder ein positives (*Result(+)*) oder negatives (*Result(-)*) Ergebnis mit. Im negativen Fall wird ein Fehlercode gemäß /52/ übergeben.

Dienst	VMD-Attribut	Wert nach Durchführung des Dienstes
VMD-Stop	VMD Logical State	LIMITED-SERVICES-PERMITTED
	VMD Physical State	PARTIALLY-OPERATIONAL
VMD-Start	VMD Logical State	STATE-CHANGES-ALLOWED
	VMD Physical State	OPERATIONAL

Bild 6.11: VMD-Attribute bei *VMD-Stop* und *VMD-Start*

Mit dem *VMD-Stop*-Dienst kann der MMS-Anwender das Einfachgerät im Betrieb unterbrechen, mit dem *VMD-Start*-Dienst wieder aktivieren. Der Start-Dienst bedient auch Einfachgeräte, die nicht von sich aus nach dem Einschalten ihre Funktion aufnehmen. Durch die genannten Dienste wird das *VMD* in definierte Zustände übergeführt (s. Bild 6.11).

### 6.2.7.2 Initiate-Dienst und -Protokoll

Der *Initiate*-Dienst beinhaltet einen Parameter (*Initiate-Request-Detail*), der für die Benutzung durch Companion-Standards definiert wurde (s.Bild 6.12).

Parameterbezeichnung	Request	Indication	Response	Confirmation
Proposed Version Number	M	M (=)		
Proposed Parameter CBB	M	M (=)		
Services Supported Calling	M	M (=)		
Additional Services Supported Calling	M	M (=)		
Negotiated Version Number			M	M (=)
Negotiated Parameter CBB			M	M (=)
Services Supported Called			M	M (=)
Additional Services Supported Called			M	M (=)

"M": Mußparameter

"=": Indication-/Confirmation- Parameter syntaktisch gleich mit Request-/Response- Parameter

Bild 6.12: Initiate-Request-Detail

a) *Proposed-Version-Number*:

Versionsbezeichnung des Companion-Standards,

b) *Proposed-Parameter-CBB*:

Angaben zur Komplexität der verwendeten Variablen (*Conformance-Building-Blocks*),

c) *Services-Supported-Calling*:

verwendete MMS-*Core*-Dienste,

d) *Additional-Services-Supported-Calling*:

verwendete, erweiternde Companion-Standard-Dienste.

Angaben zu c) und d) werden in Form von Bit-Strings gemacht: eine "1" an der entsprechenden Stelle bedeutet, daß dieser Dienst verwendet werden soll.

Der für den Sensor/Aktor-Companion-Standard gültige *Initiate-Request-Detail*-Parameter lautet:

```

InitRequestDetail ::= SEQUENCE {
    proposedVersionNumber    [0] IMPLICIT Integer16,
    proposedParameterCBB     [1] IMPLICIT ParameterSupport-Options,
    servicesSupportedCalling  [2] IMPLICIT ServicesSupport-Options,
    additionalSupportedCalling [3] IMPLICIT AdditionalSupport-Options
}

```

```

AdditionalSupportOptions ::= BITSTRING {
    vMDStop           (0),
    vMDStart          (1)
}

```

### 6.2.8 Standard-Objekte

Im folgenden werden MMS-Objekte angegeben, die für alle Einfachgeräte Gültigkeit haben. Es werden keine anderen als *Named-Variable*-Objekte benutzt, da sie für die Verwendung im Sensor/Aktorbereich mit Meß- und Stellwerten ausreichen. Der Variablentyp "Floating-Point" wurde gewählt, um möglichst alle Meßwerttypen zu erfassen und z.B. auch Werte von Kraft- und Momentensensoren (meist: 3 Kräfte, 3 Momente) in geeigneter Weise darstellen zu können. "SA\_" kennzeichnet die Zugehörigkeit zum Sensor/Aktor-Companion-Standard.

#### **SA\_MWERT** - Meßwert vom Sensor

*Object: Named Variable*

*Key Attribute: Variable Name = vmd-specific "SA\_MWERT"*

*Attribute: MMS Deletable = FALSE*

*Attribute: Type Description =*

*floating-point { format-width 32, exponent-width 8 }*

*Attribute: Access Method*

**SA\_SWERT** - Stellwert für Akteur

*Object: Named Variable*

*Key Attribute: Variable Name = vmd-specific "SA\_SWERT"*

*Attribute: MMS Deletable = FALSE*

*Attribute: Type Description =*

*floating-point { format-width 32, exponent-width 8}*

*Attribute: Access Method*

**SA\_PARANZ** - Parameteranzahl

*Object: Named Variable*

*Key Attribute: Variable Name = vmd-specific "SA\_PARANZ"*

*Attribute: MMS Deletable = FALSE*

*Attribute: Type Description = integer 8*

*Attribute: Access Method*

**SA\_PAR** - Parameter

*Object: Named Variable*

*Key Attribute: Variable Name = vmd-specific "SA\_PAR"*

*Attribute: MMS Deletable = FALSE*

*Attribute: Type Description =*

*array { numberOfElements sA-PARANZ-value,  
          elementType floating-point  
              { format-width 32, exponent-width 8}  
          }*

*Attribute: Access Method*



## 6.2.9 Konformität

Mit Hilfe von Konformitätsklassen kann der Kommunikationsumfang des Einfachgerätes auf die eigenen Bedürfnisse bzw. die des Kommunikationspartners abgestimmt werden. Insbesondere werden die für die Kommunikation mit Sensoren und Aktoren ausreichenden MMS-Funktionen explizit festgelegt. Da der vorliegende Standardisierungsentwurf für Sensoranwendungen prinzipiell geeignet sein soll, können Meßwertaufnehmer mit fortgeschrittener Leistung (mit Mikroprozessortechnik) auf ein reichhaltigeres MMS-Repertoire in Form von höheren Konformitätsklassen zurückgreifen.

### 6.2.9.1 Basis-Konformitätsklassen

Zur Mindestfunktionalität eines Einfachgeräts gehören folgende *Service-Conformance-Building-Blocks* und *Parameter-Conformance-Building-Blocks*:

*Initiate*, *Conclude*, *Abort*, *VNAM* und *NEST* = 1.

#### Sensoren:

Für die Datentransfermodi zur Lieferung von Meßwerten sind folgende *Service-Conformance-Building-Blocks* vorgesehen:

spontan/sporadisch:	<i>InformationReport</i>
spontan/zyklisch:	<i>Read</i> , <i>EventNotification</i>
auf Anforderung/sporadisch:	<i>Read</i>
auf Anforderung/zyklisch:	<i>Read</i> , <i>EventNotification</i>

Zur Erläuterung:

#### *InformationReport*:

Der MMS-Server liefert den Meßwert unaufgefordert an den MMS-Client.

*Read*: Der MMS-Server liefert den Meßwert nach Aufforderung (*Read-Request*) an den MMS-Client.

**EventNotification:**

Der *MMS-Server* teilt dem *MMS-Client* das Eintreffen eines Ereignisses mit.

**Aktoren:**

Für die Datentransfermodi zum Eintragen von Stellwerten sind folgende *Service-Conformance-Building-Blocks* vorgesehen:

sporadisch: *Write*

zyklisch: *Write*

Bild 6.13 gibt einen Überblick zu den Basis-Konformitätsklassen ("BKK").

Service-CBB \ Konformitäts-klasse	BKK1	BKK2	BKK3	BKK4
Initiate	X	X	X	X
Conclude	X	X	X	X
Abort	X	X	X	X
Information-Report	X			
Read		X	X	
Event-Notification			X	
Write				X

BKK1 : Sensor mit spontaner, sporadischer Meßwertausgabe

BKK2 : Sensor mit stimulierter, sporadischer Meßwertausgabe

BKK3 : Sensor mit zyklischer Meßwertausgabe

BKK4 : Aktor mit sporadischer oder zyklischer Stellwertversorgung

Bild 6.13: *Service-Conformance-Building-Blocks* von Basiskonformitätsklassen

**Anmerkung:**

Somit entfallen auf die einfachsten Sensor- bzw. Aktortypen genau vier *Service-CBBs* (Dienste) und 2 *Parameter-CBBs* (Angaben zum Variablentyp). Für zyklische Sensoren muß der Dienst *Event-Notification* hinzugenommen werden, da mit Hilfe des *Event-Managements* das zyklische Verhalten abgebildet wird. Der Mindestumfang von *MMS-Servern* für Aktoren vom "sporadischen" oder "zyklischen Typ" ist identisch, da in

diesem Fall **der MMS-Client** die zyklische Übertragung von Stellwerten mit Hilfe des *Event-Managements* ausführt.

**6.2.9.2 Höhere Konformitätsklassen**

Für komplexere Anwendungsfunktionen gibt es folgende *Service-Conformance-Building-Blocks*.

- Befehle: *VMD-Stop, VMD-Start,*
  - Lebenszeichen: *UnsolicitedStatus,*
  - Identifizierung: *Identify,*
  - Statusabfragen: *Status,*
  - Fehlermeldung: *UnsolicitedStatus,*
  - Parametrisierung: *Write,*
- sowie der *Parameter-Conformance-Building-Block STR1*.

Es stehen vier höhere Konformitätsklassen ("HKK") (s. Bild 6.14) zur Verfügung:

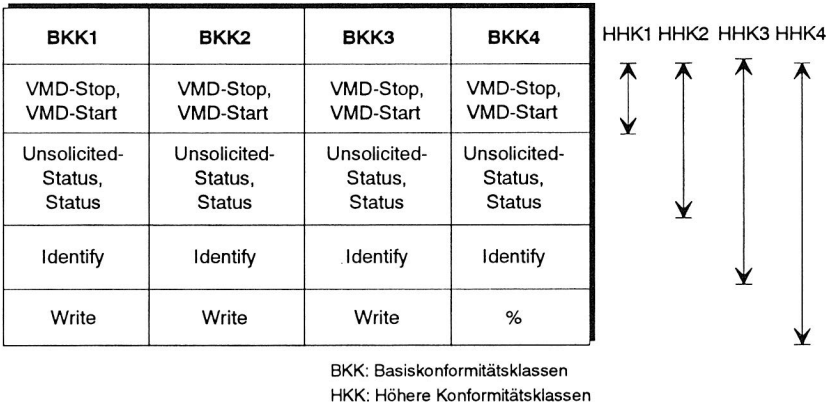


Bild 6.14: Höhere Konformitätsklassen

- HKK1 beinhaltet eine der Basiskonformitätsklassen (BKK1 oder BKK2 oder BKK3 oder BKK4) und schließt die *Service-Conformance-Building-Blocks* **VMD-Stop** und **VMD-Start** mit ein.
- HKK2 beinhaltet HKK1 sowie die *Service-CBBs* **UnsolicitedStatus** und **Status**.
- HKK3 beinhaltet HKK2 sowie den *Service-CBBs* **Identify**.
- HKK4 für Sensoren (BKK1-3) beinhaltet HKK3 und den *Service-CBB* **Write** und den *Parameter-CBB* **STR1**.

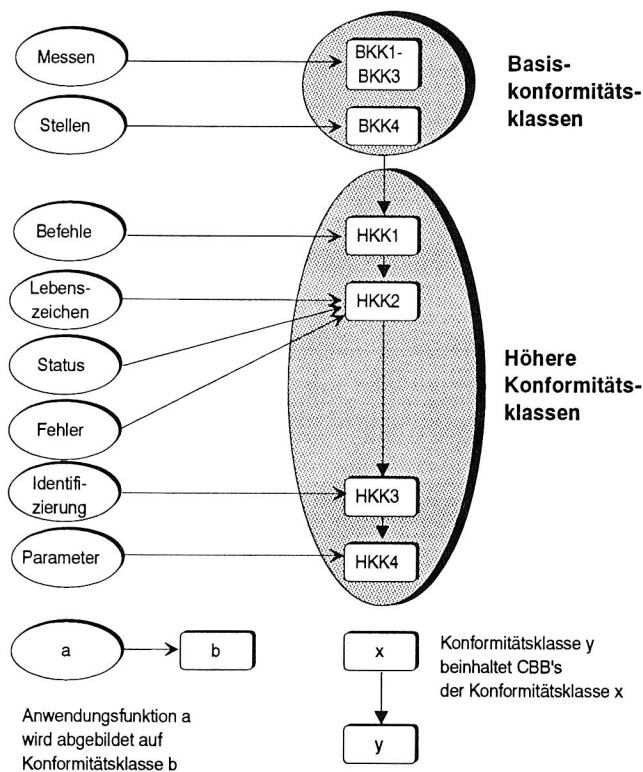
AnwendungsfunktionenKonformitätsklassen

Bild 6.15: Abbildung von Anwendungsfunktionen auf Konformitätsklassen

Bild 6.15 zeigt, welche (MMS-unabhängigen) Anwendungsfunktionen welchen Konformitätsklassen zugeordnet werden.

### 6.2.10 Anwendungsbeispiel

Als Anwendungsbeispiel wird eine Montagezelle gewählt, bei der ein Industrieroboter die Bauteile eines Lüftermotors montiert (s. Bild 6.2). Die Ankunft eines Bauteils wird vom Reflexionstaster detektiert, woraufhin eine speicherprogrammierbare Steuerung das Transportband stoppt, damit das Handhabungsgerät ein Bauteil aufnehmen kann.

Der Reflexionstaster gehört zum Typ mit spontaner, sporadischer Kommunikation und wird der Basiskonformitätsklasse 1 zugeordnet. Das zugehörige *VMD* beinhaltet die Objekte *SA\_MWERT*, *SA\_PARANZ* und *SA\_PAR*. Zur Werkstückkontrolle gehören folgende Kommunikationsphasen:

- a) Initialisierung: Abgleich der Kommunikationseigenschaften zwischen der übergeordneten speicherprogrammierbaren Steuerung und dem Reflexionstaster.
  - b) Parametrisierung: Hellschaltung beim Reflexionstaster (Nullpotential: "keine Reflexion", Betriebsspannung: "Reflexion").
  - c) Meßwertübertragung: Meldung beim Erkennen des Werkstücks.
- a) Initialisierung: Die speicherprogrammierbare Steuerung sendet dem Reflexionstaster folgende PDU:

```

initiate-RequestPDU {
    proposedMaxServOutstandingCalling 1,
    proposedMaxServOutstandingCalled 1,
    initRequestDetail {
        proposedVersionNumber 1,
        proposedParameterCBB { str1, vnam },
        servicesSupportedCalling {
            Read, Write, DefineEventCondition,
            DefineEventAction, Identify, Status },
        additionalSupportedCalling { vMDStop, vMDStart }
    }
}
```

Der Reflexionstaster antwortet wie folgt:

```
initiate-ResponsePDU {
    proposedMaxServOutstandingCalling 1,
    proposedMaxServOutstandingCalled 1,
    initRequestDetail {
        proposedVersionNumber 1,
        proposedParameterCBB { str1, vnam },
        servicesSupportedCalled { InformationReport, Write },
        additionalSupportedCalling { }
    }
}
```

- b) Parametrisierung: Die speicherprogrammierbare Steuerung stellt den Reflexionstaster dadurch auf "Hellschaltung" ein, daß sie die erste Komponente von SA\_PAR mit "1" belegt.

```
confirmed-RequestPDU {
    invokeID 1,
    write {
        variableAccessSpecification listOfVariable
        {{ variableSpecification name vmd-specific "SA_PAR", alternateAccess
            unnamed { selectAccess index 0 }}},
        listOfData floating-point '0001'H
    }
}
```

- c) Meßwertübertragung: Der Reflexionstaster meldet die Markierung am Werkstück:

```
unconfirmedPDU { informationReport
    { variableAccessSpecification listOfVariable
        {{ VariableSpecification name vmd-specific "SA_MWERT" }},
        listOfAccessResult
        { success floating-point '0000'H }
    }
}
```

## 7 Zusammenfassung und Ausblick

Durchgängige offene Kommunikation gemäß dem *Computer-Integrated-Manufacturing* ist nach heutigem Stand nur durch die Verwirklichung von MAP zu erreichen. Allgemeine Übereinstimmung besteht darin, daß die Anforderungen innerhalb von flexiblen Produktionszellen von "Full MAP" nicht erfüllt werden.

Um das Kommunikationsverhalten systematisch betrachten zu können, wurde zunächst ein Klassifikationsschema erarbeitet, das die wesentlichen Eigenschaften der zelleninternen Kommunikation wie z.B. "Offene Kommunikation", "Stationstypen" usw. darlegt.

Mit dem Leitgedanken "MAP-Konformität", wurden in einem zweiten Schritt drei für das Gerätespektrum flexibler Produktionszellen repräsentative Kommunikationsarchitekturen mit MMS und LABORINTH konzipiert. Das Kommunikationskonzept LABORINTH ist Architekturbestandteil, weil damit wichtige Kriterien aus dem Klassifikationsschema erfüllt werden können:

- Die Integration von Einfachstationen kann durch die "Zwei-Klassen-Struktur" der Stationen verwirklicht werden. Leistungsfähige Steuerungen (RC, SPS) werden auf *Steuerstationen*, Sensoren auf *Unterstationen* abgebildet.
- Teilnetze können deshalb gebildet werden, weil LABORINTH-*Steuerstationen* über *Untersysteme* hinweg kommunizieren können.
- Das dynamische An- und Abmelden von LABORINTH-*Prozessen* erlaubt Konfigurationsänderungen bei den kommunizierenden Automatisierungsgeräten im laufenden Betrieb.

Architekturvarianten "A" und "C" wurden beispielhaft realisiert und Leistungsbetrachtungen unterzogen.

Für die Lösung der Teilaufgabe "MAP-konforme Sensoren und Aktoren" wurden in vorliegender Arbeit zwei Wege eingeschlagen:

- Architekturvariante "B" ist repräsentativ für derzeitige Sensor-/Aktor-Vertreter, die nur über rudimentäre informationsverarbeitende Funktionen verfügen. Deshalb muß die Anbindung an die MAP-Welt eine "Betreuungsinstanz" herstellen, auf die die umfangreichen und komplexen Anwendungsfunktionen ausgelagert werden. Diese Lösung zeigt auf, wie die Einführung von MAP/MMS in flexible Produktionszellen stufenweise erfolgen kann.
- Die Entwicklung der "intelligenten Meßumformer" ("smart transmitters") /7/ stellt die verstärkte Unterstützung von Anwendungs- **und** Kommunikationsfunktionen durch Mikroelektronik in Aussicht. Die Leistungsfähigkeit wird aber in absehbarer Zeit aus Gründen der Miniaturisierung noch begrenzt bleiben (keine Multiprozessoren, geringe Speicherkapazität) /132/. Um die Implementierung des MAP-Anwendungsprotokolls zu erleichtern, wurde deshalb ein MMS-Companion-Standard skizziert, der die verwendeten MMS-Modelle (Objekte und Dienste) auf ein Mindestmaß beschränkt: Lese-Dienste (*Read*) für Sensoren, Schreib-Dienste (*Write*) für Aktoren.

Die in vorliegender Arbeit dargestellten Ergebnisse präzisieren die kommunikationstechnischen Anforderungen in flexiblen Produktionszellen und geben Anleitungen dafür, wie der durchgängige Informationsverbund im Produktionsbetrieb bis in die unmittelbare Nähe zum technischen Prozeß verwirklicht werden kann. Für die inhaltliche Fortführung der Arbeit kommen folgende Themen in Frage:

- Das dargelegte Schema dient dazu, Kommunikationssysteme in flexiblen Produktionszellen zu klassifizieren. Insofern können bestehende Feldbussysteme (BITBUS, Arcnet, CAN ...) anhand der aufgezeigten Kriterien verglichen und bewertet werden.
- Das Konzept der Einfach-Stationen mit MMS-Companion-Standard wurde bisher nur insoweit realisiert, als daß die Integration in die LABORINTH-Welt geschaffen wurde. Weiterführende Arbeiten können anhand der praktischen Implementierung des aufgezeigten MMS-Umfangs u.a. auch das Leistungsverhalten beurteilen.



- Für die Leistungsbewertung liegen allgemein gültige Kennwerte (Durchsatz, Antwortzeit) vor. Eine aussagekräftigere Bewertung könnte man unter Verwendung von "Benchmarks" bzw. "Metriken" vornehmen: Dazu müssen für die Fertigungsumgebung typische Kommunikationsanwendungen normiert werden.

Leitgedanke für die aufgezeigten Architekturvarianten war, durch strenge MMS-Konformität auf Anwendungsebene MAP in die Prozeßumgebung einzuführen. Den funktionalen Bedürfnissen flexibler Produktionszellen kam entgegen, daß unterhalb der Anwendungsebene beliebige Übertragungs- und Transportsysteme zugelassen sind.

Für die Zukunft ist vorstellbar, diese Vorgehensweise auch auf andere Anwendungstechnologien, wie z.B. Laborautomatisierung oder Prozeßleittechnik, zu übertragen. Die objektbasierten Sprachkonstrukte von MMS erscheinen für die Programmierung verteilter technischer Systeme generell geeignet.

## 8 Literaturverzeichnis

1. Feldmann, K., Schrödel, O.:  
Möglichkeiten und Chancen der rechnergestützten Produktion für den Mittelstand.  
Vortrag anlässlich der 12. Internationalen Fachmesse für Metallverarbeitung  
(FAMETA), Nürnberg, Juni 1990.
2. Bärnreuther, B. et al.:  
Die Standardisierung von Datenschnittstellen in der Montageautomatisierung.  
Informatik-Spektrum 12 (1989): 312-320.
3. Proc. "Standardisierung der CAD/CAM-Schnittstelle zur Bestückung von Leiterplatten". Erlangen, 9. und 10.9.1991.
4. Grabowski, H., Anderl, R., Schmidt, M.:  
STEP: Die Beschreibung von Produktstrukturen mit dem Teilmodell PSCM.  
**VDI-Z** 134, Nr. 3 (1992): 51-55.
5. Lauber, R.:  
Prozeßautomatisierung - Band 1: Automatisierungsstrukturen, Prozeßrechensysteme, Echtzeitprogrammierung, Zuverlässigkeits- und Sicherheitstechnik.  
Springer-Verlag, Berlin, Heidelberg, New York, 1989.
6. Kaminski, M., A., Dwivedi, P.:  
General Motors Manufacturing Automation Protocol.  
GM Technical Center- Warren, 1987.
7. Neumann, R., Funk, B.:  
Digitaler Temperaturregler - Intelligent und flexibel.  
Elektronik Entwicklung 10/91, 96-98.
8. DIN-Fachbericht 15:  
Normung von Schnittstellen für die rechnerintegrierte Produktion (CIM). 1987.
9. Greene, T., J., Sadowski, R., P.:  
Cellular Manufacturing Control.  
Journal of Manufacturing Systems, Vol. 2, No.2 (1983): 137-145.
10. Spur, G., Auer, B., H.:  
Die automatische Handhabung bei flexiblen Fertigungszellen.  
wt-Z. ind. Fertig. 65 (1975): 117-123.
11. Groha, A.:  
Universelles Zellenrechnerkonzept für flexible Fertigungssysteme. Dissertation,  
München, Springer 1988.

12. Herzog, U., Gora, W. et al.:  
Kommunikation in der rechnerintegrierten Fabrik.  
Proc. Fachtagung "Rechnerintegrierte Produktionssysteme", FAU, 1987
13. Kief, H., B.:  
NC/CNC-Handbuch '90.  
NC-Handbuch-Verlag, Michelstadt, 1990.
14. Decotignie, J.-D., Pleineveaux, P.:  
Field Bus in the Hierarchy of Factory Communications: the Limits of a Classical Approach.  
IEEE Workshop on Factory Communications, Gaithersburg,  
17./18.3.1987.
15. Herzog, U.:  
unveröffentlichtes Skript zur Vorlesung "Kommunikationssysteme", IMMD VII,  
Lehrstuhl Rechnerarchitektur und Verkehrstheorie, Erlangen, WS 1991/1992.
16. Lessard, A., Gerla, M.:  
Wireless Communications in the Automated Factory Environment.  
IEEE Network, Vol.2, No.3 (1988): 64-69.
17. Motorola GmbH, Produktbereich ALTAIR:  
Grundlagen der Funktechnik - Funksignale als Signalübertragungsmedium.  
Produktbegleitende Unterlagen zur CeBit '92.
18. Tanenbaum, A.,S.:  
Computer Networks. Prentice-Hall, Englewood Cliffs, 1981.
19. Wybraniec, D.:  
Broadcast/Multicast. Informatik-Spektrum (1990): 130-132.
20. Schicker, P.:  
Datenübertragung und Rechnernetze.  
Teubner-Verlag, Stuttgart, 1988.
21. Strayer, W., T., Weaver, A. C.:  
Performance Measurement of Data Transfer Services in MAP.  
IEEE Network, Vol.2, Nr. 3 (1988): 75-81.
22. Ray, A.:  
Networking for Computer-Integrated Manufacturing.  
IEEE Network, May 1988-Vol.2, No.3, P. 40-47.
23. DIN 43000:  
Informationsverarbeitung - Begriffe.  
Beuth-Verlag, Berlin, 1985.

- 
24. Rzehak, H.:  
Die Abwicklung von Realzeit-Aufträgen in MAP-Netzen.  
**atp** 31, 10 (1988): 488-495.
  25. ISO IS 7498:  
Information Processing Systems - Open Systems Interconnection - Basic Reference Model. 1984.
  26. Giese, E. et al.:  
Dienste und Protokolle in Kommunikationssystemen.  
Springer-Verlag, Berlin, 1985.
  27. Stallings, W.:  
Local Networks.  
Macmillan Publishing Company, New York, Collier Macmillan Publisher, London, 1987.
  28. Dietsch, H., Ulrich, R.:  
FDDI (Fiber Distributed Data Interface).  
Informatik-Spektrum 14 (1991): 159-160.
  29. Rose, M., T.:  
The open book: a practical perspective on OSI.  
Prentice-Hall, London, 1990.
  30. Proc. European MAP Forum "EMUG MAP Events", Vol. 1/2.  
München, 1990.
  31. MAP Workshop: "MAP Migration Strategies", SYSTEC '90, München, 1990.
  32. Watson, K., S.:  
Architecture of the Fraunhofer Conformance Test Systems.  
Proc. ENTERPRISE '88, Baltimore, 1988.
  33. Birtel, P.:  
Conformance Testing for MAP/TOP.  
Proc. ENTERPRISE '88, Baltimore, 1988.
  34. Saenger, F., Theis, M., Wieser, M.:  
Konformitätstest - ein notwendiger Schritt für den PROFIBUS auf dem Weg zur offenen Kommunikation im Feldbereich.  
**atp** 33, 1 (1991): 37-43.
  35. Katz, M.:  
Anwenderschnittstelle.  
In: Proc. VDI/VDE-GMA-Workshop "Offene Kommunikation im Feldbereich mit PROFIBUS", VDI (Hrsg.), Langen, VDI-Bericht 728, 1989.

- 
36. Katz, M.:  
Modellierung der Anwenderschicht für einen Feldbus.  
Proc. VDI/VDE-GMA-Fachtagung "Informatik für die industrielle Automation-  
INFINA 89", Karlsruhe, VDI-Bericht 732, 1989.
  37. Stadter, W.:  
Device Description Language - Ein wichtiger Beitrag zur herstellerunabhängigen  
Austauschbarkeit von Feld- und Bediengeräten.  
**atp** 34,1 (1992): 21-27.
  38. Kaminski, M., A.:  
Protocols for communicating in the factory.  
IEEE Spectrum, April 1986. 56-62.
  39. EMUG:  
Manufacturing Automation Protocol - Specification Version 3.0  
College of Manufacturing, Cranfield, 1989.
  40. Laforsch, J.:  
ENE '88: Mit MAP und TOP zur offenen Kommunikation.  
**atp** 30,10 (1988): 483-487.
  41. Lederhofer, A., Schwarz, K.:  
Communications for CIM. Open Systems Protocols and Specifications - CNMA  
Technical Overview -. CNMA-Bericht zur Systec, 1990.
  42. Dietsch, H.:  
Feldbus.  
Informatik-Spektrum 13 (1990): 217-220.
  43. DIN 19245 Teil 1:  
PROFIBUS - Übertragungstechnik, Buszugriffs- und Übertragungsprotokoll,  
Dienstschnittstelle zur Anwendungs-Schicht, Management.  
Beuth Verlag, Berlin, 1990.
  44. DIN 19245 Teil 2:  
PROFIBUS - Kommunikationsmodell, Dienste für die Anwendung, Protokoll,  
Syntax, Codierung, Schnittstelle zur Schicht2, Management.  
Beuth-Verlag, Berlin, 1990.
  45. Göddertz, J.:  
PROFIBUS-Protokolle.  
In:  
Proc. VDI/VDE-GMA-Workshop "Offene Kommunikation im Feldbereich mit  
PROFIBUS", VDI (Hrsg.), Langen, VDI-Bericht 728, 1989.

46. Rupprecht, W.:  
Analoge und digitale Signale.  
**at** 38,4 (1990): 124 - 133.
47. DIN IEC 381 Teil 1:  
Analoge Signale für Regel- und Steueranlagen; Analoge Gleichstromsignale.  
Beuth-Verlag, Berlin, 1985.
48. Strohrmann, G.:  
Automatisierungstechnik - Band I, Grundlagen, analoge und digitale Prozeßleitsysteme.  
Automatisierungstechnik - Band II, Stellgeräte, Stecken, Projektabwicklung.  
Oldenbourg, München, 1990.
49. Kennel, R., Weber, R.:  
Datenkommunikation über das Bussystem "SERCOS interface"  
Auswirkungen auf Antriebsregelungen.  
**atp** 33,7 (1991): 363-368.
50. VDI 2880 Blatt 2 Entwurf:  
Speicherprogrammierbare Steuerungsgeräte - Prozeß- und Datenschnittstellen,  
Dez. 1982.
51. Kafka, G.:  
Schnittstellen für die Datenübertragung.  
Elektronik 25/14.12.1984. 76 - 80.
52. ISO IS 9506:  
Manufacturing Message Specification.  
Part I: Service Definition; Part II: Protocol Specification. 1988.
53. ISO IS 9571:  
File Transfer, Access and Management. 1988.
54. ISO IS 8824/8825:  
Abstract Syntax Notation One/Basic Encoding Rules for ASN.1. 1987.
55. Gora, W., Speyerer, R.:  
Automatic Code Generation for ASN.1-Based Protocols.  
Proc. ENTERPRISE '88, Baltimore, 1988.
56. SEMI:  
SEMI SITF Generic Equipment Model for SECS Communications - Draft Version  
2.0, 1990.

- 
57. Secrest et al.  
SECS Communications Handbook.  
GW Associates Inc, Portola Valley, CA, 1989.
  58. Birchak, J. et al.:  
A MAP/SECS Network For Automated Semiconductor Manufacturing.  
GATEWAY, 1988.
  59. Chauhan, V.:  
Real-Time MAP: A Network Architecture Standard for Manufacturing Cells.  
Proc. IEEE Workshop on Factory Communications, Gaithersburg, 1987.
  60. IROFA:  
Presentation of FAIS Project by the Japanese Member Body.  
ISO/TC 184/SC 5/WG 2 N255, 1991.
  61. Normenausschuß Maschinenbau:  
Bericht zur 5. Sitzung des NAM-IA-UA 96.5.2 "Kommunikation und Datenaustausch" am 12.12.1991.
  62. ISO/TC 184/SC 5/WG 2:  
Technical Report of the TCCA Rapporteurs' Group of ISO/TC 184/SC 5/WG 2.  
Identifying User Requirements for Systems Supporting Time Critical Communications. März 1992.
  63. Rodd, M., G, Guo Feng Zhao, Izikowitz, I.:  
RTMMS - An OSI-Based Real-Time Messaging System.  
The Journal of Real-Time Systems, 2, 213-234.
  64. Guo Feng Zhao:  
A Real-Time Messaging System for Distributed Computer Control Systems.  
Candidature for the degree of Doctor of Philosophy, University of Wales, Swansea, 1990.
  65. Kopetz, H., Ochsenreiter, W.:  
Clock synchronization in distributed real-time systems.  
IEEE Transactins on Computers, C-36, 8 (1987): 933-940.
  66. Rzehak, H., Jäger, R.:  
Ein Meß- und Monitorkonzept zur Beurteilung der Realzeitfähigkeit eines MAP-Netzes.  
Proc. Fachtagung "Kommunikation in verteilten Systemen", Stuttgart, 1989.
  67. Garbe, K.:  
Management von Rechnernetzen.  
Teubner-Verlag, Stuttgart, 1991.

- 
68. Hesselbach, J.:  
Möglichkeiten der steuerungstechnischen Integration von Industrierobotern.  
Proc. Internationaler MHI-Kongreß - Montage, Handhabung, Industrieroboter,  
Hannover, 1985.
  69. Pleinevaux, P., Decotignie, J.-D.:  
Time Critical Communication Networks: Fieldbusses.  
IEEE Network, May 1988 - Vol.2, No. 3. 55-63.
  70. Ermlich, H.-J.:  
Sie leisten mehr als man erwartet: Speicherprogrammierbare Steuerungen auf der  
INTERKAMA 89.  
**atp** 32,2 (1990): 70 - 76.
  71. Rembold, U.:  
Entwicklung von Sensoren für die Montageautomatisierung.  
**atp**-Sonderheft Fertigungsautomatisierung 1987. 64-80.
  72. Feldmann, K.:  
unveröffentlichtes Skript zur Vorlesung "Handhabungs- und Montagetechnik",  
FAPS, Erlangen, WS 1990.
  73. Bosse, J.:  
Sensoren für den Einsatz mit Industrierobotern.  
In: Montage, Handhabung, Industrieroboter. Internationaler MHI-Kongreß,  
Hannover, 1985.
  74. Eißler, W., Knappmann, R.-J.:  
Praktischer Einsatz von berührungslos arbeitenden Sensoren.  
Expert-Verlag, Ehningen, 1989.
  75. Nolting, F.-W.:  
Projektierung von Montagesystemen.  
Dissertation, FAU Erlangen-Nürnberg. Hanser-Verlag, 1989.
  76. Bever, M.:  
OSI Application Layer - Entwicklungsstand und Tendenzen -.  
Proc. Tutorium Kommunikation in verteilten Systemen, Stuttgart, 1989.
  77. Chintamaneni, P., R. et al.:  
On Fault Tolerance in Manufacturing Systems.  
IEEE Network, May 1988 - Vol.2, No. 3.
  78. Ulloa, G., Decotignie, J.D.:  
Field Bus: A Real Alternative for Time Critical Networks.  
Proc. EFOC/LAN 88, Amsterdam, 29.6.-1.7. 1988.



- 
79. Warnecke, H.-J., Lindner, H., Gläss, W.:  
Integriertes sensorgestütztes Robotersystem.  
Robotersysteme 4 (1988): 1-8.
  80. Kleines, H., Holzer, J., Zwill, K.:  
Diskussion leistungsrelevanter Aspekte des MAP 3.0 Protokollstacks am Beispiel von MicroMAP.  
In: Wissensbasierte Systeme - Moderne Methoden der digitalen Regelung - Offene Kommunikation - Intelligente Sensorsysteme.  
Proc. Automatisierungstechnik '90, VDI-Berichte 855, September 1990, Baden-Baden.
  81. KCIM AK 2.4 03.06.1989, Internes Arbeitspapier des KCIM-Arbeitskreises 2.4.
  82. KCIM AK 2.4 504.06.89, Internes Arbeitspapier des KCIM-Arbeitskreises 2.4.
  83. Demmelmeier, F.:  
Fehlertolerante Multimikrorechnersysteme für die Prozeßautomatisierung.  
Oldenbourg Verlag GmbH, München, 1988.
  84. Maehle:  
unveröffentlichtes Skript zur Vorlesung "Fehlertolerante Rechnersysteme",  
IMMD3, Erlangen, SS 1985.
  85. Nix, H., G.:  
Fehlertolerant automatisieren durch 1-von-2-Redundanz.  
**atp 33** (1991): 296-301.
  86. Testi, F.:  
Einflüsse auf die Übertragungsleistung von Computer-Netzwerken.  
Werkstatt und Betrieb 123 (1990) 2. 119-121.
  87. Feldmann, K.:  
Rechnergestützte Diagnose von Montagesystemen.  
Montage, Heft 4/1989.
  88. Gora, W.:  
Konzept, Methoden und Werkzeuge für ein universelles Netzmanagement.  
Dissertation, Universität Erlangen-Nürnberg, 1989.
  89. ISO/TC97/SC21/WG4 N406:  
"Fault Management", 4th Working Draft, August 1987.
  90. Gora, W., Langer, R.:  
LAN Protocol Tester B5100.  
DATACOM 7/88. 22-30.

- 
91. Schröder, J.:  
Ein modulares, verteiltes Diagnose-Expertensystem für die Fehlerdiagnose in lokalen Netzen.  
**atp** 32, 11 (1990): 557-565.
  92. Dermla, A., Glöckl-Frohnholzer, J., Marcus, O., W.:  
Analyse der Protokolle in lokalen Netzen bis OSI-Ebene 4.  
**atp** 30, 3 (1988): 144-146.
  93. Schröder, J.:  
Monitoring und Diagnose in lokalen Netzen, Teil 1.  
Elektronik 7/1991. 158-164.
  94. Komischke, M.:  
Einbindung von Sensorsystemen in den Informationsfluß integrierter Produktionssysteme.  
Dissertation, TH Aachen. Fortschr.-Ber., VDI Reihe 8 Nr. 162, VDI-Verlag, Düsseldorf, 1988.
  95. DIN 19241:  
Bitserielles Prozeßbusschnittstellensystem - Teil 1-3. 1982-1988.
  96. DIN 66348 Teil 2:  
Schnittstellen und Steuerungsverfahren für die serielle Meßdatenübermittlung, Mehrpunktverbindung.  
Beuth-Verlag, Berlin, 1988.
  97. Schäper, N.:  
Feldbusse im Leistungsvergleich.  
Elektronik 6 (1991): 126-132.
  98. Henn, O.:  
Der offene Kommunikationsstandard für die Umsetzung von CIM bis in den Feldbereich: PROFIBUS.  
CIM-Management 4 (1990): 55-62.
  99. Bender, K. (Hrsg.):  
PROFIBUS - Der Feldbus für die Automation.  
Carl Hanser Verlag München Wien, 1990.
  100. Dietsch, H., Ögrük, M., E., Ulrich, R.:  
LABORINTH - Ein einheitliches Kommunikationskonzept für PDVBus, BITBUS und IEC-Bus.  
Proc. VDI/GI-Fachtagung Prozeßrechensysteme '88, Stuttgart, 1988.

- 
101. Holler, M., Heiler, K.-U.:  
PDVnet, ein Echtzeitnetzwerk für die Automatisierungstechnik.  
Interner Bericht von PEP Modular Computers GmbH und dem Labor für  
Werkzeugmaschinen der RWTH Aachen, 1988.
  102. Ögrük, M., E.:  
Exemplarische Implementierung eines PDV-Bus-Monitors.  
Interner Bericht der Informatik-Forschungsgruppe E.  
Erlangen, 1989.
  103. Patzke, M:  
Anwendungsorientiertes Feldbussystem für die Qualitätslenkung.  
QZ 33, Heft 12 (1988): 647-650.
  104. O'Grady, P.:  
Automatisierte Fertigungssysteme.  
VCH Verlagsgesellschaft, Weinheim, 1988.
  105. Bärnreuther, B., Dietsch, H., Oehlrich, C.-W., Ulrich, R.:  
Eine MMS-Implementierung in einer Feldbus-Umgebung.  
Proc. VDI/VDE-GMA-Fachtagung "Informatik für die industrielle Automation-  
INFINA 89", Karlsruhe, VDI-Bericht 732, 1989.
  106. Bärnreuther, B., Dietsch, H., Oehlrich, C.-W., Ulrich, R.:  
MAP-Anwendungsdienste über einem Kommunikationssystem für heterogene  
Feldbus-Verbunde (Teil 1).  
**atp** 33, 7 (1991): 358-363.
  107. Bärnreuther, B., Dietsch, H., Oehlrich, C.-W., Ulrich, R.:  
MAP-Anwendungsdienste über einem Kommunikationssystem für heterogene  
Feldbus-Verbunde (Teil 2).  
**atp** 33, 8 (1991): 419-425.
  108. Dietsch, H.:  
Ein verteiltes Kommunikationssystem für einen Verbund aus lokalen Mikrorech-  
ner-Netzwerken.  
Dissertation, Universität Erlangen-Nürnberg, Arbeitsbericht des IMMD, Bd. 16, Nr.  
10, Erlangen, 1983.
  109. DIN IEC 625-1:  
Byteseriell bitparalleles Schnittstellensystem für programmierbare Meßgeräte,  
1979.
  110. Fa. INTEL:  
Distributed Control Modules Databook, 1984.

- 
111. Rosenthal, R. (Hrsg.):  
Proc. NBS/IEEE Workshop on Factory Communications. NBSIR 87-3516,  
Gaithersburg, 1987.
  112. Ulrich, R., Hinze, R., Dietsch, H.:  
Erfahrungen bei der Durchsatzoptimierung eines Transputer-Netzwerks für  
ISO-OSI-Architekturen am Beispiel der LLC-Teilschicht.  
Proc. 5.GI/ITG-Fachtagung "Messung, Modellierung und Bewertung von  
Rechensystemen und Netzen", Braunschweig, 1989.
  113. Klasche, G.:  
MMS: Die Fabrikautomation hat jetzt ihre Sprache.  
Elektronik 6/1991.
  114. Schwarz, K.:  
Manufacturing Message Specification (MMS) - Übersicht über die Methoden,  
Modelle, Objekte und Dienste.  
**atp** 33, 7 (1991): 369 - 378.
  115. Redaktion Markt&Technik:  
Vielfalt gerechtfertigt.  
Markt&Technik Nr. 39, 21.9.90: 49-52.
  116. Beale, N.:  
Fibre Optics for MAP.  
Proc. ENTERPRISE '88, Baltimore, 1988.
  117. Kieli, B.:  
Fiber Optics Local Area Networks: Applications for MAP and TOP.  
Proc. ENTERPRISE '88, Baltimore, 1988.
  118. EMUG-WIG:  
Manufacturing Message Specification (MMS) over TCP/IP.  
Projektinformation der European MAP Users Group, Eindhoven, 1989.
  119. ISO/TC 184/SC 5/WG 2:  
Principles for Companion Standards to MMS. 1990.
  120. Gilles, E., D. et al.:  
Sensortechnik in der Chemie - Status und Trend.  
**atp** 28, 9 (1986): 423-431.
  121. Endress, U., H.:  
Sensoren als Fundament der Prozeßinformatik.  
**atp** 33, 4 (1991): 171-177.

- 
122. VDI/VDE 2180 Blatt 1:  
Sicherung von Anlagen der Verfahrenstechnik mit Mitteln der Meß-, Steuerungs- und Regelungstechnik; Einführung, Begriffe, Erklärungen.  
VDI-Verlag, Düsseldorf, 1986.
123. Gosebruch, H.:  
Rundschleifen im geschlossenen Regelkreis: Echtzeit-Signalverarbeitung mit neuen Sensoren und Aktoren.  
Dissertation, Univ. Hannover. VDI-Verlag, Düsseldorf, 1990.
124. Kompa, G.:  
Sensoren im MHI-Bereich - Entwicklungsstand und Trends.  
**VDI-Z** 130, Nr. 2 (1988): 42 - 54.
125. Schneider, H.-J.:  
INTERKAMA 89: Sensorsysteme für die Betriebsmeßtechnik.  
**atp** 32,2 (1990): 55-70.
126. Buxbaum, H.-J., Hidde, A., R.:  
Produktunabhängige Zellensteuerung als Erweiterung des linienbezogenen flexiblen Fabrikautomatisierungskonzepts.  
PKI Tech. Mitt. 1 (1998): 121-131.
127. Thusel, F.:  
PROFIBUS-Profile für den Sensor-Aktuator-Bereich.  
**atp** 34,4 (1992): 185-188.
128. Kleinschmidt, P.:  
Intelligente Sensorsysteme.  
Elektronik 7 (1991): 128-139.
129. Lemme, H.:  
Die Sensoren der 90er Jahre.  
Elektronik 7 (1991): 142-150.
130. Horn, W., A.:  
Sensorik und Elektronik, ein symbiotisches Verhältnis.  
**atp** 30, 5 (1988): 217-219.
131. Schott, J.:  
Modular Concept for Integration of Various Types of Sensors in Hard- and Software.  
Internes Papier, DFVLR, Wessling, 1988.
132. Madelung, O.:  
Aktuator Sensor Interface (ASI).  
SENSOR report 1 (1992): 35-36.



# Reihe Fertigungstechnik Erlangen

## Band 1

Andreas Hemberger

**Innovationspotentiale in der rechnerintegrierten Produktion durch  
wissensbasierte Systeme**

208 Seiten, 107 Bilder, 1988. Kartoniert.

## Band 2

Detlef Classe

**Beitrag zur Steigerung der Flexibilität automatisierter Montage-  
systeme durch Sensorintegration und erweiterte Steuerungskonzepte**

194 Seiten, 70 Bilder, 1988. Kartoniert.

## Band 3

Friedrich-Wilhelm Nolting

**Projektierung von Montagesystemen**

201 Seiten, 107 Bilder, 1 Tabelle, 1989. Kartoniert.

Kartoniert.

## Band 4

Karsten Schlüter

**Nutzungsgradsteigerung von Montagesystemen durch den Einsatz  
der Simulationstechnik**

177 Seiten, 97 Bilder, 1989. Kartoniert.

## Band 5

Shir-Kuan Lin

**Aufbau von Modellen zur Lageregelung von Industrierobotern**

168 Seiten, 46 Bilder, 1989. Kartoniert.

## Band 6

Rudolf Nuss

**Untersuchungen zur Bearbeitungsqualität im Fertigungssystem  
Laserstrahlschneiden**

206 Seiten, 115 Bilder, 6 Tabellen, 1989. Kartoniert.

## Band 7

Wolfgang Scholz

**Modell zur datenbankgestützten Planung automatisierter  
Montageanlagen**

194 Seiten, 89 Bilder, 1989. Kartoniert.

## Band 8

Hans-Jürgen Wißmeier

**Beitrag zur Beurteilung des Bruchverhaltens von Hartmetall-  
Fließpreßmatrizen**

179 Seiten, 99 Bilder, 9 Tabellen, 1989. Kartoniert.

## Band 9

Rainer Eisele

**Konzeption und Wirtschaftlichkeit von Planungssystemen in der  
Produktion**

183 Seiten, 86 Bilder, 1990. Kartoniert.

Band 10  
Rolf Pfeiffer  
**Technologisch orientierte Montageplanung am Beispiel der Schraubtechnik**  
216 Seiten, 102 Bilder, 16 Tabellen. 1990. Kartoniert.

Band 11  
Herbert Fischer  
**Verteilte Planungssysteme zur Flexibilitätssteigerung der rechnerintegrierten Teilefertigung**  
201 Seiten, 82 Bilder. 1990. Kartoniert.

Band 12  
Gerhard Kleindam  
**CAD/CAP : Rechnergestützte Montagefeinplanung**  
203 Seiten, 107 Bilder. 1990. Kartoniert.

Band 13  
Frank Vollertsen  
**Pulvermetallurgische Verarbeitung eines übereutektoiden verschleißfesten Stahls**  
XIII + 217 Seiten, 67 Bilder, 34 Tabellen. 1990. Kartoniert.

Band 14  
Stephan Bliermann  
**Untersuchungen zur Anlagen- und Prozeßdiagnostik für das Schneiden mit CO<sub>2</sub> - Hochleistungslasern**  
VIII + 170 Seiten, 93 Bilder, 4 Tabellen. 1991. Kartoniert.

Band 15  
Uwe Geißler  
**Material- und Datenfluß in einer flexiblen Blechbearbeitungszelle**  
124 Seiten, 41 Bilder, 7 Tabellen. 1991. Kartoniert.

Band 16  
Frank Oswald Hake  
**Entwicklung eines rechnergestützten Diagnosesystems für automatisierte Montagezellen**  
XIV + 166 Seiten, 77 Bilder. 1991. Kartoniert.

Band 17  
Herbert Reichel  
**Optimierung der Werkzeugbereitstellung durch rechnergestützte Arbeitsfolgenbestimmung**  
198 Seiten, 73 Bilder, 2 Tabellen. 1991. Kartoniert.

Band 18  
Josef Scheller  
**Modellierung und Einsatz von Softwaresystemen für rechnergeführte Montagezellen**  
198 Seiten, 65 Bilder. 1991. Kartoniert.

Band 19  
Arnold vom Ende  
**Untersuchungen zum Biegeumformen mit elastischer Matrize**  
166 Seiten, 55 Bilder, 13 Tabellen. 1991. Kartoniert.

Band 20  
Joachim Schmid  
**Beitrag zum automatisierten Bearbeiten von Keramikguß mit Industrierobotern**  
XIV + 176 Seiten, 111 Bilder, 6 Tabellen. 1991. Kartoniert.



Band 21

Egon Sommer

**Multiprozessorsteuering für kooperierende  
Industrieroboter in Montagezellen**

188 Seiten, 102 Bilder. 1991. Kartiert.

Band 22

Georg Geyer

**Entwicklung problemspezifischer Verfahrensketten  
in der Montage**

192 Seiten, 112 Bilder. 1991. Kartiert.

Band 23

Rainer Flohr

**Beitrag zur optimalen Verbindungstechnik in der  
Oberflächenmontage (SMT)**

186 Seiten, 79 Bilder. 1991. Kartiert.

Band 24

Alfons Rief

**Untersuchungen zur Verfahrensfolge Laserstrahlschneiden  
und -schweißen in der Rohkarosseriefertigung**

VI + 145 Seiten, 58 Bilder, 5 Tabellen. 1991. Kartiert.

Band 25

Christoph Thim

**Rechnerunterstützte Optimierung von Materialflußstrukturen  
in der Elektronikmontage durch Simulation**

188 Seiten, 74 Bilder. 1992. Kartiert.

Band 26

Roland Müller

**CO<sub>2</sub>-Laserstrahlschneiden von kurzglasverstärkten Verbundwerkstoffen**

141 Seiten, 107 Bilder, 4 Tabellen. 1992. Kartiert.

Band 27

Günther Schäfer

**Integrierte Informationsverarbeitung bei der Montageplanung**

195 Seiten, 76 Bilder. 1992. Kartiert.

Band 28

Martin Hoffmann

**Entwicklung einer CAD/CAM-Prozeßkette für die Herstellung  
von Blechbiegeteilen**

149 Seiten, 89 Bilder. 1992. Kartiert.

Band 29

Peter Hoffmann

**Verfahrensfolge Laserstrahlschneiden und -schweißen :  
Prozeßführung und Systemtechnik in der 3D-Laserstrahlbearbeitung  
von Blechformteilen**

186 Seiten, 92 Bilder, 10 Tabellen. 1992. Kartiert.

Band 30

Olaf Schrödel

**Flexible Werkstattsteuerung mit objektorientierten Softwarestrukturen**

180 Seiten, 84 Bilder. 1992. Kartiert.

Band 31

Hubert Reinsch

**Planungs- und Steuerungswerkzeuge zur Impliziten  
Geräteprogrammierung in Roboterzellen**

XI + 212 Seiten, 112 Bilder. 1992. Kartiert.

**Band 32**

**Brigitte Bärnreuther**

**Ein Beitrag zur Bewertung des Kommunikationsverhaltens  
von Automatisierungsgeräten in flexiblen Produktionszellen**

**XI + 179 Seiten, 71 Bilder, 1992. Kartiert.**