

Gerhard Kleineidam

*CAD/CAP: Rechnergestützte
Montagefeinplanung*

Gerhard Kleineidam

*CAD/CAP: Rechnergestützte
Montagefeinplanung*

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:	01. 02. 1990
Tag der Promotion:	30. 05. 1990
Dekan:	Prof. Dr.-Ing. G. Kuhn
Berichterstatter:	Prof. Dr.-Ing. K. Feldmann Prof. Dr.-Ing. H. Meerkamm

CIP-Titelaufnahme der Deutschen Bibliothek

Kleineidam, Gerhard:

CAD, CAP: rechnergestützte Montagefeinplanung / Gerhard
Kleineidam. Hrsg. von Klaus Feldmann. – München ; Wien :
Hanser, 1990

(Fertigungstechnik – Erlangen ; Bd. 12)

Zugl. : Erlangen, Nürnberg, Univ., Diss., 1990

ISBN 3-446-16112-0

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 1990

Herstellung: Copy Center 2000, Erlangen-Eltersdorf

Printed in Germany

V O R W O R T

Die vorliegende Arbeit entstand während meiner Zeit als wissenschaftlicher Assistent am Lehrstuhl für Fertigungsautomatisierung und Produktionssystematik an der Universität Erlangen - Nürnberg.

Herrn Professor Dr.-Ing. Klaus Feldmann, dem Leiter des Lehrstuhls, gilt mein besonderer Dank für die wohlwollende Unterstützung und großzügige Förderung bei der Durchführung meiner Arbeit.

Ebenso bedanke ich mich bei Herrn Professor Meerkamm für die aufmerksame Durchsicht der Arbeit und die Übernahme des Koreferates.

Die wertvolle Hilfe der Herren Herbert Schellig und Helmut Schuster bei der Layoutgestaltung sowie die lexikalische Durchsicht des Manuskriptes durch meinen Vater Fritz Kleineidam, haben wesentlich zum Gelingen der Arbeit beigetragen.

Den Mitarbeitern des Lehrstuhls, insbesondere den Kollegen und Studenten der Gruppe für 'Rechnereinsatz in Konstruktion und Planung', danke ich herzlich für ihre kameradschaftliche Unterstützung. Besonders zu schätzen wußte ich die wertvolle Zusammenarbeit mit meinem Kollegen und Freund Dr.-Ing., Dipl.-Wirtsch.-Ing. Rainer Eisele.

Meiner Frau Susanne und meinem Sohn Roman danke ich für ihre Liebe und die vielen kleinen Opfer, die sie mir während meiner Doktorarbeit entgegengebracht haben.

Gerhard Kleineidam

CAD/CAP : Rechnergestützte Montagefeinplanung

-Inhaltsverzeichnis-

	Seite
1. <u>Einleitung</u>	1
2. <u>Programmierung von Montagezellen in der technischen Auftragsabwicklung</u>	7
2.1 Robotik in der Montage	7
2.2 Verfahrenskette: Vom Produkt zum Steuerprogramm	12
3. <u>Manuelle und automatisierte Montagezellen</u>	17
3.1 Der manuelle Montagearbeitsplatz	19
3.1.1 Komponenten des Montagehandarbeitsplatzes	20
3.1.2 Planung und Einweisung	21
3.1.3 menschliche Leistung im Lern- und Montageprozeß	25
3.2 Die automatisierte Montagezelle	30
3.2.1 Komponenten einer Montagezelle	31
3.2.2 Geräteorientierte Programmierverfahren	37
3.2.3 Aufgabenorientierte Programmierung	42
4. <u>Anforderungen an ein aufgabenorientiertes Planungs- und Programmiersystem</u>	47
4.1 Schnittstellen	48
4.1.1 Der Anwender und seine Aufgaben	49
4.1.2 Schnittstelle zu Konstruktion und Montagegrobplanung	54
4.1.3 Schnittstelle zur Betriebsebene	57
4.2 Grundfunktionen	60
4.2.1 Entwerfen, Konstruieren und Ausarbeiten	61
4.2.2 Aufgabenbeschreibung, Programmierung	63
4.2.3 Test und Taktzeitprüfung	66

4.3	Modelle und Verwaltung	67
4.4	Einsatz von Basiswerkzeugen	70
5.	<u>CAD - integriertes Konzept für die Montagefeinplanung</u>	74
5.1	Gestaltung der Benutzeroberfläche	76
5.1.1	Dialogformen und Benutzerführung	78
5.1.2	Aufgabenbeschreibung, Programmierung	90
5.1.3	Einsatz von Grafik und Farben	96
5.2	Erzeugen des Datenmodells	98
5.2.1	Produktmodelle	102
5.2.2	Komponentenmodelle	107
5.2.3	Modell der Zelle	116
5.2.4	Framekonzept als Datenkapsel	118
5.3	Emulation von Steuerungsfunktionen	123
5.3.1	Inverse Kinematik mehrachsiger Bewegungssysteme	125
5.3.2	Bewegungen mehrachsiger Systeme	132
5.3.3	Parallele Abläufe, SPS- und Sensor- simulation	140
5.4	Simulation und Optimierung	147
5.4.1	Kollisionsvermeidung	148
5.4.2	Bahnoptimierung, Bahnplanung	151
5.5	Greif- und Bearbeitungsplanung	155
6.	<u>Prototypen und Tests</u>	161
6.1	CARo - Computer Aided Robotic System	162
6.1.1	Systemaufbau von CARo	162
6.1.2	Systembedienung	166
6.2	Anwendungen von CARo	170
6.2.1	Komplettmontage mit Gelenkrobotern	170
6.2.2	Montagelinie mit Linearachsensystem	171
6.2.3	Manuelle Montage	173
6.3	Systemtest und Testauswertung	175
7.	<u>Integration von CAP Werkzeugen in den Produktions- betrieb</u>	181
7.1	Rechnerverbund und Fabrikkommunikation	183
7.2	Personelle und organisatorische Integration	185

8. <u>Zusammenfassung und Ausblick</u>	193
9. <u>Literaturverzeichnis</u>	197
<u>Anhang :</u>	
Abkürzungen und Kurzzeichen	i
Formeln und Formelzeichen	iii
Lebenslauf	iv

1. Einleitung

In verschiedenen Teilbereichen produzierender Unternehmen -wie Konstruktion, Arbeitsplanung, Arbeitssteuerung oder Fertigung- lassen sich heute Rechner und Softwarewerkzeuge einsetzen, mit deren Hilfe die Entwicklung und Fertigung von Produkten in technischer und wirtschaftlicher Hinsicht optimiert werden kann. Trotz der starken Nachfrage und dem vorhandenen Angebot verschiedener Rechnersysteme ist der Durchdringungsgrad dieser Computertechniken in großen Teilen der Industrie bisher nur sehr schwach ausgeprägt. Die Gründe dafür beruhen sowohl auf technischen als auch auf organisatorischen und wirtschaftlichen Problemen bei der Integration solcher Techniken in gewachsene Abläufe und Unternehmensstrukturen (Bild 1).

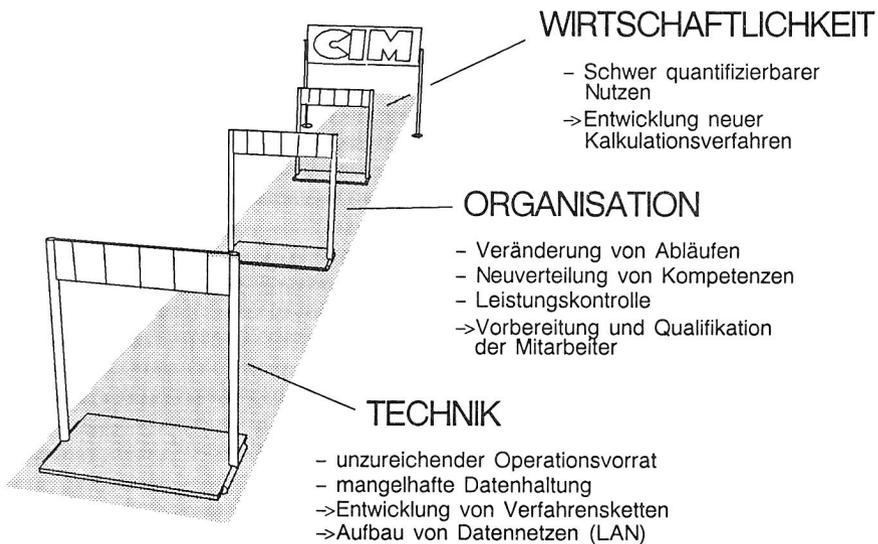


Bild 1: Problemfelder bei der Integration von rechnergestützten Methoden und Wege zu deren Lösung

Technische Probleme bei der Einführung von Computertechniken ergeben sich bei der Vernetzung und Kommunikation unterschiedlicher Systeme. So gibt es für den Konstrukteur ein Grafiksystem (CAD-System), für den Arbeitsvorbereiter Systeme zur Programmierung von NC-Werkzeugmaschinen oder für das Lager eigene Verwaltungsprogramme. Informationen werden in Form von Zeichnungen und Formularen ausgetauscht. Das technische Problemfeld ist gekennzeichnet durch unzureichende Schnittstellen zum Austausch von Daten. Es fehlen zentrale Datenverwaltungssysteme und fabrikweite Kommunikationsnetzwerke. Die Aktivitäten, die unter dem Begriff CIM zusammengefaßt werden, beschäftigen sich mit der technischen Verknüpfung der vielen isoliert arbeitenden Computerinseln in der Produktion. Bild 2 stellt global die Teilbereiche in einem produzierenden Unternehmen dar, die es in einem CIM-Verbund zu integrieren gilt.

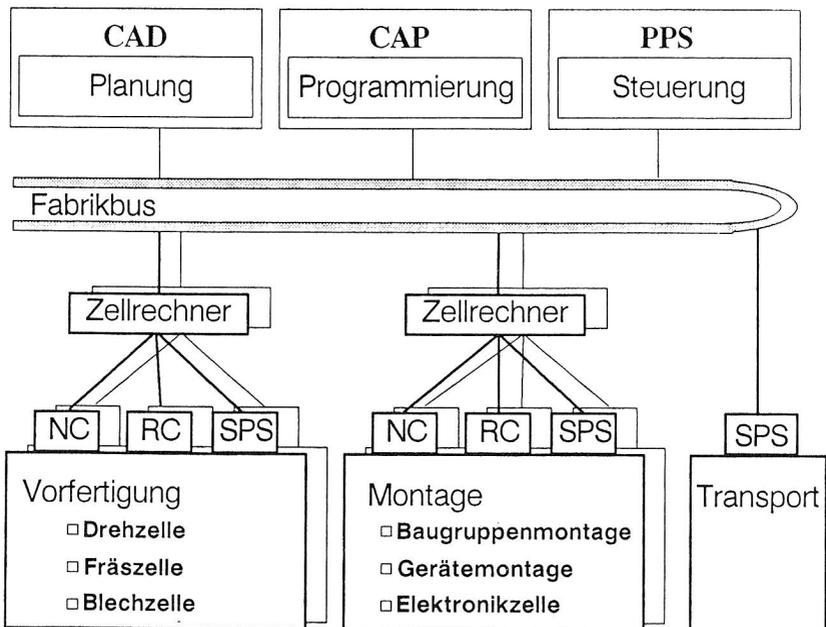


Bild 2: Struktur der rechnerintegrierten Produktion /1/

Neben dem übergeordneten Problem der Integration sind es vor allem viele technische Details, die den Einsatz des Rechners als Planungsinstrument in Produktionsbetrieben erschweren:

- Funktionale Mängel, lückenhafter Operationsvorrat,
- unzulängliche Datenmodelle und Datenhaltung,
- schlechte Anpassung an Anwenderbedürfnisse,
- Fehler in Hard- und Software,
- unzureichender Service.

Organisatorische Integrationsschwierigkeiten ergeben sich aus der mangelnden Akzeptanz der Mitarbeiter und aus Problemen bei der Veränderung von gewachsenen Abläufen und Strukturen im Unternehmen. Ursachen hierfür sind:

- mangelnde Transparenz der eingesetzten Software aufgrund der unzureichenden Qualität der Anwenderschulung, der schlechten Erlernbarkeit der Systemfunktionen, sowie durch unergonomische Systembedienung;
- mangelhafte Beratung und schlechter Support durch den Systemanbieter;
- Berührungängste der Mitarbeiter: z.B. die Angst vor dem Abbau der direkten menschlichen Kommunikation, vor dem Zwang zur Daueraufmerksamkeit und Leistungsverdichtung durch Elimination von Routinearbeiten oder die Angst vor Überwachung und Leistungskontrolle sowie vor dem Versagen beim Arbeiten mit der neuen Technik.

Aus Sicht des Unternehmers ist die Wirtschaftlichkeit des Rehereinsatzes ausschlaggebender Faktor für die Investition in moderne Rechnertechnologien, doch spielen Qualitätsverbesserungen in Planung und Produktion, sowie strategische Gesichtspunkte, wie die Sicherung von Know How im Unternehmen eine immer wichtigere Rolle. Aufgrund der bisherigen Erfahrungen bei Einführung und Einsatz von Rechnerwerkzeugen -siehe CAD-Einführung

Anfang der 80er Jahre- und den dabei aufgetretenen Problemen, ist es wichtig, zukünftig nur Entwicklungen unter einem gesamtgesellschaftlichen Konzept zu betreiben. Die Aspekte der Technik, der Organisation und der Wirtschaftlichkeit müssen dabei gleichrangig berücksichtigt werden.

Die vorliegende Dissertation wird sich dem Teilgebiet CAP innerhalb der CIM Thematik widmen und sich dabei auf den Bereich Montage und Montageplanung konzentrieren. Es soll in diesem Zusammenhang ein Beitrag zur Verwirklichung der rechnerintegrierten Arbeitsvorbereitung unter besonderer Berücksichtigung der genannten technischen und personalorganisatorischen Aspekte geleistet werden. Entwicklungen auf genanntem Gebiet müssen deutlicher als bisher den Aspekt der personellen Integrierbarkeit berücksichtigen. Dies setzt voraus, daß die neuen Werkzeuge ausbildungsgerecht und technologieorientiert bedienbar sind und auf die spezifischen Anforderungen des Anwenders zugeschnitten werden. Die hier beschriebenen Arbeiten beruhen deshalb konkret auf drei wesentlichen Entwicklungszielen:

- 1) Konzeption eines Planungswerkzeuges zur Gestaltung und Programmierung von Montagezellen,
- 2) Entwurf einer aufgaben- und technologieorientierten Kommunikationsschnittstelle zur Feinplanung und offline Programmierung von Montagezellen,
- 3) technische und personalorganisatorische Integration von rechnergestützten Planungs- und Programmierwerkzeugen in eine CIM - Umgebung.

Die Robotik zählt zu den Kernthemen der Arbeit, es wird jedoch nicht der Roboter allein, sondern das Zusammenwirken aller Komponenten in autonomen flexiblen Montageeinheiten behandelt. Dazu wird in Kapitel 2 gezeigt, welche Rolle die Robotik in der Montage spielt, welche unterschiedlichen Stoßrichtungen die Industrie und welche die Hochschulen verfolgen. Außerdem wird in Kapitel 2 die Einordnung der Montagezellenplanung in den Ablauf der technischen Auftragsabwicklung beschrieben.

In Kapitel 3 werden durch den Vergleich von manuellen und automatisierten Montagearbeitseinheiten und deren Programmierung, die Anforderungen an aufgabenorientierte Planungs- und Programmierwerkzeuge abgeleitet. Basis für diese Analogiebetrachtung ist die Erkenntnis, daß der Mensch durch seine sensorischen und motorischen Fähigkeiten sowie durch seine Intelligenz ein Höchstmaß an Flexibilität und Autonomie in der Montage garantieren kann. Diese Eigenschaften des Menschen gilt es durch verteilte Rechnerleistung in Planung und Betrieb nachzubilden. Bei der Entwicklung von Montageplanungswerkzeugen muß berücksichtigt werden, daß der Anteil manueller Montage in der Industrie durchschnittlich noch 80% ausmacht. Planungswerkzeuge müssen daher auch die Möglichkeit zur Gestaltung und Auslegung manueller Systeme bieten.

Zur Durchführung von Planungsvorgängen mit Hilfe des Rechners bedarf es dreier wesentlicher Komponenten.

1. Modellinformationen
2. Planungsregeln /-strategien
3. Methoden zur Aufgabenbeschreibung

Kapitel 4 spezifiziert die Anforderungen an diese drei Grundelemente aufgabenorientierter Programmiersysteme. Ein besonderer Schwerpunkt wird dabei die Aufgabenbeschreibung sein. Hier sollen vor allem Anforderungen aus der Kommunikationsergonomie zur Lösung von möglichen Akzeptanzproblemen oder Berührungängsten berücksichtigt werden.

In Kapitel 5 wird ein CAD-integriertes Konzept für ein Montagefeinplanungs- und Programmierwerkzeug vorgestellt, wobei Schwerpunkte auf Benutzeroberfläche, Datenmodell, Emulation von Steuerungsfunktionen und Algorithmen zur Aktionsoptimierung von Montageanweisungen gelegt werden. Die Abarbeitung einer impliziten Montageanweisung rundet das Kapitel in Form eines Beispiels ab.

Kapitel 6 stellt Prototypen vor, die im Rahmen der Forschungs-

arbeiten entstanden sind und beschreibt Testergebnisse und deren Auswertung beim probenhaften Einsatz dieser Systeme in einer Modellfabrik und in Werken der Industrie.

Der wichtige Aspekt der Integration rechnergestützter Planungswerkzeuge in die reale Produktionsumgebung wird in Kapitel 7 behandelt. Es werden Einführungsstrategien, Systemauswahl und Einsatz von Planungswerkzeugen vorgestellt. Die Erfahrung des Autors, die aus CIM-Schulungen /2/, /3/ und Rechnereinführungen gewonnen wurde, hat zu diesem Kapitel wesentlich beitragen können.

In Kapitel 8 werden schließlich die Auswirkungen beim Einsatz der behandelten Techniken auf die Arbeitsumwelt eruiert.

2. Programmierung von Montagezellen in der technischen Auftragsabwicklung

Der Themenkomplex Planung und Programmierung von Robotersystemen wird seit 1977 von zahlreichen Forschungseinrichtungen weltweit bearbeitet /4/. Seit 1985 gewannen die Entwicklungen auf diesem Gebiet durch das Aufblühen der "CIM-Aktivitäten" auch für den industriellen Einsatz zunehmend an Bedeutung. Ziel war es, das Konzept der rechnerintegrierten Produktion durch Roboterplanungs- und Programmierwerkzeuge abzurunden sowie Fertigungsbereiche zu automatisieren, in denen man ohne die Flexibilität des Roboters keine Möglichkeit sah, technisch und wirtschaftlich CIM zu verwirklichen -dies gilt insbesondere für den Produktionsbereich Montage.

Die Montage ist ein sehr kostenintensiver Produktionsbereich, der nur wenige attraktive Arbeitsplätze bietet, sich somit für den Einsatz von Automatisierungskomponenten und einer verstärkten Forschung auf diesem Gebiet eignet.

2.1 Robotik in der Montage

Auf den ersten Blick scheint es nicht schwierig zu sein Montagetätigkeiten zu automatisieren, da die Arbeitsinhalte in der Montage meist sehr einfach sind und kaum der Ausführung durch Fachpersonal bedürfen. Der erste Eindruck trügt allerdings, denn die Ausführung von Montagetätigkeiten erfordert extrem große sensorimotorische Fähigkeiten. Fähigkeiten, die bis heute von Maschinen weder in ihren erstaunlichen Eigenschaften noch in ihrer einzigartigen Kombination geleistet werden können. Ziel der Robotikforschung ist es, menschliche Fähigkeiten nachzubilden, zu erweitern und auf Maschinen zu übertragen.

Das Forschungsgebiet der Robotik (von tschech. 'robotnik' = künstlicher Mensch) beschäftigt sich mit unterschiedlichen

Teilbereichen (Bild 3), zu deren Inhalten auch Grundlagen aus angrenzenden Wissenschaften zählen –wie Mathematik, Physik, Maschinenbau, Elektrotechnik und Informatik.

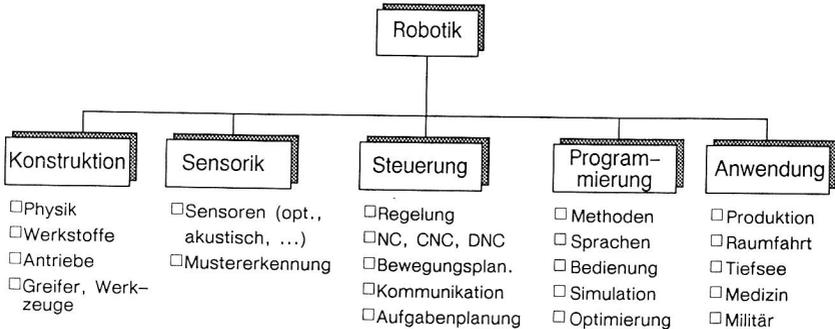


Bild 3: Forschungsgebiete und -inhalte der Robotik

Von entscheidender Bedeutung ist die sorgfältige Planung der Inbetriebnahme von Robotersystemen und deren Integration in eine bisher vom Menschen dominierte Arbeitsumwelt. Durch die Entwicklung von Modellen lassen sich automatisierte Bewegungssysteme (hier: Synonym für Roboter) mit Hilfe des Rechners simulieren. Der Einsatz von Robotern und deren Umwelt kann dadurch berechenbar und planbar gemacht werden. Drei Forschungsansätze werden dazu verfolgt.

Viele Aktivitäten konzentrieren sich allein auf die Modellierung der mathematisch physikalischen Eigenschaften von Robotersystemen und deren Programmier- und Steuerungskonzepten. Die daraus entstandenen Systeme eignen sich zur konstruktiven Auslegung von Robotern und deren Steuerungen sowie zur Berechnung der Bewegungsbahnen kinematischer Ketten. Durch die Erweiterung dieser Systeme um Ausgabemöglichkeiten zur grafischen Darstellung von Bewegungsbahnen können solche Systeme auch zur Planung von komplexen Roboterarbeitszellen eingesetzt werden. Beispiele hierfür sind Systeme wie: ROBSIM /5/, PLACE /6/, GRASP /7/, GROSIM /8/ und andere (siehe: Stand der Technik /9/, /10/).

Die Forschungsaktivitäten hierzu werden insbesondere durch Roboterhersteller, wie SIEMENS, IBM, ASEA, Unimation, KUKA, Reis, GM, VW u.v.a., sowie von Instituten für Mechanik, Steuer- und Regelungstechnik vorangetrieben.

Aus amerikanischen Forschungsarbeiten sind Konzepte und Rechnermodelle hervorgegangen, die als Grundlage für die Entwicklung moderner Roboterprogrammiersprachen dienen /11/. Diese Konzepte vernachlässigen jedoch häufig produkt- bzw. produktionsspezifische Zusammenhänge, wie exakte Geometrie, Stückzahlen oder Bearbeitungsverfahren, so daß ausschließlich abstrakte Verknüpfungen von Roboteraktionen für die Planung und Programmierung von vereinfachten Roboterzellen abgeleitet werden können. Ursprung dieses Ansatzes war, die aus der "Künstlichen Intelligenz" (KI oder engl. AI für Artificial Intelligence) stammende sog. "Klötzchenwelt" oder auch "Blocks World" /12/. Auch in Deutschland wird an der Entwicklung theoretischer Umweltmodelle zur aufgabenorientierten Programmierung von Robotersystemen gearbeitet. Ihr Einsatz bleibt aber ebenso wie der ihrer amerikanischen Vorbilder auf die Anwendung in Forschungslabors beschränkt. Beispiele für diesen Ansatz sind u.a. Sprachen wie AL /13/, AUTOPASS /14/, LAMA /15/, RAPT /16/, ROBEX /17/ oder APOM /4/. Weitere Aktivitäten und deren genaue Beschreibung sind den Quellen /18/ und /19/ zu entnehmen.

Im Gegensatz zu den beiden erstgenannten Ansätzen orientiert sich die folgende Vorgehensweise an den konkreten Forderungen aus der industriellen Praxis. Hierbei geht es grundsätzlich um die Entwicklung anwenderorientierter und integrierbarer Werkzeugkonzepte zur Planung und Programmierung von Produktionseinheiten unter einem gesamtheitlichen Fabrikkonzept (CIM). Aus dem Blickwinkel der rechnerintegrierten Produktion ist dies der einzig sinnvolle Weg, um folgende Grundforderungen angemessen zu berücksichtigen:

- Straffung des Planungsablaufs,
- Verbesserung der Qualität von Planungsergebnissen,
- anwenderorientierte Grundkonzeption

- Wieder- und Weiterverwendung von Daten, z.B. aus der Konstruktion (CAD),
- hard- und softwaretechnische Realisierung von Verfahrensketten.

In der Bundesrepublik wird vor allem die letztgenannte Vorgehensweise favorisiert. Forschungsaktivitäten hierzu finden in enger Kooperation zwischen Industrie und Hochschulen bzw. Forschungsgesellschaften statt. Die Realisierung oder Umsetzung der dabei entstehenden Konzepte in Form lauffähiger Systeme, beschränkt sich wegen der Komplexität und des Umfangs der Aufgabenstellung auf wenige Prototypen /20/ bzw. Teilmodule, die sich dennoch im probenhaften Einsatz bewährt haben. Zu nennen sind hierbei Systeme wie: COSIRO (Entwicklung des IWB an der TU-München), ROBOTICS (von Dassault Systems), ROBCAD (von Tecnomatix), ROBOTICS (von MDIS), CARo/SMS (gemeinsame Entwicklung von SIEMENS und Universität Erlangen), IGRIP (von DENEb Robotics Inc.), GRASP (von BYG) und FMSsoft (von Bosch). Die genannten Systeme werden in einigen bundesdeutschen Unternehmen zur Planung von Robotersystemen eingesetzt (siehe /2/, /10/, /20/).

Ein grundsätzlicher Fehler, der dennoch von den meisten Einrichtungen bei der Entwicklung von Planungs- und Programmierwerkzeugen für die Montage gemacht worden ist, besteht darin, daß man dem Roboter -insbesondere dem mehrachsigen Gelenkroboter- die zentrale Rolle in der automatisierten Montage beimißt. Untersuchungen haben ergeben, daß verglichen mit dem gesamten Planungs- und Installationsaufwand einer Montageanlage, der Aufbau des Teilsystems Roboter nur einen sehr geringen Anteil ausmacht (siehe Bild 17). Greiferkonstruktion, Werkzeugbau, Aufbau, Verschaltung und Programmierung von Transport und Bearbeitungseinrichtungen sowie die Sensorintegration stehen dem mit weit größerem Aufwand und letztendlich höheren Kosten gegenüber. Hinzu kommt, daß derzeit ein Trend in der Industrie festzustellen ist, der weg von den hochflexiblen meist aber zu teuren Mehrachsgeräten hin zu modularen Linearachssystemen führt.

Konsequenz kann nur sein, daß als kleinste autonome Einheit in einem Arbeitssystem nicht mehr der Roboter sondern die Zelle gesehen wird.

Definition Montagezelle :

Die kleinste autonome Einheit eines Montagesystems ist die Montagezelle. Zu ihr gehören Handhabungs- bzw. Bewegungseinrichtungen, Werkzeuge, Greifer und/oder sonstige Bearbeitungssysteme (Vorrichtungen) sowie Überwachungs- und Kontrollelemente.

(siehe auch Bild 4)

Um mittel- oder langfristig universell einsetzbare Werkzeuge für den Bereich der Arbeitvorbereitung (CAP) in der Montage verfügbar zu machen, wird intensiv an praktikablen Lösungen gearbeitet. Es bietet sich dabei an, Ergebnisse aus anderen Ansätzen der Robotikforschung zu übernehmen, wie z.B. Algorithmen zur Berechnung von Bewegungsbahnen, grafische Simulationsmodule oder Modelliersysteme.

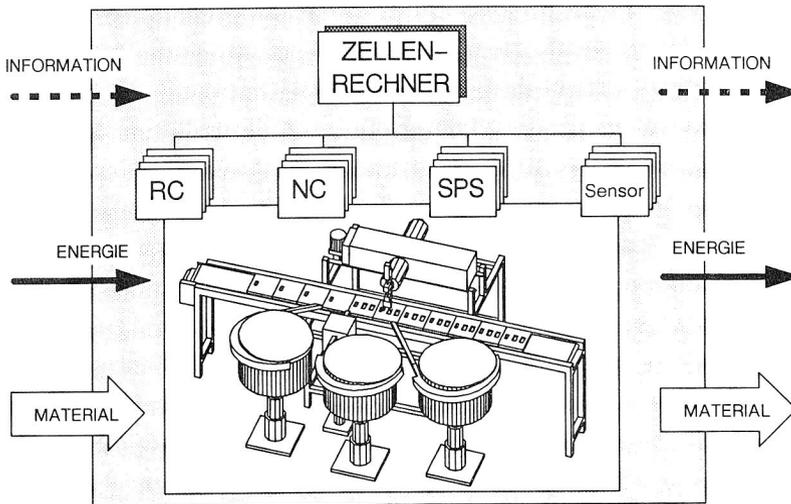


Bild 4: Die automatisierte Montageeinheit

Das größte Defizit existierender Konzepte für Planungs- und Programmiersysteme in der Montage beruht zum einen in der ungenügenden Berücksichtigung der technologischen Parameter von Montageverfahren, zum anderen in der geringen Beachtung der Ziel- bzw. Anwendergruppen, also den späteren Bedienern.

Auch technische Grenzen, wie Rechengeschwindigkeiten oder Speicherplatz von derzeit eingesetzten Arbeitsplatzrechnern sowie mangelhafte Basissoftware (Grafik, DB), verhindern noch, daß Lösungen kurzfristig auf dem Markt verfügbar sein werden. Die vorliegende Dissertation kann dennoch heute bereits zeigen, wie rechnergestützte Planungs- und Programmierwerkzeuge für die Montage aussehen sollen und wie sie eingesetzt werden können.

2.2 Verfahrenskette: Vom Produkt zum Steuerprogramm

Unter Verfahrenskette wird die durchgängige Verarbeitung von Produkt- und Fertigungsinformationen von der Konstruktion bis zum Prozeß verstanden /21/. Ausgangsbasis für eine sog. CAD-Verfahrenskette ist die Konstruktion, in der alle zur Herstellung notwendigen Produkteigenschaften in Form von Zeichnungen und Stücklisten oder Produktmodellen (im Rechner) festgelegt werden. Ein Werkzeug der Konstruktion ist das CAD-System. CAD-Systeme sind ursprünglich als reine Zeichnungswerkzeuge für den Konstrukteur entwickelt worden. Die Vielfalt der Möglichkeiten, die sich heute durch die Verarbeitung dreidimensionaler Objektdaten ergibt, reicht von der Festigkeitsberechnung von Bauteilen (FEM, BEM) über Präsentationsgrafiken und grafische Simulationen bis hin zur Übergabe von Geometriedaten an Programmiersysteme für Werkzeugmaschinen und Roboter (Bild 5).

Man unterscheidet CAD-Systeme grundsätzlich hinsichtlich ihrer 2D oder 3D Grafikfähigkeit. Für Anwendungen in der Montage ist die 3D Geometrieverarbeitung unerlässlich, denn der Montageprozeß ist geprägt von Körperbewegungen im Raum. Da dreidimensionale Grafiksysteme erst heute mit akzeptablem Antwortzeitverhalten

angeboten werden können, ist verständlich, warum erst jetzt damit begonnen wird Verfahrensketten im Bereich Montage zu entwickeln.

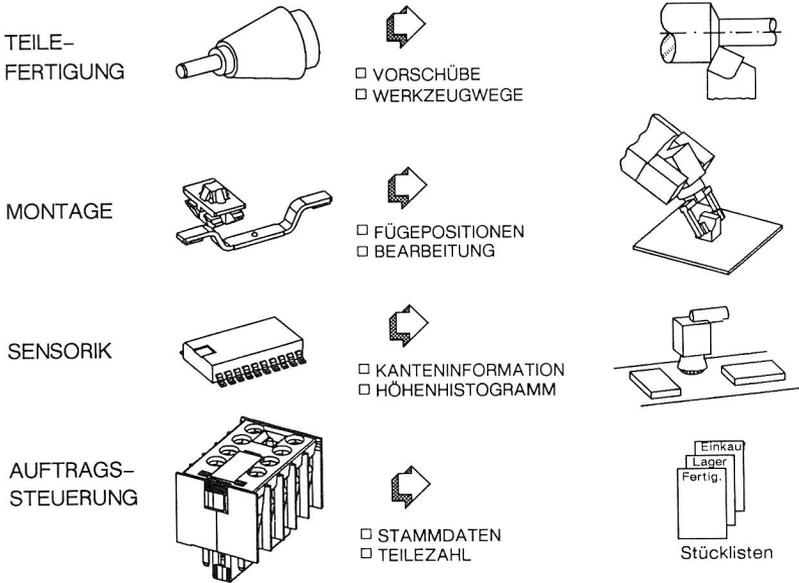


Bild 5: Beispiele für CAD - Verfahrensketten

Bekanntestes und ältestes Beispiel einer Verfahrenskette besteht in der Nutzung von CAD Daten für die Programmierung von numerisch gesteuerten Werkzeugmaschinen. Bereits seit 1960 wird daran gearbeitet, Daten aus grafischen Informationssystemen zu extrahieren und an NC-Steuerungen zu überspielen. Der Einsatz von Rechner-systemen in allen Unternehmensbereichen, hat den Ruf nach Realisierung weiterer Verfahrensketten laut werden lassen, denn die meisten der eingesetzten Systeme arbeiten isoliert, obwohl auf die gleichen Informationen zurückgegriffen werden muß. Produktdaten müssen so mehrfach eingegeben, gespeichert und verwaltet werden. Wie problematisch die Situation werden kann zeigt Bild 6.

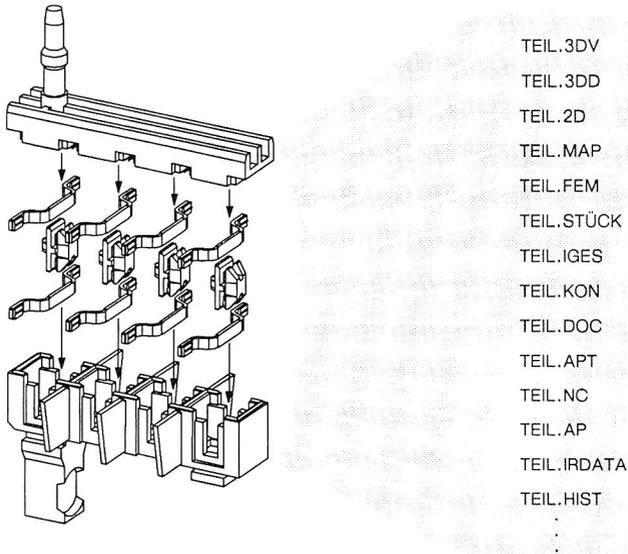


Bild 6: Verfahrensketten erfordern häufig viele Datenfiles

Verfahrensketten werden heute über Schnittstellen (siehe Bild 30) realisiert. Jede Schnittstelle wird dabei mit spezifischen Protokollen bedient, die üblicherweise als Datenfiles gespeichert werden müssen. Dadurch entsteht ein nur schwer zu verwaltendes 'Datenchaos'. Durch uneinheitliche Systembedienung und eine heterogene Softwarelandschaft wird der Versuch, echte durchgängige Verfahrensketten zu realisieren, zusätzlich erschwert.

Besonders deutlich wirkt sich dieser Mißstand im Bereich Montage und Montageplanung aus. Betrachtet man die Ablaufschritte in der Montageplanung und deren Einbettung in den Gesamtablauf der technischen Auftragsabwicklung, so erkennt man, daß hier ein Regelkreis /22/ vorliegt, der aus vielen Einzelprozessen (= Verrichtungen) besteht (Bild 7).

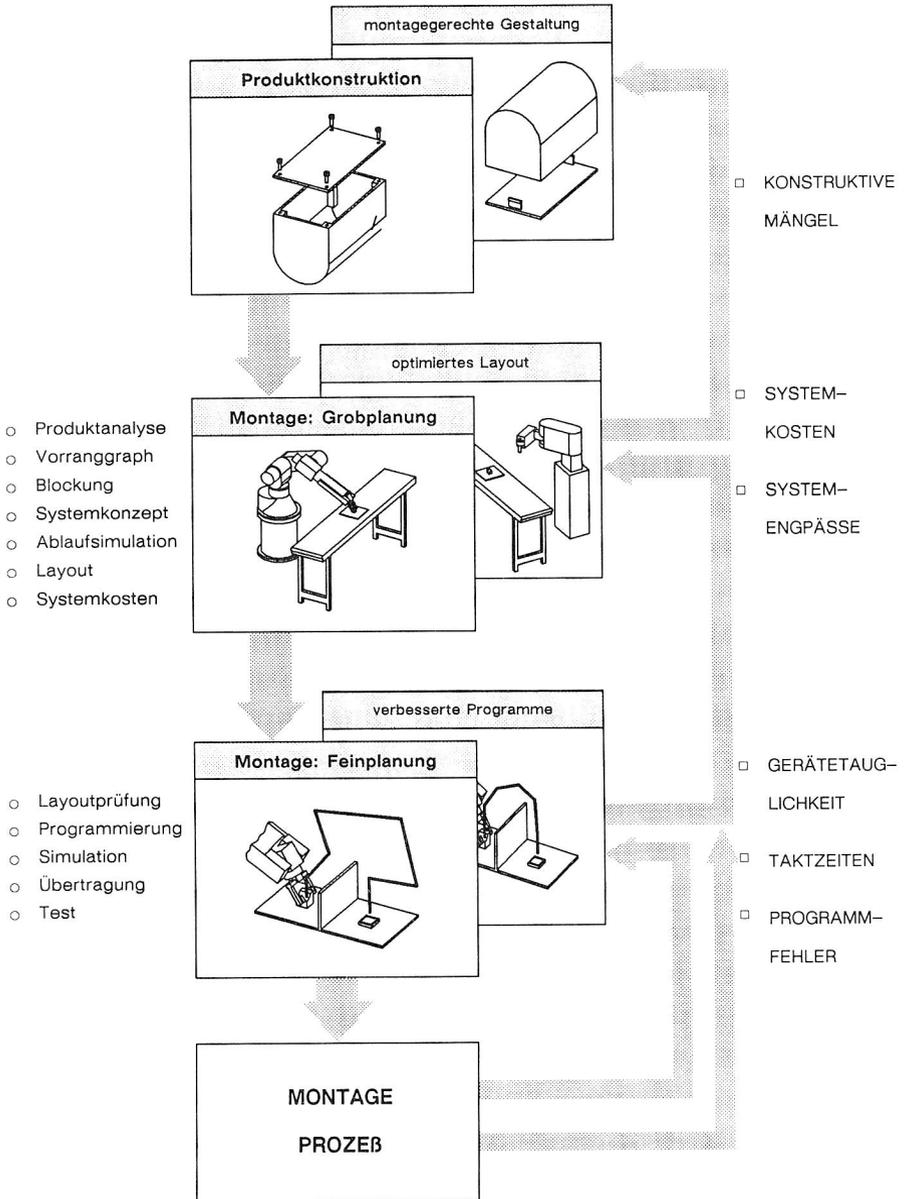


Bild 7: Regelkreis für Konstruktion und Montageplanung

Für diese Einzelprozesse stehen zahlreiche unterschiedliche Werkzeuge (Bild 8) zur Verfügung, von denen die meisten an Hochschulen entwickelt worden sind. Die einzelnen Systeme decken jeweils nur einen Teilbereich der Arbeitsplanung ab. Damit reagiert dieser Regelkreis auch entsprechend langsam und anfällig auf Störungen, wie z.B. Änderungswünsche. Die Ziele einer Verfahrenskette für die Montage bestehen demnach darin:

1. Den Informationsfluß zwischen Konstruktion, Montageplanung und -prozeß beschleunigen.
2. Die Vielzahl der Planungswerkzeuge unter einem Konzept zusammenfassen.

Einheitliche Modelle und Benutzeroberflächen für Rechneranwendungen im Bereich der technischen Auftragsabwicklung sind dafür zu entwickeln.

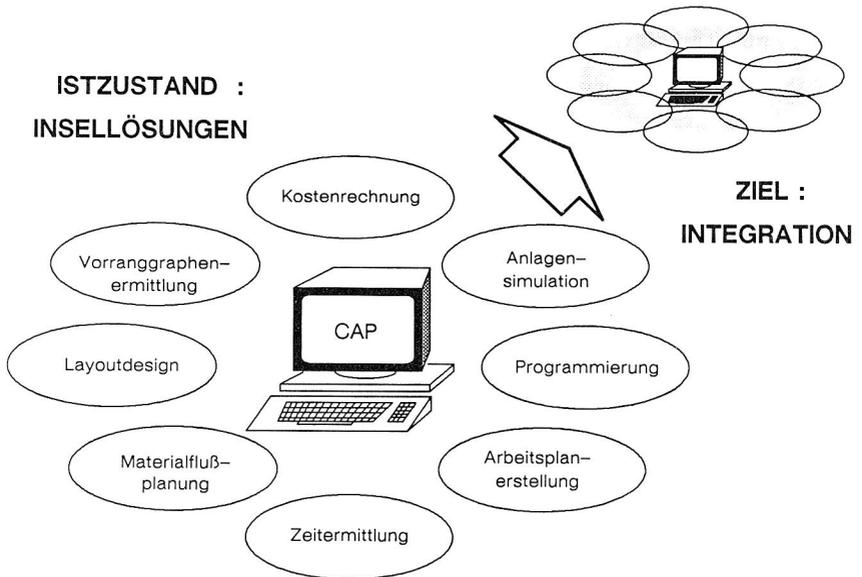


Bild 8: Heterogene Welt der CAP-Systeme in der Montageplanung

3. Manuelle und automatisierte Montagezellen

Durch den Vergleich manueller und automatisierter Montagesysteme sollen, unter Berücksichtigung von verschiedenen Montagetechnologien, Verbesserungsmöglichkeiten für die Einsatzplanung und Programmierung automatisierter Montagezellen abgeleitet werden. Es gilt hierbei die Vorteile manueller Arbeitssysteme und deren Planung auf Systeme mit weitgehend automatisierten Abläufen zu übertragen.

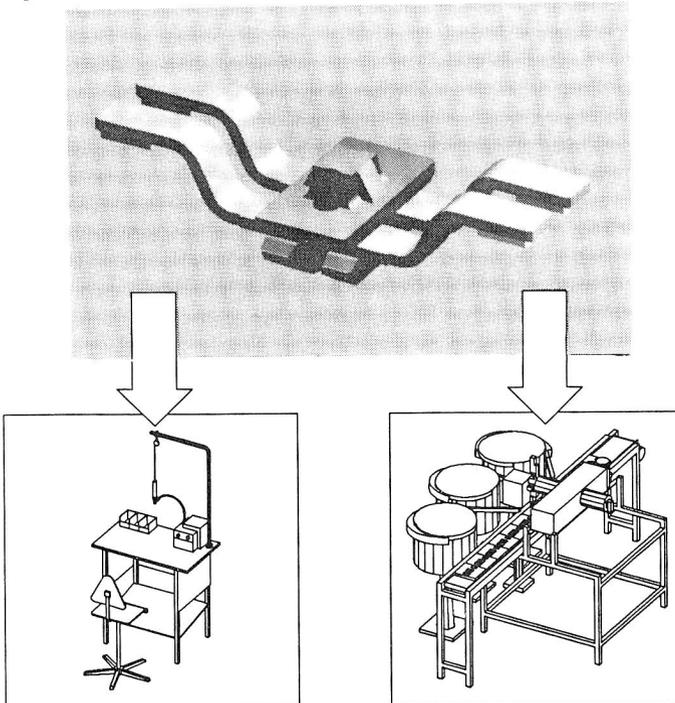


Bild 9: Beispiel eines zu montierenden Produktes und alternative Montagelösungen (entwickelt im System CARo)

Im Gegensatz zu manuellen Montagesystemen müssen fehlende Flexibilität, mangelhafte Fehlertoleranz und geringe Autonomie von Automatisierungseinrichtungen durch höheren Aufwand in der Montageplanung ausgeglichen werden. Es ist wünschenswert, Automaten

ähnlich einfach einzusetzen, d.h. zu planen, zu installieren und einzulernen, wie manuelle Arbeitssysteme. Erhöhung der Flexibilität in automatisierten Systemen verlangt schnelles und einfaches Programmieren der Montagekomponenten. Bessere Fehlertoleranz kann erreicht werden durch Planung und Programmierung von Sensorsystemen, mit deren Hilfe Strategien zur Fehlererkennung, Fehlervermeidung und Fehlerbehandlung angewendet werden können. Anlaufzeiten können mittels aufgaben- und technologieorientierter Programmierverfahren beschleunigt werden.

Planungs- oder Programmierwerkzeuge müssen so konfiguriert werden, daß sie nicht nur den momentan sehr eingeschränkten Funktionsumfang eines Industrieroboters berücksichtigen, sondern auch sensorische Fähigkeiten und lokale Intelligenz von Automatisierungseinrichtungen der Zukunft. Ziel der Forschung ist es technische Arbeitsmittel über das Maß menschlicher Fähigkeiten hinaus zu vervollkommen. Daher muß, wie in der vorliegenden Arbeit gezeigt, die Entwicklung von modernen Planungs- und Programmierwerkzeugen diese zukünftige, verbesserte Autonomie und Funktionalität der Maschinen auf Betriebsebene berücksichtigen. Der Vergleich zwischen manuellem und automatisiertem Arbeitsplatz läßt erkennen, welche Stärken bzw. Schwächen Menschen gegenüber Automaten in der Montage besitzen und wohin die Entwicklung von Montagesystemen gehen muß, um die menschliche Arbeitskraft wirkungsvoll nachzubilden. Aus dem Vergleich lassen sich für Planungs- und Programmierwerkzeuge konkret ableiten:

- die Definition der Schnittstelle zwischen Planung und Betrieb
- Modellinformationen über Montagekomponenten als Grundlage planerischer Entscheidungen,
- Anforderungen an Programmiersysteme zur Generierung von Steuerinformationen aus Sicht des Montageplaners.

Außerdem zeigt der Vergleich das Defizit, aber auch die Vorteile auf (siehe Bild 14), die automatisierte Montageeinheiten gegenüber manuellen Arbeitsplätzen heute besitzen /23/.

3.1 Der manuelle Montagearbeitsplatz

Die zentrale 'Komponente' eines manuellen Montagearbeitsplatzes ist der Mensch (siehe Bild 11). Aufgrund seiner physiologischen und mentalen Eigenschaften (siehe z.B. die Beweglichkeiten des Hand-Arm-Systems Bild 10), ist er bis heute jeder Maschine, insbesondere dem Industrieroboter überlegen. Was heute als Roboter bezeichnet wird, ist nüchtern betrachtet lediglich die mechanisch kinematische Imitation des menschlichen Armes, wobei die hochgradig adaptiven sensorischen Fähigkeiten des menschlichen Vorbildes bisher nicht annähernd nachgebildet werden konnten.

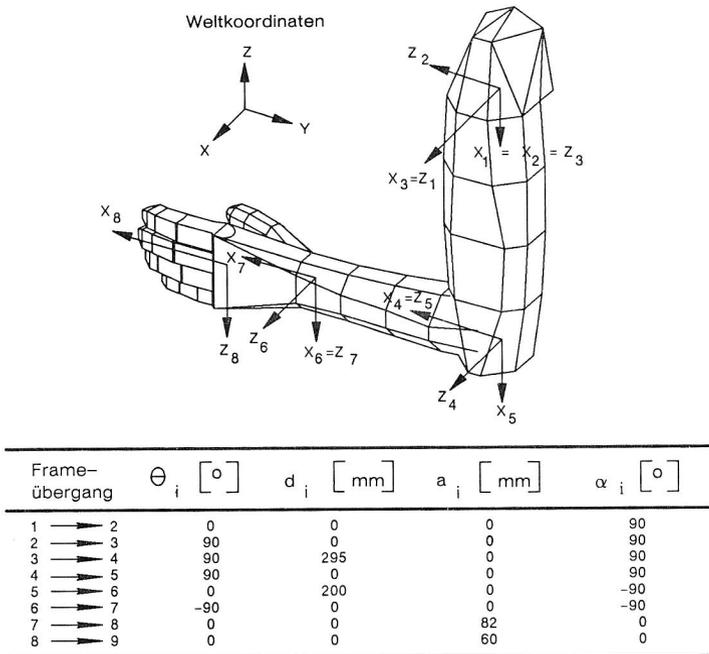


Bild 10: Kinematik des Hand - Arm Systems (Motiv aus CARo)

Die Vorteile manueller Montagesysteme sind gekennzeichnet durch:

- Flexibilität bezüglich Losgrößen und Varianten,
- geringe Investitionskosten,
- geringer Umrüstaufwand,
- kurze Anlauf- und Einlernzeiten,
- hohe Fehlertoleranz aufgrund der Prozeßoptimierung durch die Arbeitsperson.

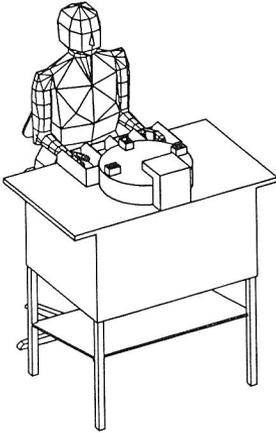
3.1.1 Komponenten des Montagehandarbeitsplatzes

Gleichgültig, welche Montageart (Massen-, Serien-, Einzelmontage) vorliegt, ob es sich um Gruppenmontage, Innen- oder Außenmontage handelt, ob eines oder mehrere Produkte bzw. Produktvarianten in einem System montiert werden, manuelle Montagesysteme besitzen stets drei unterschiedliche Komponenten /24/. Es sind dies:

- a) Mensch(en)
- b) Arbeitsobjekte
- c) Betriebsmittel/-stoffe.

Ausführendes und überwachendes Organ in einem solchen Montagesystem ist der Mensch. Seine Aufgabe es ist, Arbeitsobjekte (z.B. Einzelteile, Baugruppen) zu komplexeren Gruppen oder Fertigprodukten zusammenzubauen (=montieren).

Zur Durchführung der Montage stehen Vorrichtungen (z.B. Werkstückträger, Transporteinrichtungen), Werkzeuge (z.B. Schrauber, Pinzetten) und andere Betriebsmittel (wie Werkbank, Greifbehälter) sowie Betriebsstoffe zur Verfügung. Aufgrund seiner Fertigkeiten benötigt der Mensch zur technischen Abwicklung einer Montageaufgabe in der Regel sehr wenige Werkzeuge. Bezeichnend für den manuellen Arbeitsplatz ist vor allem die Einfachheit der eingesetzten Betriebsmittel. Jene besitzen häufig keine eigenen Steuer- bzw. Regeleinrichtungen, da der Mensch die verschiedenen Montageabläufe selbst koordiniert, überwacht und steuert.



- Keine lokalen Steuer- und Regeleinrichtungen
- relativ wenige Betriebsmittel
- Arbeitsperson führt aus:
 - + Koordination, Steuerung
 - + Transportvorgang
 - + Werkzeughandhabung
 - + Montagevorgang
 - + Informationsaustausch
 - + Qualitätssicherung

Bild 11: Kennzeichen des manuellen Montagearbeitsplatzes
(Motiv aus CARo)

3.1.2 Planung und Einweisung

Der Montageplanungsablauf für manuelle Montagesysteme ist im Normalfall weniger aufwendig als der für automatisierte Systeme. Dies liegt daran, daß der größte Teil der Funktionalität einer Montagezelle vom 'Teilsystem Mensch' ausgeführt wird. Durch die menschliche Arbeitskraft wird ein Höchstmaß an Aufgabenflexibilität sichergestellt. In der Planung muß daher erheblich weniger Aufwand betrieben werden für:

- die Auswahl und Zuordnung von Betriebsmitteln
- die Einweisung/Programmierung
- die Ermittlung der Systemkosten.

Da für manuelle Montagesysteme im allgemeinen geringere Investitionen getätigt werden müssen, kann weniger Wert auf eine ausführliche und detaillierte Montageplanung gelegt werden. Dies äußert sich in der Praxis häufig so, daß Montagesysteme nach

kurzer grober Planung aufgebaut und erst während des Produktionsablaufes optimiert werden, bzw. sich selbst durch die Fähigkeiten der Arbeitsperson optimieren; eine Strategie, die bei automatisierten Systemen wegen der geringen Aufgabenflexibilität der Teilsysteme, nicht praktikabel ist.

Trotz dieser Vorgehensweise gibt es auch Verfahren, um manuelle Montagesysteme methodisch und genau zu planen. Ein Beispiel hierfür liefern die 'Systeme vorbestimmter Zeiten (SvZ)', wie MTM oder WF. Mit diesen Methoden kann durch die Zerlegung der Montageaufgabe in ihre elementaren Bewegungen und Operationen die Ausführungszeit exakt bestimmt und die Folge der Bewegungssequenzen ausführlich definiert werden. Es steht damit ein Werkzeug zur Verfügung, das einerseits die Arbeitsgestaltung unterstützt /25/ und andererseits eine saubere Abtaktung manueller Montagesysteme ermöglicht. Die Planungsmethode nach SvZ bietet weiterhin den Vorteil, daß bei Inbetriebnahme und Einweisung auf eine standardisierte Form der Datenübergabe von der Planung an das Einricht- und Montagepersonal zurückgegriffen werden kann. Die Planungs- und Einweisungsunterlagen sind eindeutig dokumentiert und garantieren eine unverfälschte Weitergabe von Informationen. Da bei SvZ der Bewegungsvorgang auf Planungsebene vollständig analysiert und berechnet wird, bleibt es nicht mehr dem Einrichter oder dem Montierer überlassen, den Arbeitsvorgang zu gestalten.

Beim Einsatz von SvZ kann man eine gewisse Ähnlichkeit zwischen Einweisung von Arbeitspersonen und der Programmierung von Robotern erkennen, da die Zerlegung einer Bewegungsaufgabe in feinste Inkremente erfolgt. SvZ eignen sich ausgezeichnet für die elektronische Datenverarbeitung, denn das Errechnen der vielen Einzelzeiten aus Zeittabellen und die Dokumentation der Planungsergebnisse in Formularen kann leicht mit Hilfe des Rechners ausgeführt werden. Wie von Lacis /26/ gezeigt, können SvZ-Methoden durchaus auch auf die Planung automatisierter Montageanlagen angewendet werden. Verfolgt man diesen Gedanken weiter, läßt sich daraus ein erster Ansatz für die aufgabenorientierte Planung und Programmierung von Montageanlagen ableiten. Durch die starke Ver-

feinerung einer Montageaufgabe kann deren Komplexitätsgrad entscheidend verringert werden und damit möglicherweise eine rechnerische Interpretation zulassen.

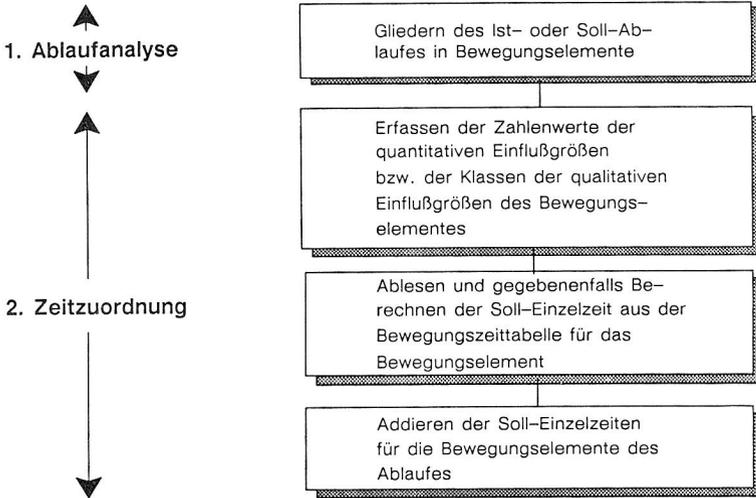


Bild 12: Montageplanung mit MTM (nach /25/)

Zur Einweisung von Arbeitspersonen in Montagesysteme werden unterschiedliche Verfahren eingesetzt. Im Idealfall geht man hierbei entsprechend der 4-Stufen-Methode nach REFA /27/ vor. Diese zeigt, welche Informationen und Informationsflüsse bei der Einweisung von Montierern gebraucht werden. Es läßt sich auf diese Weise ermitteln, welches Wissen einem System zugeführt werden muß, damit die Arbeitsperson die Arbeitsaufgabe vollständig und eigenständig ausführen kann. Diese Informationen (wie z.B. Bezeichnungen, Lage von Teilen oder Werkzeugen) können auch als Mindestanforderungen an eine Datenschnittstelle zur aufgabenorientierten Programmierung von automatisierten Montagezellen dienen. Eine solche Datenschnittstelle ließe sich u.U. auch zur Vereinheitlichung der Programmierung/Einweisung von Automaten und Arbeitspersonen einsetzen.

Häufig wird in der Montage zum Einlernen einer Montageaufgabe nach dem Prinzip 'Vormachen - Nachmachen' verfahren, wobei Arbeitspläne, technische Zeichnungen, Montageteile und der eingerichtete Arbeitsplatz zur Veranschaulichung eingesetzt werden. Erfahrene Montierer könnten allein anhand der Arbeitsunterlagen - Montageplan, Zusammenstellungszeichnung, Arbeitsplatzskizzen den Montagevorgang erlernen. Im Regelfall aber erfolgt das Erlernen einer Montage Tätigkeit aus der Kombination folgender Lehrmethoden:

- intensive Schulung (theoretisch, praktisch),
- Vormachen - Nachmachen,
- Detaillierter Montageplan,
- Arbeitsvorgang aus Zeichnung erkennbar,
- Arbeitsvorgang aus Montageplatz erkennbar,
- aus ähnlicher Arbeit ableitbar.

Das Erlernen der Montageaufgabe resultiert aus einer Verknüpfung von Informationszuführung und bereits vorhandenem Wissen, bestehend aus Erfahrung und trainierten Verhaltensmustern. Eine entscheidende Rolle spielt die Erfahrung des Montierers im Lernprozeß. Zahlreiche Fertigkeiten und Fähigkeiten hat sich der Mensch über Jahre hinweg aneignen können. Das im Laufe des Lebens erworbene Wissen ermöglicht es ihm, Arbeitsaufgaben mit einem Minimum an Informationszuführung zu verstehen und ausführen zu können. Hochgeübte Fertigkeiten bringt ein geübter Montierer von vorneherein mit:

- Erkennen von Gegenständen (Farbe, Form, Bewegung)
- Identifizieren und unterscheiden von Objekten (Lage)
- Ermitteln der Bedeutung von Objekten (Werkzeug)
- Interpretation von Szenen (akustisch, haptisch, visuell, kinästhetisch)
- Steuerung und Regelung des Bewegungsapparates (Bewegen, Kraftaufbringen)
- Behandlung von einfachen Fehlern (Bsp.: Teil liegt außerhalb der vorgesehenen Arbeitszentrums).

Es ist denkbar, daß Arbeitspersonen ausschließlich aufgrund dieser Fähigkeiten, ohne zusätzliche Anleitung bestimmte Montagearbeiten durchführen können.

3.1.3 Menschliche Leistung im Lern- und Montageprozeß

Nach /28/ wird Lernen als die durch Erfahrung und Übung (=Wiederholung, Training) relativ überdauernde Änderung von Verhaltensweisen und Handlungsmöglichkeiten bezeichnet. Das bedeutet, daß der aufgrund eines Lernvorganges gespeicherte Informationsgehalt für eine gewisse Zeit abgerufen und verarbeitet werden kann. Die Geschwindigkeit beim Speicherabruf und die Dauer der Informationsspeicherung ist abhängig von der Art der Informationsaufnahme und der Einprägung in verschiedene Gedächtnisformen. Man unterscheidet:

- Sensorisches Gedächtnis
- primäres Gedächtnis oder Kurzzeitgedächtnis
- sekundäres Gedächtnis
- tertiäres Gedächtnis

(nach Schmidt /29/).

Das sensorische Gedächtnis ist begrenzt durch die Aufnahmefähigkeit der angeregten Rezeptoren und Neuronen und wird daher häufig auch rezeptorisches Gedächtnis genannt. Die Verweilzeit der Information beträgt nur wenige Sekundenbruchteile.

Das primäre Gedächtnis erlaubt Speicherung von Informationen auch in verbaler Form bei einer Speicherzeit von mehreren Sekunden - typisches Beispiel ist die Szenenverarbeitung beim Autofahren.

Sehr große Speicherkapazität besitzt das sog. sekundäre Gedächtnis, das in der Umgangssprache fälschlicherweise als Kurzzeitgedächtnis bezeichnet wird. Hier werden Informationen durch mehrmalige Wiederholung in Form 'struktureller Engramme' /30/ eingepägt. Der Vergessenszeitraum variiert hier zwischen einigen

Minuten und mehreren Jahren. Es können komplexe räumliche und zeitliche Zusammenhänge in semantischer Form gespeichert werden. Die Zugriffszeiten sind unter Umständen sehr lang.

Werden durch sehr häufiges Wiederholen die erwähnten 'strukturellen Engramme' unauslöschar geprägt, so sind sehr schnelle Zugriffe auf gesuchte Informationen möglich. Man spricht hier von der sog. tertiären Form des Gedächtnisses oder Langzeitgedächtnis. Beispiele hierfür sind die Fähigkeit zu lesen, zu schreiben und andere Fertigkeiten, die durch jahrelanges Üben erlernt worden sind und praktisch nie vergessen werden können.

Der Unterschied zur digitalen Datenspeicherung liegt darin, daß Informationen nicht binär und eindeutig, sondern in Form von Mustern semantisch abgelegt werden. Die Summe aller gespeicherten Muster ergibt ein Wahrnehmungs-/Handlungsmodell der Welt. Jede Wahrnehmung des Menschen aktiviert die Suche nach einem passenden Muster, das bei Bedarf wiederum entsprechende Handlungen veranlassen kann. Werden keine passenden Muster gefunden, so können durch Aktivierung oder Simulation ähnlicher Wahrnehmungs-/Handlungsmuster auch neue unbekannte Situationen verarbeitet werden. Interessant ist, daß ein gefundenes Wahrnehmungs/Handlungsmuster nicht zwangsläufig die Ausführung einer Aktion zur Folge haben muß, sondern daß die Auswirkungen dieser Aktion mit Hilfe mentaler Simulationen geprüft und u.U. andere notwendige Handlungen abgeleitet werden können /31/.

Dies ist eine Fähigkeit, die Maschinen (Rechnern), aufgrund der schlecht komprimierten Daten, der Unvollständigkeit und Starrheit des Datenmodells, der mangelhaften Wahrnehmungs und Handlungsfähigkeit sowie den eingeschränkten Möglichkeiten zur Simulation von Aktionen noch lange verwehrt bleiben wird.

Die Fähigkeit zur Informationsverarbeitung wird als Intelligenz bezeichnet. Kennzeichen menschlicher Intelligenz sind breite parallele Kanäle zur Informationsaufnahme (Sinnesleistung), Informationsvorverarbeitung und -filterung, Verarbeitung von

unbekannten Zuständen durch mentale Simulationen, parallele Informationsverarbeitung, simultane Aktionsausführung, sowie hohe Zugriffsgeschwindigkeit auf hochgeübte -damit 'festverdrahtete'- Wahrnehmungs-/Handlungsmuster.

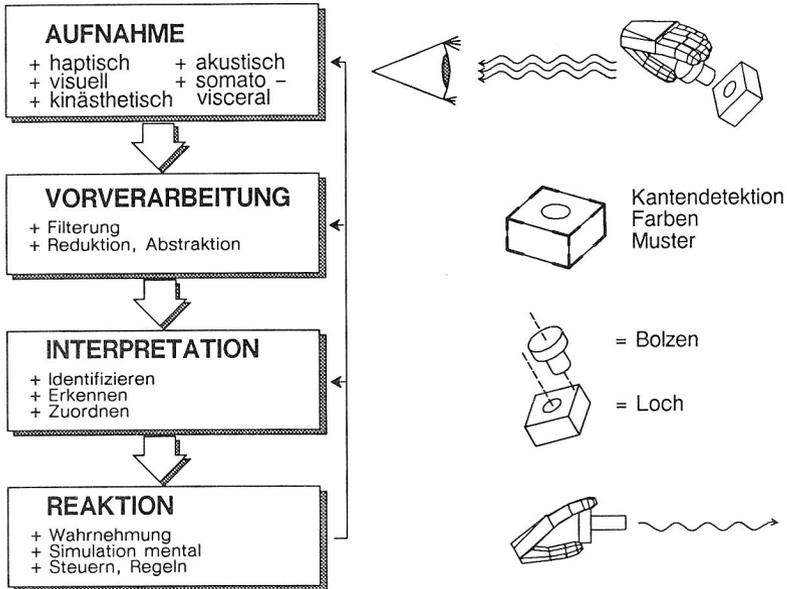


Bild 13: Mentale Verarbeitung eines Fügevorganges

Die besonderen Stärken des Menschen liegen in der Verarbeitung visueller, haptischer und akustischer Informationen. Am Beispiel der Bildverarbeitung läßt sich dies sehr gut nachweisen. Abgesehen von der reflexbedingten optimalen Einstellung des Lichteinfallendes (Variofunktionen des Auges) kann der Mensch im Takt von 10 Hertz Bilder mit einer Auflösung von 10^6 Bildpunkten aufnehmen. Bereits innerhalb der Retina (Netzhaut) findet durch 2×10^7 Neuronen eine Informationsvorverarbeitung statt, die die aufgenommenen binokularen Pixelinformationen auf ein Szenenmuster reduziert, das aus Umrissen -Kanten, Konturen-, Farben und Bewegungen besteht.

Im visuellen Cortex werden nach der Übertragung durch den Sehnerv weitere Vorverarbeitungen durchgeführt und Folgeprozesse –wie Klassifizierung, Mustererkennung, Interpretation und Handlungs-auslösung- angestoßen. Ein handelsüblicher Minicomputer benötigt zum Vergleich für die Vorverarbeitung eines 256x256 schwarz/weiß Bildes eine Zeit von ca. zwei Minuten. Der Informationsgehalt des Bildes ist dabei rund 50mal geringer und die Verarbeitungszeit um das 1200fache langsamer.

Die Ausführungen erklären, warum sich gerade im Bereich Montage Automatisierungseinrichtungen bisher nur wenig durchsetzen konnten. Ein Montageprozeß fordert im wesentlichen nämlich genau die Disziplinen, in denen der Mensch bis heute jeder Maschine weit überlegen ist:

- Erkennen und Identifizieren von Teilen,
- Ermitteln der Lage/Orientierung von Teilen in Relation zur Umwelt,
- Ausführen von Wahrnehmungs-/Handlungsmustern,
- Überwachen des Montageablaufs,
- Überprüfen der Zwischenergebnisse,
- Erkennen der Vollständigkeit und der Qualität der montierten Teile.

Menschliche Leistung läßt sich unterteilen in Leistungsbereitschaft und Leistungsfähigkeit. Das Leistungsangebot menschlicher Arbeit ist zeitlich nicht konstant und zudem individuell höchst unterschiedlich. Die Unterschiede von Arbeiter zu Arbeiter werden hervorgerufen durch: Alter, Geschlecht oder Konstitution. Auswirkungen auf das Leistungsangebot haben außerdem: Kondition, seelische Belastbarkeit, Einstellung zur Arbeit, Motivation, Herkunft und Erziehung sowie individuelle Mentalitäts- und Charaktereigenschaften /32/.

Die menschliche Leistung ist zudem starken Schwankungen ausgesetzt. Eine Rolle dabei spielen:

Motorische Leistung des Menschen

Stärken:

- hohe kinematische Freiheit
- Bewegungssteuerung und Koordination
- großes Spektrum: Feinmotorik – Grobmotorik
- dosiertes Kraftaufbringen
- hohe Anzahl von Greifmöglichkeiten der Hand

Schwächen:

- physiologische Grenzen
- Leistung, Kräfte, Wege und Geschwindigkeiten
- große oder sehr kleine Kräfte
- Dimensionalität der Ausführung (max. 7 unterschiedliche Aktionen gleichzeitig)
- Nichtlinearitäten (z.B. Koordination beschleunigter Systeme)

Sensorische Leistung des Menschen

Stärken:

- Farberkennen
- räumliches Sehen, räumliche Auflösung
- Erkennen von Bewegungen
- sensorische Bildverarbeitung
- Kraft-, Druck- und Beschleunigungsempfinden
- sensorische Kraftregelung (Servoregelung)
- parallele Informationsaufnahme (mehrere 1000 Sensoren = Rezeptoren)

Schwächen:

- physiologische Grenzen (visuell und haptisch)
- Bandbreite der Empfindungen
Licht 400 nm bis 800 nm
Schall 50 Hz bis 15000 Hz
- sonstiges Strahlungsempfinden
- Temperaturempfinden
- magnetisches und elektrisches Empfinden
- zeitliches Auflösungsvermögen
- Dimensionalität der Informationsaufnahme (maximal 7 unabhängige Vorgänge)

Mentale Leistung des Menschen

Stärken:

- sehr große Informationsspeicherung
- mentale Simulation
- Lernen
- Abstraktion
- Interpretation
- Aktionskoordination und Optimierung

Schwächen:

- Zugriffszeiten (u.U. sehr lang)
- Ermüdung (mental)
- Fehleranfälligkeit
- Vergessen

Bild 14: Stärken und Schwächen des Menschen im Arbeitsprozeß

- Tages-, Wochen-, Jahresrhythmik,
- Belastung/ Beanspruchung,
- Ermüdung/ Erholung,
- Krankheit.

Schwankungen in der Leistungsfähigkeit und Leistungsbereitschaft führen in der Produktion allerdings häufig zu schwer vorhersehbareren Fehlern oder Störungen. Zwar besitzt der Mensch die Möglichkeit, Fehler selbständig zu korregieren, doch sind Fehlaktionen in der Regel irreversibel (Zerstörung eines Teils), nicht sofort erkennbar (Fehlbestückung) oder wirtschaftlich nicht zu beheben. Die Fehleranfälligkeit menschlicher Arbeit steigt mit zunehmender Belastung bzw. Ermüdung. Aufgrund wachsender Qualitätsansprüche in der Produktion ist es daher wünschenswert, den Menschen trotz seiner enormen Fertigkeiten aus dem direkten Produktionsprozeß herauszulösen. Dies kann allerdings nur dann realisiert werden, wenn die besonderen Leistungen der menschlichen Arbeitskraft im Lern- und Montageprozeß wirkungsvoll nachgebildet werden können.

3.2 Die automatisierte Montagezelle

In Kapitel 2 ist bereits die Definition für eine flexibel automatisierte Montagezelle gegeben worden (siehe Bild 4). Danach können Zellen nicht nur Fertigungsinseln (Nester) sein, es ist auch möglich, daß sie intern eine Linienverkettung der Einzelkomponenten besitzen. Zellen selbst lassen sich zu Montagelinien verketteten oder zu Werkstätten zusammenfassen, die wiederum als Einheiten betrachtet werden können. In der Montageplanung ist es wichtig das Gesamtproblem (=Montage eines Produktes) in modulare Montageteilaufgaben zu gliedern. Im Rahmen der Montagefeinplanung werden abgeschlossene Teilaufgaben aus dem Montageablauf isoliert und durch die Gestaltung von Montagezellen gelöst (Bild 15).

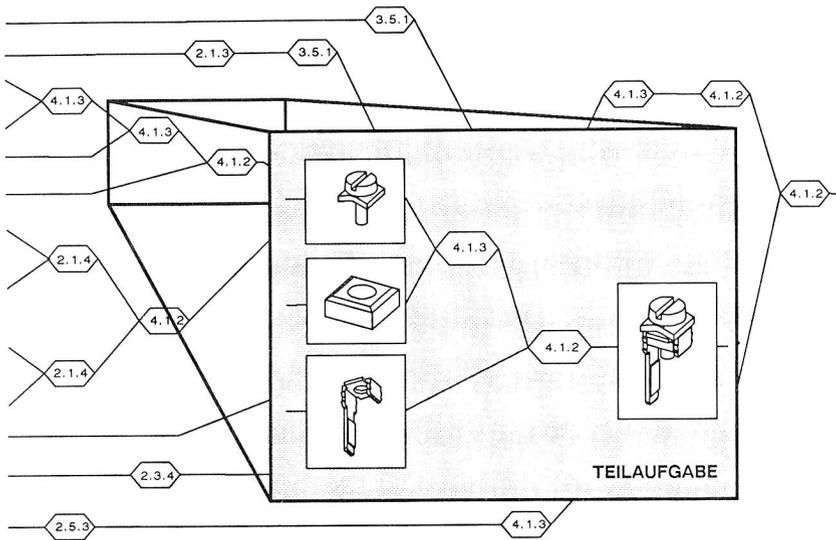


Bild 15: Teilaufgabe aus dem Montagegraph

3.2.1 Komponenten einer Montagezelle

Die Komponenten einer Montagezelle lassen sich zunächst in Steuerungs-, bzw Überwachungseinrichtung, Ausführungskomponenten und sonstige Komponenten unterteilen. Zur dritten Klasse 'sonstige Komponenten' zählen z.B. Schutzeinrichtungen oder Gestelle, die nur indirekt am Montageprozeß beteiligt, aber zur Aufgabenausführung notwendig sind. Bild 16 fächert die klassifizierende Beschreibung der Zellenkomponenten weiter auf. Betrachtet man die Zellenkomponenten systemtechnisch, so dienen die Steuerungs- und Überwachungseinrichtung der Informationsverarbeitung und die Klasse der ausführenden Komponenten zur Umsetzung von Material und Energie.

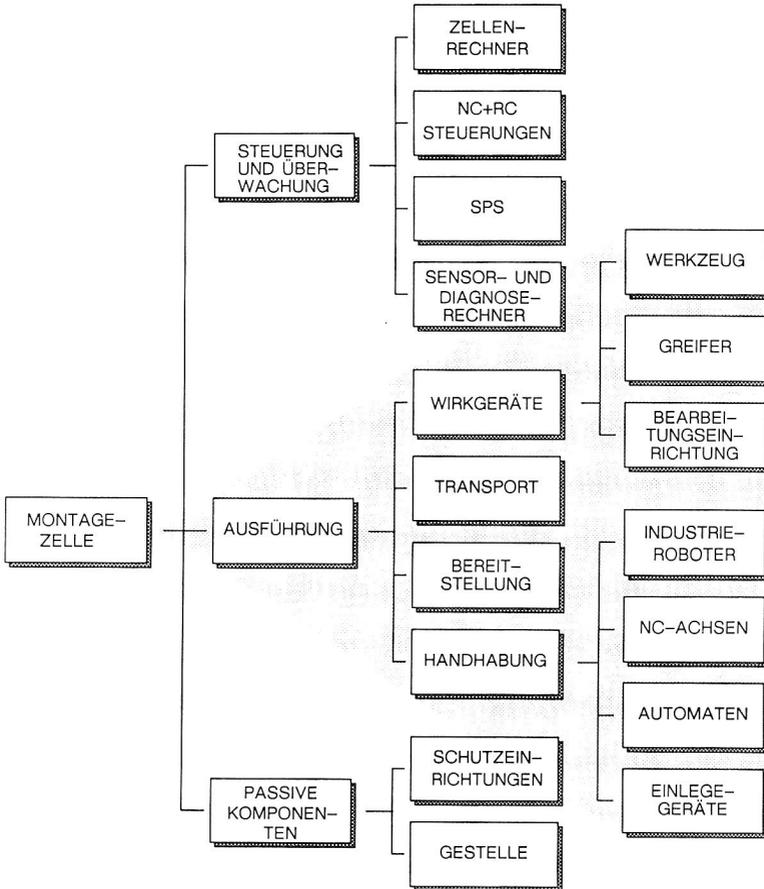


Bild 16: Komponenten einer automatisierten Montagezelle

Wesentlicher Teil einer Montagezelle ist ein übergeordnetes Leitsystem, welches hier der Einfachheit halber als Zellenrechner bezeichnet wird. Der Zellenrechner hat die Aufgabe den Gesamt Ablauf im Montageprozeß zu überwachen und bei auftretenden Problemen Störfallstrategien anzuwenden. Seine Aufgaben im einzelnen:

- Empfangen und Quittieren von Aufträgen
- Verwaltung aller Programme (für NC, RC, SPS, ...)
- Kommunikation mit den Einzelkomponenten der Zelle
- Überwachung des Ablaufs (Reihenfolge der Aktionen)
- Reaktion bei Störungen
- Kommunikation mit übergeordneten Systemen (Bedienpersonal, Montageleitrechner).

Zellrechnerarchitekturen oder standartisierte Schnittstellen zu Zellrechnern in der Montage befinden sich derzeit noch im Entwicklungsstadium. Es gibt allerdings Prototypen, welche die oben genannten Funktionen in einem Zellrechnerkonzept verwirklicht haben /33/.

Anders als bei ausreichend in der industriellen Praxis eingeführten Komponenten, wie NC-Achsen oder speicherprogrammierbaren Steuerungen, ist der Einsatz von Sensorsystemen oder Robotern schwierig, da diese technisch nicht ausgereift oder nicht standardisiert sind.

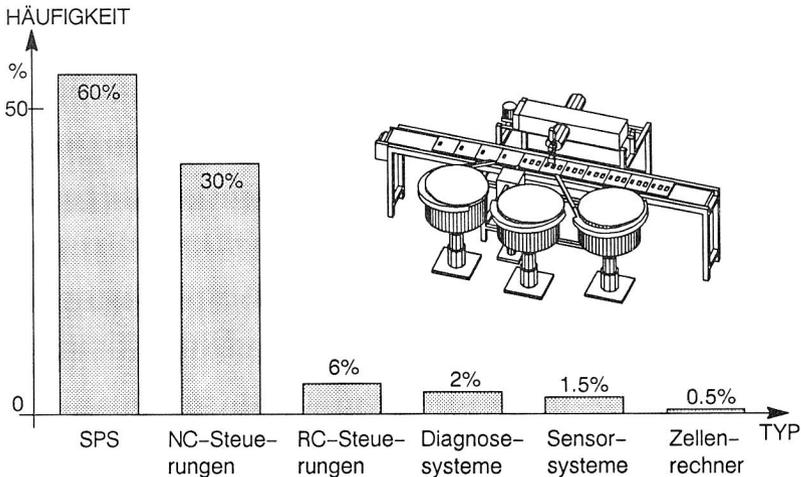


Bild 17: Verhältnis eingesetzter programmierbarer Steuer- und Überwachungseinrichtungen in der Montage (aus Verkaufszahlen eines Steuerungsherstellers)

a) Industrieroboter

Industrieroboter haben unterschiedliche Aufgaben in der Montage. Diese Aufgaben beinhalten Bewegungsabläufe, die entweder der Manipulation von Werkstücken (=Werkstückhandhabung) oder der Führung von Werkzeugen (=Werkzeughandhabung) dienen.

Industrieroboter können aus Sicht der Mechanik als Getriebe oder besser als kinematische Ketten beschrieben werden. Diese kinematischen Ketten sind in der Regel nicht einfach zu berechnen; dynamische Einflüsse bei der Steuerung der Geräte werden deshalb häufig nicht berechnet, sondern "ausgeregelt". Physikalische und regelungstechnische Details der Robotertechnik sind in der Literatur mittlerweile ausreichend vorhanden /34/, /35/, /36/, so daß an dieser Stelle keine Details erläutert werden brauchen.

Programmierbare NC-Achsen gehören im Rahmen dieser Arbeit zur gleichen Klasse wie Gelenkroboter (Klassifizierung anhand der Kinematik Kap. 5.2.2), sie sind lediglich mathematisch und steuerungstechnisch einfacher zu berechnen. Aufgrund ihrer geringeren Investitionskosten sind sie stark verbreitet. Der Einsatz von NC-Achsen hört allerdings dort auf, wo komplizierte räumliche Bewegungsabläufe erforderlich sind, wie beim Bahnschweißen, Punktschweißen, Spritzlackieren, Entgraten und bei sonstigen Bearbeitungen an freigeformten Geometrien.

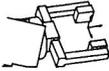
DIMENSIONALITÄT	TYP	BEISPIEL
Binäre Greifsysteme	Vakuumgreifer Zweibackensystem Greifrüssel	
Mehrfachgreifer	Mehrflächengreifer Wechselgreifer	
Gelenkgreifer	"Mechanische Hand"	

Bild 18: Klassen von Greifern

b) Greifer

Mit Hilfe von Greifern kann ein Roboter Werkstücke oder Werkzeuge aufnehmen, diese verdrehen oder versetzen und wieder loslassen. Greifer können auch aus programmierbaren Achsen bestehen, sie sind dann technisch ebenso zu behandeln wie Roboter. Komplizierte Mehrachsgreifer sind allerdings sehr teuer und werden in der industriellen Praxis kaum eingesetzt. Verbreitet sind binäre Greifersysteme die lediglich zwei Zustände kennen - offen oder geschlossen. Häufig sind auch Wechselgreifer oder Multifunktionsgreifer anzutreffen.

c) Sensoren in der Montage

Sensoren sind wesentliche Elemente zur Realisierung von zellen-internen Regelkreisen. Die Grundfunktion eines Sensors ist es, in einem vorgegebenen Takt Ist-Werte aufzunehmen. Diese werden in einer angeschlossenen Recheneinheit mit vorgegebenen Sollwerten verglichen. Abweichungen oder Übereinstimmungen können gemeldet und weiterverarbeitet werden. Beispiel: Identifikation von Teilen anhand des mitgeführten Barcodes. Zu einem diskreten Zeitpunkt liest der Sensor die Folge unterschiedlich dicker Streifen des Barcodes am Produkt und vergleicht diese mit den einprogrammierten Mustern. Findet der Sensorrechner beim Soll-Ist Vergleich das passende Muster, kann er ein entsprechendes Signal auslösen, um beispielsweise das richtige Bearbeitungsprogramm für das identifizierte Teil anzustoßen.

Auch schwerere Aufgaben lassen sich mit technischen Sensoren bewältigen. Bei der Qualitätssicherung von Leiterplatten müssen u.a. die Lage und Orientierung sowie die Anzahl der Einzelkomponenten auf der bestückten Leiterplatte überprüft werden. Der Sensor hat hier die Aufgabe, ein Ist-Muster der Platine aufzunehmen und mit dem Soll-Muster zu vergleichen. Was hier Probleme macht, ist die durch Umwelteinflüsse erschwerte Musteraufnahme sowie die Größe der zu verarbeitenden Datenmengen beim Mustervergleich.

MEßGRÖßE	SENSORBEISPIEL
Geometrie	Wegmeßsysteme, Lehren
Zeit	Zähler, Uhren
Mechanische Meßgrößen	Kraftmesser (DMS), Ultraschallsensor
Elektrische, magnetische, Meßgrößen	Elektrischer Schalter Näherungssensor (induktiv)
Thermische Meßgrößen	Thermostate
Optische Meßgrößen	Kamera (CCD), Lichtschranken Laserscanner

Bild 19: Klassen von Sensoren

d) Werkzeuge und Bearbeitungseinrichtungen

Während Rechnerkomponenten, Roboter und Greifer die Funktion des Menschen in der Montage ersetzen, besitzen Werkzeuge und Bearbeitungseinrichtungen die gleiche Funktion wie in manuellen Arbeitssystemen. Auch in der automatisierten Montage müssen spezielle, an das Produkt und die Handhabungssysteme angepaßte Werkzeuge und Vorrichtungen entwickelt werden. Eine Systematik läßt sich anhand der Aufgabe, nicht anhand der technischen Ausprägung treffen.

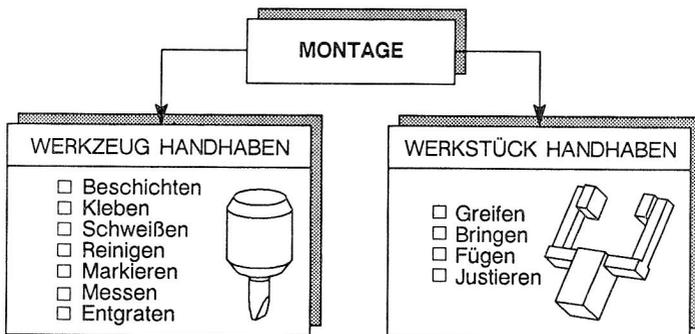


Bild 20: Aufgaben von Werkzeugen und Bearbeitungseinrichtungen

e) Transport und logischer Ablauf

Die Aufgaben des zelleninternen Materialflusses sind mit dem logischen Ablauf in der Zelle eng verwandt. Das Materialflußsystem ist die Repräsentation des Ablaufs. Unterschiedliche Systeme kommen für Transportaufgaben innerhalb von Montagesystemen zur Anwendung. Es können dies sein: Bänder, Rutschen, Rollenbahnen,

-
- Ein-/Ausschaltung unstetiger Fördereinrichtungen
 - Schaltung von Weichen bei unstetigen Fördereinrichtungen
 - Starten/Stoppen von Bearbeitungssystemen
 - Schalten binärer Glieder (Greifer auf/zu)
 - Verarbeitung analoger Signale
 - Logische Verknüpfung
 - zeitliche Synchronisation
-

Bild 21: Aufgaben von SPS

sonstige stetige oder unstetige Fördermittel. Die steuertechnische Koordination wird in flexiblen Zellen nicht mehr durch starre Festverdrahtung ausgeführt, sondern durch sogenannte Speicherprogrammierbare Steuerungen /37/.

SPS sind heute die gebräuchlichsten Steuerungen in der Automatisierungstechnik. Neben binären (digitalen) Größen können SPS auch analoge Signale verarbeiten und damit über Sensoren auf Schwellwerte bei Lärm, Hitze oder Überspannungen reagieren.

3.2.2 Geräteorientierte Programmierverfahren

Montagesysteme bestehen aus einer Anzahl verschiedener Einzelkomponenten mit sehr unterschiedlichen Programmiermöglichkeiten

bzw. Steuerungen. Zur Koordination von Roboterzellen werden verschiedene Steuerungen eingesetzt:

- Robotersteuerungen (RC)
- Sensorrechner
- numerische Steuerungen (NC)
- speicherprogrammierbare Steuerungen (SPS).

Beim Einsatz und bei der Programmierung dieser Geräte muß der Programmierer bzw. der Montageplaner aus den zu erfüllenden Montageoperationen gerätespezifische Steuerprogramme bzw. Gerätebewegungen generieren. Die so erzeugten Einzelprogramme lassen keinen Umkehrschluß auf die ursprüngliche Aufgabe mehr zu. Deshalb geht auch die Bedeutung der Montageaufgabe in den Programmen verloren. Die Merkmale dieser, auch als explizite Programmierung bekannten Methode sind:

1. Explizites Programmieren erfordert genaueste Gerätekenntnis.
2. Explizites Programmieren verlangt Kenntnisse in der Informationsverarbeitung und starkes Abstraktionsvermögen.
3. Explizite Programme können immer nur im Gesamtgefüge des Montagesystems erstellt/verändert werden.
4. Explizite Programme erlauben keine selbständige Reaktion auf Fehler und Störungen, da die Bedeutung von Ereignissen nicht in den Programmen erkennbar ist.
5. Explizit erstellte Programme sind schwer zu erzeugen und zu ändern.

Die Merkmale dieser Methode bewirken einen erheblichen Zeitverlust bei der Erstellung bzw. Änderung von Programmen, erschweren die Programmiertätigkeit und erhöhen die daraus resultierende mentale Belastung beim Programmieren. Außerdem kann die Verfügbarkeit von explizit programmierten Montageanlagen nicht durch selbstständige Fehlererkennung und -beseitigung erhöht werden.

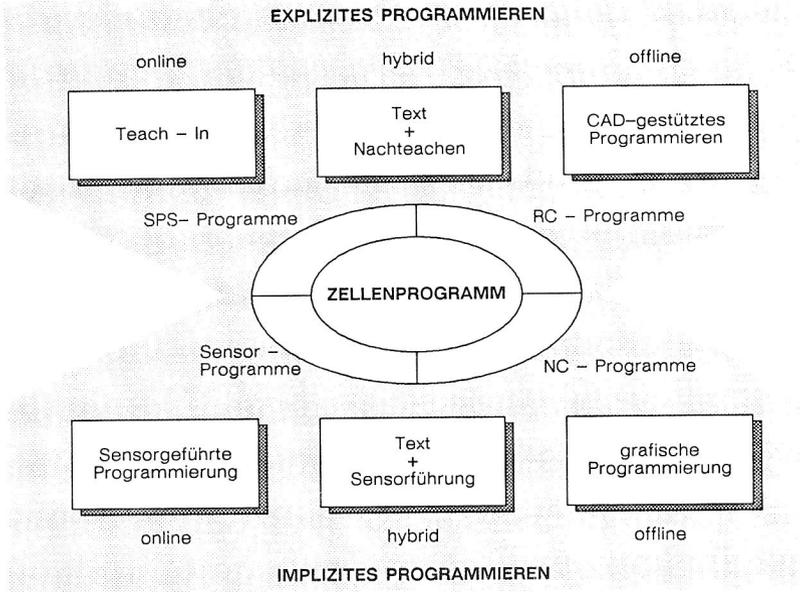


Bild 22: Programmiermethoden

Die explizite Programmierung ist geschichtlich mit der Entwicklung und Konstruktion von Automaten gewachsen, da die Gerätehersteller ihre Produkte möglichst anwendungsneutral verkaufen wollen. Die Berücksichtigung von Anwendungsproblemen bleibt den Anlagenplanern überlassen.

Bei der gerätespezifischen Programmierung werden unterschiedliche Methoden angewandt (Bild 22). Man kann dabei grob zwischen online und offline Programmierung unterscheiden. Um die Stillstands- oder Anlaufzeiten möglichst gering zu halten, werden zunehmend Offline-Programmiermethoden eingesetzt. Dennoch werden Montagegeräte vor allem Roboter, heute vorwiegend online direkt vor Ort programmiert.

a) Programmierung am Gerät (online)

Während zur Programmierung von logischen Abläufen in Montagesystemen häufig offline Programmiermethoden eingesetzt werden, erfolgt die Programmierung von Robotern und anderen Handhabungssystemen heute zu über 90% vor Ort d.h. direkt am und mit dem Gerät selbst.

Bei der Programmierung eines Roboters für eine Montageaufgabe muß die gesamte Zelle vollständig aufgebaut sein. Die komplizierten räumlichen Bewegungen des Roboters werden dann eingelernt. Aufgrund dieses Einlernvorganges wird das Verfahren auch englisch als Teach-In bezeichnet. Man unterscheidet:

- Teach-In am Roboter (Playback),
- Teach-In mit Sensorgriff und Sensorsteuerung,
- Teach-In mit Lehrgerüst (Playback),
- Teach-In mit Teach-Box,
- hybride Programmierung.

Das Teach-In am Roboter erfolgt durch manuelles Führen des Roboterarmes an definierte Positionen, deren Koordinaten durch Knopfdruck in der Steuerung gespeichert werden. Diese Art der Programmierung ist eine sehr kraftraubende Methode. Am weitesten verbreitet ist das Teach-In mit Teach-Box, bei dem über eine Steuertastatur die entsprechenden Raumpunkte angefahren werden können. Teach-In mit dem Sensorgriff und Servounterstützung erlaubt eine kraftsparende manuelle Führung des Roboters an dessen Endeffektor. Ebenso einfach ist das Programmieren mit Lehrgerüst. Lehrgerüste besitzen die gleiche Kinematik wie der Roboter selbst, lassen sich aber wegen fehlender Motoren und Aufbauten leichter führen. Verfahrenwege werden automatisch gespeichert.

Ein erster Schritt zur Offline-Programmierung ist das "Hybride Programmieren". Das Programmgerüst wird hierbei textuell erstellt, die Positionen und Verfahrenwege müssen allerdings wieder in der Zelle mit dem Roboter ermittelt werden.

Alle diese Methoden bewirken bei häufig wechselnden Produktspektren hohe Rüstzeiten bzw. Produktionsstillstand. Es wird daher intensiv an echten Offline-Systemen gearbeitet.

b) Programmierung ohne Gerät (offline)

Die Programmierung auf Planungsebene -auch Offline-Programmierung genannt- wird heute erst in wenigen Unternehmen eingesetzt.

Dem Wunsch, verschiedene arbeitsvorbereitende Tätigkeiten, wie Programmieren von Montagezellen, aus dem Produktionsbereich zu lösen und auf Planungsebene zu bearbeiten, steht der Mangel an geeigneten Werkzeugen gegenüber. In Kapitel 2 sind einige Systeme vorgestellt worden, mit denen bereichsweise Programmieraufgaben bewältigt werden können. Keines der dort genannten Systeme eignet sich jedoch für die komplette Montageprogrammierung.

Offline Programmiermethoden sind:

- Programmierung durch Schaltungsentwurf
- rein textuelles Programmieren in einer geräteorientierten Sprache
- grafisch interaktive Programmierung.

Programmierung durch Schaltungsentwurf kann auch als Programmierung durch Festverdrahtung bezeichnet werden. Abläufe in Montagezellen werden durch eine elektronische Schaltungslogik realisiert (vgl. auch Kreuzschienenverteiler).

Die nächst höhere Ebene, das textuelle Programmieren, erlaubt die Beschreibung von Abläufen in einer Sprache. Solche Sprachen können Assembler ähnlich sein oder auch Eigenschaften von höheren Computersprachen wie PLM, BASIC, PASCAL oder C besitzen. Beispiele hierfür sind: VAL II, ML oder SRCL /38/ zur Roboterprogrammierung bzw. STEP oder Programmierung nach DIN 19239 zur SPS - Programmierung /39/.

Durch den Einsatz von graphischen Systemen -zum Teil auch CAD-Systemen- können geometrische Daten und logische Ablaufanweisungen verknüpft und somit komplette Steuerprogramme erstellt werden. Grafische SPS-Programmierung kann durch Kontaktplan- oder Funktionsplandarstellung erfolgen /40/. Die grafisch interaktive NC-Programmierung von Werkzeugmaschinen gehört wegen zahlreicher verfügbarer CAD- NC Verfahrensketten zum Stand der Technik /41/.

3.2.3 Aufgabenorientierte Programmierung

Mit dem Ziel Montageprogrammierwerkzeuge für den Planungsbereich verfügbar zu machen, sind zahlreiche höhere Robotersprachen entwickelt worden, die sich vor allem durch folgende Eigenschaften charakterisieren lassen:

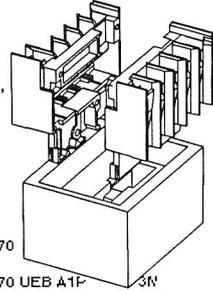
- geometrische Datentypen (Vektor, Rotation, Koordinatensystem)
- parallele, zyklische oder zeitlich verschobene Ausführung von Programmteilen
- verschiedene Bewegungsanweisungen
- digitale und analoge Ein-/Ausgabeschnittstellen.

Roboterprogramme werden jedoch mit wachsender Funktionalität der Sprache immer komplizierter und der Aufwand zur Erstellung, Wartung und Pflege immer umfangreicher. Die Folge davon ist, daß Montageprogrammierung weniger Ingenieurleistung als vielmehr den Einsatz von Computerexperten erfordert.

In den letzten Jahren findet man den Begriff "Implizite Roboterprogrammierung" in zahlreichen Veröffentlichungen. Vielfach begnügt man sich bei der Erläuterung des Begriffs, das Wort "implizit" durch den Ausdruck "aufgabenorientiert" zu ersetzen, was zwar den Begriff verdeutscht, ihn aber damit kaum verständlicher macht.

IMPLIZIT:

FAHRE "GEHÄUSEHÄLFTE" AN
 GREIFE "GEHÄUSEHÄLFTE"
 BRINGE "GEHÄUSEHÄLFTE" ZU "VORRICHTUNG"
 FÜGE "GEHÄUSEHÄLFTE" IN "VORRICHTUNG"



EXPLIZIT:

```
PTP X -408.0 Y -979.7 Z +537.7 A +104.281 B -0.210 C -17.
LIN X -407.9 Y -979.7 Z +396.7 A +104.280 B -0.210 C -179.970
GRF 1 ZU ;
LIN X -407.9 Y -979.7 Z +465.7 A +104.280 B -0.210 C -179.970 UEB A1P 3M
PTP X -1061.0 Y -25.7 Z +465.7 A +38.280 B -0.210 C -179.970 UEB A1P A2N A3M
PTP X +737.4 Y +11.2 Z +484.5 A -167.213 B -0.014 C +179.932 GND A1P A2P A
LIN X +737.4 Y +11.2 Z +400.5 A -167.213 B -0.014 C +179.932 GND A1P A2P
```

Bild 23: Vergleich impliziter und expliziter Programmierung einer Montageaufgabe

Eine weitere häufig aufzufindende Begriffserklärung wird anhand von Beispielen versucht, wie etwa durch folgenden "impliziten" Programmbefehl:

FÜGE "GEHAEUSEHAELFTE" IN "MONTAGEVORRICHTUNG"

Die Intention der Forschungsaktivitäten auf dem Gebiet aufgabenorientierter Roboterprogramme läßt sich durch solche einfachen Beispiele leicht beschreiben. Nicht der Roboter und seine Koordinatenwelt sind für den Programmiervorgang entscheidend, sondern die Aufgabe. Zwei wesentliche Mißstände bei der Anwendung heutiger Programmierverfahren lassen sich erkennen:

1. die Programmierfähigkeit an heutigen Robotersystemen ist nicht ergonomisch genug und dadurch zu aufwendig (vgl. explizites und implizites Programm Bild 23);
2. die heutigen Verfahren zur Roboterprogrammierung sind nicht genügend orientiert an gewachsenen Betriebsabläufen und sind deshalb nur mit Mühe in Produktionsbetrieben zu integrieren.

Betrachtet man heutige Roboterprogramme, so muß man erkennen, daß

die Programmierfähigkeit nichts mehr gemeinsam hat mit der zu erfüllenden Aufgabe. Die Tätigkeit des Roboters ist in abstrakter, für den Ingenieur oder Techniker schwer verständlicher Form innerhalb einer Steuerung hinterlegt. Der Fertigungsingenieur muß sich mehr mit Informatik und steuerungsspezifischen Tätigkeiten vertraut machen, was das Grundsätzliche seiner Arbeit, nämlich die kostengünstige Fertigung eines qualitativ hochwertigen Produkts, in den Hintergrund drängt.

Im folgenden soll also betrachtet werden, was unter impliziter Programmierung verstanden werden soll und welche grundsätzlichen Überlegungen für eine Forschungsarbeit auf diesem Gebiet angestellt werden müssen.

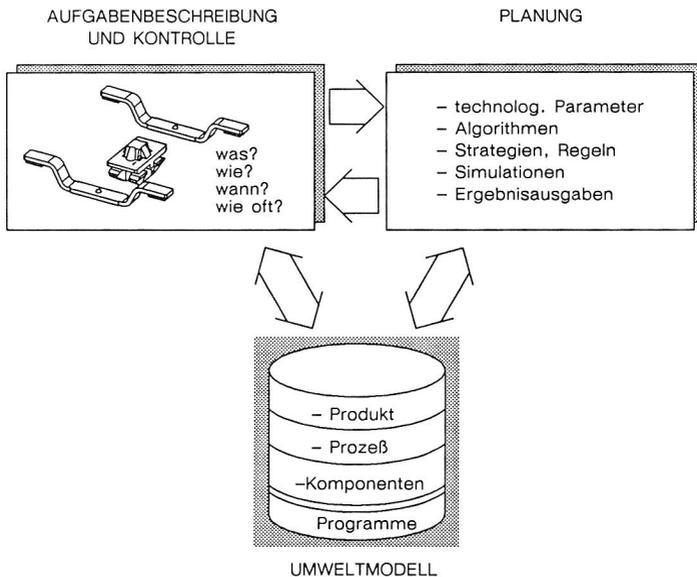


Bild 24: Grundelemente aufgabenorientierter Programmiersysteme

Ein System, das im wesentlichen aus Achsen, Gelenken, Antrieben und einigen Sensoren besteht, kann die Arbeitsaufgabe nur dann

'erlernen' und richtig ausführen, wenn drei wesentlich Bausteine als Grundelemente aufgabenorientierter Programmierverfahren vorhanden sind /42/:

1. Aufgabenbeschreibung (user-interface)
Eine Methode, um die Aufgabe für die Automatisierungseinheit zu beschreiben. Das kann z.B. die natürliche deutsche Sprache sein, das kann aber auch eine textuell syntaktisch straff formalisierte Eingabesprache sein.
2. Umweltmodell (hier PPM: ProduktProduktionsModell)
Grundwissen oder Basisinformation, die datentechnisch verfügbar sein muß, um Planungsvorgänge / Berechnungsvorgänge zu ermöglichen.
3. Planungsmodul
Modul das unter Rückgriff auf das PPM Probleme der Arbeitsaufgabe löst und letztendlich die implizite Aufgabe auf eine Folge expliziter Anweisungen abbildet.

Diese Grundelemente (Bild 24), die jedes implizite Programmierverfahren beinhalten muß, unterliegen nach Ansicht des Autors noch einigen Randbedingungen, die hauptsächlich unter dem Aspekt der Ergonomie und der Integrierbarkeit in gewachsene Abläufe und Strukturen zu sehen sind.

Aufgabenorientiertes Programmieren ist eine Problematik, die nicht alleine auf Planungsebene gelöst werden kann. Unerwartete Ereignisse, Störungen, Einflüsse der Auftragsmenge fordern im Produktionsprozeß lokale Anpassungen von Programmen an Produktionsbedingungen. Eine wichtige Ergänzung ist daher zum einen die Übertragung von Umweltinformationen in die Prozeßebene und zum anderen lokale Intelligenz durch Sensoren und Planungsprogramme. Ein Beispiel dafür liefert die Sensor gestützte Programmierung.

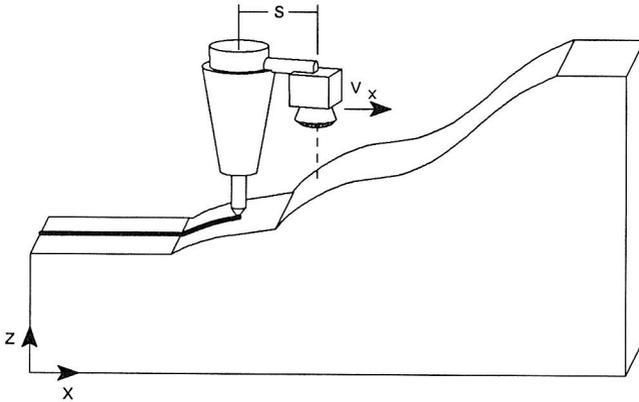


Bild 25: Abarbeitung einer aufgabenorientierten Anweisung durch lokale Intelligenz im Prozeß

Die Lösung der Aufgabe: Auftragen eines Klebstoffs, Ziehen einer Schweißnaht oder Führen eines Bearbeitungswerkzeuges, kann, wie in Bild 25 gezeigt, online durch eine vorlaufende Meßeinrichtung (Ultraschallsensor oder Scanner) auch während des Prozesses ermittelt werden.

4. Anforderungen an ein aufgabenorientiertes Planungs- und Programmiersystem

Zur Entwicklung von rechnergestützten Werkzeugen in der Montageplanung ist es grundsätzlich notwendig, eine genaue Analyse der Anforderungen, die aus der Planungstätigkeit resultieren, durchzuführen. Diese Aktivität wird in der Softwareerstellung nach /43/ als 'Definitionsphase' bezeichnet. Das Ergebnis dieser Phase läßt sich in Form eines Pflichtenheftes oder einer Anforderungsliste /44/ dokumentieren. Das Pflichtenheft beschreibt dabei in verbaler oder tabellarischer Form den funktionalen Inhalt und das Verhalten des zu entwickelnden Systems aus Sicht des späteren Benutzers. Es werden so die Schnittstellen bzw. Systemgrenzen zwischen System und Benutzer sowie zwischen System und anderen Werkzeugen in klarer und verständlicher Form wiedergegeben. Das Pflichtenheft muß frei von Konzeptideen sein, um ein weites Spektrum an Lösungsalternativen zu ermöglichen. Das Pflichtenheft kann außerdem zur Auswahl von Lösungsvarianten und zur späteren Systembewertung -Qualitätskontrolle- herangezogen werden.

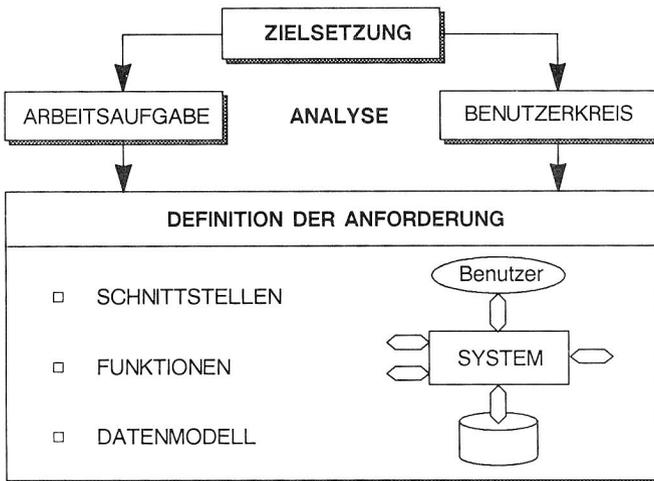


Bild 26: Vorgehensweise zur Anforderungsermittlung

In diesem Kapitel werden die wesentlichen Anforderungen, Schnittstellen an ein aufgabenorientiertes Planungs- und Programmiersystem in der Montage spezifiziert. Basis für die Spezifikation waren Fachgespräche mit Planungsgruppen aus Werken der Industrie, Roboter- bzw. Steuerungshersteller und Projektierungsexperten für automatisierte Montageanlagen.

Die globale Zielsetzung beinhaltet Zeitersparnis, Verbesserung der Planungsqualität, Sicherung von Know How und Integrierbarkeit in den gesamten Planungsprozeß.

Zeitersparnis kann zum einen durch Straffung des Planungsablaufes und zum anderen durch eine Verkürzung von Anlaufzeiten bei der Inbetriebnahme von Montagesystemen erreicht werden. Wird die Offline-Programmierung einbezogen, so ergibt sich eine Zeitersparnis durch die mögliche Verkürzung von Rüstzeiten.

Verbesserung der Planungsqualität kann gesichert werden aufgrund der Möglichkeit zum Durchspielen verschiedener Planungsvarianten und durch die Vermeidung von Planungsfehlern mit Hilfe von Werkzeugen der Simulation. Die Sicherung von Know How wird gewährleistet durch die in das System eingebrachten Informationen über Montageprozesse und Montagekomponenten, sowie durch die Speicherung bewährter Lösungen und Lösungsalternativen.

Oberster Grundsatz -nicht zuletzt wegen der übergeordneten CIM-Strategie- muß die Integrierbarkeit oder Einsetzbarkeit des Systems in den Montageplanungsprozeß sein.

4.1 Schnittstellen

Anhand der Schnittstellenbeschreibungen lassen sich Systemgrenzen definieren. Es wird gezeigt welche Informationen eingebracht werden und welche Informationen als Systemergebnisse ausgegeben werden.

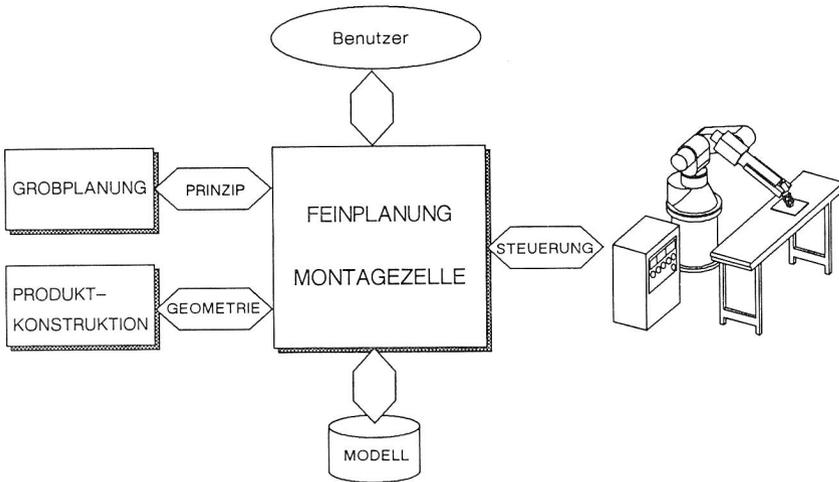


Bild 27: Schnittstellen zum Planungswerkzeug

Die wichtigste Schnittstelle eines Planungssystems ist die zum Benutzer. Das Hauptaugenmerk der Schnittstellenanalyse liegt deshalb in der Untersuchung seiner berufstypischen Ausbildung und Arbeitsweise sowie seiner Aufgaben und Tätigkeitsfelder. Nach dem Grundsatz von Nievergelt /45/ wird zunächst davon ausgegangen, daß die gesamte Aufgabe -hier Montagefeinplanung- automatisiert wird, um beim Versuch der Realisierung zu erkennen, was machbar ist und was (noch) nicht. Dies setzt voraus, daß die Tätigkeit eines Montageplaners in der Feinplanung genau beschrieben wird.

4.1.1 Der Anwender und seine Aufgaben.

Der Ablauf des Montageplanungsprozesses läßt sich unterteilen in Grob- und Feinplanung (siehe Bild 7). Die Montagegrobplanung geht aus vom Produkt und dem zu erwartenden Produktionsprogramm (Jahresstückzahlen, Auftragslose). Daraus werden abgeleitet:

- Montagereihenfolge
- Arbeitsprinzip des Systems (manuell, automatisiert, starr, verkettet, usw.)

- Grobe Ermittlung der Stückzeit
- Blockbildung (siehe /46/), Zellenbildung und Ermitteln der Taktzeiten
- Erstellung des Groblayouts (Hallengrundriß)
- Festlegung des Arbeitsprinzips der Zellen.

Im Anschluß an diese Aktivitäten, die bis zu diesem Zeitpunkt auch strategischen Entscheidungen des Managements unterworfen sind, gilt es die einzelnen Zellen detailliert zu gestalten. Hierzu werden in der Regel Fachkräfte, Spezialisten, wie Fertigungsingenieure oder erfahrene Techniker eingesetzt. Die Aufgaben der Feinplanung im einzelnen sind:

- Auswahl von Montagekomponenten und Zuordnung zur Aufgabe
- Positionieren von Komponenten im zur Verfügung stehenden Raum
- Konstruktion von Werkzeugen, Greifern, Vorrichtungen
- Erstellen von Layoutzeichnungen
- Erstellen der Stücklisten für die Montagezellen einschließlich Kleinteilen
- Festlegung des Arbeitsablaufs
- Erstellung der Schaltpläne (bei automatisierten Anlagen)
- Errechnen der Zellenkosten
- Testaufbau der Teilsysteme (Zellen)
- Verschaltung der Komponenten
- Einbau von Sensoren für MDE und Diagnose
- Programmierung der Steuerungen
- Testen des Versuchsaufbaus
- Messen der Taktzeiten, ggf. Optimierung
- Installation ins Gesamtsystem
- Starten des Systems, Anlauf im Verbund
- Anpassungen nach Anlauf der Produktion

Betrachtet man diese Aktivitäten und vor allem die Reihenfolge der Abarbeitung genauer, so erkennt man, daß ein hohes (Kosten)

Risiko mit dem frühzeitigen Kauf von Systemkomponenten und der Herstellung von Werkzeugen verbunden ist. Aus Sicht des Planers ist es daher wünschenswert, den Zeitpunkt der Investition auf einen möglichst späten Termin im Planungsablauf zu legen. Um dies zu gewährleisten, muß es ihm ermöglicht werden, ausschließlich durch eine virtuelle Testumgebung Fehler in seinem Feinkonzept auszuschalten. Durch eine Beschleunigung der Aktivitäten in der Planung können im gleichen Zeitraum mehrere Konzepte ausgearbeitet werden. Durch die Automatisierung von bestimmten Vorrichtungen kann der Anwender schließlich:

- das Planungsrisiko verringern,
- Fehler im Konzept vermeiden,
- Planungs- und Anlaufzeiten verkürzen.

Für den Montageplaner als Arbeitnehmer ergeben sich folgende Vorteile:

- Reduzierung der Belastung, Verantwortung
- Steigerung der Motivation durch größeren Erfolg.

Da das zu konzipierende Werkzeug die gesamte Arbeitsumgebung des Feinplaners repräsentieren soll, müssen alle genannten Aufgaben durch den Rechner unterstützt oder verwaltet werden. Dabei darf es z.B. keine Rolle spielen, ob ein manuelles oder teil- bzw. vollautomatisiertes Montagesystem geplant werden soll. Die Komplexität der Arbeitsaufgabe darf sich insgesamt nicht erhöhen und der Arbeitsablauf muß im großen und ganzen der gleiche bleiben. Zudem müssen die Fähigkeiten und Fertigkeiten der Anwender genutzt bzw. vom System bevorzugt angesprochen werden. Diese besonderen Fähigkeiten basieren bei Montageplanern auf maschinenbaulichen, fertigungstechnischen und elektrotechnischen Kenntnissen, sowie einer stark grafikorientierten Arbeitsweise (Zeichnung ist die Sprache des Ingenieurs). Informatikkenntnisse können in der Regel nicht vorausgesetzt werden. Das Werkzeugsystem muß in der Einlernphase Hilfestellungen geben, damit für den Anwender sehr schnell Arbeitserfolge sichtbar werden. Bei zunehmender

Gewöhnung an die Systembedienung soll das System in der Lage sein, sich der steigenden Arbeitsgeschwindigkeit anzupassen (z.B. durch reduzierte, effizientere Eingabemöglichkeiten und andere Lösungswege).

AUFGABENANGEMESSENHEIT

- Alle Planungsaktivitäten werden durch das System unterstützt
- Verwendung ingenieurorientierter graphischer Arbeitsweisen
- Beibehalten des üblichen Planungsablaufs
- Keine zusätzliche Belastung durch Systembedienung

SELBSTERKLÄRUNGSMÖGLICHKEIT

- Hilfe-Funktionen jederzeit verfügbar, aber nur bei Bedarf sichtbar
- Eingabevorschläge bei Fehlbedienung
- Kenntnisse des Systemaufbaus sind nicht erforderlich

STEUERBARKEIT

- Arbeitsgeschwindigkeit soll vom Montageplaner bestimmt werden
- Mehrere Formen der Steuerung wählbar
- Aktionen frei wählbar; Kreativität weiten Spielraum lassen
- Frühere Zustände wiederherstellen (UNDO, REDO, RECOVER)

ERWARTUNGSKONFORMITÄT

- Einheitlichkeit der Bedienung
- Rückmeldung nach jeder Aktion
- Systemzustand jederzeit erkennbar

FEHLERTOLERANZ

- Sinnvolle Fehlermeldungen, akustische und optische Markierung
- Keine undefinierten Systemzustände
- Korrekturmöglichkeiten

Bild 28: Forderungen an ein System zur Montagefeinplanung aus Benutzersicht (nach DIN 66234 Teil 8 /47/)

Die Benutzerumgebung wird sich an bereits existierenden und im Umfeld der Montageplanung eingesetzten Softwaresystemen orientieren. Die Oberfläche eines CAD-Systems wird hierbei eine entscheidende Rolle spielen, da ohnehin eine datentechnische Kopplung zur

Produktkonstruktion vorgesehen werden muß und auch in der Montageplanung konstruktive Aufgaben zu erfüllen sind. Die Übergänge von Systembaustein zu Systembaustein -gleichgültig ob nach oben oder unten- dürfen keine spürbare Änderung der Systembedien-
 ung zur Folge haben. Nur so kann sichergestellt werden, daß der Planer problemorientiert arbeiten kann, ohne sich um interne Funktionsabläufe und Systemdetails kümmern zu müssen.

DV-KENNTNISSE

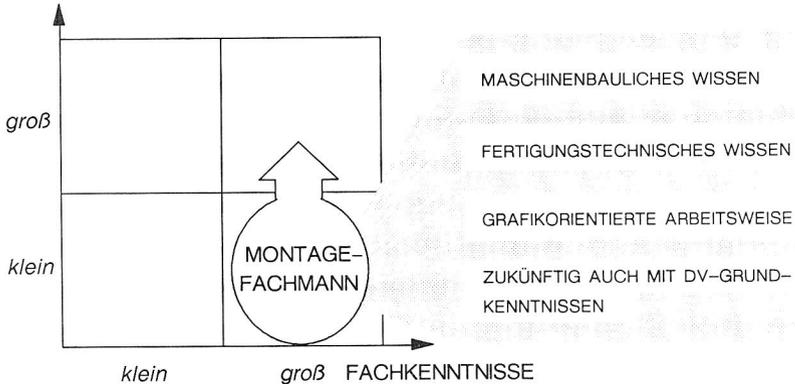


Bild 29: Zuordnung von Montageplaner und Benutzerklassen

Aufgrund heute verfügbarer Hardware kann gefordert werden, daß das System mehrprozeßfähig ist. Es besteht so die Möglichkeit, daß gleichzeitig unterschiedliche Teilaufgaben rechnergestützt bearbeitet werden können. Die Programmausführung sowie die Ein- und Ausgabe von Daten kann dadurch parallel erfolgen, Eingriffe des Benutzers während einer grafischen Simulation (z.B. Zoomvorgang während einer Roboterbewegung) werden so ermöglicht.

Mit Hilfe von Dialogprotokollen, Fehler- und Ergebnisprotokollen sowie Hardcopy- und Plotfunktionen hat der Benutzer die Möglichkeit seine Arbeit automatisch zu dokumentieren und zu archivieren. Dialogprotokolle können z.B. dazu dienen, eine Folge von Aktionen zu wiederholen oder in veränderter Form ablaufen zu lassen.

Hilfestellungen müssen der Erfahrung des Systembedieners angepaßt werden und sich dynamisch der Arbeitsweise des Benutzers anpassen. Es sollen dazu vier Formen der Unterstützung angeboten werden /48/:

- passive Hilfe: das System gibt Hilfestellung aufgrund einer expliziten Forderung des Benutzers,
- aktive Hilfe: das System leistet Hilfestellung, wenn Fehler festgestellt werden,
- statische Hilfe: gibt Auskunft über feste Strukturen im System,
- dynamische Hilfe: bei der Hilfestellung wird der aktuelle Systemzustand berücksichtigt.

Das Systemverhalten bei Fehlern soll flexibel sein, so daß im einfachsten Fall Meldungen den Grund des Fehlers erläutern bis hin zur aktiven Führung des Benutzers mittels beispielhafter Bediensequenzen oder konkrete Vorschläge zur Wahl einer veränderten Strategie.

4.1.2 Schnittstelle zur Konstruktion und Montagegrobplanung

In der Montagefeinplanung werden eine Reihe von Informationen aus vorgelagerten Bereichen weiterverarbeitet. Schnittstellen ergeben sich deshalb zur Produktkonstruktion und zur Montagegrobplanung (siehe Bild 7 und Bild 27). Daten, die aus der Produktkonstruktion stammen, betreffen vor allem geometrische und eine Reihe technologischer Informationen:

- Abmessung der Einzelteile
- Toleranzen
- Werkstoffe
- Topologie (Produktaufbau)
- Montagemakros (z.B. Fügehilfen).

In der Konstruktion werden solche Daten mittels CAD erzeugt und

verarbeitet. Zur Weiterverarbeitung in anderen Bereichen kann grundsätzlich auf zwei Weisen verfahren werden:

1. Direktes Aufsetzen der Anwendung auf das CAD
2. Übergabe von Daten über eine Schnittstelle.

Variante 1 ist grundsätzlich der technisch bessere Weg, weil von vorneherein auf gleiche Datenstrukturen, Bedienoberflächen und gleiche Grundsoftware zurückgegriffen werden kann. Auf der anderen Seite bringt diese Möglichkeit einige Nachteile mit sich. Ein direktes Aufsetzen von verschiedenen Anwendungen auf ein CAD-System setzt voraus, daß der CAD-Anbieter und Entwickler alle benötigten Applikationen, mit der geforderten Funktionalität anbieten, anpassen, pflegen und schulen kann. Der CAD-Anbieter erhält damit eine erhebliche Schlüsselstellung im Unternehmen des Kunden. Für das Unternehmen kann diese Abhängigkeit fatale Auswirkungen haben: Fehler können die gesamte technische Auftragsabwicklung lahmlegen, Systemaufrüstung muß nach Vorgabe des Anbieters und nicht nach eigenem Ermessen durchgeführt werden. Bestimmte Anwendungen, die später aktuell gebraucht werden, können nicht oder nur mit dem Willen oder Zutun des Anbieters eingepaßt werden. Bedenken sollte man auch, daß der CAD-Anbieter seine Produkte vom Markt nehmen könnte oder sogar selbst aufgrund neuer Hardware vom Markt verschwindet. Aufgrund strategischer Überlegungen -Kompatibilität mit Kunden und Zulieferern- wird die Variante 2 häufig bevorzugt. Die erfolgreiche Anbindung an CAD über Schnittstellen setzt allerdings voraus, daß alle im CAD erzeugten und zur Übergabe relevanten Daten über eine solche Schnittstelle übergeben werden können. Schnittstellen zu CAD sind derzeit weit gefächert (NC-Programmierung, Finite Elemente Berechnung, Formenbau, Dokumentation, PPS, u.a.). Als wichtige Standards haben sich dabei IGES (Initial Graphics Exchange Specification) und VDAFS (Verein Deutscher Automobilhersteller Flächenschnittstelle) herauskristallisiert.

Schnittstellen unterliegen beim Austausch von Daten erheblichen Restriktionen. Eine Standardschnittstelle kann nur das kleinste

gemeinsame Vielfache der Informationen übertragen, welche von unterschiedlichen Systemen erzeugt und verarbeitet werden.

STANDARD	ORGANISATION	BEMERKUNG
IGES	National BUREAU of Standard (U.S.)	seit 1980
XBF-2	CAM-I (U.S.) Computer Integrated Manufacturing International	seit 1982
VDAFS	Verein Deutscher Automobilhersteller (D)	seit 1983
SET	AEROSPATIALE (F)	seit 1984
PDDI	US Air-Force (U.S.)	seit 1984
ESP	National BUREAU of Standards (U.S.)	seit 1985
TAP	DIN (D)	seit 1986
CAD*I	Commission of the European Community (EG)	in Arbeit
STEP	ISO	weltweit in Arbeit

Bild 30: Schnittstellen zu CAD-Systemen /49/

Bis heute ist es daher über diese Standards nicht möglich, Volumeninformation, Produkttopologie mit Attributen und Toleranzen sowie anderen technologischen Informationen, wie sie in der Montagefeinplanung benötigt werden, normiert zu übergeben. Mit der Entwicklung und internationalen Normung von STEP kann sich dies in Zukunft ändern /49/. STEP sieht die Entwicklung eines Referenzmodells vor, das folgenden Anforderungen genügt:

- Verarbeitung von 2D und 3D Geometrien
- FEM - Kopplung
- Datenbankkopplung
- Einbettung produktdefinierender Daten für verschiedene Anwendungen, wie Maschinenbau, Elektrotechnik, Elektronik, usw.

- Kopplung von allen Unternehmensbereichen zur Übergabe von Produktinformationen.

Aus der Montagegrobplanung müssen folgende Informationen an das zu konzipierende System weitergegeben werden:

- Beschreibung der Montageteilaufgabe
- Grobes Arbeitsprinzip der geplanten Montageeinheit (manuell, automatisiert, Linen- bzw Nestfertigung)
- Maximal zulässige Taktzeit
- Bezeichnung und Menge der ein- oder ausgehenden Teile
- Vorgegebene Komponenten und Anschlußgrößen (z.B. Montageband oder FTS-Bahnhof)
- zur Verfügung stehender Raum.

Wird in der Grobplanung rechnergestützt gearbeitet, so können, aufgrund der Erstellung des Hallenlayouts mit CAD, Zellenabmessungen oder Anschlußgrößen weiterverarbeitet werden. Alle anderen Eingabegrößen müssen vom Benutzer nachträglich eingegeben werden. Es ist vorgesehen, sowohl Montagegrob- als auch Montagefeinplanung unter einem einheitlichen Konzept, mit einer Bedienoberfläche zu integrieren. Dies setzt voraus, daß das zu konzipierende Werkzeug als Modul in ein übergeordnetes Werkzeugsystem für die gesamte Montage eingebettet werden kann. Damit würden alle Informationen, die in der Grobplanung erzeugt werden, auch dem Feinplanungssystem zur Verfügung stehen.

4.1.3 Schnittstelle zur Betriebsebene

Eine der Grundforderungen für ein geeignetes Werkzeug zur Montagefeinplanung ist die Einbeziehung der Programmieraktivitäten in das System. Durch die Erstellung von Programmen auf Planungsebene lassen sich Stillstands- und Rüstzeiten minimieren. Außerdem kann die erstellte Software auf geeignete Weise dokumentiert und archiviert werden. Voraussetzung für die Realisierung der

Offline-Programmierung ist die Verwendung einer geeigneten Schnittstelle zwischen Planungsebene und Betriebsebene.

In der Praxis eingeführte Schnittstellen zu programmierbaren Automatisierungskomponenten bieten z.B. Roboterprogramme oder Logikprogramme für speicherprogrammierbare Steuerungen. Beispiele hierfür sind:

IRL	Normvorschlag einer höheren Roboterprogrammiersprache /50/
IRDATA	nach VDI-Richtlinie 2863 zur Programmierung von Robotern
CLDATA	nach DIN 66215 und ISO 8632 zur Programmierung von NC-Maschinen, vorwiegend Werkzeugmaschinen
NC-CODE	nach DIN 66025. Maschinennaher Code für NC-Maschinen, vorwiegend Werkzeugmaschinen
DIN 19239	Normierte Schnittstelle für die SPS-Programmierung
VAL (Unimation), SRCL (SIEMENS), u.a.	Herstellerspezifische Roboterprogrammiersprachen
STEP5 (SIEMENS) u.a.	Herstellerspezifische SPS-Programmiersprachen

Konzepte höherer Programmiersprachen wie IRL /50/ sind heute noch nicht von Steuerungsherstellern übernommen worden. Dennoch ist der Ansatz, anwendungsorientierte Programmierschnittstellen zu schaffen, ein sinnvoller Weg, um vor allem das Erstellen, Ändern, Übertragen und Testen von Programmen zu vereinfachen. Grundsätzlich sollte jedoch eine Trennung zwischen umweltbeschreibenden Informationen und ablaufbeschreibenden Informationen vorgenommen werden. Man erreicht auf diese Weise, daß man auf eingeführte Schnittstellen, welche ausschließlich explizite Beschreibungsformen für Montageoperationen verwenden, zurückgreifen kann (Stufe 1, Bild 31), ohne auf eine spätere Erweiterung für implizite

Anweisungen zu verzichten (Stufe 2, Bild 31). Dazu müssen vom Programmiersystem aus aufgabenorientierten impliziten Montageanweisungen explizite Programme generiert und parallel dazu Umweltinformationen, wie Bauteilgeometrien, Abmessungen im Arbeitsraum, sowie globale Steuerstrategien in Form eines Zellenprogrammes mitgeliefert werden.

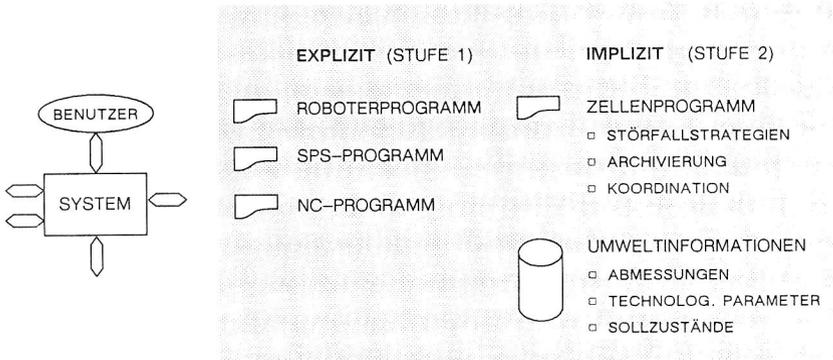


Bild 31: Schnittstellen zur Betriebsebene

Globale Steuerstrategien sollen beispielsweise im Störfall durch Diagnoseprogramme und Referierprogramme Fehler im System erkennen und beheben. Ein solches Störfallprogramm für den Zellenrechner kann wie folgt arbeiten:

1. Abfragen aller registrierbaren aktuellen Signale
2. Vergleich des Signalmusters (IST) mit dem SOLL-Muster zu diesem Zeitpunkt (Umweltinformationen)
3. Auswahl eines Referierprogrammes oder Senden einer Störmeldung an das Bedienpersonal.

Neben Störfallprogrammen hat ein Zellenrechner Strategien zum Umrüsten beim Auftragswechsel oder zur Abwicklung des Produktionsauftrages.

Da es heute keine Standards für Zellenrechner, deren Funktionalität und Programmierung gibt, kann der oben beschriebene Aspekt nur fiktiv im Konzept behandelt werden. Für eine integrierte Kopplung von Planungsebene und Betriebsebene müssen neben der Verwendung von SW-Schnittstellen auch hard- und firmwareorientierte Forderungen erfüllt werden, wie:

- Kommunikation über Fabriknetz (MAP /51/)
- Systeminitiierte Übergabe von Steuerprogrammen an den Montagerechner
- Verteilte Archivierung von Programmen in Fertigungsdatenbanken
- Modellabgleich anhand von Meßdaten auf dem Fertigungsbereich
- Rückgabe von Programmen aus dem Fertigungsbereich

4.2 Grundfunktionen

Mit der Beschreibung der Schnittstellen wird implizit die Gesamtaufgabe eines zu entwickelnden Systems festgelegt. Zur genauen Spezifikation ist es notwendig, die Gesamtaufgabe in Teilfunktionen zu gliedern und detailliert zu beschreiben. Für ein Werkzeug zur Montagefeinplanung lassen sich drei Funktionsblöcke definieren, die im späteren Konzept als Systemmoduln betrachtet werden können. Bild 32 stellt diese drei Blöcke und deren planungsrelevante Ausgabeinformation dar. Alle drei Module sollen auch unabhängig voneinander arbeiten können und den Planer bei seinen Einzelaufgaben unterstützen.

Aus diesem Grund lassen sich dem Aufgabenblock 'Modellieren', die Tätigkeiten:

- Zellenentwurf,
- Vorrichtung, Werkzeug, Konstruktion,
- Stücklistengenerierung,
- Zeichnungserstellung.

zuordnen. Wobei der Stücklistengenerierung eine erhebliche Bedeutung zukommt, da diese Grundlage für die Ermittlung von Systemkosten und die Bestellung von Zellenkomponenten ist.

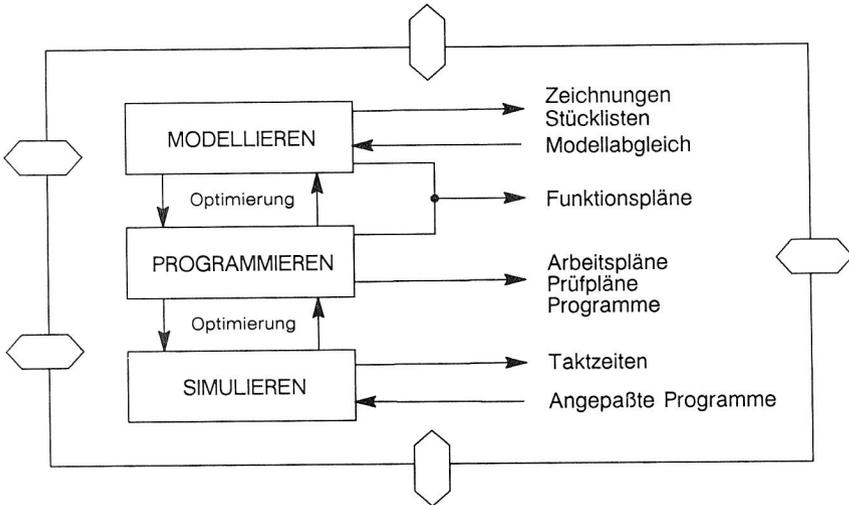


Bild 32: Funktionalität eines Werkzeuges zur Montagefeinplanung

Der Funktionsblock 'Programmieren' deckt sich mit den Aufgaben zur Erstellung von Arbeitsunterlagen und Steuerprogrammen. Das Testen des Zellenkonzepts und zugehöriger Programme läßt sich dem Block 'Simulieren' zuordnen. Hierbei soll das System eine virtuelle Testumgebung zur Verfügung stellen, damit nicht bereits beim Testen von Programmen Kosten für den Zellenaufbau und den Kauf von Komponenten entstehen. Entsprechend der funktionalen Aufteilung in Bild 32 lassen sich die folgenden Kapitel gliedern.

4.2.1 Entwerfen, Konstruieren und Ausarbeiten

Der Entwurf einer Montagezelle ergibt sich aus der Wahl von Montagekomponenten, deren Positionierung auf der vorgegebenen Produktionsfläche und der Konstruktion von Werkzeugen und Vorrich-

tungen. Dokumentiert wird der Zellenentwurf anhand von bemaßten Layoutzeichnungen, Stücklisten und evtl. in Form eines maßstäblichen Modells.

Bei der klassischen Vorgehensweise wird dabei auf bewährte Techniken, wie Kataloge, Zeichenbrett und Modellbaukästen zurückgegriffen. Für eine rechnergestützte Vorgehensweise lassen sich folgende Anforderungen formulieren:

Entwerfen:

- Eröffnen eines Projektes
- Erzeugen einer Zelle, Vergabe eines Zellennamen
- Laden von Komponenten aus Bibliotheken
- Positionieren von Komponenten
- Verändern des Layouts, Löschen von Komponenten

Konstruieren

- rechnergestützter Entwurf von Werkzeugen
- rechnergestützter Entwurf von Greifern
- rechnergestützter Entwurf von Vorrichtungen

Beim Konstruieren müssen alle geometrischen und technologischen Informationen, die bei Programmierung und Simulation notwendig sind, eingegeben werden, sofern diese nicht aus vorhandenen Produktinformationen abgeleitet werden können.

Ausarbeiten

- Zeichnungen erstellen
- Stücklisten schreiben
- Erzeugen von Präsentationsgrafiken

Unter Ausarbeiten wird im wesentlichen das Erstellen von Zeichnungen und Stücklisten verstanden. Zeichnungen werden in zweidimensionalen Ansichten nach DIN 6774 erstellt und bemaßt. Für den Fertigungsmittelbau müssen Fertigungszeichnungen von Werkzeugen, Greifern und Vorrichtungen, sowie bemaßte Layoutdarstellungen für die Aufbauteams der Montagezellen erzeugt werden.

Besonders wichtig ist die Qualität von Zeichnung bei Fremdvergabe von Anlagen und Montagezellen an Zulieferfirmen. Stücklisten von Montagezellen dienen zum einen als Grundlage für die Bestellung von Systemkomponenten und für die Errechnung der Systemkosten. Stücklisten können von manchen CAD-Systemen automatisch erstellt werden. Da allerdings bei der rechnergestützten Auslegung einer Montagezelle nicht bis ins Detail konstruiert wird -keine Schrauben an Gestellen, keine Darstellung der Antriebe von Bändern- müssen entsprechende (Klein)Teilezuschläge vorgenommen werden und auch in den Bestellstücklisten berücksichtigt werden.

Als weiteres Ergebnis der Ausarbeitung können die Hauptabmessungen der Montagezelle ausgegeben werden. Diese Information ist für die Gestaltung der Fabrikhallenlayouts von entscheidender Wichtigkeit (=Informationsrückfluß in die Montagegrobplanung).

Durch den Einsatz grafischer Rechnerwerkzeuge (CAD) lassen sich Fertigungs- bzw. Montagezellenlayouts schnell und bei Bedarf auch in mehreren Varianten konfigurieren. Die Darstellung von Zellen auf geeigneten Computerbildschirmen kann sich auch in der Montagefeinplanung als wertvolle Hilfe erweisen /52/. Farbschattierte Layoutbilder erlauben es nicht nur, schwer erkennbare Stellen zu verdeutlichen oder Kollisionen sichtbar zu machen, sondern dienen als realitätsnahe Präsentationsgrafiken auch der Entscheidungshilfe für Kunden oder Management. Als Zünglein an der Waage werden solche Darstellungen, anhand derer auch dem technischen Laien die Funktion einer Montagezelle erläutert werden kann, den Ausschlag bei der Annahme oder Ablehnung eines Layoutentwurfs geben.

4.2.2 Aufgabenbeschreibung, Programmierung

Neben der Layouterstellung muß die Möglichkeit bestehen, Montageaufgaben so zu beschreiben, daß der Montagevorgang mit Hilfe des Rechners verarbeitet und anhand des internen Modells von Produkt und Zelle simuliert werden kann.

Die Aufgabenbeschreibung muß dem Sprachgebrauch und der Terminologie von Montagefachkräften angepaßt sein. Es eignen sich Formen der Aufgabenbeschreibung nach SvZ, wie sie bei der Vorgabezeitermittlung und Analyse von manuellen Arbeitssystemen bereits eingesetzt werden. Für die Aufgabenbeschreibung bei automatisierten Anlagen müssen rechnerintern die Einflußgrößen anders behandelt werden als dies beispielsweise bei den bekannten Verfahren wie MTM oder WF durchgeführt wird. Die Ausführung von Bewegungen und die wirkenden Einflüsse werden großteils numerisch berechnet oder aus Zeittabellen gelesen. Die Verwendung von Aufgabenbeschreibungen gemäß SvZ hat den Vorteil, daß sie zum einen dem Arbeitsvorbereiter bekannt und vertraut sind und zum anderen aufgrund der starken Verfeinerung des Montagevorganges eine realistische Möglichkeit bieten, Montageaufgaben rechnergestützt zu lösen.

Ein weiterer entscheidender Vorteil besteht darin, daß die Aufgabenbeschreibung, die anschließende Simulation, die Arbeitsplan- und Programmgenerierung sowohl für manuelle als auch für automatisierte Montagezellen eingesetzt werden kann. Ansätze zu dieser Vorgehensweise finden sich bei /53/, /54/.

Damit die Aufgabenbeschreibung am Rechner einfach zu formulieren und möglichst fehlerfrei zu erstellen ist, werden drei Grundforderungen an die Art der Eingabetechnik gestellt:

1. mehrfache Eingabemöglichkeit - textuell, symbolisch, grafisch oder kombiniert.
2. Zuordnen der Tätigkeit durch Selektion der auszuführenden Komponente auf dem Grafikschild.
3. Sofortige Kontrolle der Anweisung mittels grafischer Simulation des Aufgabensegments.

Die Aufgabenbeschreibung muß neben Aufgabensequenzen auch die Formulierung paralleler Abläufe und logischer Verknüpfungen zulassen. Zudem muß mit Hilfe eines Editors, der ebenfalls textuell, symbolische oder grafische Werkzeuge besitzt, das Hinzufügen, Ändern oder Löschen von Sequenzen oder Kontrollstrukturen

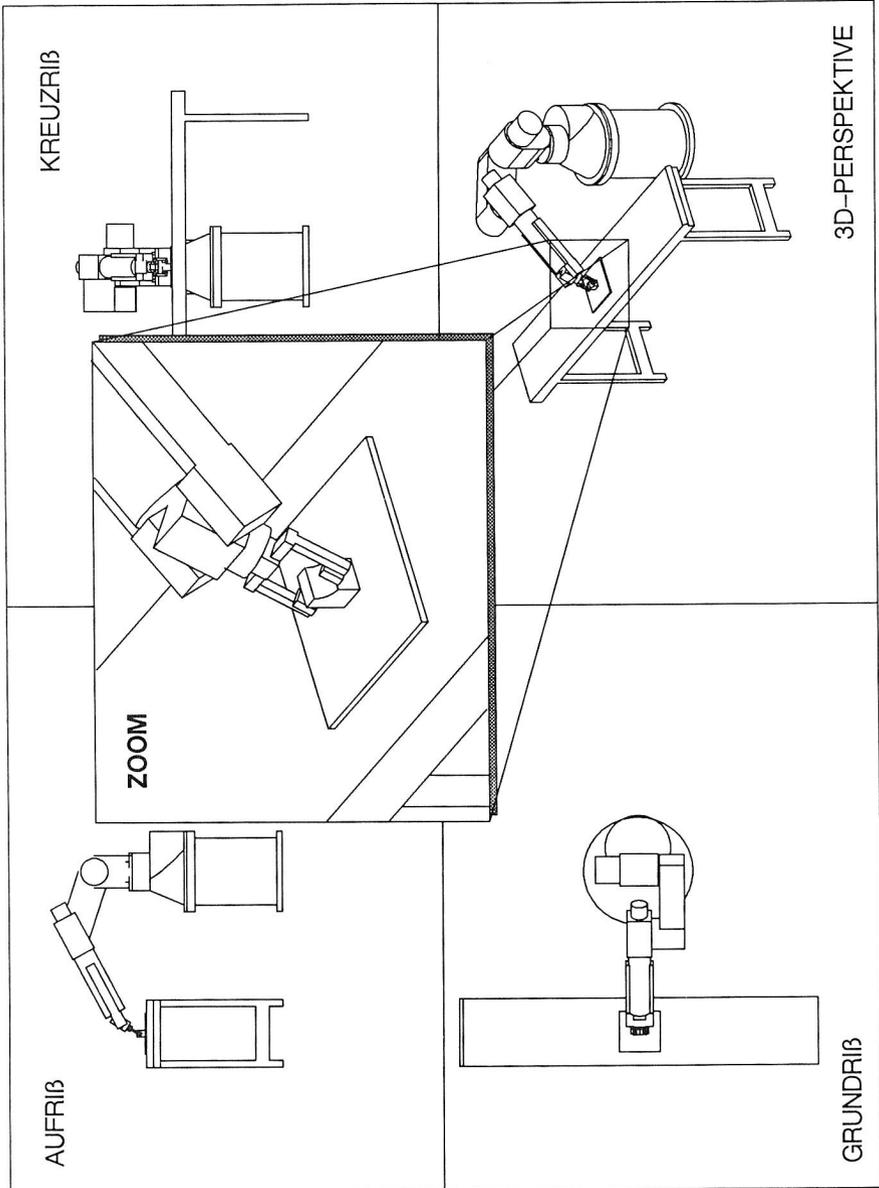


Bild 33: Grafische Darstellungsmöglichkeiten während der Aufgabensimulation

ermöglicht werden. Aus der Aufgabenbeschreibung müssen Steuerungsprogramme für die Montageteilsysteme generiert werden. Geeigneterweise sollen sich diese Steuerungsprogramme an verfügbaren Standards orientieren (siehe Kapitel 4.1.3). Es müssen aber auch Übersetzer, Prozessoren für herstellungsspezifische Programmiersprachen in Planungssystem eingebaut werden können (auswechselbare Prozessoren und Postprozessoren).

4.2.3 Test und Taktzeitermittlung

Zum Testen von Layoutentwürfen muß eine virtuelle Testumgebung geschaffen werden, in denen die formulierten Aufgaben simuliert werden können. Dazu sind folgende Funktionen zu realisieren:

- graphische Aufgabensimulation (3D)
- TRACE - Funktion
- Numerische und graphische Ausgabe
- Simulationsprotokolle und deren Auswertung
- Zeitmessung, Stoppuhrfunktion
- automatische Kollisionskontrolle
- interaktive Bildschirmmanipulation
- Zeitlupe-, Zeitrafferfunktion
- Meßfunktionen
- beliebiges Unterbrechen der Simulation.

Die Simulation soll aufsetzend auf vorhandenen 3D-Zellenmodellen, sowohl höhere aufgabenorientierte Programme verarbeiten können, als auch konventionelle explizite Steuerprogramme. Die grafische Darstellung muß parallel zur Simulationsrechnung, quasi in Echtzeit erfolgen, um eine interaktive Beurteilung des Programms oder sogar Änderungen im Programm während der Laufzeit zu ermöglichen. Die Zellen sollen während der Bewegungssimulation in drei Modi darstellbar sein:

- farbiges Drahtmodell
- Körpermodell mit Ausblendung verdeckter Kanten
- farbschattiertes Körpermodell.

Während der Simulation muß es möglich sein, den Betrachterstandpunkt zu verändern ohne den Programmablauf zu unterbrechen. Dazu sind folgende Bildschirmmanipulationsfunktionen notwendig:

- Simulation in mehreren beliebigen Ansichten
- Perspektivische oder isometrische Darstellung
- Drehen und Verschieben des Betrachterstandpunktes
- Zoomfunktion
- Ausblenden von Objekten
- Clipping
- Zuordnen von Farben (z.B. aktive Elemente rot)

Wesentlich ist die Auswertung der Simulationsergebnisse. Für die Montageplanung ist es besonders wichtig festzustellen, ob die geforderten Taktzeiten eingehalten werden können. Dazu sollen, ähnlich wie bei der REFA-Zeitaufnahme, Stoppuhrfunktionen vorhanden sein. Über die mitlaufenden Protokolle können Weg/Zeitdiagramme erstellt und Einzelzeiten der Aufgabenbeschreibung zugeordnet werden. Damit ist neben der Taktzeitermittlung, einer automatischen Arbeitsplanung auch die rechnergestützte Logikprogrammierung möglich.

Durch die Verfolgung der Bewegungsbahnen und einem zuschaltbaren Kollisionsdetektor sollen Programmfehler, die zur Zerstörung von Zellenkomponenten oder Bauteilen führen können, aufgespürt werden. Dabei werden Kollisionen während des Simulationslaufs akustisch angezeigt und optisch sowie in Form eines Kollisionsprotokolls angezeigt.

4.3 Modelle und Verwaltung

Viele Rechneranwendungen verwenden heute Dateisysteme zur Speicherung von (Modell-) Informationen. Dabei erzeugt jeder Benutzer eine Reihe von Files, deren Verwaltung und Organisation ihm selbst überlassen bleibt (Bild 6). Wichtige Informationen können von anderen Benutzern nicht wiederverwendet werden, da die eige-

nen Anwenderprogramme nicht auf existierende Dateien zugreifen können oder bestimmte Daten wegen einer fehlenden Ordnung gar nicht erst gefunden werden. Dieses Problem beruht auf der Verwendung unterschiedlicher Datenmodelle (Schemata), die für jedes Anwenderprogramm individuell in Form von Dateien gespeichert sind. Mit Hilfe von Datenbanken wird schon lange versucht, eine Trennung von Daten und Anwenderprogrammen zu verwirklichen (siehe Bild 34).

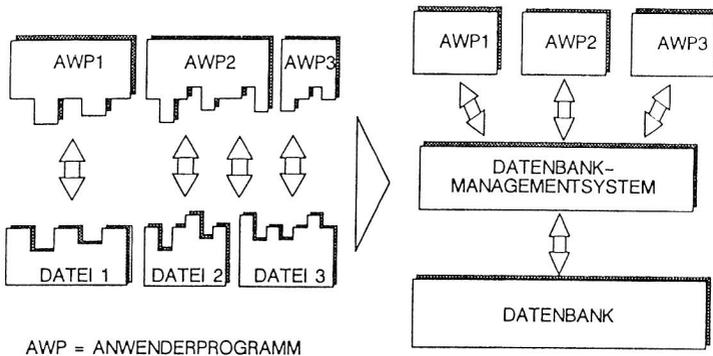


Bild 34: Vereinigung der Daten in einer Datenbank /55/

Für die Montagefeinplanung ist es wichtig, daß Daten, die aus der Konstruktion, der Montagegrobplanung, der Betriebsebene oder vom Benutzer kommen, in Form eines konsistenten Datenmodells verarbeitet und verwaltet werden können. An die Datenorganisation werden daher folgende Forderungen gestellt.

1. Zentralisierung der Daten:

Es muß sichergestellt werden, daß über das Datenverwaltungssystem auf alle Informationen, die für ein bestimmtes Objekt (oder Planungsprojekt) wichtig sind, zugegriffen werden kann. Redundanz der Speicherung von Daten ist zu vermeiden.

2. Wahrung der Datenunabhängigkeit:

Es darf nicht darauf ankommen, wer, wann und durch welches System planungsrelevante Informationen

(z.B. Produktgeometrie) erzeugt und gespeichert werden. Entscheidend für die Weiterverarbeitung soll alleine die Sicht des Anwenders, in diesem Fall des Montagefeinplaners sein.

3. Datenintegrität:

Aufgrund mehrfachen Zugriffs auf gleiche Daten muß dafür gesorgt werden, daß Widerspruchsfreiheit und damit Datenkonsistenz gewährleistet sind. Im gleichen Zusammenhang müssen Daten vor Verlust, beabsichtigter oder unbeabsichtigter Zerstörung sowie vor unerlaubten Zugriffen geschützt werden.

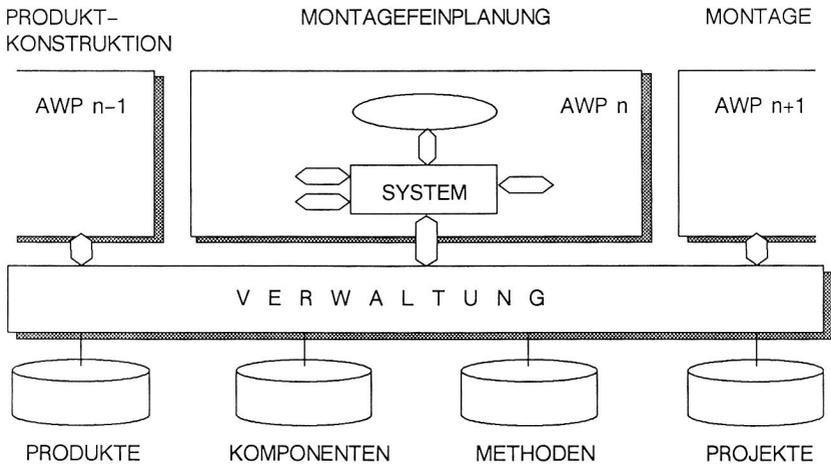
Gespiegelt am Planungsablauf bedeutet dies konkret, daß Produktinformationen, wie die 3D-Geometrie, Werkstoffdaten, Toleranzen, Varianten aus der Konstruktion übernommen werden und, ohne die Konsistenz zu gefährden, um montagerelevante Eigenschaften erweitert werden können -z.B. zum Greifen, Kennzeichnen von Klebbehältern.

Alle zu einer Montageanlage gehörenden Zellenentwürfe sind unter einem Projekttitel zusammengefaßt. Die Verwaltung des Entwicklungsstandes, der Entwurfsversion oder -variante bleibt dem Verwaltungssystem überlassen. Durch die Projektverwaltung muß auch die Zuordnung von Zellen zu Planungsergebnissen, wie Steuerprogrammen, Simulationsprotokollen getroffen werden.

Kataloginformationen sind aus einer Bibliothek abrufbar. Hierbei sollen standardisierte und herstellereigenspezifische Montagegeräte und -einrichtungen durch einfache Suchmechanismen gefunden und für den Zellenentwurf verwendet werden können. Eine solche Bibliothek könnte z.B. aus Sammlungen von Handhabungssystemen, Transporteinrichtung, Gestellen usw. bestehen.

Auch die Verwaltung von Methoden und Berechnungsprogrammen für die Simulation von Montageabläufen und die Interpretation von aufgabenorientierten Anweisungen sollen durch das Datenbanksystem -in diesem Fall Methodenbanksystem- ermöglicht werden. Methoden

können beispielsweise aus Planungsregeln, wie sie bei Greifanweisung oder Bearbeitungsanweisungen -Kleben, Schrauben, Entgraten, Schweißen, Fügen u.a.- benötigt werden. Unterschiedliche Berechnungsprogramme werden benötigt für die Bahnberechnungen, Schwingungsrechnung, Festigkeitsrechnung. Durch die Abtrennung der Methoden vom eigentlichen Anwenderprogramm (hier System zur Montagefeinplanung) bleibt das Gesamtkonzept modular erweiterbar. Auf diese Weise können nachträglich neue Sprachinterpretierer, Bahnplanungsregeln, Steuerungsstrategien oder Postprozessoren eingepaßt werden. Entscheidend hierbei ist, daß dazu Methoden oder Programme zur Laufzeit des Planungswerkzeuges auswertbar sein müssen.



AWP = ANWENDERPROGRAMM

Bild 35: Aufgaben der Datenorganisation

4.4 Einsatz von Basiswerkzeugen

Rechnergestützte Feinplanung von Montagezellen ist nur eine der zahlreichen Aktivitäten zum Thema CIM. Um nicht für jeden CIM-Baustein erneut sämtliche Funktionen zu implementieren, ist es

sinnvoll auf bewährte eingeführte SW-Werkzeuge und Techniken zurückzugreifen. In diesem Kapitel werden daher Softwarelösungen vorgestellt, die als Basis für ein System zur Montagefeinplanung dienen können und damit gleichzeitig die Verbindung zu anderen Systemen im Produktionsbetrieb herstellen (siehe Bild 36).

Mit seinen zahlreichen Funktionen zum Modellieren und Visualisieren von geometrischen Modellen eignet sich ein offenes CAD-System als Grundlage für planerische Aufgaben in der Montage. Die wesentlichen Forderungen an ein Datenverwaltungssystem sind im vorangegangenen Kapitel bereits definiert worden. Diesen Forderungen stehen heute einige bewährte Datenbanklösungen gegenüber, die jedoch nur Teile der geforderten Eigenschaften erfüllen können. Diese klassischen Datenmodelle sind nach Eigner /56/:

- hierarchische,
- netzwerkartige,
- relationale.

In Datenbanken können einzelne Daten aber auch ganze Datensätze unter einem internen Schema gespeichert und verwaltet werden. Soll die Verwaltung von Programmen und Methoden mit Hilfe einer Datenbank durchgeführt werden, so können beispielsweise Programmnamen, Zeiger oder Trigger auf bestimmte Objekte (=Programme, Methoden) in der Datenbank hinterlegt und bei Bedarf freigegeben bzw. ausgelöst werden.

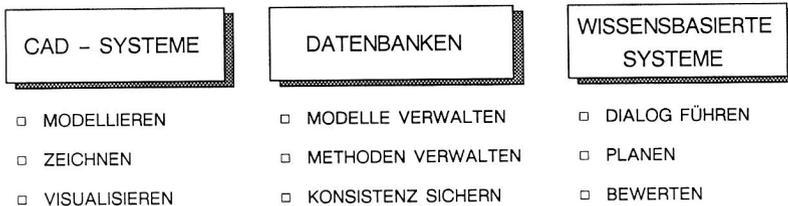


Bild 36: Basiswerkzeuge für ein System zur rechnerunterstützten Montagefeinplanung

Klassische Datenbanken machen einen konzeptionellen Entwurf des Datenbankmodells erforderlich. Dieser ist stark orientiert an rechnerinternen Strukturen und Architekturen. Ziel hierbei ist in erster Linie die effiziente Speicherung und Verwaltung von Daten und nicht die anwendungsorientiert benutzernahe Bedienung. Die Lösung wird derzeit in Form "semantischer Datenmodelle" /55/ gesucht, doch sind auf diesem Gebiet noch keine praxisbewährten Systeme eingeführt worden.

Wesentliche Impulse für rechnergestütztes Planen gehen von den Aktivitäten der "künstlichen Intelligenz" aus. Der entscheidende Unterschied zu klassischen Formen der Informationsverarbeitung besteht darin, daß kein Unterschied gemacht wird, zwischen Daten und Programmen. Der unmittelbare Vorteil davon ist, daß Programme ebenso wie Daten zur Laufzeit eines übergeordneten Programmes erzeugt, verändert und zudem auch gestartet werden können. Aus diesem Grund wäre es denkbar, daß sich ähnlich wie im menschlichen Gehirn, planerische Entscheidungen durch interne (rekursive oder iterative) Simulationen herbeiführen lassen. Weitere wichtige Eigenschaften, die von Systemen der künstlichen Intelligenz erwartet werden, sind:

Benutzerorientiertes Dialogverhalten:

- Verarbeitung aufgabenorientierter Anweisungen (Sprache)
- Selbsterklärungsfähigkeit

Wissensbasiertes Planen:

- Greifplanung
- Bearbeitungsplanung
- Bahnplanung

Bewerten von Lösungen:

- Layoutbewertung (Kosten, Anlaufzeit)
- Programmbewertung (Energie, Ausführungszeit)

Als praktikable Werkzeuge haben sich Programmiersprachen wie LISP

/57/ oder Prolog, sowie Expertensystemschalen z.b. KEE oder WWS /58/ erwiesen. An Werkzeugen zur wissensbasierten Anpassung von Benutzerschnittstellen, sog. Metabenutzerschnittstellen, wird geforscht.

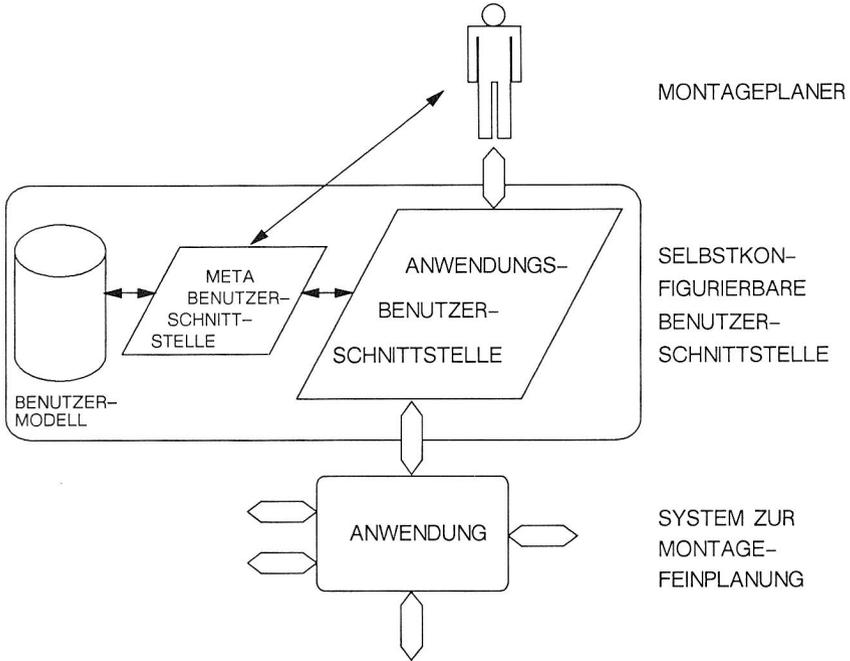


Bild 37: Wissensbasierte Anpassung des Anwendersystems an den Benutzer nach /59/

5. CAD - integriertes Konzept für die Montagefeinplanung

Anhand der Anforderungen an ein aufgabenorientiertes Programmsystem -siehe Kapitel 4- läßt sich ein Konzept entwickeln, das auf ein offenes 3D-CAD System aufbaut. Dadurch wird einerseits eine einfache datentechnische Verknüpfung zwischen Konstruktion und Arbeitsvorbereitung ermöglicht und andererseits wesentliche Funktionen durch gleiche Bedienung und Datenspeicherung ausgeführt (siehe Bild 38). Die Vorteile davon sind:

1. keine redundante Dateneingabe, da Produktinformationen übernommen werden können
2. Kompatibilität der geometrischen Informationen von Produktkonstruktion und Werkzeug-/Vorrichtungskonstruktion
3. Ähnlichkeit bei der Systembedienung
4. Möglichkeit zu einer integrierten Datenhaltung
5. keine künstlichen Schnittstellen und damit kurze Informationswege und kein Informationsverlust.

Bei der Entwicklung eines aufgabenorientierten Programmsystems kann daher u.U. die Zeit zur Realisierung von geometrischen Modellierungsfunktionen oder grafischen Ausgabemöglichkeiten eingespart werden.

AUFGABEN DES CAD-SYSTEMS IM KONZEPT

- GEOMETRISCHES MODELLIEREN
 - MESSEN
 - POSITIONIEREN
 - VISUALISIEREN
 - BEMÄßEN
 - ZEICHNUNGSERSTELLUNG
 - BESCHREIBUNG DER TOPOLOGIE
-

Bild 38: Aufgaben des CAD Systems im Konzept

Wichtig ist, daß es sich bei dem auszuwählenden CAD-System um einen Modul handelt, auf dessen Funktionen und Daten über klare Schnittstellen zugegriffen werden kann (z.B. C - Schnittstelle zum CAD Kern).

Die Benutzeroberfläche des CAD - Systems und des Planungswerkzeuges sollen in Aufbau und Struktur ähnlich sein, sodaß auch aus Sicht des Anwenders kein Bruch beim Wechsel in das ein oder andere Modul entsteht.

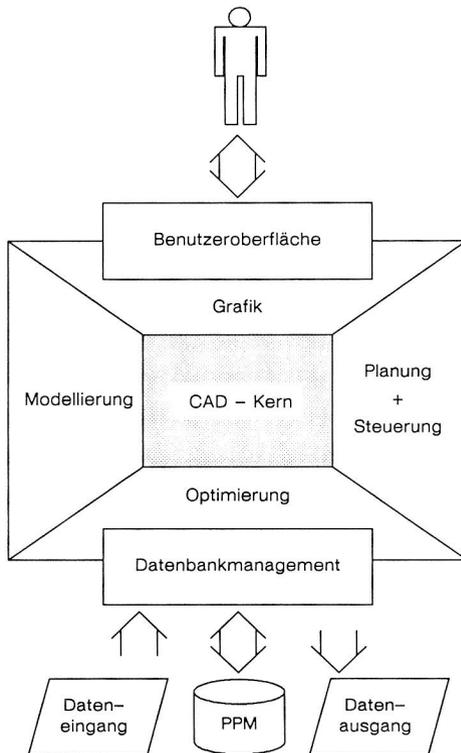


Bild 39: CAD integriertes Konzept

Ein weiterer wichtiger Baustein ist ein Datenverwaltungssystem, das den unterschiedlichen Teilsystemen über externe Schemata

Zugriffe auf die gesammelten Modelldaten PPM erlaubt. Es bietet sich zu diesem Zweck die Einbettung eines Datenbanksystems an, das neben Datenkonsistenz auch für Datensicherheit sorgt. Der Entwurf der Datenbank und der internen Schemata ist nicht Teil der hier vorliegenden Arbeit (siehe dazu /55/), dagegen werden durch die Konzeption des Systems externe Schemata beschrieben und damit die Anforderungen an den Datenbankentwurf spezifiziert.

Bild 39 zeigt in grober Form Aufbau und Zusammenwirken der Teilsysteme in einem CAD integrierten aufgabenorientierten Programmiersystem. Diese grobe Gliederung erlaubt eine Zerlegung des Gesamtsystems in kleinere Bausteine, die nach der Festlegung von Schnittstellen getrennt entwickelt werden können. Das Kapitel 5 ist dazu entsprechend gegliedert:

- | | |
|-------------------------------------|------------|
| 1. Benutzeroberfläche und Grafik | (Kap. 5.1) |
| 2. Modellierung und Datenverwaltung | (Kap. 5.2) |
| 3. Planung und Steuerung | (Kap. 5.3) |
| 4. Optimierung | (Kap. 5.4) |

5.1 Gestaltung der Benutzeroberfläche

Mit der Entwicklung der Benutzeroberfläche steht oder fällt die Einführung eines Systems in die praktische Anwendung. Dies gilt vor allem für Einsatzgebiete, in denen es nicht darum geht, große Datenmengen einzugeben, sondern kreative Arbeit mit Hilfe des Rechners sinnvoll zu unterstützen. Montageplanung erfordert geistig kreative Arbeit von Ingenieuren oder Technikern; Systeme, die hier unterstützen, müssen sich in hohem Maße an den speziellen Bedürfnissen der Planer orientieren. Dies ist der Grund, warum in dieser Arbeit großer Wert auf eine eingehende Aufgaben- und Benutzeranalyse gelegt wird.

Neben den direkten Anforderungen aus der Planungstätigkeit sind zudem die allgemeinen arbeitswissenschaftlichen Erkenntnisse bei der Gestaltung des 'user - interface' zu berücksichtigen. Das

Anspruchsvolle daran ist die Verknüpfung der in zahlreichen Veröffentlichungen /48/, /60/ und Normen /47/ dargestellten kommunikationsergonomischen Grundlagen mit der eigentlichen Anwendung -in diesem Fall der Montagefeinplanung.

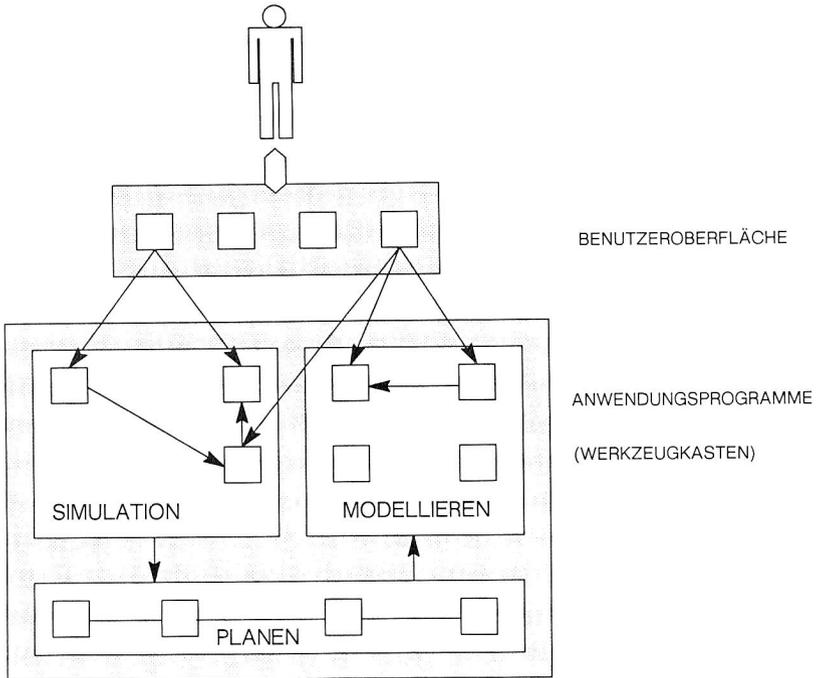


Bild 40: Trennung von Benutzerschnittstelle und Anwenderprogrammen.

Die Gestaltung der Benutzeroberfläche kann unabhängig von der Implementierung der Einzelfunktionen gestaltet werden. Voraussetzung dafür ist, daß eine Trennung von Benutzeroberfläche und Anwenderprogrammen durch geeignete Schnittstellen vollzogen wird. Programme und Datensätze zur Ausführung einzelner Aktionen müssen dazu quasi als Werkzeugkasten benutzerunabhängig zur Verfügung stehen. Technisch kann dies in Form einer konsequenten objektorientierten Programmierung verwirklicht werden.

Die bisher übliche Vorgehensweise zur Entwicklung von Computerwerkzeugen geht aus von der Hardware, entwirft passende Software dafür und fügt zuletzt noch die Benutzeroberfläche hinzu. Diese Art der Entwicklung sozusagen von "innen nach außen" (Fischer und Herzog in /48/) hat dazu geführt, daß Software vom Ansatz her gar nicht benutzergerecht gestaltet werden kann. Es ist deshalb grundsätzlich zu empfehlen, mit dem Entwurf der Benutzerumgebung zu beginnen und danach SW und HW zu konzipieren -Entwicklung von "außen nach innen".

Im folgenden geht es darum, aus der Vielzahl möglicher Dialogformen die am besten geeignete auszuwählen und zu gestalten, sowie eine geeignete Sprache zur Beschreibung der Montageaufgabe zu entwickeln, welche einerseits leicht verständlich, einfach zu editieren und andererseits für die Simulation vom Rechner schnell und fehlerfrei zu verarbeiten ist. Zuletzt wird gezeigt, wie mit Hilfe von Grafik und Farben Ergebnisse der Aufgabensimulation oder der expliziten Zellenprogrammierung in ansprechender und durchschaubarer Form ausgegeben werden können.

5.1.1 Dialogformen und Benutzerführung

Der Dialog ist die höchste Ebene der kommunikativen Schnittstelle zwischen Mensch und Maschine. Je komfortabler dieser Dialog für den Anwender gestaltet wird, desto stärker werden Speicherplatz, Rechengeschwindigkeit und damit das Antwortzeitverhalten des Anwendungssystems beeinträchtigt. Mit steigender Rechnerleistung und sinkenden Preisen für Hardware verringert sich jedoch dieser Nachteil.

In der Informationsverarbeitung kann man mehrere Formen des Rechnerdialogs unterscheiden:

- Transaktionscodes
- Kommandosprachen
- Menü- bzw. Auswahlssysteme

- Maskensysteme
- direkte Manipulation
- natürliche Sprache.

Dazu kommen noch einige Sonderformen - wie Handschriftteingabe, Sprach- oder Bildverarbeitung. Die genannten Dialogformen können auch hybrid, d.h. kombiniert eingesetzt werden. Die Wahl der geeigneten Dialogform orientiert sich entscheidend an der zu erfüllenden Arbeitsaufgabe. Durch Antwort auf die folgenden 4 Fragen kann eine Auswahl getroffen werden:

1. Verläuft die zu erfüllende Aufgabe stets nach dem gleichen starren Muster? (Funktionsumfang)
2. Wird von der Dialogform universelle Einsetzbarkeit verlangt?
3. Verlangt die Aufgabe gelegentliches oder häufiges Arbeiten mit dem System? (Häufigkeit)
4. Müssen gleichzeitig mehrere Funktionen zur Verfügung stehen? (Parallelität)

Ein Lagerverwaltungssystem beispielsweise, das stets gleiche Aufgaben zu erfüllen hat, ausschließlich für die Einlagerung und Entnahme von Gütern eingesetzt wird, aber ständig im Einsatz ist, läßt sich geeigneter Weise durch ein Maskensystem bedienen, bei dem Lagergutkennung, Art, Menge, Lagerort und Lagerzeit in eine Bildschirmmaske eingetragen werden können.

Die Tätigkeit des Montageplaners ist dagegen geprägt von vielen unterschiedlichen Verrichtungen, bei denen der Rechner zahlreiche Funktionen, Werkzeuge parallel zur Verfügung stellen muß. Die Arbeitsaufgabe ist offen und fordert die Kreativität des Planers. Der Dialog mit dem Rechner kann daher nur benutzerorientiert (im Gegensatz zu rechnerorientiert) erfolgen und muß über ein breites Spektrum von Funktionsaufrufen verfügen. Die Dialogform 'direkte Manipulation' eignet sich am besten für den Aufgabenbereich der Montageplanung.

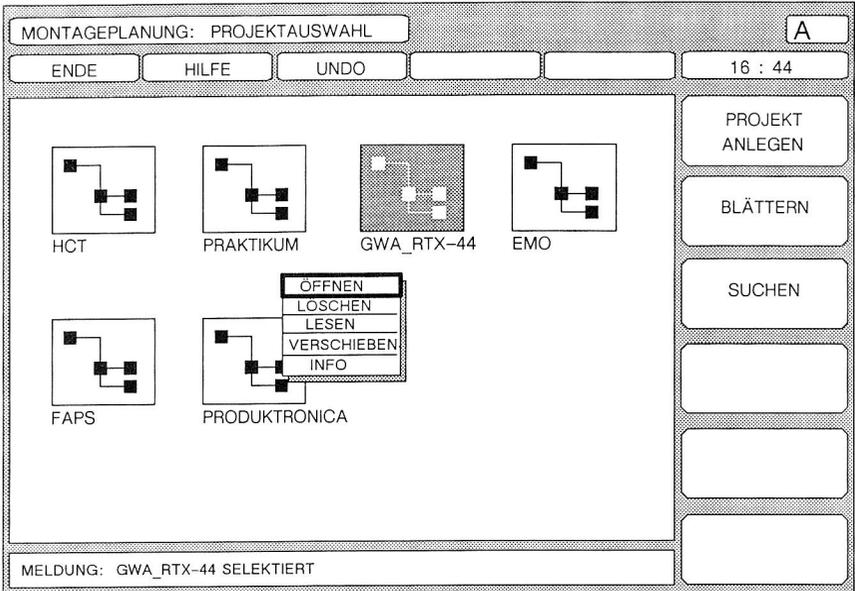


Bild 41: Dialogform 'direkte Manipulation' am Beispiel des Eingangsmenus der Montageplanung

Bild 41 stellt das oberste Benutzerfenster der Montageplanung dar. Verschiedene Techniken der 'direkten Manipulation' sind erkennbar. Symbolische Darstellung von Objekten (Ikonen) -hier Projekte zur Planung von Montagesystemen- Pop Ups, Bildschirmmenü, Statusinformationen und Meldungsfenster sind typische Merkmale einer Oberfläche, die nach diesem Prinzip arbeitet. Üblicherweise werden solche Systeme, die der neuen Generation von Bildschirmarbeitsplätzen zuzuordnen sind, ausgerüstet mit einem hochauflösenden Rasterbildschirm (u.U. auch farbig mit mindestens 1024x1024 Pixel und mindestens 8 Farbebenen) sowie dem auf das Terminal einwirkenden Zeigegerät, das hier eine erheblich wichtigere Rolle für die Systembedienung spielt als etwa die Alphatastatur. Neben Bildschirm, Tastatur, Potentiometerleisten oder Funktionstastaturen dienen folgende Zeigeeinstrumente als wesentliche Hardwarekomponente:

- Maus
- Joystick
- Rollkugel
- Lichtgriffel
- Grafiktablett
- Berührempfindliche Flächen.

Bild 42 zeigt beispielhaft den Hardwareaufbau für eine praktische Systembedienung.

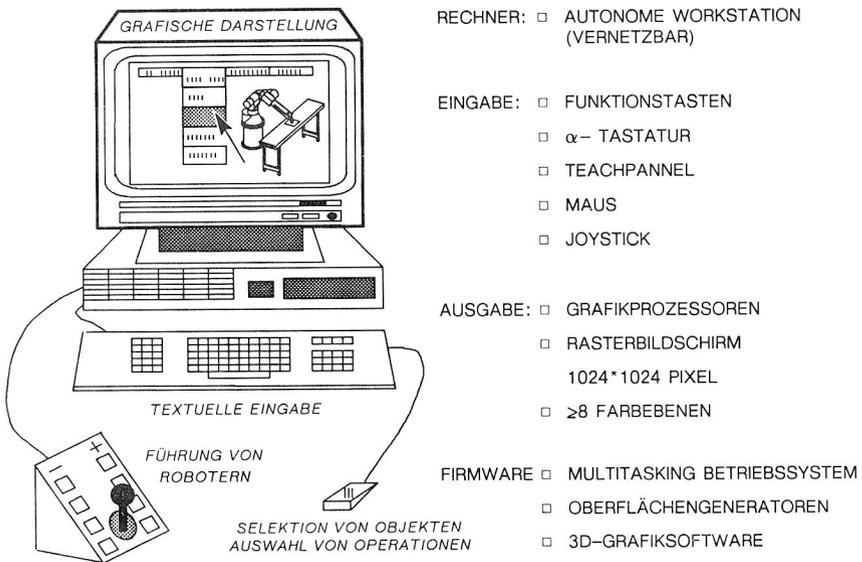


Bild 42: Ausstattung für die rechnergestützte Planung mittels 'direkter Manipulation'

Die Gestaltung und Aufgabenangemessenheit einer Benutzerumgebung zeigt sich, wenn der Benutzer das vorgeschlagene Modell seiner neuen Arbeitsumgebung akzeptiert. Die 'direkte Manipulation' ermöglicht eine schnelle Gewöhnung, da Systemobjekte und Funktionen sichtbar bedient werden können (WYSIWYG: What You See Is What You Get) /61/. Jedes Element der Oberfläche kann nach Selektion

Auskunft über die Möglichkeiten wirksamer Anwendungen auf sich selbst oder sonstige Informationen über sich liefern. Operationen können unmittelbar ausgelöst und auch jederzeit wieder rückgängig gemacht werden (undo, redo).

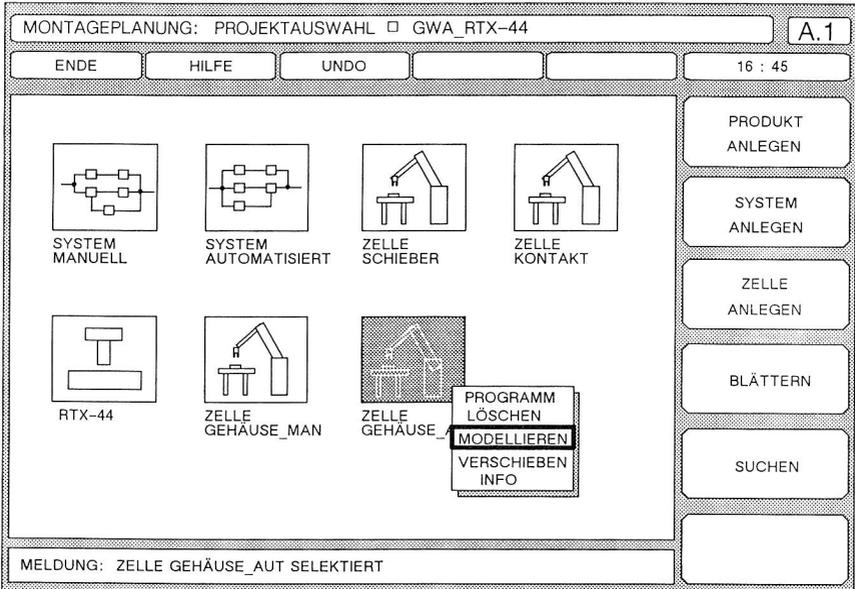


Bild 43: Objekte in einem Montageplanungsprojekt

Ausgehend von 'Top' -dem Hauptmenü- der Montageplanung wird die gewählte Technik im folgenden speziell für die Feinplanung weiter beschrieben. Die Benutzeroberfläche wie in Bild 41 dargestellt ist weitgehend selbsterklärend. Prinzipiell sind alle Systemfenster in 4 Bereiche eingeteilt:

1. Statusfeld
2. allgemeine Systembedienungsfunktionen
3. Feld zur Objektmanipulation
4. Systemmeldungen

Durch das Statusfeld wird in knapper Form der aktuelle System- und Prozeßzustand beschrieben. Diese Beschreibung enthält den Namen des aktuellen Objektes -in Bild 43 ist dies das Projekt GWA RTX 44- einen Hinweis auf den Stamm und die Ebene in der Systemhierarchie sowie eine farbige Kennzeichnung, ob der aktuelle Prozeß arbeitet (rot) oder eingabebereit ist (grün).

Allgemeine Systembedienfunktionen sind in jeder Systemebene vorhanden und unterscheiden sich kaum voneinander. Ihre Bedienung und Bedeutung ist grundsätzlich gleich, kann sich aber in Ihrer Auswirkung nach dem gegenwärtigen Systemzustand richten.

Beispiel:

ENDE

Dieses Menüfeld bewirkt grundsätzlich den Abbruch des aktuellen Programms bzw. das Schließen des gegenwärtigen Objektes.

HILFE

Dieses Menüfeld orientiert sich dagegen am Status bzw. am gegenwärtigen Zustand des Prozesses und gibt dazu Auskunft. Hat es beispielsweise nach der letzten Aktion eine Fehlermeldung gegeben, wird dies berücksichtigt. Die Hilfestellung gibt dann Hinweise, wodurch der Fehler entstanden ist und wie er vermieden werden kann.

Das größte Feld (Arbeitsfeld) dient der eigentlichen Objektmanipulation. Dieser Fensterausschnitt kann -je nach Systemzustand- mit Ikonen belegt sein oder wie beim Simulieren von Bewegungsabläufen mit grafischen 3D-Objekten (siehe Bild 44). Durch Auswahl mit Hilfe eines Zeigergerätes -z.B. durch einen mausgesteuerten Cursor- werden Objekte entsprechend ihren Eigenschaften manipuliert.

In Bild 43 sind die verschiedenen Objekte dargestellt, die eine Bedeutung für ein Montageplanungsprojekt besitzen. Man unterscheidet:


```

      OFFNEN PROJEKT  GWA_RTX44
## PROJEKT GWA_RTX44 GEOFFNET                10.41.32
      SELEKTIEREN ZELLE "GEHAUSE MANUELL"
## ZELLE "GEHAUSE MANUELL" SELEKTIERT        10.41.34
      PROGRAMMIEREN SIMULIEREN
## MODELL DER ZELLE "GEHAUSE MANUELL" WIRD GELADEN 10.41.35
## CARo ZUR PROGRAMMIERUNG UND SIMULATION WIRD
## GESTARTET                                  10.41.37
## CARo BEREIT                                10.41.40
      . . .

      FAHRE GEHAUSEHAELFTE R AN
## FEHLER: STELLUNG KANN LINEAR NICHT ERREICHT WERDEN
## ANDERE BEWEGUNGSSTRATEGIE WAEHLLEN        10.52.01
      FAHRE GEHAUSEHAELFTE R AN
## FEHLER: STELLUNG KANN LINEAR NICHT ERREICHT WERDEN
## ANDERE BEWEGUNGSSTRATEGIE WAEHLLEN
## WOLLEN SIE EINEN SYSTEMVORSCHLAG? JA/NEIN 10.53.05
## ANFAHRSTRATEGIE FREI WAEHLLEN
## SYSTEM WAEHLETT AB JETZT EINE ANFAHRSTRATEGIE 10.53.06
      . . .

      CARo BEENDEN
## SOLL CARo WIRKLICH BEENDET WERDEN? JA/NEIN 11.23.31
      JA, CARo BEENDEN
## CARo BEENDET                               11.23.32
      . . .

# xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
# x ENDE DER SITZUNG VOM 25.08.1989          10.30.01 UHR   x
# x                                     AM 25.08.1989    11.47.42 UHR   x
# xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx

```

Grafisch interaktives Modellieren von Montagezellen

Die wesentlichen Funktionen zur Modellierung von Montagezellen bestehen aus den Programmodulen:

- o Zellenkomponente erzeugen
- o Generieren einer Zelle
- o Ergänzen einer Zelle
- o Modifizieren einer Zelle.

Diesen Hauptfunktionen stehen alle Routinen eines 3D-volumenorientierten CAD-Modellierers zur Verfügung. Geeignet hierbei ist z.B. das CAD-Kernsystem PARASOLID das volumenorientiert arbeitet, zusätzlich die Gestaltung von Freiformflächen erlaubt und beliebigen Wechsel zwischen Draht-, Flächen- und Volumendarstellung zuläßt. PARASOLID ist ein von SIEMENS und McDONNELL DOUGLAS entwickelter Modellierer, der ab 1990 in den CAD-Systemen SIGRAPH und UNIGRAPHICS zur Verfügung steht. Durch die Verwendung eines marktgängigen CAD-Systems können zahlreiche Funktionen für die Modellierung von Montagezellen verwendet werden. Bild 44 zeigt, wie die Oberfläche eines Zellenmodellierers aussehen kann.

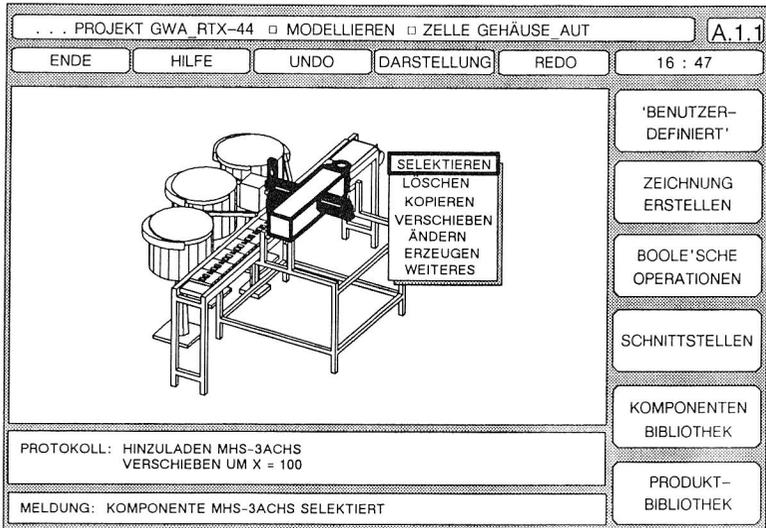


Bild 44: Oberfläche eines Zellenmodellierers

Dem Benutzer steht ein umfangreiches Kommandorepertoire zur Verfügung, das über eine einheitliche Benutzeroberfläche ansprechbar ist. Geometrische und topologische Zusammenhänge sowie technologische Eigenschaften werden in einer konsistenten Datenstruktur abgespeichert (Kapitel 5.2). Zur Zeichnungserstellung können beliebige Ansichten, Risse oder Schnitte ausgewählt und nach DIN bemaßt und beschriftet werden.

Die wesentlichen Funktionen aus dem CAD sind:

- Erzeugen von Körpern
- Verknüpfen von Körpern (Boole'sche Operationen)
- Modifizieren und Positionieren von Elementen
- Zeichnen
- Informieren (z.B. Abmessungen, Eigenschaften).

Das Besondere des Modellierers liegt im Aufbau der komplexen Zellenstruktur und in der Anbindung von technologischen Parametern und Berechnungsroutinen zu einzelnen Systemkomponenten. Bei

der Modellierung eines Roboters müssen -falls dieser nicht bereits in der Komponentenbibliothek verfügbar ist- eine Reihe von Eigenschaften und Verweisen auf Rechenvorschriften zugeordnet werden. Da der Aufbau des technologischen Modells einer Komponente teilweise schwierig ist, wird in diesem Fall ein rechnerinterner Dialog in Form eines Variantenprogrammes eingesetzt, der alle relevanten Komponentendaten vom Benutzer anfordert.

Beispiel einer Protokolldatei zur Robotermodellierung:

```

. . .
ERZEUGEN EINES ROBOTERS
## SIE ERZEUGEN EINEN ROBOTER
## WAEHLEN SIE DEN KINEMATISCHEN TYP AUS          16.35.22
      TRRA1
## SIE ERZEUGEN EINEN ROBOTER VOM TYP TRRA1
## INFORMATION: ES EXISTIEREN BEREITS 4 ROBOTER DIESES
##          TYP IN UNSERER BIBLIOTHEK          16.35.51
. . .

## BITTE GEBEN SIE DIE ANSCHLAEGE VON ACHSE 2 IN GRAD
## EIN          16.51.10
##          ACHSANSCHLAG_MIN = -162 ; ACHSANSCHLAG_MAX = +162
## ACHSE2: ACHSANSCHLAG_MIN = -162 ;
##          ACHSANSCHLAG_MAX = +162
## GEBEN SIE DAS MAXIMALE DREHMOMENT VON ACHSE 2 IN
## NM EIN          16.51.21
. . .

```

In schwierigen Phasen der Modellierung kann das System den Benutzer durch verschiedene Hilfsstufen, die bis zum komplett automatischen Ablauf eines Modellierungsvorganges gehen, zum Ziel führen. Es bietet die Möglichkeit an, durch Editieren eines vorgeschlagenen Modells das gewünschte Ergebnis zu erhalten.

Beispiel:

```

. . .
## DIES IST VORSCHLAG FUER EINEN ROBOTER VOM TYP TRRA1
## SIE KOENNEN NUN GEOMETRIE UND TECHNISCHE DATEN DES
## ROBOTERS ACHSWEISE MIT DEN UEBLICHEN MANIPULATIONS-
## FUNKTIONEN VERAENDERN          08.45.33

```

Programmieren und Simulieren von Montagezellen

In bisher bekannten Systemen (siehe Kap. 2) konnten nur einzelne Komponenten einer Zelle programmiert und simuliert werden. Durch die Modellierung von Verschaltungen der Teilsysteme zu einer

vollständigen autonom arbeitenden Montagezelle ist es möglich einzelne Komponenten zu programmieren und gleichzeitig durch Senden und Empfangen von Signalen den Gesamttablauf zu steuern. Das Ergebnis einer vollständigen Zellenprogrammierung besteht demnach aus einer Anzahl von Einzelprogrammen und einem übergeordneten Zellenprogramm, das den Gesamttablauf ereignisorientiert und zeitlich koordiniert.

Die Simulation erlaubt es, Einzel- und Zellenprogramme, Programmsequenzen und -zeilen sowie Bewegungseingaben über Funktionstasten zu verarbeiten und die entsprechenden Ergebnisse grafisch und alphanumerisch anzuzeigen. Die Programme lassen sich schrittweise im TRACE-Modus abarbeiten und gleichzeitig debuggen (Fehler beseitigen). Sequenzen können dabei auch rückwärts simuliert werden.

Es besteht die Möglichkeit die Modelle graphisch in folgenden Darstellungsarten zu simulieren:

- Drahtdarstellung
- Hidden-Line Darstellung
- Farbschattierte Flächendarstellung.

Auch hier wird auf die vom CAD zur Verfügung gestellten Routinen zurückgegriffen. Zusätzlich können die abgefahrenen Bahnen graphisch durch Ausbildung einer 'temporären graphischen Struktur' mitverfolgt werden. Diese 'temporäre Struktur' läßt sich als Linienzug oder als Schlauch mit kreisförmigem Querschnitt zur Vereinfachung von Kollisionsuntersuchungen (visuell oder automatisch) verwenden.

Zu beachten ist, daß je nach gewählter Darstellungsart, die Antwortzeiten des Systems schwanken können. Die farbschattierte Flächendarstellung benötigt beispielsweise umfangreiche Beleuchtungsmodelle zur Berechnung der am Bildschirm erscheinenden Farbpixel. Dennoch kann heute mit Hilfe von leistungsfähigen grafischen Rechnern ein ansprechendes Antwortzeitverhalten auch bei

dieser Art der Präsentation gewährleistet werden, so daß auch hier grafisch interaktiv gearbeitet werden kann.

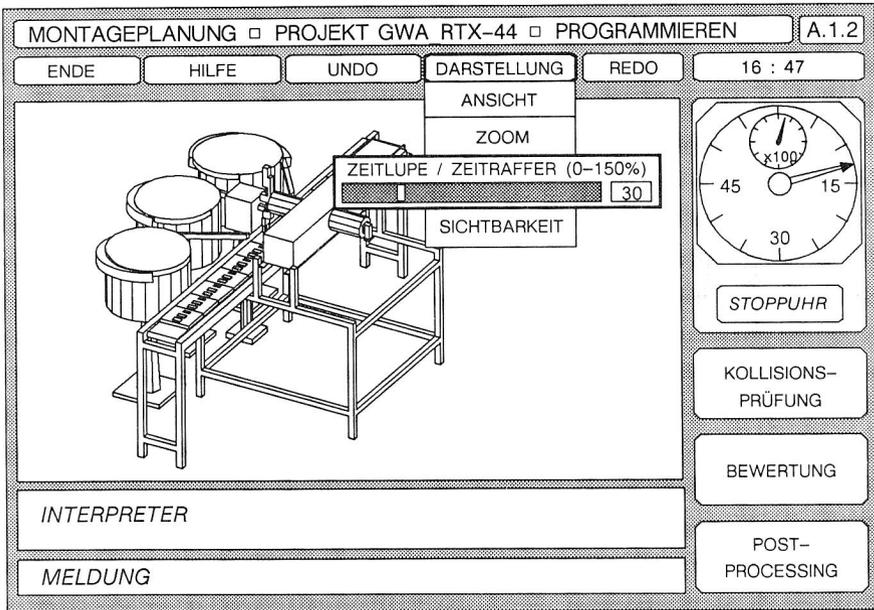


Bild 45: Programmieren und Simulieren von Montagezellen

Wichtigstes Werkzeug zur Messung von Einzel- und Fortschrittszeiten ist die mitlaufende Stoppuhr (siehe Bild 46). Für den Montagefeinplaner ist die Einhaltung der vorgegebenen Taktzeit wichtigstes Kriterium für die Bewertung seines Zellenentwurfs.

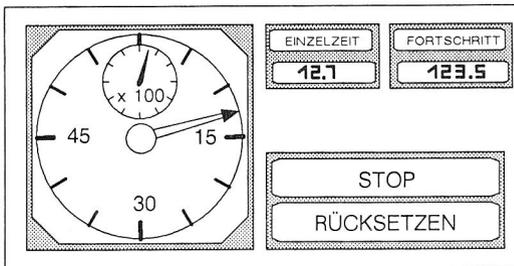


Bild 46: Stoppuhr zur Messung der simulierten Taktzeit

Die Einstellung von Zeitlupe oder Zeitraffer, welche durch einen sog. Scroll-Bar realisiert ist, hat keinen Einfluß auf die berechnete Dauer eines Simulationslaufes. Der Anwender kann jedoch durch die Wahl dieser Funktion das Antwortzeitverhalten des Systems beeinflussen, indem er die Intervalle für Berechnungsschritte und Bildaufbau vergrößert oder verkleinert.

- Zoomen,
- Verdrehen, Verschieben des Betrachterstandpunktes,
- Darstellung aus mehreren beliebigen Sichten,
- Ein- und Ausblenden von Objekten,
- Clipping

sind Funktionen, die ebenfalls durch das Basissystem CAD zur Verfügung gestellt werden und für eine komfortable Visualisierung der graphischen Simulation eingesetzt werden können.

5.1.2 Aufgabenbeschreibung, Programmierung

Mit Hilfe einer formalen Aufgabenbeschreibung definiert der Montageplaner die einzelnen Schritte, die zur Durchführung der gestellten Montageaufgabe notwendig sind. Seiner gewohnten Tätigkeit entsprechend, kann sich diese Beschreibungsform an Methoden der Arbeitsplanerstellung oder der Einweisung von Personal am Montagearbeitsplatz orientieren. Die Aufgabenbeschreibung soll implizit erfolgen. Dazu ist es notwendig anzugeben:

- welche Zellenkomponente
- welche Tätigkeit
- mit welcher(n) Baugruppe(n) (Teil)
- an welchem Ort ausführt.

Dabei sind auch Kontrollstrukturen einzubringen, die aussagen:

- welche Operationen parallel ausgeführt werden,
- wie oft sich bestimmte Vorgänge wiederholen und
- welche Ereignisse zu veränderten Reaktionen führen.

Aus Entwicklersicht treten zudem die Forderungen hinzu, daß die verwendete Sprache syntaktisch und semantisch eindeutig interpretierbar sein muß und daß der eingegebene Quellcode sich in die entsprechenden Codes der Zielsysteme (IRDATA, CLDATA, SPS-Code nach DIN 19239) übersetzen läßt.

Als Eingabeelemente lassen sich für die zu beschreibenden Tätigkeiten der Montage die in Bild 47 dargestellten Gruppen nach DIN 8593 zusammenfassen.

ZUBRINGEN:	<input type="checkbox"/> bunkern	<input type="checkbox"/> ordnen	<input type="checkbox"/> eingeben
	<input type="checkbox"/> magazinieren	<input type="checkbox"/> drehen	<input type="checkbox"/> ausgeben
	<input type="checkbox"/> weitergeben	<input type="checkbox"/> wenden	<input type="checkbox"/> positionieren
	<input type="checkbox"/> abzweigen	<input type="checkbox"/> schwenken	<input type="checkbox"/> spannen
	<input type="checkbox"/> zusammenführen	<input type="checkbox"/> zuteilen	<input type="checkbox"/> entspannen
KONTROLLIEREN:	<input type="checkbox"/> messen	<input type="checkbox"/> prüfen	
JUSTIEREN:	<input type="checkbox"/> einstellen	<input type="checkbox"/> abstimmen	<input type="checkbox"/> eichen
HILFSFUNKTIONEN:	<input type="checkbox"/> reinigen	<input type="checkbox"/> bedrucken	<input type="checkbox"/> kühlen
	<input type="checkbox"/> entgraten	<input type="checkbox"/> markieren	
	<input type="checkbox"/> anpassen	<input type="checkbox"/> erwärmen	
FÜGEN:	<input type="checkbox"/> durch Zusammenlegen: <ul style="list-style-type: none"> ▪ einlegen ▪ einhängen 	<input type="checkbox"/> durch Umformen: <ul style="list-style-type: none"> ▪ falzen ▪ vernieten 	
	<input type="checkbox"/> durch Füllen: <ul style="list-style-type: none"> ▪ tränken 	<input type="checkbox"/> durch Stoffverbinden: <ul style="list-style-type: none"> ▪ schweißen ▪ löten ▪ kleben 	
	<input type="checkbox"/> durch An- und Einpressen: <ul style="list-style-type: none"> ▪ teilen ▪ schrauben ▪ pressen 	<input type="checkbox"/> durch andere Haftverfahren: <ul style="list-style-type: none"> ▪ binden ▪ nähen 	
	<input type="checkbox"/> durch Urformen: <ul style="list-style-type: none"> ▪ umgießen ▪ umpressen 		

Bild 47: Montagetätigkeiten nach DIN 8593

Die Art der Eingabe orientiert sich am Systembenutzer. Um dabei die Akzeptanz für das Programmiersystem zu erhöhen, soll die Eingabetechnik dem gewohnten Arbeitsablauf und dem natürlichen Sprachgebrauch entsprechen. Die folgenden Möglichkeiten der Auf-

gabenbeschreibung orientieren sich im Aufbau an der Grammatik der deutschen Sprache und sind deshalb leicht zu erlernen. Satzglieder sind:

- Subjekt : Zellenkomponente, die eine Aktion ausführen soll
- Prädikat: beschreibt die auszuführende Verrichtung
- Objekt : Gegenstand an dem oder mit dem eine Verrichtung durchgeführt wird
- Ort : die Stelle, an der die Verrichtung durchgeführt/beginnen soll.

Mit diesen Sprachelementen kann textuell, symbolisch, graphisch oder in einer hybriden Form ein Zellenprogramm zusammengestellt werden. Beispiel für die schrittweise textuelle Programmbeschreibung:

Subjekt:	Prädikat:	Objekt:	Ort:
Manutec r3	entnehmen	2-Finger-Greifer	Greifermagazin
Manutec r3	entnehmen	Bauteil B1	Wendelförderer

Kontrollstrukturen und logische Anweisungen sind aus dem Montagevorranggraphen ableitbar und werden entweder automatisch durch das System oder vom Benutzer durch folgende Spracherweiterungen eingebracht:

- verrichten n mal
- wenn ... dann ... sonst ...
- verrichten bis ...
- verrichten während ...

Solche Kontrollstrukturen sind Sprachelemente aus bekannten Programmiersprachen und stellen kein Problem bei der Verarbeitung dar. Größere Schwierigkeiten ergeben sich bei der Interpretation impliziter Ablaufsequenzen, denn hier muß Umwelt- und Technologiewissen verarbeitet werden. Dazu muß das Programmiersystem entweder alle notwendigen Informationen aus den rechnerinternen Modellen ableiten oder vom Benutzer nachfordern (z.B. an welcher

Position und mit welcher Kraft ein Greifer aus dem Magazin genommen werden kann s.o.).

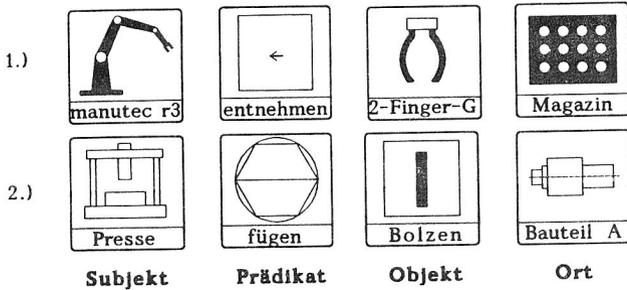
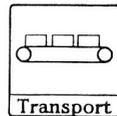
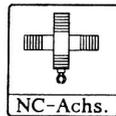
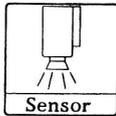
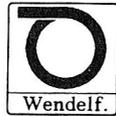
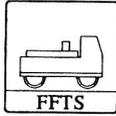
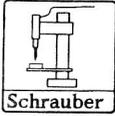


Bild 48: Symbolisch unterstützte Eingabe über Auswahlmenüs

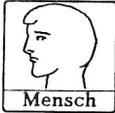
Der Vorteil einer rein textuellen Eingabetechnik besteht in der starken Ähnlichkeit mit der Erstellung eines Arbeitsplanes. Ein entscheidender Nachteil ist, daß der Benutzer die gesamten Bezeichnungen (Namen) von Subjekten, Prädikaten, Objekten und Orten genau kennen bzw. diese über das System erfragen muß. Zudem hat sich gezeigt, daß sich bei textuellen Eingaben Schreibfehler einschleichen, die auch von 'intelligenten' Programmiersystemen häufig nicht verarbeitet werden können; in der Regel melden solche Systeme nur die Ursache des Fehlers. Man kann Fehler bei textueller Eingabe dadurch vermeiden, daß die einzugebenden Satzglieder über Menüleiste ausgewählt und zu Programmzeilen zusammengefügt werden. Das System sorgt dabei für die korrekte lexikalische Formulierung und den richtigen syntaktischen Aufbau des Programmes.

Die symbolische Art der Eingabe hat gegenüber der rein textuellen Eingabe den Vorteil, daß die Sprachelemente einfach erkannt und ausgewählt werden können. Einlernzeit und Eingabezeiten lassen sich durch diese Technik reduzieren. Bei der Auswahl der in Bild 48 dargestellten Ikonen, kann man sich an den Symbolen aus der VDI Richtlinie 2860 (siehe Bild 49) orientieren.

- Zellenkomponenten:



- Mensch:



- Tätigkeiten:

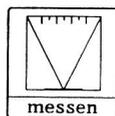
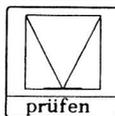
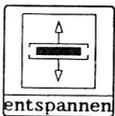
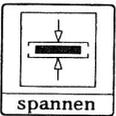
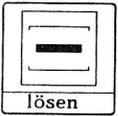
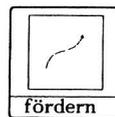
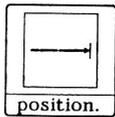
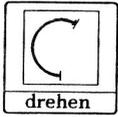
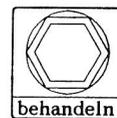
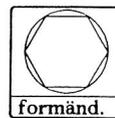
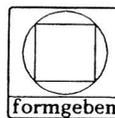
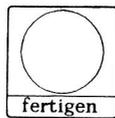
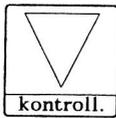
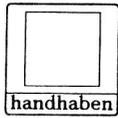


Bild 49: Ikonen für die Montage nach VDI 2860

Da viele Tätigkeiten sich schwer oder gar nicht durch diese Symbole beschreiben lassen, ist es notwendig weitere Symbole selbst zu definieren (z.B. Entnehmen).

Die dritte Möglichkeit, Subjekte, Objekte und Orte zu identifizieren bzw. zu selektieren, besteht in der interaktiven Beschreibung mit Hilfe der 3D-Animation. Hierbei ist die zu programmierende Montagezelle in dreidimensionaler Darstellung auf dem Bildschirm vorhanden. Prädikate, also Verrichtungen, werden wieder über Menüleiste ausgewählt. Der Vorteil dieser Vorgehensweise besteht vor allem darin, daß man Zellenkomponenten oder Produktteile nicht kennen sondern nur erkennen muß. Außerdem wird durch diese Art der Programmierung eine Simulation der eingegebenen

nen Anweisungen möglich. Die Aufgabe muß also nicht von Anfang bis Ende formuliert werden, sondern kann schrittweise (interaktiv) zusammengestellt werden (siehe Bild 50).

So kann die ausführende Zellenkomponente über einen mausgesteuerten Cursor (Zeigen und Picken) selektiert werden. Das Prädikat kann ausgewählt werden aus den möglichen Manipulationen, die auf das selektierte Element anwendbar sind. Ebenso wie Subjekte können Objekte und Orte graphisch gepickt werden. Die vorgestellte Methode läßt sich vergleichen mit der Einweisung eines Montagearbeiters, bei der durch Zeigen, Anfassen (=Selektieren) von Montagekomponenten die Aufgabe schrittweise erklärt wird. Um die Vorzüge aller dargestellten Arten zur Aufgabenbeschreibung zu vereinigen, kann man die Verfahren kombinieren (hybride Technik) oder/und dem Benutzer selbst die Entscheidung überlassen, mit welcher Technik der Aufgabenbeschreibung er arbeiten möchte.

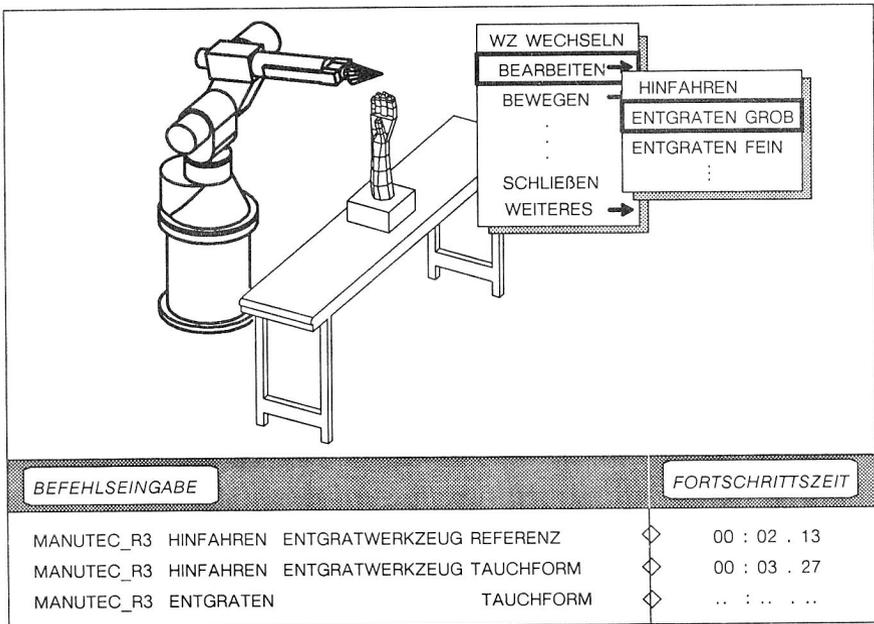


Bild 50: Grafisch interaktive Programmierung

Wenn die Aufgabe vollständig formuliert ist, kann durch einen Simulationslauf die Vollständigkeit und Richtigkeit des Montagevorganges geprüft werden. Eine schrittweise Abarbeitung (TRACE) des Zellenprogramms soll die Möglichkeit bieten, zu jedem Zeitpunkt das Programm zu verändern.

Entspricht der Montageablauf allen geforderten Bedingungen, so müssen für die einzelnen Steuerungen der Zelle die passenden Programme erzeugt und zur Übertragung vorbereitet werden. Für automatisierte Zellen wird dazu ein Satz von Einzelprogrammen generiert, für manuelle Arbeitssysteme ein Arbeitsplan.

Der Weg zurück von der Betriebsebene zur Planung soll dadurch gewährleistet sein, daß auch existierende explizite Einzelprogramme simuliert und in komplette Aufgabenbeschreibungen eingebunden werden können. Damit soll sichergestellt werden, daß

- existierende Programme aus der Produktion wiederverwendet werden können und
- auf Betriebsebene veränderte Programme zum Abgleich in der Planung verwendet werden können.

5.1.3 Einsatz von Grafik und Farben

CAD-Systeme, die auf der Basis der 3D-Modellierung arbeiten, finden in zunehmendem Maße Eingang in modernen Produktionsbetrieben. Gerade die Methoden und Werkzeuge der dreidimensionalen Modellierung und Farbschattierung werden heute längst nicht mehr als reine Spielerei betrachtet. So wurden in den letzten Jahren zahlreiche Anwendungen gefunden, die den Nutzen bei der Verwendung von Farbgrafiken mit 3D-CAD-Objekten verdeutlichen. Farben bieten die Möglichkeit, komplexe Geometrien am Bildschirm verständlich abzubilden.

Montageanlagen, die in ihren Abmessungen weit größer sind als die Dimension eines normalen Grafikbildschirmes, lassen sich nur mit

Hilfe von Farben akzeptabel darstellen.

Das komplizierte Bewegungsverhalten flexibler Handhabungssysteme, z.B. das von Industrierobotern, kann bei der Planung einer Montagezelle von ausschlaggebender Bedeutung sein. Durch Simulation der Bewegungsvorgänge im Montageprozeß können Montageabläufe berechnet und grafisch visualisiert werden. Die in /52/ erläuterte Methode zur Abbildung von Bildinformationen in Bitmaps ermöglicht die Animation von Bewegungsabläufen in Form von "Bitmap-Filmen". Der Rechenaufwand bei der Erstellung solcher Filme ist allerdings erheblich, so benötigt eine Bildfolge von 20 Elementen (=Simulationsschritten) für die in Bild 51 dargestellte Montagezelle eine Rechenzeit von mehreren CPU-Stunden und ein Speichervolumen von ca. 24 MB unter Verwendung einer 32 Bit CAD - Workstation. Durch den Einsatz graphischer Standards (PHIGS), grafischer Prozessoren und einfacher Schattierungsalgorithmen (Gouraud Shading) können heute bereits online Farbsimulationen auf CAD-Workstations durchgeführt werden.

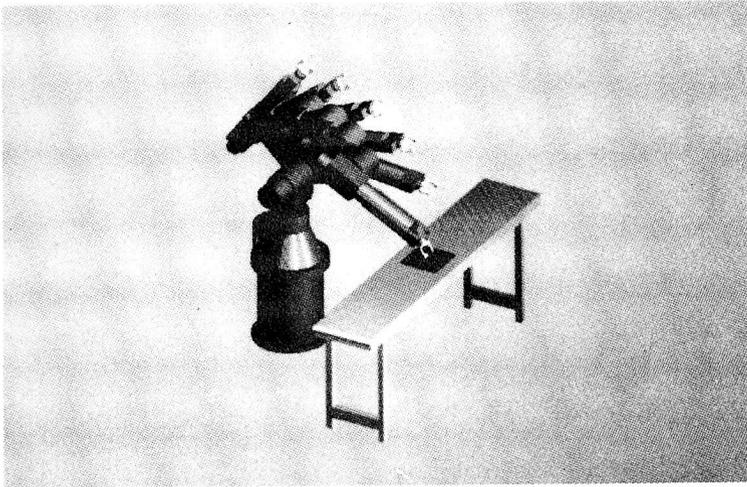


Bild 51: Bewegungsabschnitt in einem "Montagefilm" mit Kollision

5.2 Erzeugen des Datenmodells

Der Leistungsumfang rechnergestützter Systeme ist abhängig von der Qualität des zugrundeliegenden Datenmodells. Dies gilt insbesondere für Systeme zur aufgabenorientierten Planung und Programmierung von Montagezellen.

In Kapitel 3 wurden die Komponenten von Montagezellen beschrieben. Für die datentechnische Verarbeitung müssen sämtliche Informationen über diese Komponenten strukturiert und in rechnerverständlicher Weise programmiert werden. Dieser Vorgang wird als Modellieren bezeichnet. Je genauer das verfügbare Datenmodell ist, desto umfangreicher sind die Möglichkeiten der Rechnerunterstützung.

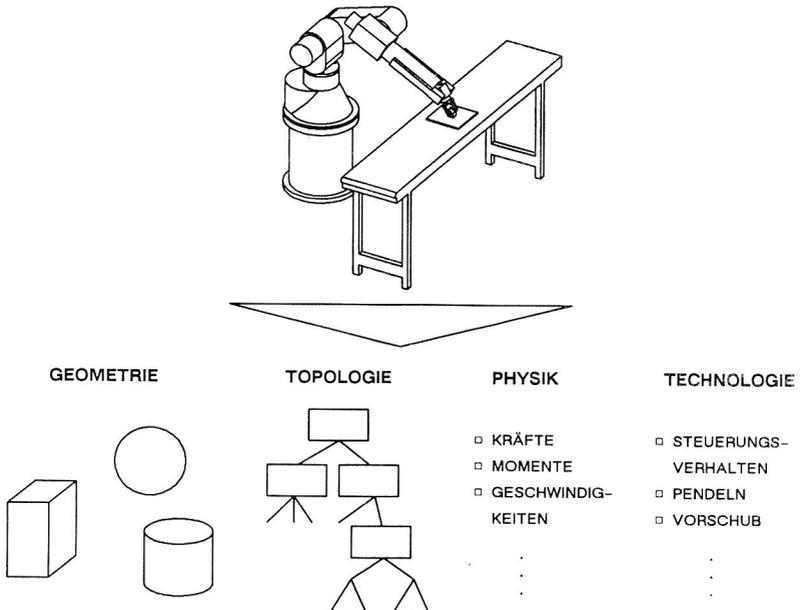


Bild 52: Abbildung der realen Welt auf ein rechnerinternes Modell

Bei der Modellierung einzelner Objekte ist nicht nur die geometrische Information zu berücksichtigen, sondern vorallem die

technologischen und physikalischen Eigenschaften der Objekte. In den folgenden Kapiteln werden die Datenmodelle für

- Montageobjekte (Produktmodell)
- Montagekomponenten (Komponentenmodell)
- Montagezellen (Zellenmodelle)

unter dem eingangs erläuterten CAD-integrierten Konzept beschrieben. Zuletzt wird die Möglichkeit der Verwaltung solcher Modelle in einer relationalen Datenbank erörtert. Zunächst werden einige Vereinbarungen getroffen, die für die Modellierung von Zellelementen von wesentlicher Bedeutung sind.

MATRIX BESCHREIBUNG	FRAME	BEZUGSSYSTEM
T_k	Körperbezugsframe	Umweltsystem
I	Position eines Teils in einer Baugruppe (Instanz)	Baugruppe
T_G	Greifframe	Körperbezugsystem Werkstück
T_{Gr}	Greiferframe	Körperbezugsystem Werkzeug
T_B	Bearbeitungsframe	Körperbezugsystem Werkstück
T_P	Prüfframe	Körperbezugsystem
A	Achsframe i+1	Achssystem i
T_{KW}	Koppelframe Werkzeug	Körperbezugsystem Werkzeug
T_{KR}	Koppelframe Roboter	Achssystem n
T_{TCP}	TCP – Frame u.a.	Koppelframe Werkzeug
T_F	Freies Frame	Umweltsystem

Bild 53: Klassen von FRAMES

Unter Frame (dt. Rahmen) wird im Bereich der künstlichen Intelligenz (KI) eine Möglichkeit der Wissensrepräsentationen

verstanden. Frames können beschreibende und prozedurale Informationen beinhalten. Entscheidend ist, daß Vererbungsmechanismen sowie dynamische Veränderungen der Frameeinträge (Slots) während der Laufzeit von Zugriffen durchgeführt werden können.

Im Gegensatz dazu wird im Bereich der Roboterkinematik ein Frame als ein kartesisches Rechtskoordinatensystem definiert. Es impliziert die Existenz eines Bezugskoordinatensystems, bezüglich dessen es beschrieben ist. Ein solches Frame wird durch die Position \underline{p} seines Ursprungs und die Orientierung seiner Einheitsvektoren im Bezugskoordinatensystem beschrieben und kann mit dem zugehörigen Körper mitbewegt werden.

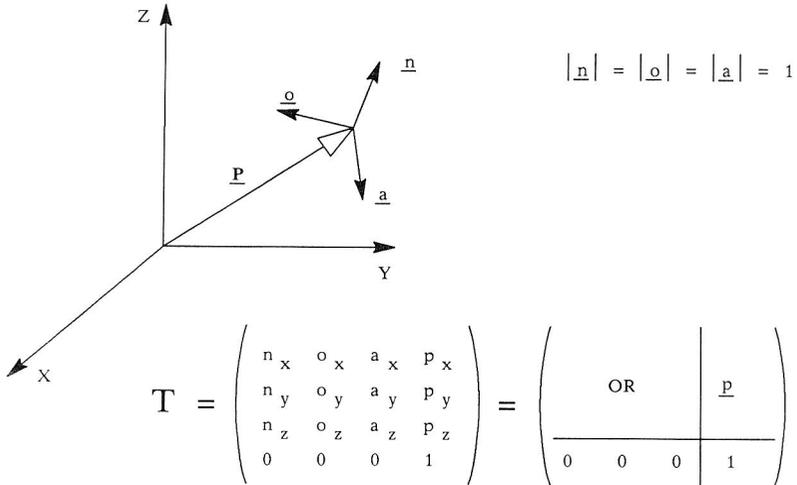


Bild 54: Beschreibung eines Frames durch die Transformation T

In der vorliegenden Arbeit besitzen Frames weitere Eigenschaften (siehe Bild 53). Greifframes beispielsweise beschreiben die Stellen, an denen Werkstücke gegriffen werden können. Das Greifframe bezieht sich dabei auf das werkstückeigene Körperbezugsframe. Das kartesische Dreibein des Greifframes legt durch die z-Koordinate auch die Anfahrrichtung des Greifers bzw. der Greifelemente fest (siehe Bild 65). Außerdem wird die Auswahl von

Greifern durch weitere Eigenschaften des FRAMES unterstützt. Um Frames geometrisch im Raum zu definieren oder Positionsveränderungen während einer Bewegungssimulation durchzuführen, muß eine mathematische Beschreibungsform gewählt werden, die einfach zu erzeugen ist und welche mit den Grundrechenarten der Vektoralgebra behandelt werden kann.

Die Lage und Orientierung eines Teils bezogen auf ein Ursprungs-koordinatensystem läßt sich durch eine Transformation T darstellen (siehe Bild 54).

Besonders wichtig ist diese Form der Beschreibung für die Berechnung von verketteten Systemen -wie System: Roboterachse 1 bis Roboterachse n, Greifer, Werkstück. Die Verkettung von FRAMES kann mathematisch durch Matrizenmultiplikation berechnet werden.

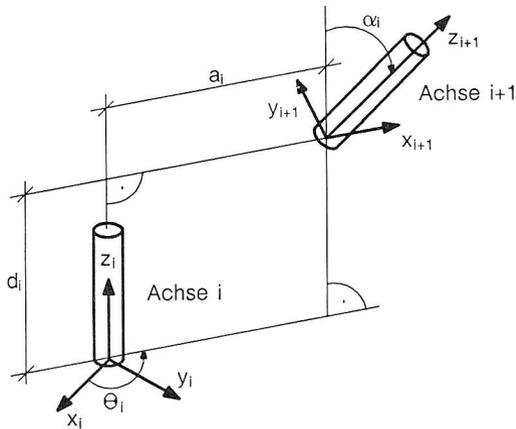


Abb 55: Framebeschreibung nach Denavit-Hartenberg

Durch das Denavit - Hartenberg Verfahren /62/ lassen sich offene kinematische Gelenkketten beschreiben und berechnen. Folgende Regeln sollen zur Beschreibung solcher Ketten dienen:

- Ein rotatorischer Freiheitsgrad ist durch eine Dre-

hung um die z-Achse des entsprechenden Achsframes definiert.

- Ein translatorischer Freiheitsgrad entspricht einer Verschiebung entlang der z-Achse des zugehörigen Achsframes.

Die Transformation nach Denavit-Hartenberg ist durch eine Folge von Rotationen und Translationen bestimmt. Es gilt gemäß Bild 55:

1. Drehung um die Achse z_i mit dem Winkel θ_i
2. Verschiebung entlang der Achse z_i um den Abstand d_i
3. Translation entlang der Achse x_{i+1} um den Abstand a_i
4. Drehung um die Achse x_{i+1} mit dem Winkel α_i

Zusammengefaßt lassen sich diese vier Operationen in einer 4x4 Transformationsmatrix A_i darstellen:

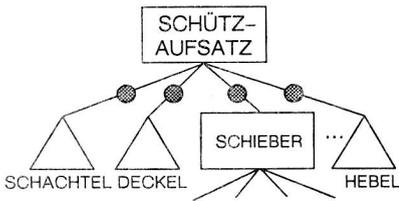
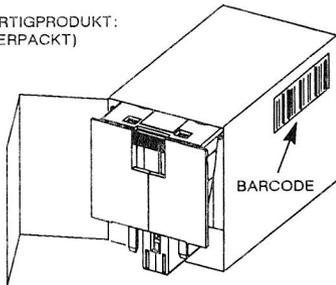
$$\left[\begin{array}{ccc|c} \cos \theta_i & -\cos \alpha_i \sin \theta_i & \sin \alpha_i \sin \theta_i & \alpha_i \cos \theta_i \\ \sin \theta_i & \cos \alpha_i \cos \theta_i & -\sin \alpha_i \cos \theta_i & \alpha_i \sin \theta_i \\ 0 & \sin \alpha_i & \cos \alpha_i & d_i \\ \hline 0 & 0 & 0 & 1 \end{array} \right]$$

wobei hier, wie in Bild 55, Lage und Orientierung von Achse $i+1$ bezogen auf die Achse i direkt aus der Transformationsmatrix abgelesen werden.

5.2.1 Produktmodelle

Produkte werden heute bereits mit Hilfe von CAD-Systemen konstruiert. Man kann daher geometrische Informationen -falls diese rechnerintern dreidimensional verfügbar sind- als gegeben voraussetzen.

FERTIGPRODUKT:
(VERPACKT)



IDENT. : RTX44-xxx
 NAME : SCHÜTZAUFSATZ
 ELEMENTTYP : ERZEUGNIS
 VÄTER : TOP
 GEOMETRIE : xxx.2D
 xxx.3D
 MONTAGE : ARBEITSPLAN Nr.xxx
 GREIFEN : $T_G, FRAME_{xxx}, PARAM.$
 $T_G, FRAME_{xxx}, PARAM.$
 ...
 PRÜFEN : $T_{QS}, FRAME_{xxx}, PARAM.$
 QS
 ERKENNUNG : BARCODE,WERT
 GEWICHT : Σ GEWICHT (SÖHNE)
 STABILE LAGEN: STABILITÄTSRECHNUNG
 ...
 SÖHNE :
 SCHACHTEL, IDENTxxx, T_K, I
 DECKEL, IDENTxxx, T_K, I
 SCHIEBER, IDENTxxx, T_K, I
 ...
 HEBEL, IDENTxxx, T_K, I

Bild 56: Modell eines Erzeugnisses

Anders ist dies bei technologischen und physikalischen Eigenschaften von Produkten. Um aufgabenorientierte Programmanweisungen durchführen zu können, müssen die gespeicherten Produktmodelle über wesentlich mehr als nur geometrische Informationen verfügen. Solche Daten betreffen z.B.

- Positionen zum Greifen oder Handhaben
- zulässige Kräfte beim Handhaben und Bearbeiten
- physikalisches Verhalten vor, während und nach dem Einwirken (z.B. Deformation)
- Stabile Lagen
- Reflexionseigenschaften bei aktiven oder passiven Beleuchtung und vieles mehr.

Bild 56 zeigt neben der geometrischen Information auch die Topologie einer Baugruppe. Montageprodukte werden unterschieden nach:

- Erzeugnis (Bild 56)

- Baugruppe und (Bild 57)
- Einzelteil. (Bild 58)

Erzeugnisse, Baugruppen und Einzelteile sind untereinander verzeigert, so daß sich in der Struktur des Datenmodells der logische Aufbau des Produktes widerspiegelt. Als Attribut kann zu der Produktstruktur jedem Montageteil die Position im Montagevorangeordneten Graphen zugeordnet werden. Informationen zur Montage der Teile sind in Form von Verweisen auf Bearbeitungsprozeduren oder in Form von Attributen an das Datenmodell gebunden.

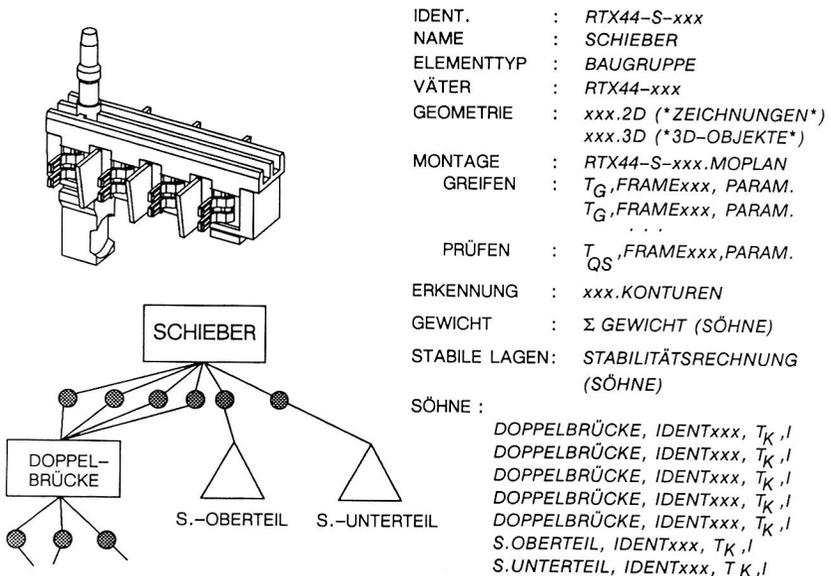
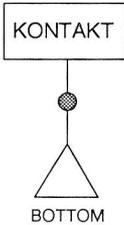
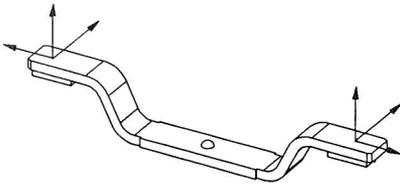


Bild 57: Modell einer Baugruppe

Durch den Programmiervorgang wird das aus der Konstruktion vorgegebene Datenmodell kontinuierlich erweitert. Das Produktmodell ist somit einem ständigen Änderungs- bzw. Erweiterungsprozeß unterworfen. Dies setzt eine flexible dynamische Datenstruktur voraus, wie sie beispielsweise in zur Laufzeit veränderbaren Systemen angewendet wird.



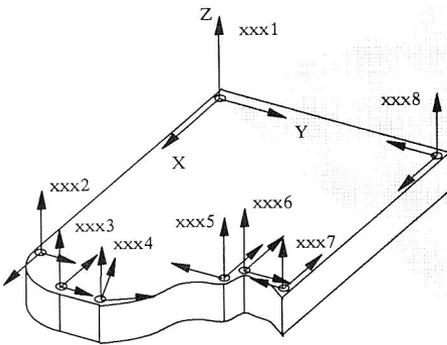
IDENT.	:	RTX44-D-K-xxx
NAME	:	KONTAKT
ELEMENTTYP	:	EINZELTEIL
VÄTER	:	RTX44-D-xxx RTX44-D-yyy
GEOMETRIE	:	xxx.2D (*ZEICHNUNGEN*) xxx.3D (*3D-OBJEKTE*)
FERTIGUNG	:	ARBEITSPLAN.FERPLAN
GREIFEN	:	$T_G, FRAME_{xxx}, PARAM.$ $T_G, FRAME_{xxx}, PARAM.$...
PRÜFEN	:	$T_{QS}, FRAME_{xxx}, PARAM.$
ERKENNUNG	:	xxx.KONTUREN
GEWICHT	:	0.002 kg
STABILE LAGEN:	:	STABILITÄTSRECHNUNG (FEM)
SÖHNE	:	BOTTOM

Bild 58: Modell eines Einzelteils

Für die aufgabenorientierte Programmierung ist die Anbindung von Greif- und Bearbeitungsframes von entscheidender Wichtigkeit /63/. Durch diese Frames werden nicht nur die geometrischen Positionen zum Greifen oder Bearbeiten vorgegeben, sondern auch Hinweise zum Anfahren oder zur technischen Ausführung des Bearbeitungsvorganges. Die Dimensionalität eines Greifframes -1,2 oder n- gibt an, ob die Greifposition sinnvoll durch einen Einflächen- oder Mehrflächengreifer angefahren werden kann. Dies sei am Beispiel: Greifframe (Typ: 1-dimensional) und anhand des Bearbeitungsframes (Typ: Kleben) dokumentiert.

Vakuumgreifer mit einer Saugereinheit oder Magnetgreifer mit einem Greifpol können für den Greifvorgang an diesem Werkstück in Betracht gezogen werden. Gebräuchlich und weit verbreitet sind Greifersysteme, die an zwei Stellen -z.B. Zweifingergreifer- fassen und eine reibschlüssige Verbindung herstellen. Von entscheidender Bedeutung für das Funktionieren der Verbindung

Kleben, Schweißen, Löten oder Entgraten sind Fertigungsverfahren, die entlang räumlichen Bahnen auszuführen sind. Die Bearbeitungsframes für diese Operationen müssen deshalb den Verlauf dieser Bahnen wiedergeben. Zusätzlich müssen Bearbeitungsvorschriften wie Geschwindigkeit beim Klebeauftrag, Schweißnahtform, Lötmaterial oder maximale Gratdicke dem Frame zugeordnet werden.



```

IDENT.       : xxx1
FRAME        : BEARBEITEN
ELEMENTTYP   : KLEBEN
BEWEGUNGS-  : LINEAR, ZIRKULAR,
ART          : SPLINE
ENDPUNKT     : POSITION_xxx2
STÜTZPUNKT   : NIL
GESCHWINDIG- : 0,02 m/sec
KEIT, TANGENTIAL
KLEBSTOFF    : Wyyy
FILMDICKE   : 4 mm * 0.5 mm
  
```

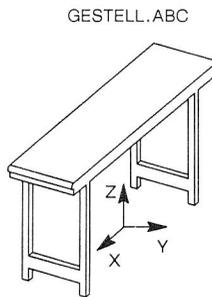
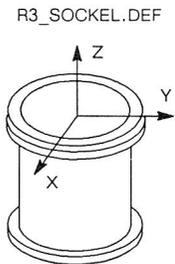
Abb 60: Bearbeitungsframes

Ein Klebeframe impliziert die Normalenrichtung der Klebstoff (z-Achse) und der tangentialen Bewegung beim Klebeauftrag (y-Achse). Einige Daten (entities) der Frames können aus dem geometrischen Modell ermittelt werden -wie z.B. die Bewegungsart beim Kleben. Andere Frameeinträge müssen beim Modelliervorgang oder beim nachfolgenden Programmieren hinzugefügt werden.

5.2.2 Komponentenmodelle

Gemäß Kapitel 3 besitzen Montagezellen verschiedene Klassen von Komponenten (siehe Bild 16). Komponenten -wie Abdeckungen, Schränke, Gestelle- die weder agierend noch steuernd am Montageprozeß beteiligt sind, können als einfache, passive Körper betrachtet werden.

Da Robotersysteme die technologisch kompliziertesten Geräte in Montagesystemen sind, ist auch deren Modellierung am aufwendigsten. NC-Bewegungsachsen oder andere Bewegungssysteme -wie FTS, Pick- und Placegeräte, u.a.- können im Modell als Untermenge von Robotern behandelt werden, da ihre grundsätzliche Aufgabe, nämlich das Bewegen von Werkstücken oder Werkzeugen, die gleiche und nur die Zahl ihrer Bewegungsfreiheiten kleiner ist. Man unterscheidet bei Bewegungssystemen zwischen offenen und geschlossenen kinematischen Ketten. Für beide Typen gibt es Steuerungsalgorithmen und Datenstrukturen die eine Simulation der Bewegungsabläufe zulassen.



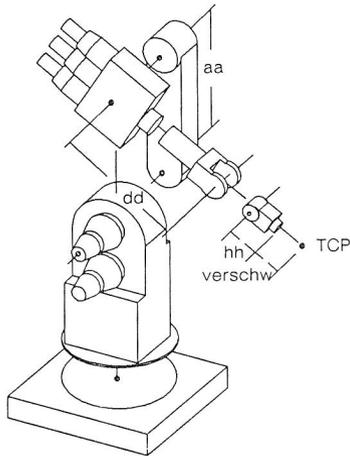
EIGENSCHAFTEN:

- NAME BZW. IDENT.
- KÖRPEREIGENES
BEZUGSFRAME (T_K)
- VOLUMENGEOMETRIE

Bild 61: passive Zellenkomponenten

Eine kinematische Kette ist nach VDI 2860/2861 "eine zusammenhängende Struktur mehrerer starrer Körper", die durch Gelenke miteinander verbunden sind. Ein charakteristisches Merkmal der einzelnen Körper einer kinematischen Kette ist deren Freiheitsgrad. Industrieroboter bilden kinematische Ketten, wobei deren Freiheitsgrad dem Freiheitsgrad des letzten Körpers der kinematischen Kette entspricht.

Für eine Modellierung ist es notwendig, folgende Informationen über das Bewegungssystem zu kennen:



ACHSE	θ	a	d	α	$\dot{\theta}_{\min}$	$\dot{\theta}_{\max}$	$\ddot{\theta}_{\max}$
1 \Rightarrow 2	θ_1	0°	0°	-90°	-162.5°	162.5°	$171^\circ/\text{s}$
2 \Rightarrow 3	θ_2	aa	0°	0°	-110°	110°	$86^\circ/\text{s}$
3 \Rightarrow 4	θ_3	0°	0°	90°	-145°	145°	$300^\circ/\text{s}$
4 \Rightarrow 5	θ_4	0°	dd	-90°	-187.5°	187.5°	$194^\circ/\text{s}$
5 \Rightarrow 6	θ_5	0°	0°	90°	-120°	120°	$242^\circ/\text{s}$
6 \Rightarrow TCP	θ_6	0°	hh+verschw	0°	-270°	270°	$212^\circ/\text{s}$

Bild 62 : Kinematisches Modell des Typs RRRRA3

- Identifikation, Name oder Bezeichnung
- Steuerungs- oder Bewegungstyp
- Zahl der Bewegungsachsen, Freiheiten
- Antriebsmomente, Massen, Maximalgeschwindigkeiten
Grenzstellungen jeder einzelnen Achse
- Bewegungsart jeder Achse (rotatorisch, translato-
risch, Bewegungsschraube)
- Werkzeuganschlußmöglichkeiten.

Anhand einer Marktbetrachtung erkennt man, daß sich viele Roboter- oder Bewegungssysteme in den genannten Eigenschaften stark ähneln bzw. sich nur durch einige variable Größen -wie unter-

schiedliche Antriebsmomente der Achsen oder unterschiedliche Grenzstellungen- unterscheiden. Betrachtet man beispielsweise die Folge von rotatorischen und translatorischen Gelenken bei mehreren bekannten Robotern, so zeigt sich, daß durch 10 kinematische Typen über 90% aller eingesetzten Robotersysteme beschrieben werden.

FABRIKAT	MODELL	ROBOTERKLASSE	HANDGELENKSKLASSE
AEG	SERVATOR	TRR	C2
ASEA	IRb 90 M/2	RRR	A3
ASEA	IRb 60/2	RRR	B2
ASEA	IRb 8200	TTT (PORTALGERÄT)	A1
BOSCH	SR 400...SR 800	TRR	C2
IBM	7535	TRR	C2
KUKA	IR 161	RRR	A3
KUKA	IR 360/8.0	RRR	A3
KUKA	IR 360/8.2	TRR	A3
KUKA	IR 400	TTT (PORTALGERÄT)	A1 ODER A2
MANUTEC	r3	RRR	A3
MANUTEC	h5	TRR	B1
MANUTEC	m3	RTT	A2
PANASONIC	PANAROBO H II	TRR	C2
PANASONIC	PANAROBO A III M	TTT (SÄULENGERÄT)	B1
PANASONIC	PANAROBO VI	RRR	A3
REIS	RR 625	TRR	A3
UNIMATION	PUMA-SERIE	RRR	A3

Bild 63: Fabrikate gängiger Robotertypen

Für eine Modellierung mit möglichst großer Gültigkeit ist es daher sinnvoll, sich auf diese wenigen gängigen Klassen von kinematischen Systemen zu beziehen. In Bild 63 werden einige Fabrikate mit ihrer kinematischen Zuordnung vorgestellt. Am Beispiel eines signifikanten Typs soll im folgenden das kinematische Modell von Robotern erläutert werden.

Der Typ RRRA3 ist ein 6-Achs-Gelenkroboter mit ausschließlich rotatorischen Achsen. Fabrikate diesen Typs findet man in Bild 63. Die kinematische Struktur läßt sich nach Denavit-Hartenberg durch die in Bild 62 stehenden Parameter beschreiben. Es ergeben

sich daraus die folgenden Transformationsmatrizen A_1 bis A_6 , für die gilt:

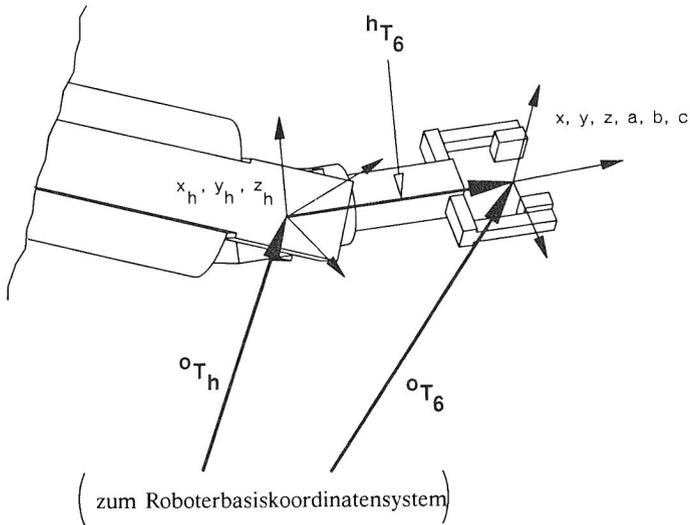


Bild 64: Geometrie des kinematischen Modells eines 6-Achs-Roboters. Zerlegung der Transformation zum TCP in Arm- und Handgelenk

$$A_1 = \begin{bmatrix} c_1 & 0 & -s_1 & 0 \\ s_1 & 0 & c_1 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5.1)$$

$$A_2 = \begin{bmatrix} c_2 & -s_2 & 0 & aa & c_2 \\ s_2 & c_2 & 0 & aa & s_2 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix} \quad (5.2)$$

$$A_3 = \begin{bmatrix} c_3 & 0 & s_3 & 0 \\ s_3 & 0 & -c_3 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5.3)$$

$$A_4 = \begin{bmatrix} c_4 & 0 & -s_4 & 0 \\ s_4 & 0 & c_4 & 0 \\ 0 & -1 & 0 & dd \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5.4)$$

$$A_5 = \begin{bmatrix} c_5 & 0 & s_5 & 0 \\ s_5 & 0 & -c_5 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5.5)$$

$$A_6 = \begin{bmatrix} c_6 & -s_6 & 0 & 0 \\ s_6 & s_6 & 0 & 0 \\ 0 & 0 & 1 & hh \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5.6)$$

Für die Berechnung der TCP-Position durch 0T_6 bezogen auf die Roboterbasis rechnet man also:

$${}^0T_6 = {}^0T_3 \cdot {}^3T_6 = (A_1 \cdot A_2 \cdot A_3) \cdot (A_4 \cdot A_5 \cdot A_6) \quad (5.7)$$

wobei

$${}^0T_3 = \begin{bmatrix} c_1c_{23} & -s_1 & c_1s_{23} & aa c_1c_2 \\ s_1c_{23} & c_1 & s_1s_{23} & aa s_1c_2 \\ -s_{23} & 0 & c_{23} & -aa s_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5.8)$$

$${}^0T_3 = \begin{bmatrix} c_4c_5c_6 - s_4s_6 & -c_4c_5s_6 - s_4s_6 & c_4s_5 & c_4s_5 & (hh+verschw) \\ s_4c_5c_6 + c_4s_6 & -s_4c_5s_6 + c_4c_6 & s_4s_5 & s_4s_5 & (hh+verschw) \\ -s_5s_6 & s_5s_6 & c_5 & c_5 & (hh+verschw)+dd \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (5.9)$$

Man kann nun aus 3T_6 die Verschiebung dd ausklammern und erreicht damit anschaulich eine Zerlegung der Gesamttransformation 0T_6 in eine Positionstransformation 0T_h zum Handpunkt des Roboters und eine Transformation hT_6 zum TCP (siehe Bild 64). Diese Form der Beschreibung erleichtert, wie später dargestellt, das Berechnen der 'inversen Kinematik'.

0T_h und hT_6 errechnen sich zu:

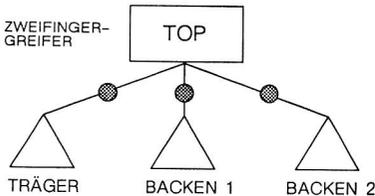
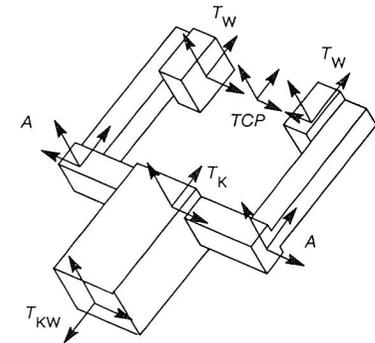
$${}^0T_h = \begin{bmatrix} c_1c_{23} & -s_1 & c_1s_{23} & aac_1c_2 + ddc_1s_{23} \\ s_1c_{23} & c_1 & s_1s_{23} & aas_1c_2 + dds_1s_{23} \\ -s_{23} & 0 & c_{23} & -aas_2 + ddc_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5.10)$$

$${}^0T_3 = \begin{bmatrix} c_4c_5c_6 - s_4s_6 & -c_4c_5s_6 - s_4s_6 & c_4s_5 & c_4s_5 & (hh+verschw) \\ s_4c_5c_6 + c_4s_6 & -s_4c_5s_6 + c_4c_6 & s_4s_5 & s_4s_5 & (hh+verschw) \\ -s_5s_6 & s_5s_6 & c_5 & c_5 & (hh+verschw) \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (5.11)$$

Greifer sind Werkzeuge mit der Funktion, Teile oder andere Komponenten zu greifen. Greifer können ebenso wie Roboter eigene Kinematik besitzen. Diese kann äußerst kompliziert -siehe künstliche Hand /64/- oder auch sehr einfach sein (siehe z.B. Zweibackengreifer). Magnetgreifer, Sauggreifer, Greifrüssel besitzen dagegen keine beweglichen Elemente, sondern ermöglichen den Greifvorgang durch andere physikalische Effekte. Bild 65 zeigt die

einfache Geometrie und das technologische Modell eines Greifers.

Wesentlich an Greifern sind die Stellen an denen Werkstücke oder andere Komponenten gegriffen werden. Diese Stellen sind identifizierbar durch Greiferframes, deren Lage geometrisch durch die Transformation T_w bezogen auf das Greiferbezugssystem (Körperframe) beschrieben ist. Greiferframes besitzen Informationen über die Technik des Zugreifens (z.B. magnetisch, Kfg = Kohäsion fester Körper, Unterdruck), die Richtung des Anfahrens -hierbei zeigt die Normale des Greiferframes in Anfahrriichtung- und den gegenwärtigen Zustand des Greifers (Greifer im Eingriff, Greifer nicht im Eingriff).



IDENT. : GR 72.xxx
 NAME : xxx
 ELEMENTTYP : ZWEIFINGERGREIFER
 GEOMETRIE : xxx.2D
 : xxx.3D
 ZUSTÄNDE : 0 AUF, 1 ZU
 INPUT/OUTPUT : 1 DIGIT
 KOPPLUNG : IDENT., T_{KW} , TYP xxx
 TCP-VERSCHIEBUNG : TCP
 MAX. TRAGLAST : 0.5 KG
 GEWICHT : Σ GEWICHT (SÖHNE)
 KINEMATIK :
 BACKEN 1, A, Δ d, F, ρ , T , PNEUMATIK
 BACKEN 2, A, Δ d, F, ρ , T_W , PNEUMATIK
 VÄTER : TOP
 SÖHNE :
 BACKEN 1, IDENTxxx, T_K
 BACKEN 2, IDENTxxx, T_K
 TRÄGER, IDENTxxx, T_K

$F = \text{ANPREBKRAFT} \begin{bmatrix} N \\ \end{bmatrix}$
 $\rho = \text{REIBBEIWERT} \begin{bmatrix} - \\ \end{bmatrix}$

Bild 65: Modell eines Greifers

Werkzeuge oder Bearbeitungseinrichtungen können ebenfalls eine eigene Kinematik besitzen -z.B. schwenkbare Düse beim Lackieren, Mehrfachwerkzeug, angetriebene Werkzeuge (Bohrer) oder Werkzeuge die Werkstückverformungen herbeiführen sollen. Auch bei Werkzeug-

gen sind solche Stellen von Interesse, an denen ein Werkstückeingriff erfolgt. Diese Stellen sind durch die Transformation T_w , bezogen auf das werkzeugeigene Körperkoordinatensystem, beschrieben. Neben der geometrischen Position der Einwirkframes besitzen diese zusätzliche Angaben über die Art des Einwirkens - kleben, bohren, schrauben, biegen, entgraten, usw. - einschließlich der zugehörigen technischen und physikalischen Parameter. Bild 66 gibt ein Beispiel für ein Schraubwerkzeug.



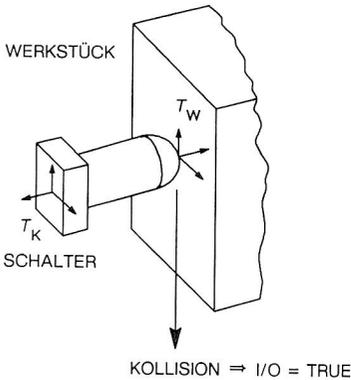
IDENT.	: SCHRAUB.xxx
NAME	: xxx
ELEMENTTYP	: SCHRAUBER.xxx
GEOMETRIE	: xxx.2D xxx.3D
KOPPLUNG	: IDENT. T_{KW} , TYP xxx
INPUT/OUTPUT	: 1 DIGIT
SCHRAUBEN	: ABC
TCP-VERSCHIEBUNG	: TCP
WERKZEUGEINGRIFF	: T_w
AXIALKRAFT	: F
ANZUGSART	: xxx
ANZUGSMOMENT	: M
FÜGEHILFEN	: PASSIV

Bild 66: Modell eines Schraubwerkzeuges

Sensoren und Schalter sind Komponenten in Montagesystemen, die durch ihre physikalische Funktion Informationen aus dem Montageprozeß holen und für eine Verarbeitung zur Verfügung stellen. Die Funktion einiger Sensoren kann speziell bei Montageverrichtungen gut durch ein geometrisches Ersatzmodell simuliert werden. Eine Lichtschranke beispielsweise läßt sich durch einen dünnen Zylinder abbilden (Bild 78). Bei der späteren Simulation kann eine Kollision mit diesem 'körperlichen Sensorstrahl' als Unterbrechung interpretiert und verarbeitet werden. Ebenso können belie-

bige Schalter bei Berührung (= Kollision von Volumenmodellen) Signale an den Simulationsprozeß senden (Bild 67).

Der Meßbereich von Sensoren, wie CCD-Kamera oder Laserscanner, kann als geometrischer Körper gekennzeichnet werden (Kegel, Kugel, Quader, Zylinder); entscheidend ist dabei die im Modell hinterlegte Funktion -z.B. bei Kollision: Szene aufnehmen. Diese Funktion bewirkt, daß die errechenbaren Sollgrößen -z.B. Höhenhistogramm des im Bereich befindlichen Teils- aufgenommen und im Zellenprogramm hinterlegt werden (siehe Kapitel 5.3.3). Sie dienen dem Sensor zum Soll-Ist-Vergleich bei der Mustererkennung im realen Produktionsprozeß.



IDENT. : *SENSOR.xxx*
 NAME : *xxx*
 ELEMENTTYP : *SCHALTER BINÄR*
 GEOMETRIE : *xxx.2D*
 xxx.3D
 INPUT/OUTPUT : *1 DIGIT*
 EINGRIFF : T_W
 AUSLÖSEKRAFT: *F*

Bild 67: Modell eines Schalters

5.2.3 Modell der Zelle

Aus den einzelnen Komponenten und Produktbaugruppen lassen sich komplexe Strukturen, sprich Zellen, generieren. Solche Modellzellen sind nach dem bekannten Vater-Sohn Prinzip aufgebaut. Dieses Prinzip repräsentiert im wesentlichen die kinematische Struktur und Topologie einer Montagezelle. Nur durch diese Struktur ist es möglich, einzelne Frames (Roboter, NC-Achsen) zu selektieren und individuell zu manipulieren, ohne dabei den logischen Aufbau der Zelle zu zerstören. Konsequenterweise lässt sich die-

ses Prinzip noch erweitern, indem Zellen selbst wieder als Einzelframes betrachtet und zu übergeordneten Systemstrukturen zusammengefaßt werden. Damit lassen sich dann auch ganze Fertigungssysteme bzw. Fabrikhallen konfigurieren und simulieren. Die Realisierung eines solchen frameorientierten Vater-Sohn Konzeptes wird im folgenden Kapitel behandelt. Zunächst soll aber am Beispiel einer einfachen Bestückzelle (siehe Bild 68) der Aufbau und Informationsgehalt einer Zelle beschrieben werden.

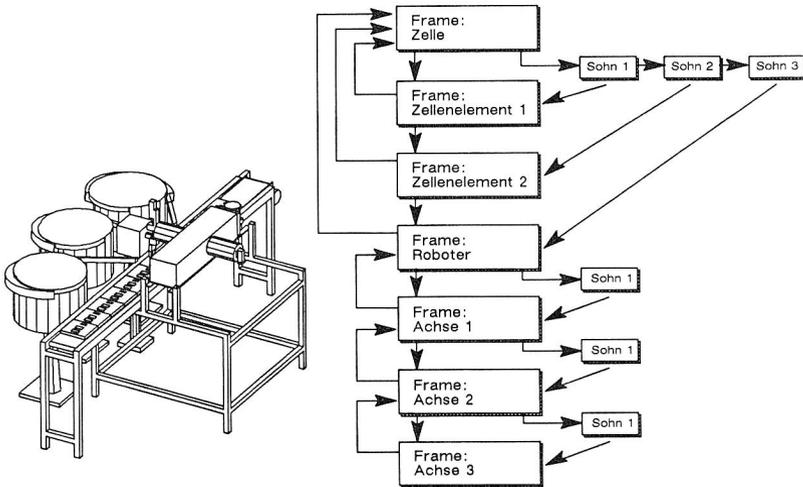


Bild 68: Modell einer Bestückzelle

Jede Zelle enthält, wie alle anderen Objekte auch, eine eigene Identität, gegeben durch Identifikationsschlüssel (Nummerungssystem) und einen Namen. Gleichzeitig werden Informationen über den Typ der Zelle -SMD-Bestücker, Entgratzelle, usw.- und das bearbeitete Produkt bzw. die bearbeiteten Produkte gegeben.

Die Bezeichnung und Lage der Söhne in der Zelle sind wesentlicher Bestandteil des Datenmodells. Weitere Informationen sind:

- Teilequellen und Teilesenken
- Hauptabmessungen
- Zellensteuerungstyp (z.B. PC mit FMC-Software /65/)
- Art und Anzahl der zu fertigenden Produktvarianten
- Schnittstellen zu übergeordneten Systemen.

Zellen können zwar, wie beschrieben, selbst wieder zu größeren Systemen zusammengefaßt werden. Bei der Programmierung und Simulation zeichnet sich eine Zelle gegenüber allen anderen Objekten jedoch dadurch aus, daß sie selbst statisch ist und damit das einzig wirklich feste Koordinaten Bezugssystem besitzt. Dieses Bezugssystem wird als Umweltkoordinatensystem betrachtet. Die Lage aller 'Söhne' d.h. aller Objekte in einer Zelle lassen sich aus Transformationen bezogen auf dieses Umweltsystem berechnen. Mit diesem Koordinatensystem werden somit auch die absoluten Anfahrpositionen der Bewegungssysteme festgelegt.

5.2.4 Framekonzept als Datenkapsel

Im vorangegangenen Kapitel wurde grob beschrieben, welche Informationsgehalt das Datenmodell für eine rechnergestützte Montageplanung besitzen muß. Hier geht es darum zu zeigen, wie dieses Datenmodell rechentechnisch gefaßt und gespeichert werden kann. Aufgrund der Fülle von Daten drängt sich die Forderung nach Einsatz einer Datenbank zur Verwaltung der Modellelemente auf. In /55/ wird gezeigt, wie sich diese Forderung auf Basis eines relationalen Schemas verwirklichen läßt.

Der Einsatz einer Datenbank dient der Integration von Daten; sie ermöglicht die Verwendung von Daten durch mehrere Anwendungsprogramme. Das vorgestellte Konzept zur Feinplanung und Programmierung von Montagezellen ist nur eine der Anwendungen zur Entwicklung und Herstellung von Produkten, die auf gemeinsamen Datenbestand zugreifen muß. Es sollen auch Konstruktion, Vertrieb, Lager, Einkauf, Fertigung an eine Datenbank angeschlossen sein. Für die hier vorgestellte Anwendung zur Planung und Programmie-

rung bietet die Datenbank folgende Bibliotheken an:

- Bibliothek von Produkten, deren Varianten und Versionen sowie zugehörigen Arbeitsplänen
- Bibliothek von Montagekomponenten, deren Geometrien und technologischen Eigenschaften
- Projektbibliothek zur Verwaltung von Montagezellen, deren Zeichnungen, Stücklisten und zugehörigen Steuerprogrammen.

Die Verwaltung der Datenbank zeigt allerdings das generelle Problem von Verwaltungssystemen auf. Zu viele Zugriffe und zu große Datenmengen machen ein schnelles interaktives Arbeiten -wie es etwa die offline Programmierung erfordert- auf der Datenbank unmöglich. Die Datenbank kann daher nur als abgekoppeltes Verwaltungssystem dienen. Man bedient sich daher eines sogenannten "views" /66/ auf das Datenmodell. Ein "view" extrahiert die für die Anweisung notwendigen Daten aus der Datenbank und bildet diese auf eine zur interaktiven Bearbeitung geeignete Struktur ab (Bild 69).

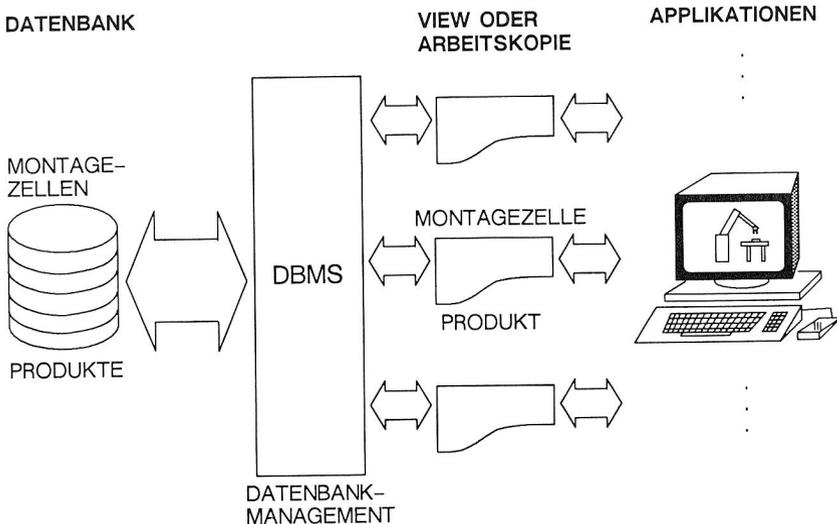


Bild 69: View auf das Modell einer Zelle in der Datenbank

Dazu wird zunächst eine Trennung von grafischen und technologischen (nicht grafischen) Daten vorgenommen. Zur schnellen Bildmanipulation eignet sich die Abbildung der Geometrie auf dem grafischen Standard PHIGS (Programmers Hierarchical Interactive Graphics System) oder ähnlich wie GKS (Graphical Kernel System) oder GMR 3D (Graphic Metafile Ressource, Fa. Apollo) sich zur hardwareunterstützten Verarbeitung grafischer Objekte -wie Linie, Kurve im Raum- eignet. PHIGS-Modelle können im Dreidimensionalen durch einfache Zoom-, Rotations- oder Verschiebefunktionen manipuliert werden. Farbschattierung, Flächenüberschneidungen, Z-Buffering und Clipping erlauben eine gute visuelle Darstellung und Untersuchung von Objekten am Bildschirm.

Bei den nichtgrafischen Informationen der Modelle kann auf keine Standardstrukturen zurückgegriffen werden. Denkbar wären zwar Datenstrukturen wie sie in framebasierten Expertensystemen oder in Systemen mit objektorientiertem Ansatz vorzufinden sind, doch erweisen sich solche Systeme, ähnlich wie die relationale Datenbank, als zu langsam, um den zeitkritischen Anforderungen eines grafisch interaktiven Planungssystems zu genügen.

Der hier beschriebene Ansatz sieht deshalb die Verwendung von sog. Datenkapseln /67/ vor, die intern als Recordstrukturen mit doppelt verzweigten Listen realisierbar sind. Die hierarchischen Abhängigkeiten von Elementen in einer Zelle werden durch diese Verzweigung abgebildet. Die Datenstruktur ist folgendermaßen aufgebaut:

- Jedem Zellenelement wird ein Record zugewiesen, der als Frame bezeichnet wird und alle elementtypischen Informationen birgt.
- Die Frames werden in einer baumförmigen Struktur und zusätzlich in Form einer Liste angeordnet. Dabei bilden die Frames die Knoten des Baumes.
- Die hierarchische Abhängigkeit wird im Aufbau des Baumes nach dem Vater-Sohn Prinzip realisiert.
- Die Wurzel eines Baumes ist das Frame 'Zelle'.

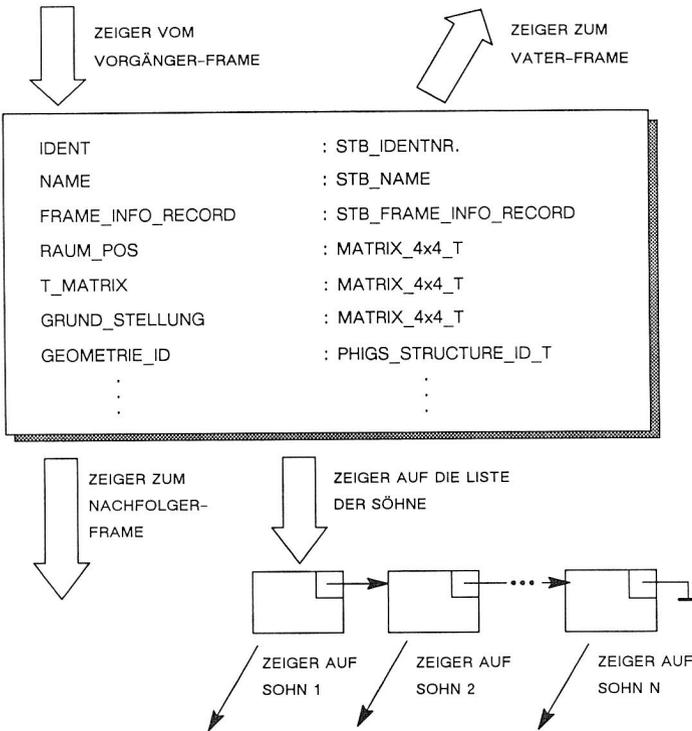


Bild 70: Aufbau eines Frames

Wichtigstes Kennzeichen einer Datenkapsel ist, daß Zugriffe auf Modellinformationen nur über spezielle Zugriffsroutinen möglich sind. Dadurch wird sichergestellt, daß nur erlaubte Operationen zu Veränderungen am Modell führen. Diese Routinen lassen sich in drei Kategorien untergliedern:

1. Aufbau der Datenstruktur
2. Abfragen und Verändern von Daten in der Struktur
3. Löschen der Datenstruktur.

Zur ersten Kategorie gehören Programme, die aus der Datenbank alle relevanten Informationen einholen und die beschriebene Datenstruktur generieren; außerdem werden die gleichen Programme für Operationen beim Simulieren und Programmieren benötigt, wie etwa beim Greifen eines Teils -hier muß nämlich das gegriffene Objekt an die zugreifende Struktur als Sohn angebinden werden. Das Greifen erfordert zuvor auch eine Aktion durch Routinen der dritten Kategorie, denn das gegriffene Objekt muß aus seiner vorherigen Position im 'Zellenbaum' gelöscht werden.

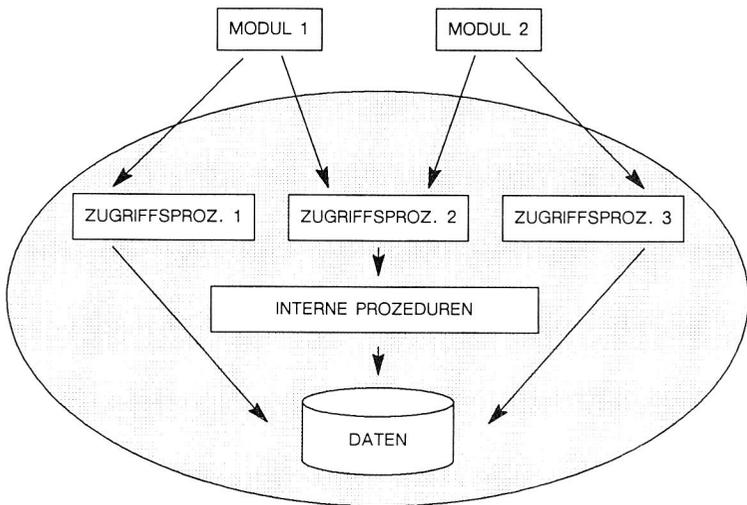


Bild 71: Aufbau einer Datenkapsel

Zugriffe zum Abfragen oder Verändern von Einträgen in der Datenstruktur -Kategorie 2- erlauben es, einzelne Frames zu selektieren, zu identifizieren, deren Position zu verändern oder verschiedene Einträge in der Liste der Eigenschaften vorzunehmen. Sollen gleichzeitig Zugriffe auf mehrere Frames sowie deren Söhne, Enkel und Urenkel durchgeführt werden -z.B. Selektieren eines Roboters- so werden die einzelnen Frames nach dem Prinzip 'depth-first' (dt. zuerst in die Tiefe) abgearbeitet.

5.3 Emulation von Steuerungsfunktionen

Um Montagesysteme zu simulieren, muß man ein mathematisches Modell der Steuerungsfunktionen definieren. Diese Form des Modellierens, die man in der Regel durch Erstellung von Programmen vollzieht, wird als Emulation bezeichnet.

Eine besondere Bedeutung kommt hierbei den Bewegungssystemen zu, wie Robotern, Achsen oder Transporteinrichtungen. Für die Bewegung von Mehrachssystemen müssen umfangreiche Rechenprogramme zur Hintransformation, Rücktransformation /68/ oder Bahnplanung entwickelt und implementiert werden. Häufig ist die Emulation für jeden Typ von Bewegungssystem individuell zu erstellen. Zwar gibt es Ansätze /69/, die allgemeine numerische Lösungen verfolgen, beispielsweise für die Rücktransformation bei Robotern, doch reichen weder Genauigkeiten noch Rechengeschwindigkeiten aus, um hier akzeptable Ergebnisse zu erzielen.

Zudem stimmen die Bewegungen von realen Geräten aufgrund unterschiedlicher Steuerstrategien nicht mit den numerisch ermittelten Bewegungen überein.

Ein praktikabler Ansatz ist es, Klassen von Bewegungssystemen zu formulieren und Module mit passenden Berechnungsprogrammen für jede Klasse zu implementieren. Auf diese Weise erhält man einen modularen erweiterbaren Programmbaustein, der in ein System zur Planung und Programmierung von Montagesystemen eingebunden werden kann (Bild 72). Vom Prinzip ist dieser generalisierte Berechnungsmodul nichts anderes als eine Programmbibliothek.

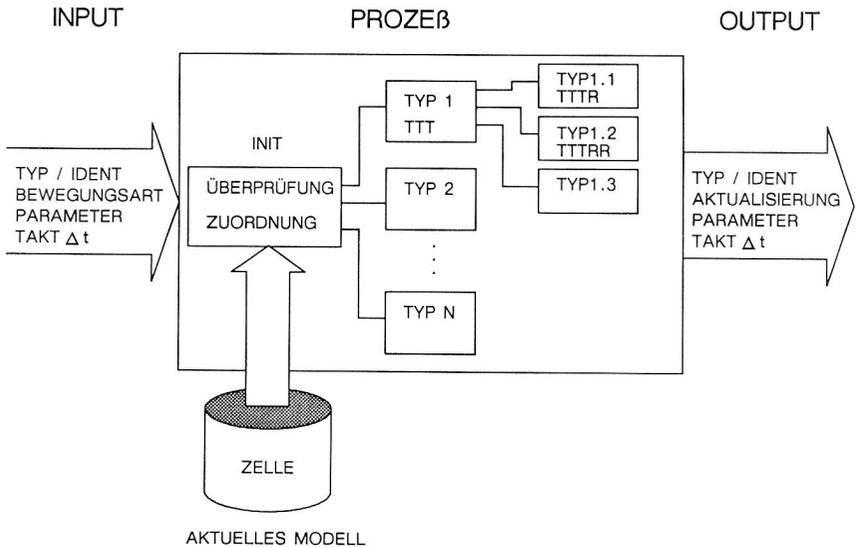


Bild 72: Baustein zur Berechnung von Bewegungen beliebiger kinematischer Systeme

Außer den kinematischen Berechnungen soll es auch möglich sein, dynamisch, d.h. massen-, kraft- und momentabhängige Simulationen durchzuführen, um genauere Ergebnisse ermitteln zu können. Dazu dienen Rechenprogramme zur exakten dynamischen Berechnung, wie in /70/ beschrieben.

Neben Bibliotheksprogrammen zur Berechnung kinematischer und dynamischer Zustände müssen auch steuerungsspezifische Programme vorhanden sein. Steuerungsspezifische Programme berechnen Bewegungsarten, wie PTP, LINEARES, ZIRKULARES Fahren, behandeln Mehrdeutigkeiten bei koordinierten Mehrachs-bewegungen und ermitteln das Bewegungsverhalten von Sonderbewegungen, wie Pendeln, Überschleifen, Stottern usw..

Außer den Steuerungsfunktionen von Bewegungseinrichtungen gibt es in Montagezellen noch andere zu steuernde Komponenten. Die

Emulation von Signalgebern, Kontakten, Schaltern oder komplexen Sensorsystemen ist zur vollständigen Simulation von Montageeinheiten unerlässlich. Es muß auch möglich sein, den Gesamttablauf - der aus vielen Einzelprozessen besteht- zu beschreiben und zu simulieren.

In den beiden folgenden Kapiteln werden Lösungen für die Detailprobleme bei der Bewegungs- und Ablaufsimulation in Montagezellen beschrieben und an Beispielen demonstriert.

5.3.1. Inverse Kinematik mehrachsiger Bewegungssysteme

Zur Durchführung von koordinierten Bewegungen mehrachsiger Systeme ist es notwendig, das Problem der Rücktransformation (auch inverse Kinematik) zu beherrschen. Rücktransformation bedeutet die Umrechnung von in Weltkoordinaten (siehe 5.2.2) gegebenen Punkten und deren Bewegungen im Raum in aktuelle Achsstellungen und Antriebsmomente bzw. Beschleunigungen des Bewegungssystems.

Beispiel:

Ein Teil soll von einem Roboter mit konstanter Geschwindigkeit entlang einer Geraden bewegt werden.

Zur Lösung der Aufgabe ist es notwendig, zu diskreten Zeitpunkten Positionen auf der Bewegungsgeraden in Roboterkoordinaten zu errechnen und gleichzeitig die Geschwindigkeitsprofile der Achsen zur Aufrechterhaltung der konstanten Bahngeschwindigkeit anzugeben. Bei realen Robotersystemen werden die Geschwindigkeiten der Roboterachsen geregelt. Beim Simulieren sollen physikalische Einflüsse (Reibung, Gravitation) vernachlässigt werden und Sollvorgaben als Ist-Werte übernommen werden. Da die inverse Kinematik die Grundlage für Bewegungsplanung, -durchführung und -optimierung (Kap. 5.4) ist, wird diese im folgenden anhand des Robotertyps RRR3A beispielhaft vorgerechnet. Die kinematische Beschreibung des Typs RRR3A erfolgt gemäß den technologischen Modells (siehe Kapitel 5.2.2.) und den Gleichungen (5.1) bis (5.11).

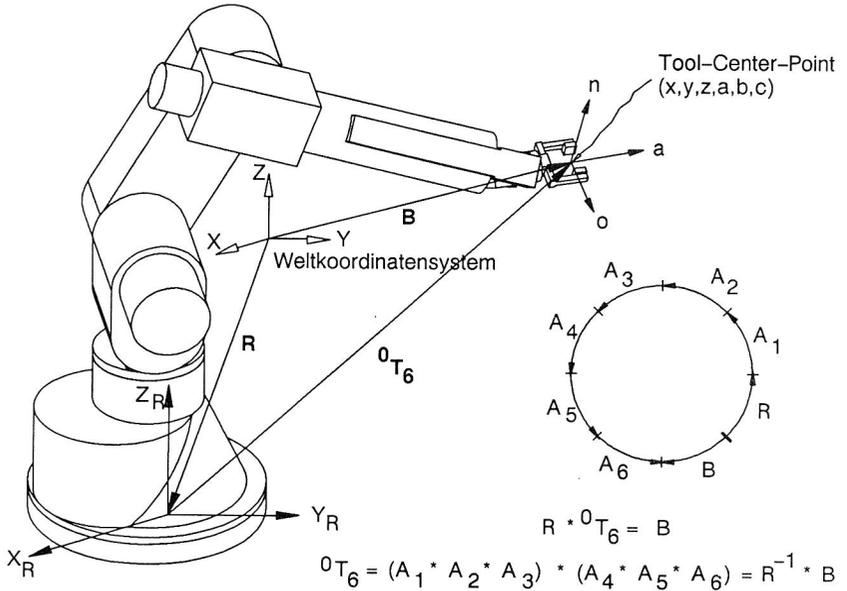


Bild 73: Rücktransformation eines Gelenkroboters.

Der anzufahrende freie Frame (=Raumpunkt) liefert die Ortskoordinaten x , y und z sowie die Orientierungsvektoren \underline{n} , \underline{o} und \underline{a} gemäß Bild 54.

Zur Bestimmung der Achswinkel θ_1 bis θ_6 kann man in zwei Teilschritten vorgehen. Man berechnet zunächst die Lage des Handpunktes x_h , y_h , z_h (=Punkt dessen Position bei Veränderung der Achsen 4, 5 und 6 fix bleibt) und daraus die ersten drei Achswinkel /68/. Im zweiten Schritt aus aT_6 werden dann θ_4 , θ_5 und θ_6 errechnet.

$$x_h = x - a_x \quad (5.12)$$

$$y_h = y - a_y \quad (5.13)$$

$$z_h = z - a_z \quad (5.14)$$

Dabei sind x , y , z , \underline{n} , \underline{o} , \underline{a} die auf den Roboterursprung bezogenen Raumkoordinaten. Lage und Orientierung des Handpunktes erhält man aus Gleichung 5.10:

$$x_h = aac_1c_2 + ddc_1s_{23} = c_1 (aac_2 + dds_{23}) \quad (5.15)$$

$$y_h = aac_1c_2 + dds_1s_{23} = s_1 (aac_2 + dds_{23}) \quad (5.16)$$

$$z_h = ddc_{23} + aas_2 \quad (5.17)$$

x_h , y_h und z_h sind aus 5.12 - 5.14 bekannt. Man hat damit drei Gleichungen für drei Unbekannte θ_1 , θ_2 und θ_3 . Zur Berechnung von θ_1 bildet man den Quotienten der Gleichungen 5.15 und 5.16:

$$\frac{y_h}{x_h} = \frac{s_1 (aac_2 + dds_{23})}{c_h (aac_2 + dds_{23})} = \frac{s_1}{c_1} \quad (5.18)$$

Daraus folgt:

$$\theta_1 = \arctan2 (y_h, x_h) \text{ bzw.} \quad (5.19)$$

$$\theta_1 = \arctan2 (y_h, x_h) + k * \Pi \quad (5.20)$$

(mit $k \in \mathbb{Z}$)

Durch Multiplikation von (5.15) mit c_1 und (5.16) mit s_1 und seitenweiser Addition der Gleichungen erhält man:

$$dd s_{23} + aa c_2 = c_1 x_h + s_1 y_h = A \quad (5.21)$$

Mit Gleichung 5.17 erhält man das neue Gleichungssystem zur Errechnung der Winkel θ_2 und θ_3 :

$$\text{aus (5.21)} \quad dd s_{23} = A - aac_2 \quad (5.22)$$

$$\text{aus (5.17)} \quad dd c_{23} = z_h - aas_2 \quad (5.23)$$

Quadrieren der Gleichungen und seitenweises Addieren liefern:

$$dd^2 = A^2 + z_h^2 + aa^2 - 2aa (A c_2 + z_h s_2) \quad (5.24)$$

durch Umformen erhält man:

$$B \cos \theta_2 + C \sin \theta_2 = D \quad (5.25)$$

mit

$$B = \frac{A}{\sqrt{A^2 + z_h^2}} \quad (5.26)$$

$$C = \frac{z_h}{\sqrt{A^2 + z_h^2}} \quad (5.27)$$

$$D = \frac{A^2 + z_h^2 + aa^2 + dd^2}{2aa \sqrt{A^2 + z_h^2}} \quad (5.28)$$

θ_2 ergibt sich damit zu:

$$\theta_2 = \arctan 2 \left[\frac{A}{z_h} \right] \pm \arctan 2 \left[\frac{\sqrt{1 - D^2}}{D} \right] \quad (5.29)$$

Durch Quadrieren von (5.17) und (5.21) und anschließendem seitenweisen Addieren erhält man:

$$\begin{aligned} s_3 = (s_{23}c_2 - c_{23}s_2) &= \frac{A^2 + z_h^2 - dd^2 - aa^2}{2 aa dd} \quad (5.30) \\ &= E \end{aligned}$$

Da die Arcustangensfunktion über den gesamten Definitionsbereich stetig ist, ist es sinnvoll auch in (5.30) zur Ermittlung von θ_3 den Sinus durch den Tangens zu ersetzen. Mit

$$\tan \theta_3 = \frac{\sin \theta_3}{\sqrt{1 - \sin^2 \theta_3}} \quad (5.31)$$

ergibt sich schließlich für den letzten der Handwinkel:

$$\theta_3 = \pm \arctan 2 \left[\frac{E}{\sqrt{1 - E^2}} \right] \quad (5.32)$$

Die Lösung ist nicht definiert, wenn E (und damit der Sinus θ_3) größer als 1 ist, d.h.

$$\frac{A^2 + z_h^2 - aa^2 - dd^2}{2 aa dd} > 1 \quad (5.33)$$

$$A^2 + z_h^2 > aa^2 + 2 aa dd + dd^2 \quad (5.34)$$

Nachdem die Armwinkel eins bis drei bestimmt sind, berechnet man die Elemente der Transformationsmatrix 0T_h . Es gilt:

$${}^hT_6 = ({}^0T_h)^{-1} {}^0T_6 \quad (5.35)$$

Man multipliziert die Inverse der Rotationsmatrix von 0T_h (wegen der Orthogonalität der Rotationsmatrizen ist die Inverse gleich der Transponierten) von links an die Orientierungsmatrix und erhält die Rotationsmatrix von hT_6 .

$$\begin{bmatrix} nw_x & ow_x & aw_x \\ nw_y & ow_y & aw_y \\ nw_z & ow_z & aw_z \end{bmatrix} = \begin{bmatrix} c_1 c_{23} & s_1 c_{23} & -s_{23} \\ -s_1 & c_1 & 0 \\ c_1 s_{23} & s_1 s_{23} & c_{23} \end{bmatrix} \begin{bmatrix} n_x & o_x & a_x \\ n_y & o_y & a_y \\ n_z & o_z & a_z \end{bmatrix} \quad (5.36)$$

Nun können die restlichen drei Winkel (θ_4 , θ_5 und θ_6) berechnet werden. Es gilt (vgl. 5.11):

$$aw_x = c_4 s_5 \quad (5.37)$$

$$aw_y = s_4 s_5 \quad (5.38)$$

Um das Auftreten des bereits erwähnten $\alpha 5$ -Problems zu verhindern, wird geprüft, ob $aw_x^2 + aw_y^2 < 0.00001$. Bei Unterschreitung des Grenzwertes wird $\theta_5 = 0$ gesetzt und Achse 4 bzw. θ_4 festgehalten; damit ist die Lösung wieder eindeutig.

Andernfalls bestimmt man:

$$\theta_4 = \arctan2 (aw_y, aw_x) \quad (5.39)$$

oder

$$\theta_4 = \arctan2 (aw_y, aw_x) + k * \pi \quad (5.40)$$

mit $k \in \mathbb{Z}$

Mit $aw_z = c_5$ erhält man

$$\theta_5 = \pm \arctan2((c_4aw_y, s_4aw_y), aw_z) \quad (5.41)$$

Nach der Berechnung von θ_5 kann auch θ_4 ermittelt werden. Nach 5.11 folgt weiterhin

$$nw_z = -s_5c_6 \quad (5.42)$$

$$ow_z = s_5s_6 \quad (5.43)$$

und damit

$$\theta_6 = \arctan2(s_5ow_z, -s_5nw_z) \quad (5.44)$$

Die Mehrdeutigkeiten der Lösungen von θ_1 bis θ_6 lassen sich dadurch einschränken, daß man die Verfahrbereichsgrenzen des speziellen Roboters abprüft. In der Regel lassen die rotatorischen Achsen eines Gelenkroboters maximal eine Vollkreisbewegung zu, also $-\pi < \theta < \pi$ bzw. $0 < \theta < 2\pi$.

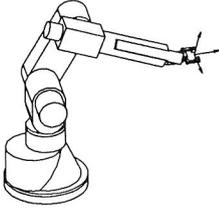
Weitere Mehrdeutigkeiten werden beim Programmieren realer Geräte durch das Abprüfen des 'Winkelstatus' ausgeschlossen. So werden beim Teachen von Robotern nicht nur Lage und Orientierung des angefahrenen Punktes gespeichert, sondern auch Informationen über die Art der Stellung der Roboterachsen zueinander.

Für den vorgestellten Robotertyp RRR3A kann die Roboterstellung, falls wie oben dargelegt jede Achse zwischen $-\pi$ und $+\pi$ dreht, durch einen 3Bit Winkelstatus eindeutig beschrieben werden. Das erste Bit gibt Auskunft, ob Grund- oder Überkopfposition (siehe Bild 74) eingenommen werden soll. Die Regel zur Auswahl der Lösung von θ_1 lautet hier:

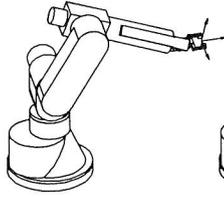
In der Grundposition müssen die Vorzeichen von θ_1 und y_h (aus 5.13) übereinstimmen (Bit 1 = 0), in der Überkopfposition sind die Vorzeichen von θ_1 und y_h entgegengesetzt (Bit 1 = 1).

Das zweite Bit legt fest ob θ_3 positiv oder negativ ist und läßt nach Ermittlung von θ_3 auch die Auswahl des entsprechenden Armwinkels θ_2 zu (siehe Bild 74 obere/untere Position). Das dritte Bit legt den Winkel θ_5 und damit auch θ_4 und θ_6 fest.

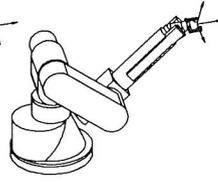
Raumposition des TCP : $x = 947$ $y = 362$ $z = 788$ $a = 138$ $b = -11$ $c = 127$



Überkopfstellung
 $\theta_2 + \theta_3$ oben
 $\theta_5 > 0$



Grundstellung
 $\theta_2 + \theta_3$ oben
 $\theta_5 < 0$



Grundstellung
 $\theta_2 + \theta_3$ unten
 $\theta_5 > 0$

Achse 1 :	$\theta_1 = -160$	$\theta_1 = 20$	$\theta_1 = 20$
Achse 2 :	$\theta_2 = -30$	$\theta_2 = 30$	$\theta_2 = 127$
Achse 3 :	$\theta_3 = -80$	$\theta_3 = 80$	$\theta_3 = -80$
Achse 4 :	$\theta_4 = -150$	$\theta_4 = -150$	$\theta_4 = 10$
Achse 5 :	$\theta_5 = 20$	$\theta_5 = -20$	$\theta_5 = 80$
Achse 6 :	$\theta_6 = 40$	$\theta_6 = 220$	$\theta_6 = 67$

Bild 74: Anfahren einer Position in verschiedenen Stellungen des Roboters

Die Wahl des richtigen Winkelstatus ist bei der Offline-Programmierung ein Problem, das zur Lösung einer Strategie bedarf. Bei der Gestaltung von Montagezellen können aus dem Modelliersystem zwar Lage und Orientierung eines anzufahrenden Teils ermittelt werden. Die Achskonfiguration des Roboters und die Bewegungsbahn muß aber geplant werden (siehe Kapitel 5.4.2).

5.3.2 Bewegungen mehrachsiger Systeme

Mehrachsige Bewegungssysteme sind entsprechend der ihnen zur Verfügung stehenden Freiheitsgrade in der Lage, unterschiedliche Positionen im Raum einzunehmen. Im vorausgegangenen Kapitel wurde gezeigt, wie die in kartesischen Weltkoordinaten anzufahrende Stellung in Gelenkwerte -also Roboterkoordinaten- umgerechnet werden kann. Hier wird nun gezeigt, auf welchen Bahnen mehrachsige Systeme von einer Stellung A zu einer Stellung B gelangen. Grundsätzlich unterscheidet man folgende Bewegungsarten:

- Punkt zu Punkt Bewegen (PTP)
- inkrementales Bewegen (MP)
- lineares Bewegen (LIN)
- zirkulares Bewegen (ZR)
- sonstiges interpolierendes Bewegen (CP)

Technologieabhängige Sonderbewegungen, die sich aus den eben genannten Bewegungsgrundformen zusammensetzen, werden in Kapitel 5.4.2 behandelt.

Die einfachste Bewegungsform ist die PTP-Bewegung. Hierbei werden alte und neue Stellung des Bewegungssystems verglichen und danach alle Achsen gleichzeitig, aber unabhängig voneinander um den errechneten Differenzbetrag bewegt. Es ergibt sich beim realen Gerät ein nicht vorausberechneter Bahnverlauf mit variierenden Beschleunigungen und Geschwindigkeiten im Geräteendpunkt (TCP). Beim Bewegen eines Modelles ist jedoch zu jedem Zeitabschnitt Δt das Bewegungssystem vollständig berechnet. Es können für kritische Bewegungsabschnitte im PTP-Bahnverlauf mit Hilfe angeschlossener Programme Festigkeitsrechnungen, auftretende Kräfte und Momente im TCP, am Werkstück oder in den Gelenken ermittelt werden.

Da bei PTP-Bewegungen aufgrund unterschiedlicher Wege und unterschiedlicher Antriebsgeschwindigkeiten die Achsen die Zielposition nicht gleichzeitig erreichen, ergeben sich in der Regel

Bewegungsbahnen mit starken Richtungsänderungen und damit unerwünschten Beschleunigungsspitzen im TCP. Deshalb wird durch unterschiedliche Methoden versucht, die PTP - Bewegung zu optimieren (siehe 5.4.2).

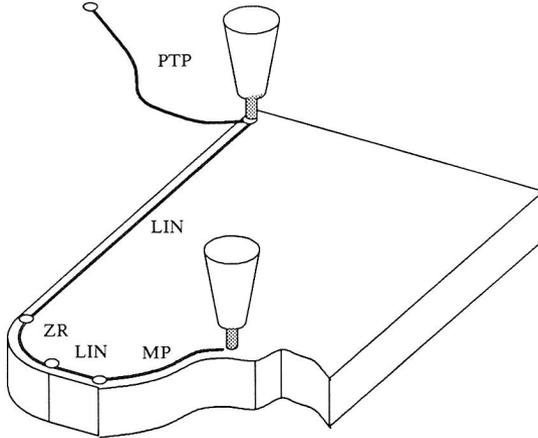


Bild 75: Bewegungsarten von Robotern (vgl. Bild 60)

Die PTP - Bewegung ist die am häufigsten verwendete Bewegungsart, da sie zum einen wenig Rechenaufwand bedarf und zum anderen die Bewegungsmöglichkeiten der einzelnen Achsen voll nutzen kann. Für die Programmierarbeit des Montageplaners bedeutet dies konkret, daß sich mit PTP - Bewegungen die kürzesten Bewegungszeiten erreichen lassen. Beim Einsatz von Robotern in der Montage stellen kurze Taktzeiten ein großes Problem dar. Es wird daher häufig beobachtet, daß aufgrund von Taktzeitproblemen trotz kritischer Bewegungsoperationen PTP-Bewegungen koordinierten aber langsamen Bahnbewegungen vorgezogen werden.

Die PTP-Bewegung läßt sich simulieren, indem für jede Achse ein Beschleunigungs- bzw. Geschwindigkeitsprofil vorgegeben wird (siehe Bild 76). Aus Geschwindigkeitsprofil und zurückzulegendem Weg $\Delta\theta$ bzw. Δd läßt sich die Bewegungsdauer berechnen und für die grafische Simulation in kleine Zeitabschnitte unterteilen. Die Annahme von Geschwindigkeitsprofilen (siehe Bild 76) ist eine

Modellvereinfachung. Bei realen Systemen ergibt sich der Bahnverlauf aufgrund der Antriebsmomente der Achsmotoren sowie den dynamischen Einflüssen (Kräfte, Momente) angrenzender Achsen bzw. Achssysteme. Die Bahnverläufe von Modell und realem Gerät können aufgrund dieses Sachverhaltes, insbesondere bei hohen Geschwindigkeiten, erhebliche Abweichungen aufweisen. Größere Genauigkeiten bei der Simulation können nur durch ein besseres Modell, das dynamische Größen einschließt /70/, erreicht werden. Erweiterte Modelle sind allerdings zum einen für jedes Bewegungssystem individuell zu erstellen. Zum anderen machen lange Rechenzeiten bei komplexen Modellen die grafisch interaktive Bewegungssimulation bei heutiger Rechnerleistung unmöglich.

Die inkrementelle Bewegung von Robotern entspricht einer Direktsteuerung der Bewegungsachsen durch Vorgabe kleiner Bewegungsschritte in einem von der Steuerung vorgegebenem Takt. Dieser Takt ist im wesentlichen abhängig von der Datenübertragung, bei der die eingelernten Zwischenpositionen vom Datenspeicher der Steuerung direkt an die Antriebsregler weitergegeben werden. Für die Montage ist die MP-Bewegung nur im Zusammenspiel mit Sensoren interessant, wenn nämlich im Montageprozeß -z.B. beim Auftragen von Klebstoff auf eine beliebige Kontur- die Positionswerte vom Sensor ermittelt und direkt an die Bewegungssteuerung weitergegeben werden. Die MP-Bewegung läßt sich einfach simulieren, wenn man den Verarbeitungstakt der Steuerung kennt.

Für die Montageprogrammierung ist die lineare Bewegung die wichtigste Bewegungsart. Zum einen ergibt die Bewegung entlang einer Geraden kürzeste Wege, zum anderen sind lineare Bewegungen für den Programmierer einfach zu verstehen und zu überwachen. Für den Montageprozeß selbst, z.B. bei Fügeoperationen, ist die Linearbewegung technologisch unerläßlich. Andererseits stellt diese Bewegungsart hohe Anforderungen an die Steuerungstechnik. Der Interpolator inkrementiert die Bewegungsbahn und muß für jede inkrementale Position auf der Bewegungsgeraden eine Rücktransformation rechnen. Vergleicht man die Rechenzeiten von PTP und LIN bei der Berechnung einer Bewegung zwischen den Posi-

tionen A und B, so ergibt sich in etwa ein Verhältnis von 1 : 100 /34/. Ein weiteres Problem ergibt sich, wenn -und das geschieht häufig bei 6-achsigen Gelenkrobotern- die Stellung des Roboters degeneriert (z.B. siehe Typ RRRA3 mit $\theta_5 = 0$). Diese Degenerationen lassen sich auch beim realen Gerät nicht voraussehen, deshalb bietet das Simulieren mit einem grafischen Modell den Vorteil, extreme Situationen bei Linearbewegungen ohne Schaden zu erkennen. Bahngeschwindigkeiten lassen sich am realen Gerät nicht abschätzen, da diese in komplexer Abhängigkeit zu den begrenzten Geschwindigkeiten der einzelnen Achsen stehen.

Die Interpolation selbst ist mathematisch einfach zu berechnen. Zunächst wird der Vektor \underline{a} und der Abstand $|\underline{a}|$ zwischen Position A und Position B ermittelt und mit Hilfe der daraus errechneten Verfahrszeiten und dem Interpolationstakt t_T inkrementiert.

$$\underline{a} = \underline{p}_B - \underline{p}_A = \begin{bmatrix} x_B \\ y_B \\ z_B \end{bmatrix} - \begin{bmatrix} x_A \\ y_A \\ z_A \end{bmatrix} \quad (5.45)$$

$$|\underline{a}| = \sqrt{(x_B - x_A)^2 + (y_B - y_A)^2 + (z_B - z_A)^2} \quad (5.46)$$

Die Zahl der Bewegungsinkremente n kann man beispielsweise aus dem Verhältnis von Gesamtverfahrzeit t_{LIN} (siehe 5.50) und Berechnungstakt t_T ermitteln:

$$n = \frac{t_{LIN}}{t_T} \quad (5.47)$$

somit gilt für die aktuelle Zwischenposition auf der Bewegungsgeraden:

$$\underline{p}_{\text{aktuell}} = \underline{p}_{\text{aktuell}-1} + \frac{1}{n} \underline{a} \quad (5.48)$$

Für eine lineare Bewegung mit konstanter Geschwindigkeit über den größten Teil der Geraden müssen auch die Geschwindigkeits- und Beschleunigungswerte der einzelnen Achsen überprüft und die

Überschreitung von Grenzen korregiert werden. Das Problem wird in der Literatur mehrfach behandelt (/68/, /71/). Die Simulation der Linearbewegung ist im Gegensatz zur Steuerung und Regelung des realen Gerätes einfach. Es genügt, die Bewegungsinckremente als PTP-Bahnen zu berechnen, um damit Überschreitung von Grenzgeschwindigkeiten in den Achsen zu erkennen und durch Reduzieren der Bahngeschwindigkeit zu vermeiden. Bei Linearrobotern (Typ TTT) sind die Weginkremente der Achsen $\Delta d_i = 1/n d_i$ konstant. Es ergibt sich somit bei konstanter Achsgeschwindigkeit auch eine konstante Bahngeschwindigkeit. Bei Robotern mit Drehgelenken wird dieser Zusammenhang schwierig und macht eine differentielle Rechnung wie bei /34/ notwendig. Für den gesamten Geschwindigkeitsverlauf einer gesteuerten Bahnbewegung kann in der Simulation wie beim Geschwindigkeitsprofil der Achsen, eine 'Geschwindigkeitsrampe' (Bild 76) angenommen werden. Dieses Geschwindigkeitsprofil entspricht weitestgehend dem realen Fahrverhalten und kann deshalb grundsätzlich bei allen Bewegungsarten, auch Zirkularbewegung und anderen CP-Bewegungen, in der Simulation eingesetzt werden.

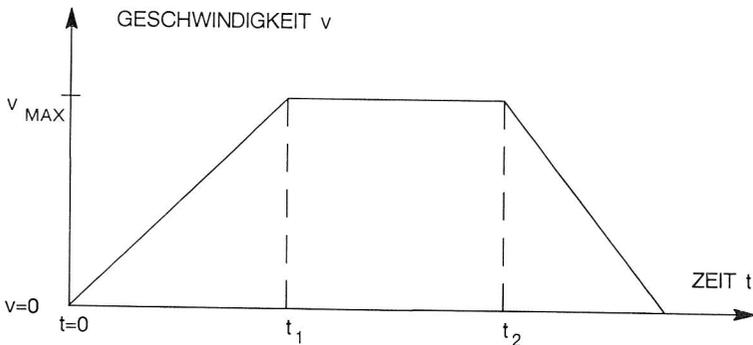


Bild 76: Geschwindigkeitsprofil zur Simulation von Bewegungen

Die Gesamtdauer einer Bewegung ermittelt sich zu

$$T = \frac{A S}{V_{\max}} + \frac{1}{2} \left[\frac{1}{c_1} + \frac{1}{c_2} \right] \quad (5.49)$$

Für die Linearbewegung gilt dann bei gegebener Steigung der Rampe und der Entfernung aus 5.46

$$t_{\text{LIN}} = \frac{|\underline{a}|}{V_{\text{LIN}}} + \frac{1}{2} \left[\frac{1}{c_1} + \frac{1}{c_2} \right] V_{\text{LIN}} \quad (5.50)$$

Im Fall, daß die Endgeschwindigkeit V_{max} nicht erreicht wird, (also $t_2 \neq t_1$) gilt als Mindestdauer nach /72/

$$T = \sqrt{\left[\frac{1}{c_1} + \frac{1}{c_2} \right] 2 \Delta s} \quad (5.51)$$

Das gleiche gilt für krummlinige Bahnkurven, wenn s als Bogenlänge und c_1 und c_2 als Tangentialbeschleunigungen bzw. Verzögerungen aufgefaßt werden.

Die Zirkularbewegung wird vom Prinzip ebenso wie die Linearbewegung gerechnet. Als Eingabegröße ist jedoch neben Anfangspunkt A und Endpunkt B des Kreises ein weiterer Stützpunkt C bzw. Mittelpunkt M und Radius r anzugeben. Im folgenden wird die Zirkularbewegung über drei Punkte A, B, C mit den Stützvektoren \underline{p}_a , \underline{p}_b und \underline{p}_c in Weltkoordinaten gerechnet.

$$\underline{p}_a = \begin{bmatrix} x_a \\ y_a \\ z_a \end{bmatrix} ; \underline{p}_b = \begin{bmatrix} x_b \\ y_b \\ z_b \end{bmatrix} ; \underline{p}_c = \begin{bmatrix} x_c \\ y_c \\ z_c \end{bmatrix} \quad (5.52)$$

Die von den drei Punkten A, B, C aufgespannte Ebene im Raum wird zunächst in die xy Ebene abgebildet, in der man über die explizit gegebene Kreisgleichung interpolieren kann.

Dazu bildet man zwei komplanare Vektoren $\underline{h}_a = \underline{p}_a - \underline{p}_c$ und $\underline{h}_b = \underline{p}_b - \underline{p}_c$ welche die Kreisebene aufspannen. Es wird eine Orthogonalmatrix gesucht, so daß \underline{h}_a und \underline{h}_b in die xy -Ebene abgebildet werden. Man erhält folgendes Gleichungssystem:

$$\begin{bmatrix} h_{ax} & h_{ay} & h_{az} \\ h_{ex} & h_{ey} & h_{ez} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad (5.53)$$

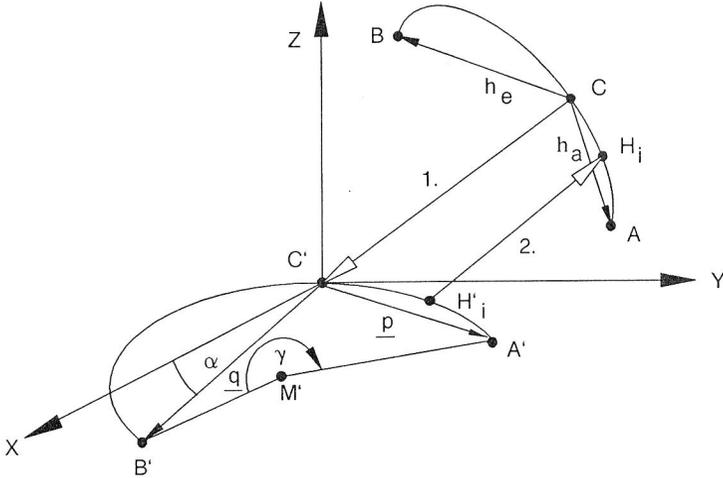


Bild 77: Zirkularinterpolation durch Abbildung der Kreisbahn in der Ebene

Nach Lösen des Gleichungssystems durch freie Wahl von $x_3 = 1$ und Orthonormalisierung der Abbildung nach dem Schmidt'schen Verfahren /73/, ergibt sich die Abbildung der drei Kreispunkte in xy -Ebene durch die Translation entlang \underline{p}_c in den Ursprung und die Rotation mit der Rotationsmatrix \underline{Q} :

$$\underline{Q} = \begin{bmatrix} \frac{x_2^2 + 1}{u} & \frac{-x_1 \cdot x_2}{u} & \frac{-x_1}{u} \\ 0 & \frac{1}{v} & \frac{-x_2}{v} \\ \frac{x_1}{w} & \frac{x_2}{w} & \frac{1}{w} \end{bmatrix} \quad (5.54)$$

$$\text{mit } x_1 = \frac{h_{ay} h_{ex} - h_{az} h_{ey}}{h_{ax} h_{ey} - h_{ay} h_{ex}} \quad (5.55)$$

$$x_2 = \frac{h_{az} h_{ex} - h_{ax} h_{ez}}{h_{ax} h_{ey} - h_{ay} h_{ex}} \quad (5.56)$$

$$\text{und } u = \frac{1}{\sqrt{(x_1^2 + x_2^2 + 1)(x_2^2 + 1)}} \quad (5.57)$$

$$v = \frac{1}{\sqrt{x_2^2 + 1}} \quad (5.58)$$

$$w = \frac{1}{\sqrt{x_1^2 + x_2^2 + 1}} \quad (5.59)$$

Der Vorteil der Abbildung durch die Orthonormalmatrix \underline{O} wird bei der Rückrechnung der interpolierten Kreispunkte in den Raum sichtbar, es gilt nämlich:

$$\underline{O}^{-1} = \underline{O}^T \quad (5.60)$$

Da die Transponierte einfach durch Spiegelung der Matrixelemente an der Matrixdiagonalen erhalten wird, ist die Umkehrabbildung einfach durchzuführen.

Zur Interpolation in der Ebene werden die Vektoren \underline{h}_a und \underline{h}_b auf die Vektoren $\underline{p} = (p_x, p_y, 0)^T$ und $\underline{q} = (q_x, q_y, 0)^T$ abgebildet. Mittelpunkt $\underline{m} = (m_x, m_y, 0)^T$ und Radius r des Kreises errechnen sich gemäß der Kreisgleichung in /73/ zu:

$$m_x = \frac{1}{2} \frac{p_y (q_x^2 + q_y^2) - q_y (p_x^2 + p_y^2)}{q_x p_y - p_x q_y} \quad (5.61)$$

$$m_y = \frac{1}{2} \frac{q_y (p_x^2 + p_y^2) - p_y (q_x^2 + q_y^2)}{q_x p_y - p_x q_y} \quad (5.62)$$

$$r = \sqrt{m_x^2 + m_y^2} \quad (5.63)$$

Der Segmentwinkel γ errechnet sich zu:

$$\gamma = \arctan2(q_y - m_y, q_x - m_x) - \arctan2(q_y - m_y, p_x - m_x) \quad (5.64)$$

Die Interpolation kann nun durch Inkrementierung des Segmentwinkels γ in $\Delta\gamma$ und Berechnung der Stützpunkte H_i' erfolgen. Dabei sind die Stützpunkte H_i' beschrieben durch den Vektor $\underline{h}_i = (h_x', h_y', 0)^T$.

$$h_x' = m_x + r \cdot \cos(\alpha + \Delta\gamma) \quad (5.65)$$

$$h_y' = m_y + r \cdot \sin(\alpha + \Delta\gamma)$$

mit

$$\alpha = \arctan2(p_y - m_y, p_x - m_x) \quad (5.66)$$

Die gesuchten Raumpunkte A_i auf dem Kreis beschrieben durch den Vektor $\underline{h}_i = (h_x, h_y, h_z)^T$ ergeben sich endlich zu:

$$\underline{h}_i = \underline{Q}^T \cdot \underline{h}_i' + \underline{p}_c \quad (5.67)$$

Die TCP-Geschwindigkeit auf dem Kreisbogen kann auf die gleiche Weise errechnet werden wie bei der Linearinterpolation, nur daß hier nicht der Weg zwischen den Punkten A und B sondern der Winkel γ ($< A'M'B'$) gemäß vorgegebener Geschwindigkeitsrampe inkrementiert wird.

5.3.3 Parallele Abläufe, SPS- und Sensorsimulation

Neben kinematischen Komponenten wie Robotern, Greifern, Transporteinrichtungen sind weitere Elemente an der Erfüllung einer Montageaufgabe in einer Montagezelle beteiligt. Dazu gehören vor allem Sensoren und Signalgeber unterschiedlicher Art. Diese Zellenkomponenten wirken im wesentlichen ablaufsteuernd im Montageprozeß, können aber auch die Bewegungen anderer Systeme beeinflussen.

Ein optischer Sensor beispielsweise kann die Funktion haben, ankommende Teile zu erkennen, um dann eine bestimmte Arbeitsgangfolge anzustoßen. Er kann aber auch die Teile exakt identifizieren, deren Lage und Orientierung im Raum ermitteln, um damit den Bewegungsablauf eines Handhabungsgerätes direkt steuern.

In der Simulation wird die Sensorfunktion einer Lichtschranke - aber auch anderer Sensoren- mit Hilfe der Kollisionsprüfung eines geometrischen Körpers, welcher den Meßbereich repräsentiert, mit den zu erkennenden Teilen realisiert (siehe Bild 67 und Bild 78).

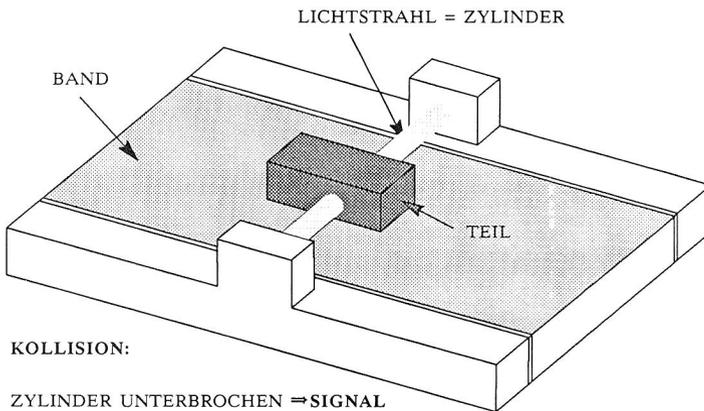


Bild 78: Simulation der Lichtschrankenfunktion

Für die Offline-Programmierung komplexer Sensoren und deren Rechnern, können aus den vorhandenen Modellen a priori Informationen abgeleitet werden. In Forschungslaboratorien der SIEMENS AG wird beispielsweise ein Sensor entwickelt /74/, der aus einem Höhenhistogramm und dem Schnittmuster mit einem Zylinder oder einer Kugel, Lage und Orientierung eines Teils errechnen kann, wenn zuvor ein Vergleichsmuster vorhanden war. Das Ziel der offline Programmierung ist es, dieses Vergleichsmuster, anhand dessen der bezeichnete Sensor jedes Teil eindeutig identifizieren und nach Lage und Orientierung beschreiben kann, aus dem vorhandenen Datenmodell abzuleiten.

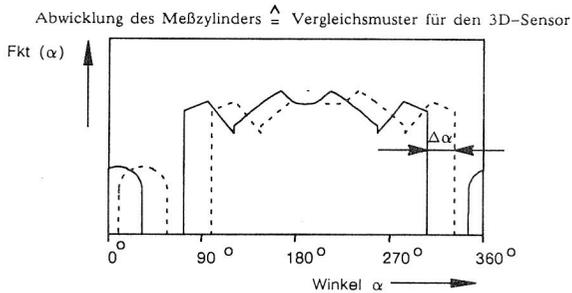
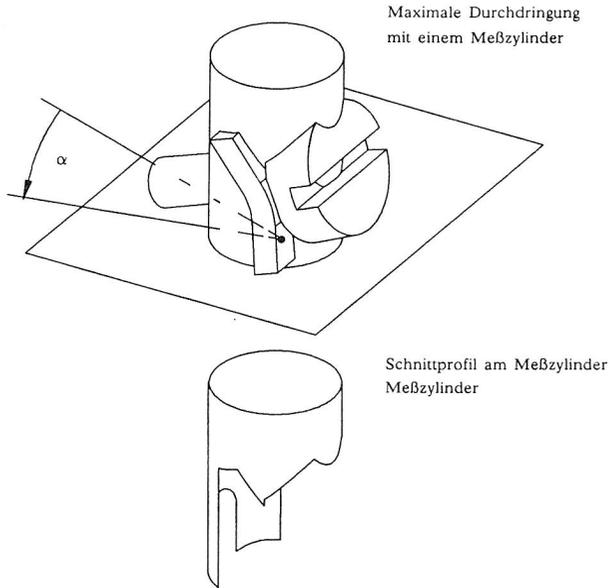


Bild 79: Vergleichsmuster aus dem CAD-Modell zur Bestimmung der Orientierung eines Körpers mit einem 3D-Sensor.

Das Prinzip des Sensors (siehe Bild 79) ist einfach. Zunächst wird ein Höhenhistogramm des im Meßbereich befindlichen Teils ermittelt -z.B. durch Laserscanning. Aus dem Höhenhistogramm wird der Volumenschwerpunkt ermittelt. Ein Zylinder, dessen vertikale

Achse (vertikal = parallel zu den Höhenwerten des Histogramms) durch diesen Volumenschwerpunkt stößt, wird im Radius so variiert, daß sich eine maximale Schnittfläche zwischen ihm und dem als Histogramm dargestellten Körper ergibt. Wickelt man, wie in Bild 79 gezeigt, diesen Zylinder ab, so ergibt sich ein eindeutiges Muster, das -egal wie der Körper verdreht wurde- stets gleich ist.

Durch die Berechnung des Schwerpunktes kann die Lage des Teils und durch das Verschieben des abgewickelten Zylinders die Orientierung ermittelt werden. Simuliert man den Vorgang mit einem CAD-Modell, so läßt sich das Vergleichsmuster auf ähnliche Weise herleiten. Dazu ist es notwendig, den Körper, in der zu erwartenden stabilen und definierten Lage, ebenfalls mit einem entsprechenden Hohlzylinder (maximale Schnittfläche) durch eine Boole'sche Operation zu verschneiden. Die Zylinderfläche kann dann -ebenfalls mit einer CAD-Operation- abgewickelt werden und das so gewonnene Vergleichsmuster z.B. als IGES-Datei an den Sensorrechner übermittelt werden. Bild 79 stellt diesen Vorgang dar und zeigt im unteren Teil das aus dem CAD-Modell abgeleitete Vergleichsmuster und das vom Sensor gemessene Ist-Muster (gestrichelte Linie).

In vielen Montageabläufen sind es nicht komplexe Sensoren, sondern viele einfache Schaltvorgänge, die unterschiedliche Einzelprozesse anstoßen oder stoppen. Die Erfüllung einer Montageaufgabe wird durch viele dieser parallel arbeitenden sich gegenseitig beeinflussenden Einzelprozesse ausgeführt. Simulation einer Montagezelle bedeutet deshalb auch Verschaltung und Koordination paralleler Einzelprozesse und Schaltvorgänge. Auch Benutzereingriffe, wie kurzes Unterbrechen, Bildmanipulation müssen während der Simulation verarbeitet werden.

Der parallele Ablauf der Geräteprozesse erfolgt zu diskreten Zeitpunkten. Hat ein Prozeß A seine aktuelle Programmzeile abgearbeitet, müssen Blockierungsbedingungen überprüft werden - z.B. kann ein anderer Prozeß auf die Fertigstellung von A warten,

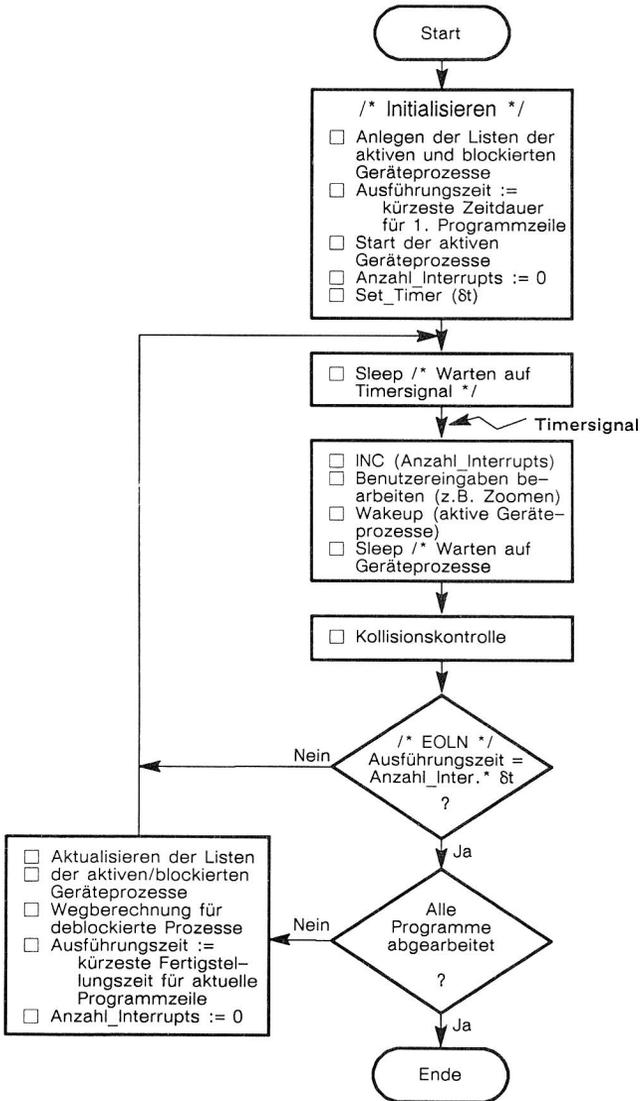


Bild 80: Programmablaufplan des Steuerprozesses zur Koordination parallel arbeitender Einzelprozesse

oder die nächste Anweisung für A lautet "Warte auf Prozeß X". Weiter müssen zu den diskreten Zeitpunkten Kollisionsüberprüfungen erfolgen sowie Benutzereingaben verarbeitet werden. Diese koordinierende Tätigkeit übernimmt der Steuerprozeß (siehe Bild 80). In einer Initialisierungsphase werden

- die Prozeßlisten angelegt (aktive und blockierte),
- die kürzeste Fertigstellungszeit für die erste Programmzeile ermittelt,
- die Initialisierung der Geräteprozesse angestoßen sowie evtl. Rückmeldungen verarbeiten (z.B. Fehler bei Initialisierung),
- die Anzahl der angelaufenen Interrupts auf 0 gesetzt,
- der Timer auf die Zeit Δt gesetzt.

Anschließend blockiert sich der Steuerprozeß und wartet auf ein Signal der Uhr: Es werden Benutzereingaben verarbeitet, die Geräteprozesse "geweckt" sowie eine Kollisionskontrolle durchgeführt. Hat einer der Geräteprozesse seine aktuelle Programmzeile abgearbeitet, werden die Blockierungsbedingungen überprüft und die Liste der aktiven und passiven Prozesse neu angelegt. Anschliessend geht der Steuerprozeß wieder in den "Sleep"-Zustand. Der Ablauf der Geräteprozesse wird also vollständig vom Steuerprozeß gesteuert (siehe Bild 81). Nach der Initialisierung warten die Geräteprozesse auf ein "Weck"-Signal des Steuerprozesses, führen dann einen Bewegungsschritt durch und blockieren sich wieder.

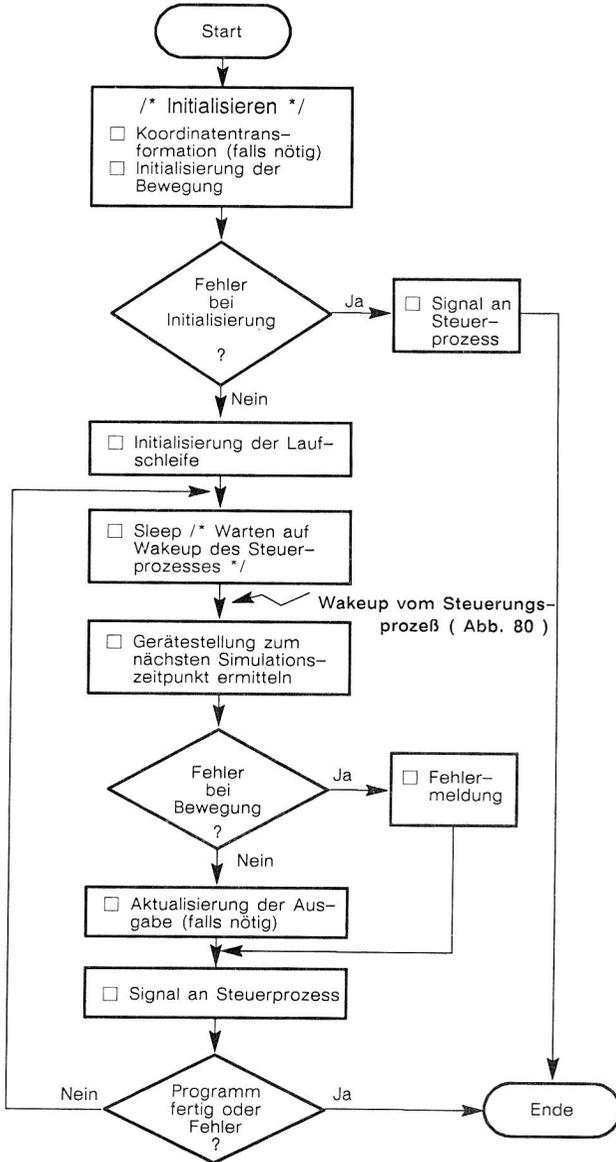


Bild 81: Programmablauf eines Geräteprozesses

5.4 Simulation und Optimierung

Zur Optimierung von Planungsergebnissen lassen sich aus speziellen Bibliotheksprogrammen Regelwerke und Berechnungsalgorithmen verwenden, die Layout und Zellenprogramm bewerten bzw. optimieren.

Optimieren bedeutet neben der Sicherung unmittelbarer Schäden und Störungen (Kollisionskontrolle) auch Wahrung von arbeitsplatzergonomischen Vorschriften (bei manuellen Systemen), von sicherheitstechnischen Regeln sowie der Gestaltung kostengünstiger (=weg-, zeit-, energieoptimierter) Arbeitsabläufe.

Die sicherheitstechnischen Aspekte können beispielsweise durch Kollisionsberechnungen während einer Ablaufsimulation oder durch das Eintragen von Schutzräumen als 3D-CAD-Körper (siehe Bild 82) berücksichtigt werden.

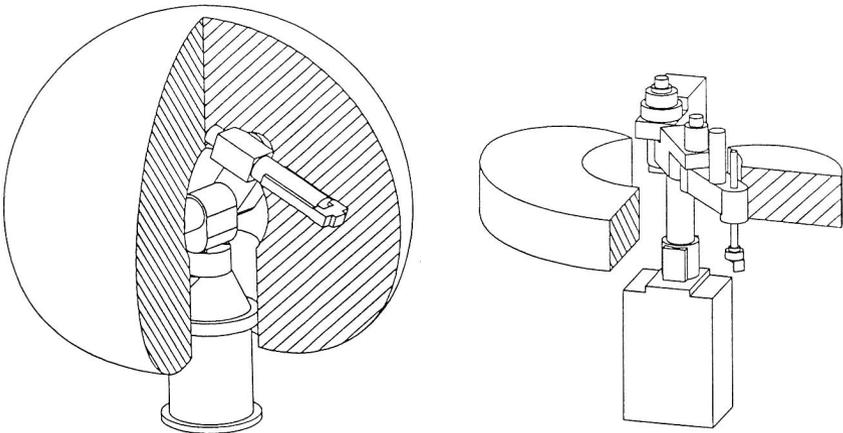


Bild 82: Gefahrenbereiche von Bewegungssystemen zur Systemoptimierung.

Die Layoutoptimierung kann u.a. durch eine einfache Formel vorgenommen werden, indem man nach dem kompletten Programmablauf die zurückgelegten Verfahrenswege eines Handhabungsgerätes minimiert.

Die Verschiebung Δp_{HHS} eines Gerätes in die optimale Lage im Raum errechnet sich zu:

$$\Delta p_{\text{HHS}} = 1/n \sum_{i=1}^n g_i \underline{x}_i \quad (5.68)$$

wobei die Vektoren \underline{x}_i die Anfahrpositionen des HHS beschrieben und g_i die Häufigkeit mit der diese Stellen angefahren werden. In den folgenden Kapiteln werden weitere Spezialgebiete der Optimierung von Montagezellen erläutert.

5.4.1 Kollisionsvermeidung

Beim Einsatz von Automatisierungseinrichtungen in der Fertigung oder Montage spielt die Kollisionsüberwachung eine wichtige Rolle. Das Problem in der Produktion besteht vor allem darin, Kollisionen zu erkennen, bevor sie auftreten. Zudem muß festgestellt werden, ob Kollisionen erwünscht bzw. sinnvoll (bei Fügeoperationen) oder unerwünscht (bei Schäden an Einrichtungen) sind und wie auf eine mögliche Kollision reagiert werden soll. Ein wesentlicher Vorteil der Kollisionserkennung mittels Computermodell besteht darin, daß man Körperzusammenstöße rechnerisch ohne Schaden zulassen kann, um dann eine Strategie zu wählen, die beim nächsten Versuch diese Kollision vermeidet. Ist bei einer Simulation eine Körperüberschneidung aufgetreten, so kann der letzte Bewegungsschritt revidiert werden und ein anderer Weg gesucht werden. Das Problem der Früherkennung entfällt deshalb bei einer Simulation. Mit Hilfe von Volumenrechnungen, Abstandsmessungen und anderen geometrischen Berechnungen lassen sich auch Strategien zur Kollisionsvermeidung einfacher realisieren. Grundsätzlich unterscheidet man zwischen **Kollisionserkennung** und **Kollisionsvermeidung**. Bei expliziter Roboterprogrammierung werden keine Berechnungsprogramme zur Kollisionsvermeidung benötigt, da

die Bahnen definiert (explizit) vom Programmierer vorgegeben werden. Die Bahnplanung oder Pfadsuche spielt bei impliziter Programmierung jedoch eine erheblich wichtigere Rolle. Zahlreiche Veröffentlichungen beschäftigen sich mit Strategien der Wegsuche /75/ /76/. Grundlage für die Überprüfung einer Strategie ist jedoch die Kollisionserkennung. Es bieten sich mehrere Möglichkeiten gemäß Bild 83 an:

- visuelle Überprüfung mittels grafischer Simulation,
- zeitlich diskretisierte Abfrage von Körperüberschneidungen (Boole'sche Operationen, z.B. im CAD),
- mathematische Berechnung des "Bewegungsschlauchs" und anschließendes Auffinden von Überschneidungen,
- visuelle Überprüfung in der realen Montagezelle.

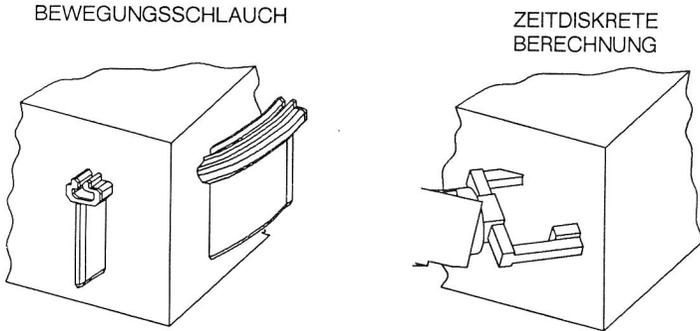
Technisch einfach zu realisieren ist bei einer ohnehin zeitdiskretisierten Simulation von Montageabläufen mit Volumenmodellen, eine schrittweise Überprüfung auf Kollision. Bei einer Montagezelle mit n Körpern bedeutet eine vollständige Kollisionsuntersuchung (jeder Körper wird mit jedem auf Kollision überprüft) bei m Simulationsschritten:

$$\text{Zahl der Untersuchungen} = m \cdot n \cdot (n-1)/2 \quad (5.59)$$

Bei herkömmlichen CAD-Workstations kann eine Kollisionsuntersuchung zweier einfacher Körper bis zu einer Sekunde in Anspruch nehmen. Am Beispiel einer Montagezelle mit 81 Körpern müssen bei einer einfachen Roboterbewegung (Linearbewegung: 100 mm, Geschwindigkeit: 50 mm/s) und einem Rechenintervall von 10 Rechnungen pro Sekunde, $20 \cdot 81 \cdot 40 = 64.800$ Rechnungen durchgeführt werden, was einer Rechenzeit von 18 Stunden entspräche.

Der Rechenaufwand läßt sich erheblich reduzieren, indem man Zellelemente ausschließt, die niemals kollidieren können, z.B. ortsfeste Körper untereinander. Körper die außerhalb der Schutzzräume von Bewegungssystemen liegen, brauchen nicht überprüft zu werden; dazu ist zwar eine zusätzliche einmalige Rechnung not-

wendig, die sich aber bereits bei wenigen Berechnungsschritten lohnend auswirkt. Setzt man r als Anzahl aller beweglichen Zellelemente ein (Achsen, Werkzeuge, Werkstücke), s als statische Elemente, die im Schutzraum der bewegten Zellenkomponenten liegen und p als Peripherieelemente, die vollständig außerhalb stehen, so bleiben



VISUELLE PRÜFUNG BEI GRAFISCHER SIMULATION

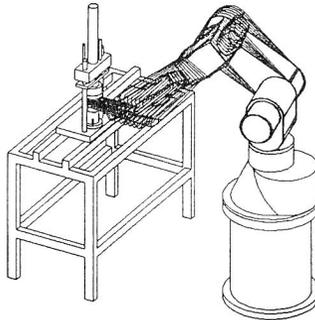


Bild 83: Volumenorientierte Offline-Kollisionskontrolle

$$(s+p) + m \cdot [r/2 \cdot (r-1) + r \cdot s] \quad (5.70)$$

Rechenschritte. Verwendet man das obige Beispiel ($r=12$, $s=20$, $p=49$), so reduziert sich die Rechenzeit auf ca. 1.7 Stunden. Durch weitere Regeln (z.B. Ausschließen von Werkstücken die gefügt werden, oder von Robotern deren Arbeitsbereiche nicht über-

lappen) läßt sich diese Art der Kollisionsprüfung weiter vereinfachen.

Das Konzept der zeitdiskretisierten Kollisionsuntersuchung ist umfassend und ermöglicht es, nicht nur Kollisionen zwischen Robotern und starren Komponenten zu erkennen, sondern läßt sich auch auf Roboter im koordinierten Betrieb anwenden.

5.4.2 Bahnoptimierung, Bahnplanung

Die Bahnoptimierung verfolgt das Ziel, den Arbeitsraum von Bewegungssystemen zu vergrößern, beispielsweise durch Strategien zum Wechsel des Winkelstatus oder geschickte Auswahl eines Anfahrweges bei PTP-Bewegungen. Bahnplanung läßt sich unterteilen in:

- Bewegungsgrobplanung und
- Bewegungsfineplanung.

Bei groben Bewegungsabläufen -Anfahren, Transportieren, Ruheposition einnehmen- spielt die zeit- und energieoptimale Nutzung der gesteuerten kinematischen Systeme die wichtigste Rolle. Geometrische und physikalische Einflüsse der Umwelt, wie Kollisionen mit der Peripherie, wirksame Kräfte auf das bewegte Teil am Endeffektor und Bewegungsdauer, sind zu berücksichtigen. Feinmotorische Bewegungen unterliegen anderen Einflußgrößen. Hier kommt es in hohem Maß auf die Art der auszuführenden Operation (Montagevorrichtung) an -Teile fügen, Werkzeug positionieren, kleben, entgraten, schweißen usw. Es sind definierte Bahnen zu berechnen, die im Normalfall keine Alternativen wie bei der Grobmotorik zulassen- Pendeln um eine Schweißnaht, orthogonale Werkzeugpositionierung beim Schrauben, definierte Fügebewegung beim Montieren eines Teiles. Implizite Anweisungen wie Hinfahren, Bringen, gehören zu den Grobbewegungen; Fügen, Greifen, Loslassen zu den Feinbewegungen.

Bei PTP-Bewegungen ergeben sich häufig starke Geschwindigkeits-

gradienten und damit große Kräfte, da jede Achse einer kinematischen Kette, ohne Rücksicht auf die Bewegung einer anderen Achse, mit maximalem Drehmomentverlauf aus der Anfangs- in die Zielstellung fährt. Optisch liefern solche unkontrollierten PTP-Bewegungen einen unstetigen und damit ruckhaften Bahnverlauf im Raum. Knicke in der Bahn treten nämlich dann auf, wenn eine Bewegungsachse der kinematischen Kette ihr Ziel erreicht hat und plötzlich anhält. Abhilfe kann durch sog. synchronisierte PTP-Bewegungen erfolgen, bei denen für jede Bewegung gilt, daß alle Achsen gleichzeitig losfahren und gleichzeitig den Zielpunkt erreichen. Eine solche Bewegung kann durchgeführt werden, wenn die Geschwindigkeiten aller Achsen reduziert werden, indem sie sich an der 'langsamsten' Achse orientieren.

BEWEGUNGSART	PLANUNGSAUFGABE	ANWENDUNG
GROBMOTORIK	Wegoptimierte, zeitoptimierte und energieoptimierte Bahn ermitteln	Hinfahren
	Kollisionsfreiheit von HHS und Umwelt sichern	Bringen (=Transportieren)
	Kräfte am Endeffektor regulieren	
FEINMOTORIK	Bewegungsbahnen exakt berechnen	Fügen (Schrauben, Schweißen, usw.)
	Prozeßparameter berücksichtigen	Greifen
	Sensorik einplanen	Loslassen

Bild 84: Planen von Bewegungsabläufen

Für die implizite Programmierung ist ein weiteres Problem durch Bahnoptimierung zu lösen. Bei einer Anweisung -z.B. "Fahre Stützframe an"- kann dessen Lage und Orientierung aus dem rechnerinternen Modell gelesen werden. Die Bahn auf dem das Bewegungssystem fahren soll, ist aber nicht bekannt. Bewegungssysteme können eine vorgegebene Stellung nicht nur in unterschiedlichen Achskonfigurationen einnehmen, sondern sich auch auf unterschiedlichen Bahnen dorthin bewegen. Durch eine Berechnung vor Ausführung der Aktion, kann der Weg nach folgenden Kriterien optimiert werden:

- kürzeste Zeit,
- kürzester Weg,
- geringste Energie.

Zudem muß sichergestellt werden, daß die gewählte Strategie eine kollisionsfreie Bahn (siehe Kapitel 5.4.1) ermöglicht. Die Ermittlung des optimalen Weges für eine grobmotorische Bewegung (PTP) läßt sich auf ein rein rechnerisches Problem zurückführen. Zunächst müssen alle n möglichen Lösungen der Rücktransformation ermittelt werden. Dann werden alle PTP-Bewegungen von der Ausgangsstellung zur Zielstellung durchgerechnet und gemäß den Zielkriterien bewertet und gewichtet.

Beispiel:

Für einen sechssachsigen Gelenkroboter vom Typ RRRA3 gibt es normalerweise 8 verschiedene Achskonfigurationen, um eine bestimmte Position im Raum einzunehmen. Durch die sowohl in positive als auch negative Richtung drehenden Gelenkachsen, ergeben sich $8 \cdot 2^6 = 512$ Anfahrwege. Aus diesen 512 Anfahrwegen kann über das Kriterium Zeit- oder Wegminimum eine Rangreihe errechnet werden, deren einzelne Elemente, beginnend beim Optimum simuliert und damit auf Kollisionen oder andere Restriktionen getestet werden.

Der Vorteil, die Berechnungen bereits in der Offline-Planung durchzuführen, besteht darin, daß anstelle einer Bewegungsanweisung in kartesischen Koordinaten eine Bewegungsanweisung in Achskoordinaten (=Roboterkoordinaten) erfolgen kann. Das bedeutet eine Rechenzeiterparnis in der Robotersteuerung und liefert zudem die Gewißheit, daß die programmierte Bahn tatsächlich wie geplant abgefahren wird (den meisten Robotersteuerungen sind nämlich eigene Bahnplanungsstrategien implementiert; diese lassen die Bestimmung einer Bahn offline nicht zu, selbst wenn es möglich ist, den gewählten Zielpunkt mit Lage, Orientierung und Winkelstatus zu übertragen).

Auch für feinmotorische Bewegungen, wie beim linearen oder zirkularen Bahnfahren, sind Strategien notwendig. Aus dem Modell eines

Teils, das mit einem Kleberauftrag versehen werden soll, kann zwar die abzufahrende CP Bahn ermittelt werden, nicht aber die Reaktion des werkzeugführenden Handhabungssystems, wenn dieses an technische Grenzen (Endanschläge, Wechsel des Winkelstatus, Geschwindigkeitslimits der Achsen) stößt. Auch hierbei ist es sinnvoll, in der Planung Vorgänge zu simulieren, um Probleme frühzeitig zu erkennen. Ausgereifte Simulatoren können zudem Lösungen bei einigen Problemen selbst errechnen.

Beispiel: Wechsel des Winkelstatus bei Linearbewegung.

Dieses Problem betrifft vor allem Gelenkroboter. Da es gilt, Achspositionen aus trigonometrischen Ausdrücken zu ermitteln, können Schwierigkeiten entstehen, wenn ein Gelenkroboter beim Abfahren einer Bewegungsgeraden in einer oder mehreren Achsen einen Nulldurchgang erfährt. Bei der Rücktransformation werden für jede Achse mehrere mögliche Stellungen errechnet. Durch den Winkelstatus (z.B. Achse positiv oder negativ) wird die Lösung eindeutig bestimmt. Wenn also der Winkelstatus beim Start der Linearbewegung eine bestimmte Achskonfiguration vorschreibt, können keine Nulldurchgänge ausgeführt werden, es sei denn, es wird mittels einer Strategie ermittelt, wie der Winkelstatus sinnvoll zu ändern ist (siehe dazu /77/).

Aus Technologiebibliotheken können physikalische und montagerelevante Parameter für feinmotorische Bewegungsbahnen abgeleitet werden. Solche Bibliotheken enthalten Angaben für das Fahren von Schweiß- und Klebebahnen, aber auch Angaben über Schraub- oder andere Fügeoperationen. Mittels der eindeutigen Identifikation der im Produktmodell angegebenen Bearbeitungsframes (siehe Kapitel 5.2) können die entsprechenden Angaben, wie notwendige Bahngeschwindigkeit, Bewegungsart (Pendeln) oder Beschleunigungsverhalten für das kinematische System abgeleitet werden.

5.5 Greif- und Bearbeitungsplanung

Die Bearbeitung impliziter Anweisungen vollzieht sich in einem Wechselspiel zwischen planenden und kontrollierenden Systembausteinen, die steten Zugriff auf statische und veränderliche Informationen des in Kapitel 5.2 beschriebenen Datenmodells besitzen. Bild 85 gibt das funktionale Zusammenwirken und den Systemaufbau zur Bearbeitung impliziter Anweisungen wieder. Der vorgeschlagene Systemaufbau erlaubt eine modulare und schrittweise Entwicklung der einzelnen Systembausteine.

Das Konzept des Systems erlaubt auch den Einsatz einzelner Bausteine und ermöglicht damit z.B. Offline-Programmierung in verschiedenen Komplexitätsstufen. In Baustein 1 werden einzelne Kommandos oder Anweisungsblöcke interpretiert:

- ...
1. WÄHREND ...
 2. TEIL A FÜGEN (IN) TEIL B
 3. TEIL C BRINGEN (IN DAS) ARBEITSZENTRUM
- ...

Es erfolgt die Zerlegung in Teilaufgaben und die Zuordnung der Zellenkomponenten aus dem Zellenmodell, welche die gestellten Aufgaben erfüllen können (Baustein 2). Für die Einzelkomponenten werden - immer noch implizite - Programmsequenzen generiert. Aus den impliziten Programmsequenzen können mittels komponentenspezifischer Planungsmodule explizite Programme generiert werden. Dazu wird in Baustein 3, dessen Arbeitsweise an einem Beispiel noch genauer erläutert wird, die Teilaufgabe für jede Komponente gelöst. Die Sequenzen werden durch einen Regelkreis aus Simulation (Baustein 4) und Kontrolle (Baustein 5) generiert und auf Korrektheit geprüft.

In einem Simulator (Baustein 6), der parallele Prozesse verarbeiten kann (Kapitel 5.3), wird das Zusammenwirken der einzelnen Komponenten berechnet und schließlich durch den Baustein 7 überprüft. Beim Auftreten von Störungen können alternative Programmsequenzen angefordert werden. Falls es dem System mit Hilfe

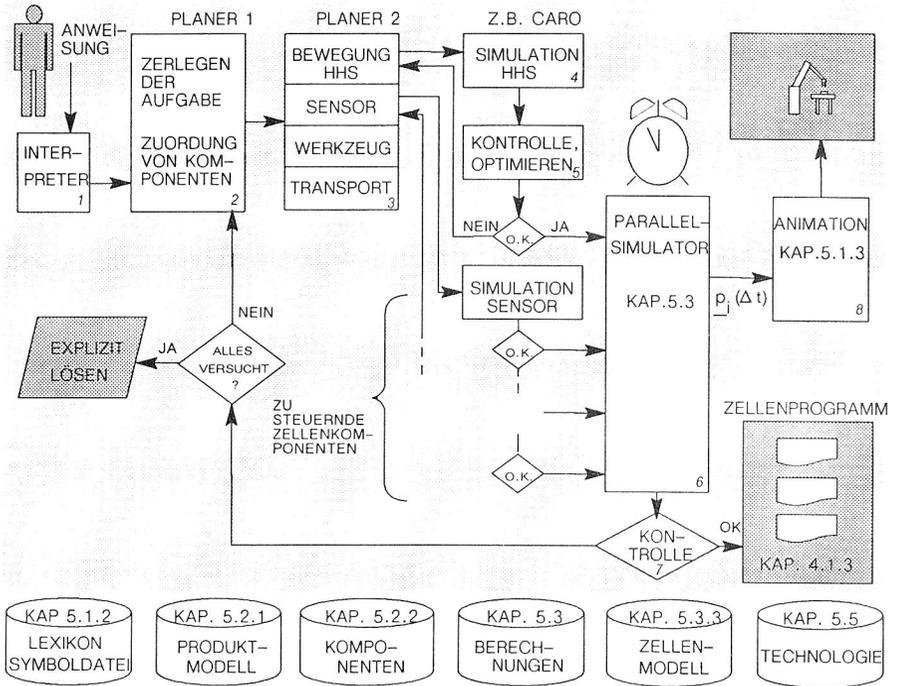


Bild 85: Zusammenwirken der Systemmodule zur Verarbeitung impliziter Programmanweisungen

beider Regelkreise nicht gelingt, eine Lösung zu finden, werden vom Systembediener neue Informationen oder evtl. explizite Lösungen angefordert. Der Systembediener hat im übrigen die Möglichkeit, die Ergebnisse von Simulationsrechnungen grafisch und numerisch zu verfolgen (Baustein 8), um bei Bedarf den Planungs- und Simulationsvorgang zu unterbrechen.

Handelt es sich bei der verwendeten Zelle um einen manuellen Arbeitsplatz, so wird statt eines Steuerprogrammes ein Arbeitsplan durch das System generiert. Dieser enthält neben der detaillierten Aufgabenbeschreibung auch Zeiten für jeden einzelnen Schritt.

Interessant ist die Schnittstelle zu Baustein 6, denn die eingehenden Programme, für jede zu steuernde Zellenkomponente, bestehen aus expliziten Codes, wie sie auch in der Produktion eingesetzt werden. Das bedeutet, daß über diese Schnittstelle auch existierende Steuerprogramme eingebracht und simuliert werden können (für Roboter kann diese Schnittstelle beispielsweise IRDATA sein). Ausgehend von diesem Baustein kann also auch eine explizite Zellenprogrammierung ausgeführt werden.

Genauer soll im folgenden 'Planer 2' (=Baustein3 in Bild 85) und dessen Zusammenwirken mit dem Datenmodell sowie dem Simulationsmodul zur Berechnung der Abläufe einer Zellenkomponente betrachtet werden. Diese eingehendere Betrachtung soll am Beispiel einer Fügeaufgabe -ausgeführt von einem Roboter und einem Greifwerkzeug ausgeführt werden- durchgespielt werden. Die vollständige implizite Anweisung (siehe Bild 86) lautet:

"Teil A fügen (in) Teil B".

Daraus werden durch Planer 1 zwei implizite Anweisungssequenzen für das Handhabungssystem der aktuellen Montagezelle und das Greifsystem generiert.

Für das Handhabungssystem sieht diese Sequenz folgendermaßen aus:

1. Roboter XYZ HINFAHREN (ZU) TEIL A ZUM GREIFEN MIT
GREIFER UVW
2. Roboter XYZ WARTEN BIS GREIFER UVW FERTIG
3. Roboter XYZ HINFAHREN (ZU) TEIL B ZUM FÜGEN MIT
GREIFER UVW
4. Roboter XYZ WARTEN BIS GREIFER UVW FERTIG
5. Roboter XYZ HINFAHREN (IN) SICHERHEITSPPOSITION 1

Planer 2 muß aus diesen impliziten Fahrbefehlen ein Programm in Form expliziter Anweisungen für den Roboter XYZ erzeugen. Für die Anweisung 1 wird dies im Ablaufplan gemäß Bild 87 ersichtlich.

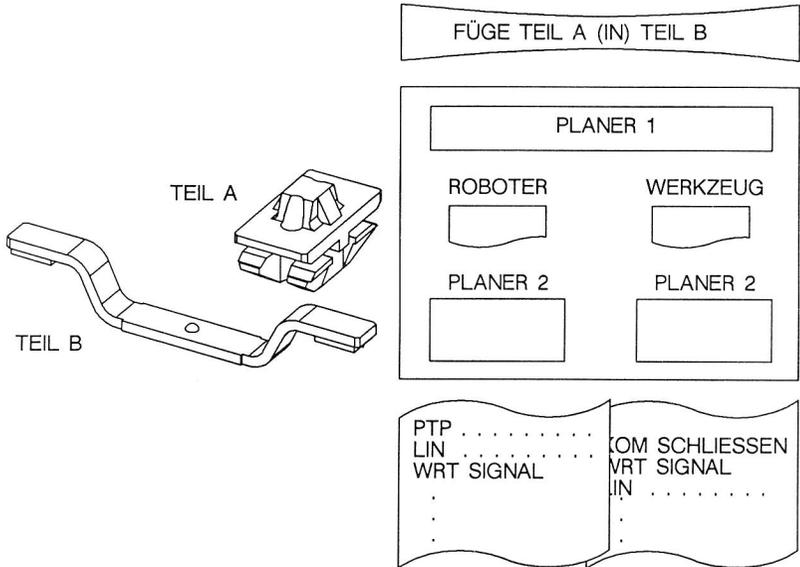


Bild 86: Lösen einer impliziten Montageanweisung

Zunächst müssen alle Komponenten, die bei der Lösung des Teilproblems beteiligt sind, identifiziert werden. Mit der Identifikation stehen alle im Datenmodell gespeicherten geometrischen und technologischen Informationen zur weiteren Auswertung zur Verfügung. Über die aktuelle Lage des Roboters und des Greifers kann die IST-Position des TCP berechnet werden (siehe Bild 73). Zudem läßt sich aus der Lage des zu greifenden Teils mit dessen Greifframe T_G und aus der Transformation vom Greiferangriffsframe T_w zum TCP (aus dem Greifermodell) die SOLL-Stellung des Roboters errechnen (siehe Bild 53, 59, 65). Da nur im Ausnahmefall direkt aus der IST-Stellung des Roboters in die Zielstellung gefahren werden kann, hat Planer 2 die Aufgabe eine sinnvolle Zerlegung der Bewegung vorzunehmen. Dazu wird eine grobmotorische Bahn geplant, deren Endlage sich aus den maximalen Abmessungen des Teils und der idealen Anfahrriichtung zum Greifframe ermitteln läßt. Die Grobbewegung soll nach Möglichkeit mittels einer einzigen kollisionsfreien PTP-Bewegung möglich sein.

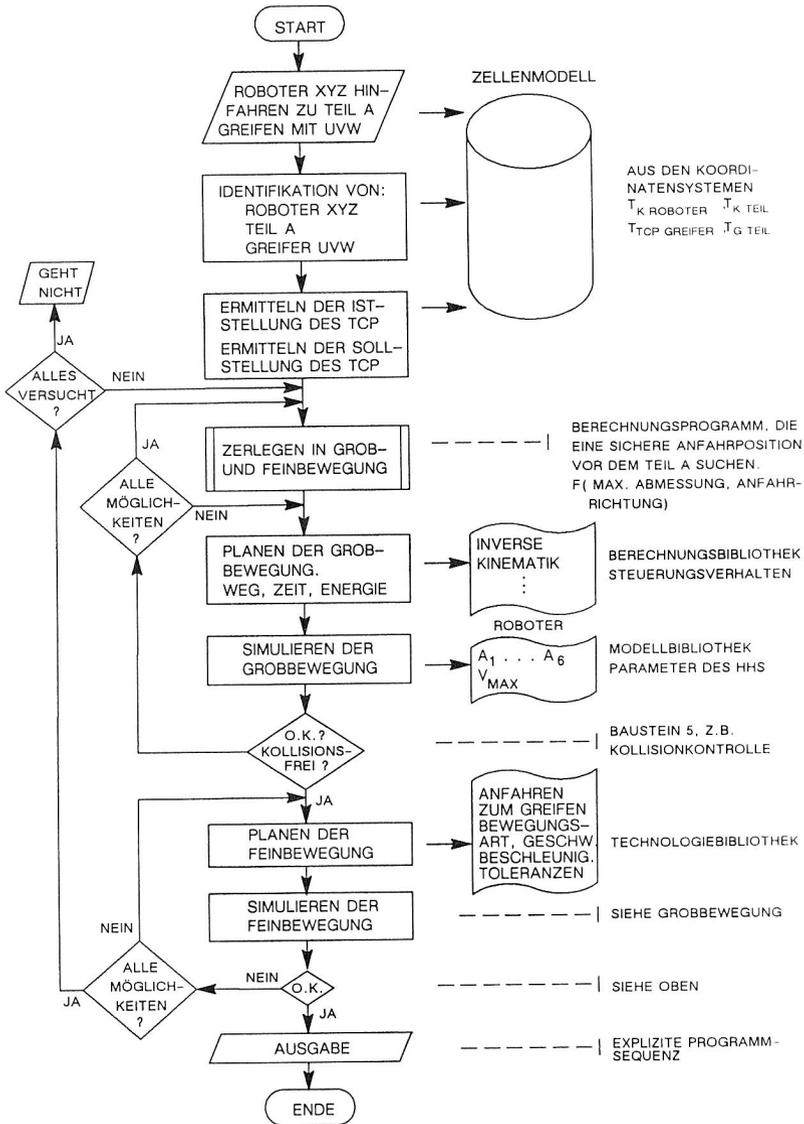


Bild 87: Lösen einer impliziten Anweisung mit PLANER_2

In schwierigen Fällen sind ein oder mehrere Stützpunkte zu ermitteln, die einen kollisionsfreien Weg zur Ausgangslage der Feinbewegung garantieren. Die Feinbewegung zum Greifen kann im ersten Ansatz über einen Linearsatz versucht werden. Die genaue Bahn ist aber aus dem technologischen und geometrischen Modell von Teil A zu ermitteln. Toleranzen, Anfahrgeschwindigkeiten und Beschleunigungsverhalten der Feinbewegung sind in der Technologiedatei 'Greifen von Werkstücken' enthalten. Die Charakteristika der Paarung Werkstück - Greifer werden dabei in Form von Regeln berücksichtigt, z.B. wenn der Greifer im geöffneten Zustand überall eine Mindestdistanz Δs vom Werkstück besitzt, kann maximale Anfahrgeschwindigkeit gefahren werden.

In einem Regelkreis aus Simulation und Kontrollmechanismen wird aus den möglichen Lösungen der Bewegungsbahnen die optimale ausgewählt. Im einfachsten Fall erstellt Planer 2 aus der impliziten Anfahrweisung einen PTP-Satz und einen Linearsatz.

6. Prototypen und Tests

Aus den theoretischen Arbeiten zur Anforderungsbeschreibung und Grundkonzeption für anwender- und aufgabenorientierte Planungs- und Programmiersysteme sind zu Testzwecken Module eines Prototypen realisiert worden. Die implementierte Software wird heute für Ergonomieuntersuchungen, studentische Ausbildung sowie zur Projektierung von Montageanlagen im Rahmen von Industrieprojekten eingesetzt.

Als Hardwarebasis dienen Arbeitsplatzrechner (workstation), die sich im Konstruktionsbereich als CAD - Rechner bewährt haben. Durch die verfügbare 32bit Architektur, die Möglichkeit zum 32 MByte Hauptspeicherausbau, spezielle arithmetische und grafische Koprozessoren sowie ein mehrprozeßfähiges Betriebssystem (UNIX, AEGIS) können Anforderungen, wie in den Kapiteln vier und fünf beschrieben, weitgehend realisiert werden.

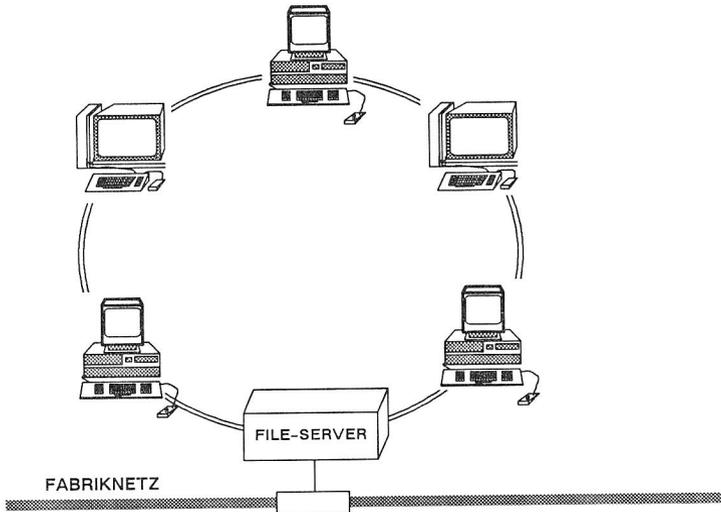


Bild 88: Rechnerkonfiguration

Softwaretechnisch werden Grundbausteine für die Menügestaltung (DIALOGUE) und die 3D Objektdarstellung (GMR3D, PHIGS) eingesetzt. Zur geometrischen Modellierung von Produkten und Produktionskomponenten werden die CAD - Systeme SIGGRAPH und ROMULUS genutzt.

Aufgrund hard- und softwaretechnischer Grenzen sowie personalkapazitiver Engpässe konnte nur ein Teil der in den vorangegangenen Kapiteln beschriebenen Anforderungen in Form des Systems CARo realisiert werden. Im folgenden werden die Anwendungen mit CARo, Systemtests, deren Auswertung sowie der Programmier- und Simulationsmodul in CARo beschrieben.

6.1 CARo - Computer Aided Robotic System

6.1.1 Systemaufbau von CARo

CARo (I) ist ein Werkzeug zur grafisch interaktiven Layoutplanung und Programmierung von Montagesystemen /21/. Auf der Basis eines CAD - Systems, einer Modelldatenbank und eines Moduls zur Programmierung und Simulation können Roboterarbeitszellen gestaltet, programmiert und getestet werden. Schließlich ist es möglich, die offline erstellten und getesteten Programme an Robotersteuerungen in der Fertigung zu übergeben . Die wesentlichen Eigenschaften von CARo (I) lassen sich wie folgt zusammenfassen:

- geometrisch technologische Modellierung von Robotern,
- Archivieren von Modellen in Bibliotheken,
- Gestalten von Montagesystemen, beliebige Variationen,
- Einbauuntersuchungen am Produktmodell,
- Erreichbarkeitsstudien mit den Montagesystemmodellen,
- Programmierung durch grafisches Teachin kombiniert mit textueller Eingabe der Programmlogik,
- Test und Optimierung der Roboterprogramme am Grafikbildschirm,
- Taktzeiterfassung,
- Übergabe der Roboterprogramme via Fabriknetz (Bild 88)

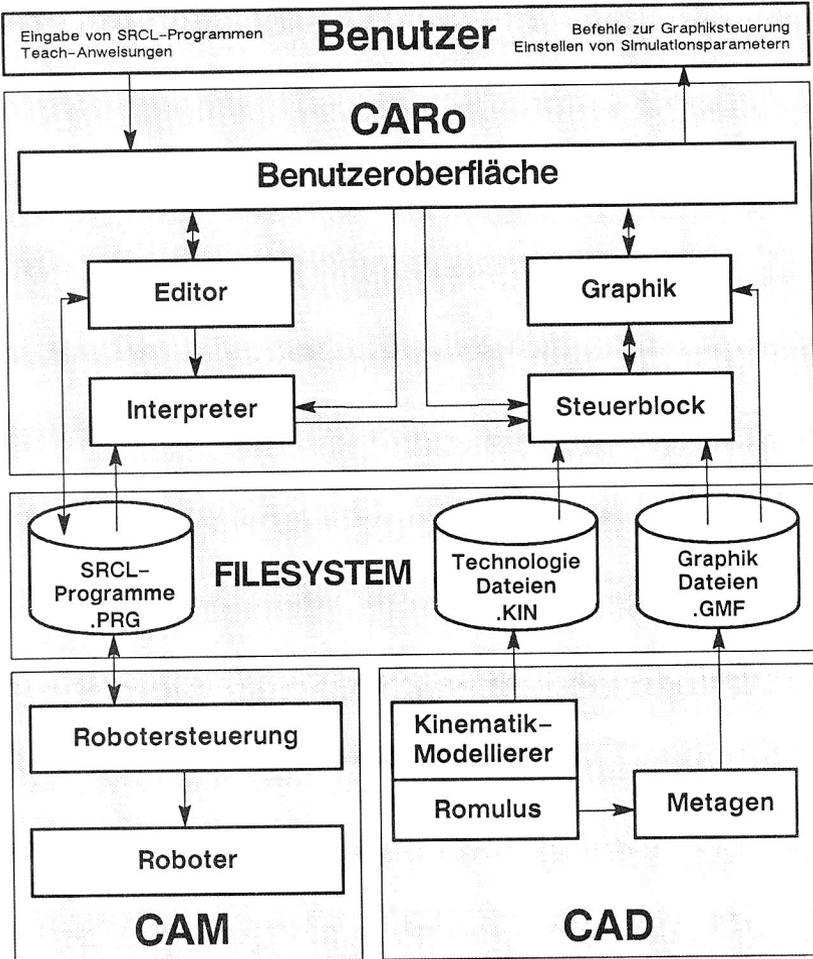


Bild 89: Systemaufbau von CARo

Weitere wesentliche Systemmerkmale sind die integrierte Datenhaltung und eine menügeführte Benutzeroberfläche. Bild 89 verdeutlicht den Aufbau und die Arbeitsweise von CARo (I), (II). Das System gliedert sich dabei logisch in die drei Bereiche: Zellenmodellierung (CAD), Planung (CAP) sowie Programm (CAM).

Das in CARo installierte CAD - System erlaubt auf einfache, d.h. dem CAD - Konstrukteur gewohnte Weise das geometrische Modellieren von Montageelementen, wie Handhabungsgeräten, Transportsystemen, Werkzeugen, Greifern und selbstverständlich auch Werkstücken. Durch das Anbinden zusätzlicher montagespezifischer Informationen an die geometrischen Modelle können wesentliche Entscheidungen in den späteren Phasen der Montageplanung von Rechenprogrammen unterstützt werden. Schon während des Modelliervorganges ist es mit CARo möglich, solche Informationen auszuwerten, um beispielsweise Einbauuntersuchungen durchführen und Fügeoperationen simulieren zu können (Bild 90).

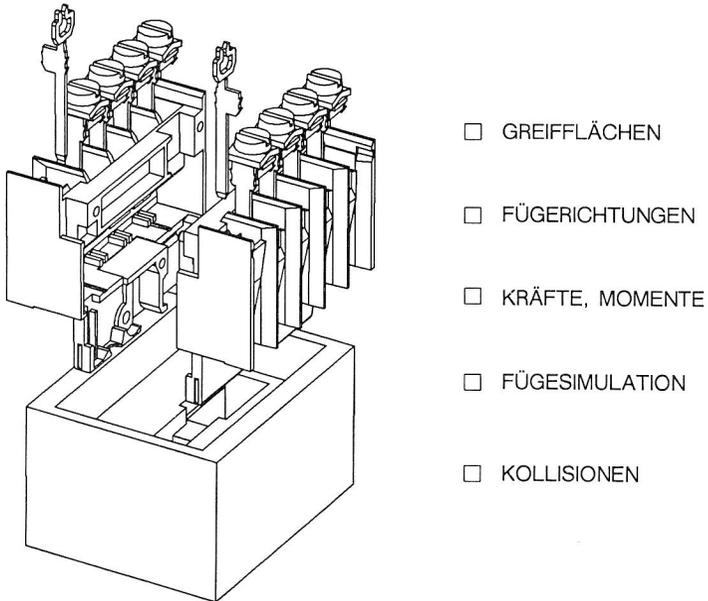


Bild 90: Montageuntersuchungen am Produktmodell in CARo

Für eine sinnvolle rechnergestützte Planung reicht die geometrische Information der diversen Systemkomponenten nicht aus. Deshalb besitzt CARo ein Modul zur technologisch klassifizierenden Beschreibung von Montageelementen. Darin wird unter anderem festgelegt, welche Beweglichkeiten bzw. welche kinematischen

Parameter bestimmte Zellenkomponenten besitzen. Alle Montagekomponenten sind hier klassifizierbar und parametrierbar. Für einen Industrieroboter werden beispielweise Typ, Tragkraft, Positioniergenauigkeit sowie Zahl, Lage, Geschwindigkeiten, Beschleunigungen und Verfahrbereiche der Achsen eindeutig festgelegt. Werkzeuge werden als solche gekennzeichnet und lassen sich einem Handhabungssystem zuordnen. Sonstige Komponenten werden geometrisch modelliert und als beweglich oder unbeweglich klassifiziert.

Geometrische und technologische Beschreibung bilden zusammen das Modell einer Komponente. Jede dieser Komponenten wird in einer Bibliothek abgelegt und verwaltet. CARo unterscheidet grob folgende Objektklassen:

- Handhabungssysteme (Roboter, kartesische Achsen),
- Werkzeuge (wie Greifer, Schrauber, Schweißzangen),
- Werkstücke,
- Werkstückträger,
- Transportsysteme (wie Bänder, Förderfahrzeuge),
- sonstige Montagekomponenten (wie Steuerungen, Tische, Portale, Gerüste).

Im folgenden Schritt, nämlich der Programmierung mit Bewegungs- und Ablaufsimulation, kann nur mit vereinfachten geometrischen Modellen (Draht- oder Flächenmodellen 1. Ordnung) gearbeitet werden (siehe Bild 97). Durch den in CARo enthaltenen Volumenmodellierer lassen sich die während des Programmiervorgangs aufgezeichneten Arbeits- und Simulationsprotokolle auch in 3D-Vollkörperdarstellung, farbschattiert oder mit Ausblendung verdeckter Kanten, am Bildschirm aufzeichnen (Bild 91 und 51). Die rechenzeitintensive Simulation mit Vollkörpern liefert dabei nicht nur eine eindrucksvolle Animation von Montagevorgängen auf dem Monitor, sondern erlaubt auch die Durchführung automatischer Kollisionskontrollen. Zu jedem diskreten Zeitpunkt der vorangegangenen Programmsimulation werden beim Verarbeiten der Simulationsprotokolle alle in der Montagezelle vorhandenen Körper auf mög-

liche Überschneidungen untersucht. Kollisionen werden in der Vollkörperanimation farblich gekennzeichnet und durch Signalton angekündigt.

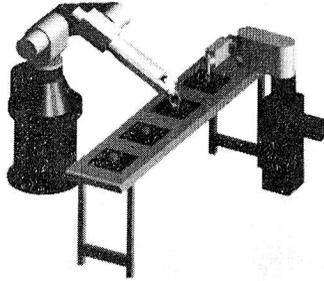


Bild 91: Kollisionsuntersuchungen mit CARo - Volumenmodellen

Während die erste Version von CARo nur die explizite Programmierung von Robotern erlaubt, ist CARo(II) um ein Modul zur aufgabenorientierten Roboterprogrammierung sowie um Programmiermöglichkeiten für andere Zellenkomponenten -wie SPS oder Sensoren- erweitert worden. Neben der Anwendung für automatisierte Montagezellen können mit CARo(II) manuelle Arbeitsplätze geplant und simuliert werden. In einer weiteren Ausbaustufe wird es möglich sein, MTM-Zeitstudien mit Hilfe des Modells durchzuführen und aus Bewegungsanweisungen Arbeitspläne zu generieren. CARo (II) verarbeitet zwar noch nicht das in Kapitel 5.2 beschriebene Datenmodell, berücksichtigt jedoch die speziellen Anforderungen der Systembediener unter ergonomischen und funktionalen Gesichtspunkten.

6.1.2 Systembedienung

Der Modelliermodul von CARo setzt auf einem CAD-System auf, entweder SIGGRAPH 3D (Fa. SIEMENS) oder ROMULUS (SHAPE DATA). Dem

Systembenutzer steht daher die gewohnte CAD-Funktionalität zum geometrischen Gestalten von Systemkomponenten und Produkten zur Verfügung. Technologische Informationen, die nicht in die CAD-Struktur übernommen werden können, lassen sich mit Hilfe der Prozedur "Zellenmodellierer" an die Modelle binden. Der Benutzer wird in dieser Prozedur durch einen rechnerinitiierten Dialog geführt, dabei wird sichergestellt, daß alle zur Simulation notwendigen Informationen eingebracht werden.

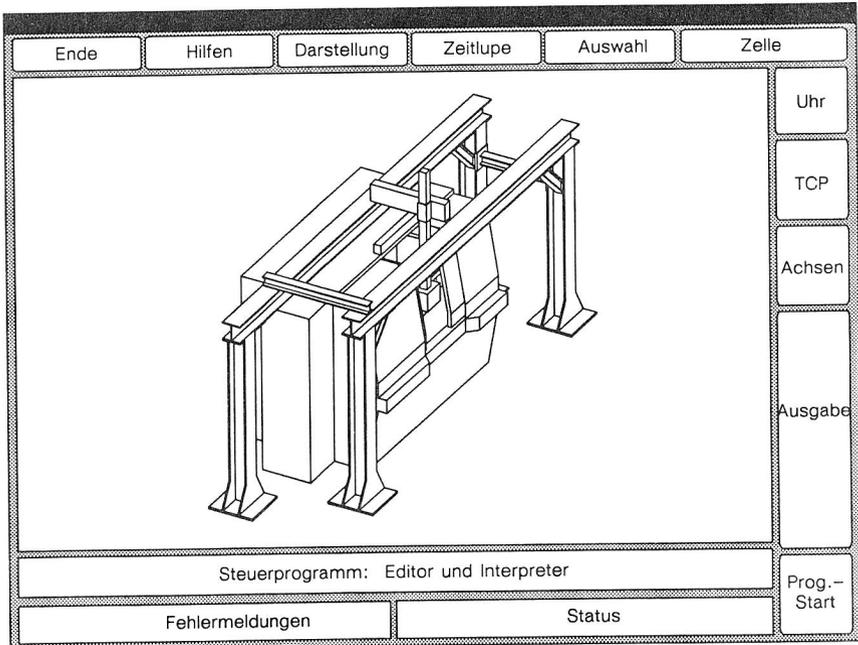


Bild 92: Programmierung und Simulation einer Maschinenbeschickung in CARo

Der Programmier- und Simulationsmodul ist, ebenso wie der Modellerteil, durch ein Bildschirmmenü zu bedienen (siehe Bild 92). Das Menü wird auf einem Fenster (engl. window) auf dem Bildschirm dargestellt. Über einen Bildschirmzeiger (engl. Cursor), lassen sich die einzelnen Operationen direkt oder über auftauchende Untermenüs (engl. Pop-Ups) auslösen. Die Arbeitsweise von

CARo entspricht aus Gründen des Antwortzeitverhaltens und aufgrund fehlender Entwicklungstools nicht vollständig der im Konzept vorgeschlagenen objektorientierten Arbeitsweise. Die Systembedienung erlaubt dennoch eine rasche Einarbeitung, ein befriedigendes Antwortzeitverhalten und damit die interaktive Bedienung des Systems.

Das Fenster der Oberfläche ist -wie in Bild 92 gezeigt- grob in drei Bereiche unterteilt:

1. Grafikpad zur Visualisierung von Bewegungen in quasi Echtzeit,
2. Editorpad zum Erstellen und Testen von Programmen und Abläufen,
3. Systemfunktionen zum Ablesen/Einlesen von Simulationsparametern oder zum Bewegen/Überwachen der Zellelemente.

Im Grafikpad werden die zu programmierenden Montagezellen visualisiert. Dort können einzelne Komponenten auch selektiert werden und Modellinformationen abgefragt werden. Das Editorpad erlaubt die Formulierung von Steuerprogrammen, die entweder in einer konkreten Robotersprache oder in Form einer impliziten Kommandofolge erstellt werden. Explizite Bewegungsanweisungen von Robotern lassen sich einfach in den Editor eintragen, indem die gewünschte Roboterstellung über Funktionstasten angefahren und mit der zugehörigen Bewegungsart einfach ins Programm übernommen wird (PASTE-Funktion). Das Editorpad wird auch bei der Interpretation von Steuerprogrammen eingesetzt. Es dient dann zur Wiedergabe der aktuell abzuarbeitenden Programmzeile oder Programmsequenz. Dadurch ist es möglich, Programme schrittweise zu interpretieren und bei Bedarf während des Laufs zu ändern (TRACE-Funktion). Neben dem vollständigen expliziten Befehlssatz einer Robotersprache akzeptiert CARo folgende implizite Anweisungen:

- FAHRE 'FRAME' AN
- GREIFE 'FRAME'.

CARo kann ein Programm, das viele dieser impliziten Anweisungen besitzt, in ein explizites Programm wandeln und testen.

Zahlreiche Systemfunktionen erlauben die Manipulation der Darstellung. So lassen sich geometrische Objekte in verschiedenen Ansichten und aus unterschiedlichen Richtungen abbilden. Zoomen, Verschieben und Verdrehen des Betrachterstandpunktes über Tastatur gehören ebenso dazu, wie die Darstellung in verschiedenen Rissen und Schnitten nach DIN. Das Bildschirmmenü erlaubt weiterhin die Einstellung von Zeitlupe und Zeitraffer bei der Simulation von Bewegungen. Um Taktzeiten zu messen und um eine Vorstellung vom Realzeitverhalten des simulierten Modells zu erhalten, kann eine Stoppuhr eingeschaltet werden, die die berechneten Zeitinkremente summiert und damit Einzelzeiten und Fortschrittszeiten wiedergibt.

Wichtigstes Bedienelement neben der Funktionstastatur ist die Maus, mit der in Abhängigkeit ihrer Lage Systemfunktionen ausgelöst, Komponenten selektiert oder Positionen im Editor angesteuert werden können. Bild 93 zeigt die Arbeitsweise der Maus.

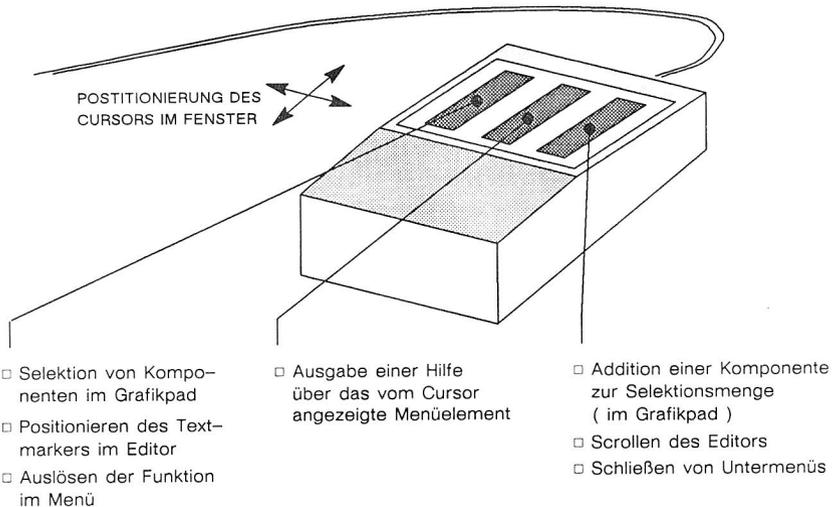


Bild 93: Funktionen der Maus in CARo

Über das Menüfeld "Ausgabe" lassen sich erstellte und getestete Steuerprogramme entweder im Verwaltungssystem speichern oder via Fabriknetz an die entsprechenden realen Geräte übertragen.

6.2 Anwendungen von CARo

Im Rahmen von Montageplanungsprojekten konnte CARo mehrfach wirkungsvoll eingesetzt werden (siehe /78/). Neben der Ermittlung von Taktzeiten hat sich CARo vor allem als wertvolle Hilfe bei der Auswahl von Planungsalternativen erwiesen. Am Beispiel mehrerer Projekte die teilweise in Zusammenarbeit mit der Industrie ausgeführt worden sind, sollen verschiedene Anwendungen von CARo vorgestellt werden.

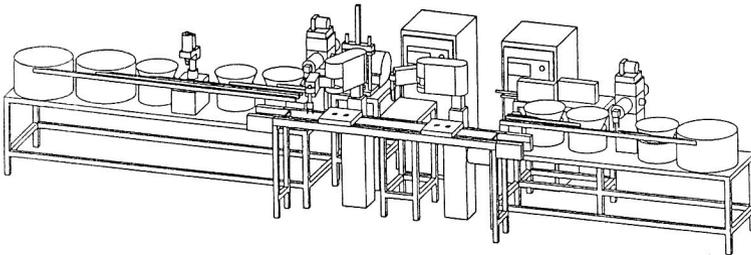
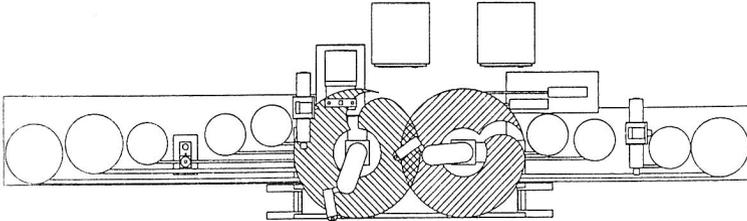
6.2.1 Komplettmontage mit Gelenkrobotern

In Bild 94 ist eine Roboterzelle mit 2 Gelenkrobotern und 2 NC Achsen dargestellt. Diese Zelle soll den in Bild 90 dargestellten Schütz montieren. Roboter 1 (links) hat dabei 7 und Roboter 2 (rechts) parallel dazu 8 Einzelaufgaben zu verrichten. Mit CARo sollte:

1. der Montageablauf optimal synchronisiert werden,
2. die Montagezeit ermittelt werden,
3. kritische Bewegungsabläufe im gemeinsamen Kollisionsbereich beider Roboter untersucht werden und
4. eine Stückliste der verwendeten Systemkomponenten erstellt werden.

Neben der Simulation konnte die Anwendung von CARo selbst getestet werden. Es hat sich herausgestellt, daß die grobe Gestaltung des Zellenaufbaus innerhalb eines Tages ausgeführt werden kann, sofern alle Montagekomponenten in der Datenbasis verfügbar sind. Da bei dieser Aufgabe keine korrekten offline Programme erstellt werden mußten, war keine Detailkonstruktion

von Greifern oder Vorrichtungen notwendig. Die Projektierung und der Test der Montageanlage sowie die Erstellung der technischen Zeichnungen beinhaltete das Sammeln von Unterlagen (Produkt, Vorranggraph), die Auswahl und Positionierung der Montagekomponenten im CAD, das Bemaßen und Beschriften des Layouts durch CAD-Funktionen, das grobe Programmieren und Abtasten der beiden Scara Roboter und die Dokumentation des Ergebnisses. Das Projekt konnte nach 15 Manntagen abgeschlossen werden.



- | | |
|----------------------------|----------------------------|
| 2 Manotec h5 Roboter | 2 Siemens-MHS-2Achsgeräten |
| 1 Biegepresse | 1 Eindrückvorrichtung |
| 9 Vibrationswendelförderer | 4 Tische |
| 1 Doppelgurtförderer | 2 Werkstückträger |

Bild 94: Komplettmontage (modelliert und getestet in CARo)

6.2.2 Montagelinie mit Linearachsensystem

Alternativ zu der bezeichneten Komplettmontage mit Gelenkrobotern wurde die Herstellung des gleichen Teils in einer Montagelinie geplant, in der vor allem programmierbare NC-Achsen eingesetzt werden sollten. Zu untersuchen war insbesondere die in Bild 95 dargestellte Zelle zur Bestückung eines Werkstückträgers mit

diversen Baugruppen eines Schütz (Bild 90). Die Zelle wurde als kritische Station der Linie betrachtet und mußte daher hinsichtlich Taktzeit überprüft werden.

Die geforderte Taktzeit in der Linie betrug 13,6 Sekunden. Bei der Simulation des Bewegungsablaufes mit CARO wurde im ungünstigsten Fall (=reine Linearbewegungen, nicht optimierte Bewegungsfolge) eine Taktzeit von 10,52 Sekunden und im optimalen Fall eine Taktzeit von 9,87 Sekunden errechnet. Zudem konnte festgestellt werden, daß durch eine geschickte Greiferkonstruktion (Mehrfachgreifer statt Wechselgreifer) die Ausführungszeit der Zelle noch erheblich verkürzt werden könnte. Dies wurde allerdings aufgrund einer bereits vorhandenen Sicherheit von 27% nicht weiter erörtert.

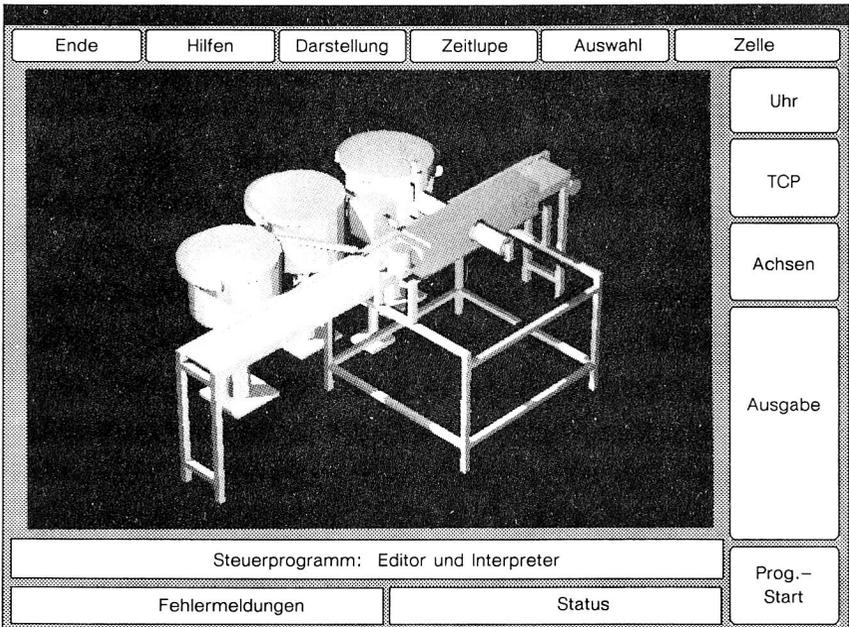


Bild 95: Linearachsroboter zum Bestücken (Motiv aus CARO)

6.2.3. Manuelle Montage

Auch die Gestaltung und Simulation manueller und teilautomatisierter Arbeitssysteme ist mit CARo möglich. Für den in Bild 96 dargestellten Arbeitsplatz konnten folgende Punkte rechnergestützt ermittelt werden:

1. Gestaltung des ergonomischen Arbeitsplatzlayouts aus Komponenten in der Komponentenbibliothek.
2. Erstellen von Zeichnungen und Stücklisten durch CAD-Funktionalität.
3. Montagezeitermittlung durch Simulation des Bewegungsablaufs.
4. Ablaufprogrammierung und Synchronisation der Teilver- und entsorgung.

Der in Bild 96 dargestellte teilautomatisierte Arbeitsplatz dient zur Sonderbestückung von Leiterplatten. Während der eigentliche Bestückvorgang samt Entnahme der Leiterplatten aus dem Zwischenspeicher manuell erfolgt, ist die gesamte Transportaufgabe in der Zelle durch ein ausgeklügeltes System bestehend aus Rollenbahnen, Aufzug, Schiebereinrichtung und einem FTS mit Hebeeinrichtung automatisiert gelöst.

Die Besonderheit bei der Anwendung von CARo auf manuelle Arbeitssysteme besteht in der Art der Zeitermittlung. Bewegungsabläufe des Menschen sind mit einem Modell nur schwer nachvollziehbar, Zeiten und Geschwindigkeiten analytisch kaum berechenbar. Die absolut korrekte Simulation des Bewegungsablaufes wird auch in Zukunft kaum möglich sein, da der Steuerungs- und Regelmechanismus des Menschen nicht bekannt bzw. nicht genügend erforscht ist.

Bei einer rechnergestützten Ermittlung der Montagezeit kann aber nach der Methode der Systeme vorbestimmter Zeiten vorgegangen werden. Dabei lassen sich Einflußgrößen -wie Bewegungsdistanz, Werkstückgewichte, Greifbedingungen- aus dem Modell ableiten oder vom Systembenutzer erfragen und Zeiten aus rechnerintern gespei-

cherten Tabellen ermitteln /79/, /80/. Die zusätzliche nicht ganz korrekte Bewegungssimulation -nicht exakt wegen des vereinfachten Menschmodells- liefert dann Informationen, ob die Montageaufgabe vollständig ausgeführt wird, ob bestimmte Bewegungen zu ungünstigen Körperhaltungen führen und ob die Bewegungen entfernt genug von Gefahrenstellen erfolgen.

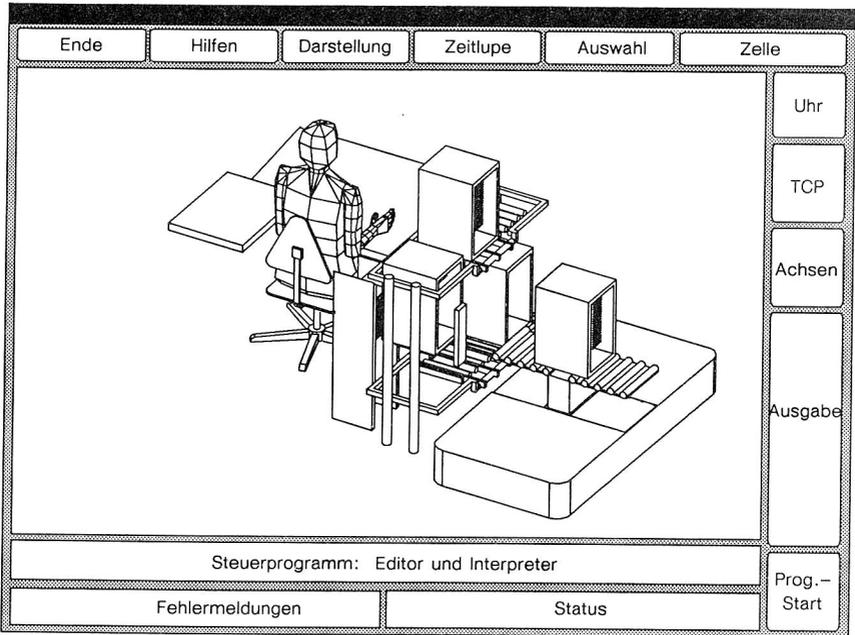


Bild 96: Teilautomatisierter Montagearbeitsplatz (Motiv aus CARO)

Wie bei Roboterarbeitszellen können natürlich im Modell verschiedene Bereiche -wie Arbeitszentrum, Einhandzone, aber auch Blickfeld und Umblickfeld- eingeblendet werden. Auf diese Weise kann durch CARO auch die ergonomische Arbeitsplatzgestaltung unterstützt werden.

6.3 Systemtest und Testauswertung

Für Projektarbeiten und die in 6.2 beschriebenen Anwendungen wurde CARo von Experten mit ausgeprägten DV- und Ingenieurkenntnissen bedient. Aufgrund der Forderung, daß das Planungswerkzeug auch Nicht-DV-Spezialisten in der Montage zur Verfügung stehen soll, mußte unter kommunikationsergonomischen Gesichtspunkten auch getestet werden, wie schwer oder leicht sich CARo erlernen läßt und welche Probleme beim Einsatz in der Praxis auftreten können. Schwierig war es dabei ausreichend Versuchspersonen verschiedener Benutzerklassen (siehe Bild 29) zu finden und in angemessenem Zeitraum zu testen. In einer Studie /81/ konnte an der Universität Erlangen-Nürnberg eine Versuchsreihe mit 66 Fertigungsingenieuren (in Ausbildung), 36 Informatikstudenten (in Ausbildung) und 5 CARo-Experten durchgeführt werden. Ziel des Versuchs war es festzustellen, ob das System CARo technologieorientiert bedienbar ist, von den Versuchspersonen in vollem Umfang innerhalb einer begrenzten Zeit erlernt werden kann und nach der Einlernphase als Planungswerkzeug akzeptiert wird. Zudem sollte festgestellt werden, ob die Funktionalität von CARo den Montageplaner wirkungsvoll unterstützen kann. Folgende Messungen wurden durchgeführt:

- Erlernbarkeit, Einarbeitungszeit (Messen)
- Akzeptanz spezieller Funktionen (Zählen)
- Auswirkung des Antwortzeitverhaltens auf die
Arbeitsgeschwindigkeit (Vergleichen)
- Arbeitszufriedenheit (Bewerten)

Zusätzlich wurden Systemfehler, Systemabstürze, Fehlbedienungen und sonstige Störungen registriert. Die Reaktion der Testpersonen auf Fehlermeldungen oder andere Systemreaktionen (z.B. Bedienblockierung während eines langen Rechengangs) wurden qualitativ erfaßt und beurteilt. Es wurden weiterhin Vergleiche angestellt, wie sich die Motivation -also die Bereitschaft den Versuch durchzuführen- auf das Arbeitsergebnis auswirkt.

Für den Versuch wurde unter den o.g. Zielen und Meßkriterien eine abgegrenzte Aufgabe entwickelt, die folgende Punkte beinhaltet:

1. Theoretische Einarbeitung anhand der Bedienerführung und der Versuchsbeschreibung.
2. Kurze praktische Vorführung des Systems.
3. Überprüfung der Vorkenntnisse anhand eines Testats.
4. Gezieltes Einlernen anhand eines Leitfadens.
5. Test mit Zeitmessungen, Zählungen und Bewertungen.
6. Befragung und Beurteilung anhand eines Fragenkataloges.

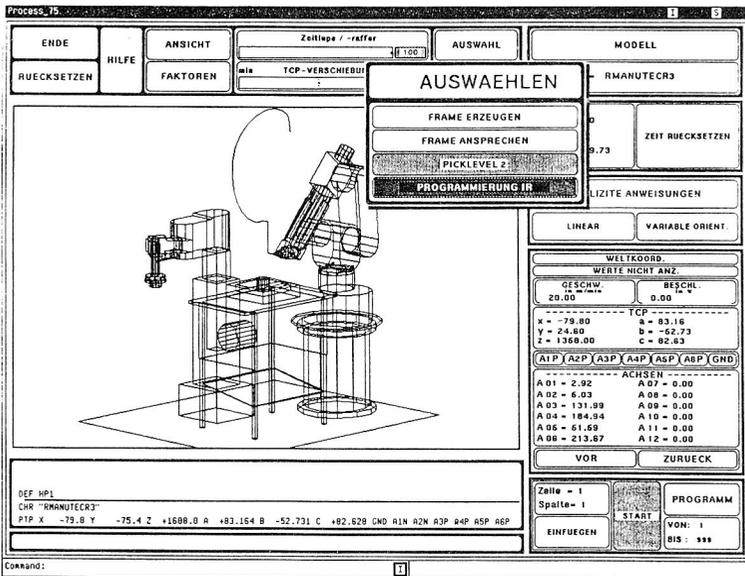


Bild 97: Testaufgabe : CARo zur Planung eines Handhabungs- und Klebevorgangs (Bildschirmkopie)

Die Punkte 1-4 dienen der Schaffung möglichst gleicher Voraussetzungen (Systemkenntnis) für den eigentlichen Versuch. Das Testat (Punkt 3) dient außerdem der Beurteilung der Motivation, mit der die Probanden an den Versuch herangehen. Im Versuch selbst (Punkt 5) soll eine einfache aber typische Montageaufgabe mit zwei Industrierobotern programmiert und simuliert werden.

Dazu mußten folgende Tätigkeiten mit CARo ausgeführt werden:

- Überprüfung des Arbeitsraumes eines 6-Achs-Gelenkroboters,
- Zuführen eines Werkstückes mit einem Scara Roboter,
- Setzen eines Klebepunktes mit einem 6-Achs-Gelenkroboter und Klebepistole,
- Zusammenfassen der Aktionen im Steuerprogramm,
- Messen der Zeiten durch Programmsimulation.

Die gesamte Aufgabe (siehe Bild 97) wurde zunächst von Experten ausgeführt. Diese konnten bereits nach etwa 90 Minuten alle Aufgaben vollständig abschließen. Die Bearbeitungszeiten der anderen Versuchsteilnehmer variierte aufgrund unterschiedlicher Motivation und unterschiedlicher Leistungsfähigkeit zwischen 100 und 180 Minuten. Erstaunlich war, daß lediglich 6% der Probanden den Versuch erfolglos nach 180 Minuten abbrechen mußten. Der Grund für den Abbruch lag weniger in der mangelnden Befähigung als vielmehr in der stark emotionalen Ablehnung des Versuchs. Dabei konnte nicht festgestellt werden, ob diese Ablehnung sich grundsätzlich gegen jede Art der Rechneranwendung richtet oder speziell gegen diesen Versuch.

Die Motivation und die Leistungsfähigkeit der Versuchsteilnehmer wurde anhand des Testats festgestellt. Die Anzahl der Fehler auf einfache, im multiple-choice Verfahren gestellte Fragen, gab Aufschluß über die Lernbereitschaft und die Lernfähigkeit. Beide Faktoren wirken sich stark auf Bearbeitungszeit und Qualität des Arbeitsergebnisses aus.

Bild 98 liefert mehrere Aussagen. Zum einen zeigt das Diagramm den starken Einfluß der inneren Einstellung und der Lernfähigkeit auf die Bearbeitungszeit. Zum anderen ist die zeitliche Spanne der Bearbeitungszeiten dargestellt. Die Unterscheidung zwischen Informatikern und Fertigungstechnikern liefert die Aussage, daß Spitzenzeiten am ehesten von Ingenieuren mit technologieorientierter Arbeitsweise zu erreichen sind.

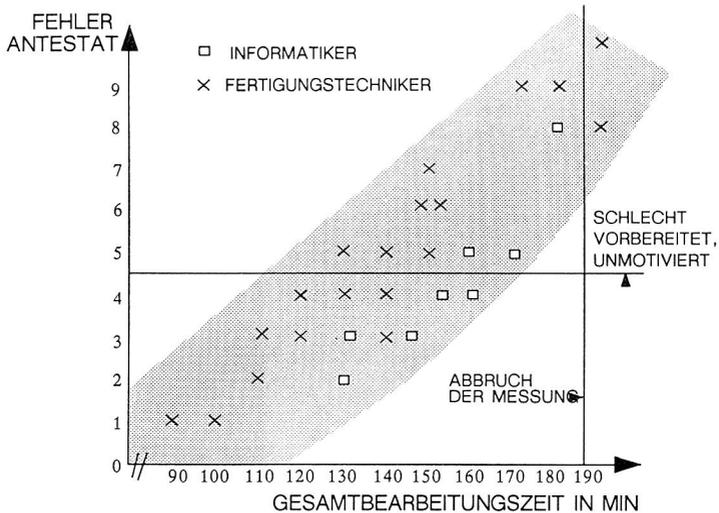


Bild 98: Einfluß der Vorbereitungszeit und der inneren Einstellung auf die Bearbeitungszeit

In enger Relation zur Bearbeitungszeit steht die Arbeitszufriedenheit, welche letztendlich über die Akzeptanz eines Systems Aufschluß gibt. Durch einen Fragenkatalog und ein Bewertungsschema konnte CARo von den Versuchsteilnehmern bewertet werden. Bewertungskriterien waren u.a.:

- Arbeitsumgebung (Hardwareergonomie)
- Antwortzeitverhalten,
- Systemreaktion auf Fehler,
- mentale Belastung,
- Eignung für bestimmte Montageanwendungen, etc.

65% der Versuchsteilnehmer zeigten sich zufrieden bis sehr zufrieden. Nur wenige Probanden vergaben schlechte Noten, was deutlich auf Probleme bei der Bearbeitung einzelner Aufgaben zurückzuführen war (Bild 99). Überraschend war, daß auch Probanden mit schlechter Vorbereitung und anfangs geringer Motivation das System CARo schließlich mit guten Noten bewertet haben.

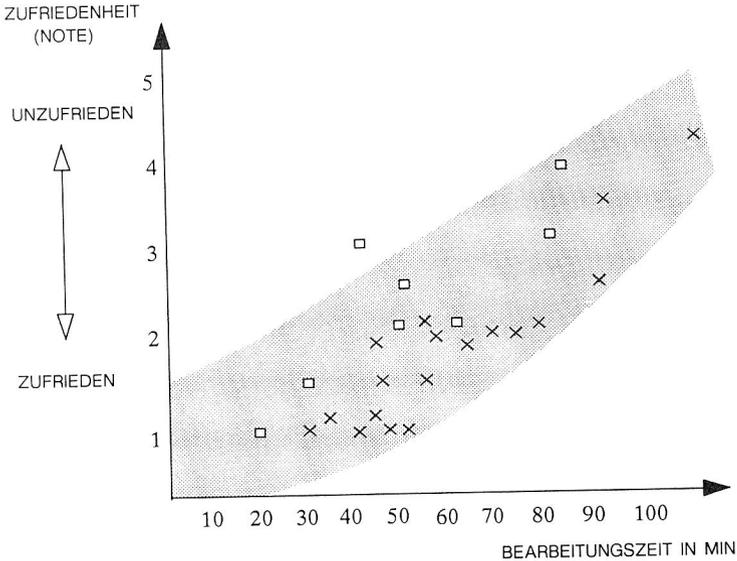


Bild 99: Arbeitszufriedenheit zu Bearbeitungszeit

Von unterschiedlicher Bedeutung war die Frage, inwieweit bei einem Planungswerkzeug für die Montage von Grafikfunktionen Gebrauch gemacht wird und wie sich deren Gebrauch meßbar positiv auf die Qualität des Arbeitsergebnisses auswirkt. Die Qualität des Arbeitsergebnisses wurde anhand der im Versuch zu messenden Taktzeit beurteilt. Je kürzer die gemessene Taktzeit für das erzeugte Steuerprogramm, desto besser das Planungsergebnis. Bild 100 zeigt, daß bei häufiger Verwendung von Bildmanipulationsfunktionen das Arbeitsergebnis verbessert werden kann. Diese Aussage untermauert die Forderung nach starken grafischen Fähigkeiten von Planungswerkzeugen für Ingenieure. Interessant ist auch, daß Bildmanipulationsfunktionen offenbar stärker von Fertigungstechnikern als von Informatikern genutzt werden. Diese Erkenntnis konnte auch durch die Art der Programmierung gewonnen werden, welche von den zwei unterschiedlichen Benutzergruppen zur Aufgabenlösung eingesetzt werden. CARo bietet nämlich die Möglichkeit sowohl durch grafisches Teach-In Roboter zu programmieren,

als auch durch textuelle Eingabe des Steuercodes. Während 71% der Informatiker der textuellen Eingabe den Vorzug gaben, taten dies nur etwa 20% der Fertigungsingenieure. Nach der Einlernphase wurde allerdings auch festgestellt, daß größtenteils die gemischte Programmierung -also textuelle und grafische - von beiden Gruppen eingesetzt wurde.

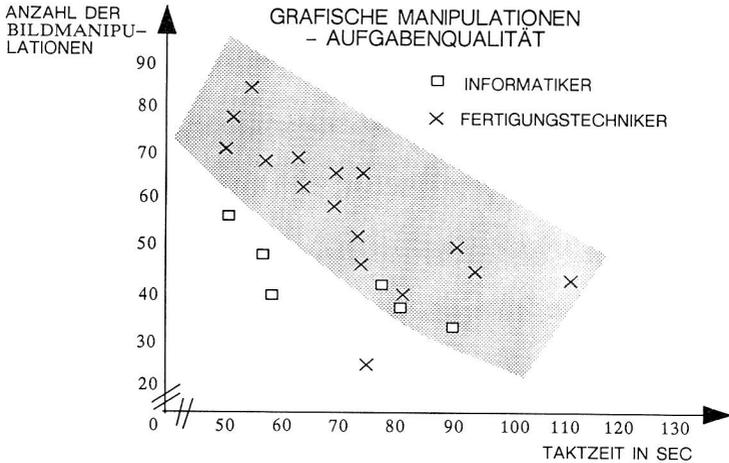


Bild 100: Einfluß der grafischen Operationen auf das Arbeitsergebnis

Zusammengefaßt ergab die Versuchsauswertung, daß die gestellte Montageaufgabe mit CARo geplant werden kann, daß 94% der Versuchspersonen nach kurzer Zeit in der Lage waren, diese Aufgabe zu erfüllen. Dabei zeigt sich die starke Auswirkung der Einführungsphase auf die spätere Anwendung. Ingenieurgerechtes Arbeiten setzt verstärkten Einsatz von Grafik voraus; Rechnerwerkzeuge wie CARo können mit ihren umfangreichen grafischen Fähigkeiten als effektive Werkzeuge auch in der Montageplanung eingesetzt werden. Die schnelle und einfache Methode der Taktzeitermittlung durch Simulation von Bewegungsprogrammen anstelle der Berechnung mit Weg/Zeit Diagrammen, sowie das gefahrlose Arbeiten mit Robotermodellen statt der nicht ungefährlichen Programmierung am teuren realen Gerät, trägt zur Sicherung der Arbeitszufriedenheit bei.

7. Integration von CAP-Werkzeugen in den Produktionsbetrieb

Die Integration von CAP-Werkzeugen in den Produktionsbetrieb kann nicht allein durch die Entwicklung technisch ausgereifter Rechnersysteme erreicht werden. Diese sind lediglich Grundvoraussetzung für eine erfolgreiche Einführung. Integration bedeutet vielmehr Anpassung von verfügbaren Systemen an das individuelle Anforderungsprofil eines Unternehmens. Dabei spielen die Aufbauorganisation, Abläufe in der technischen Auftragsabwicklung, vorhandene Systeme und Maschinen, sowie die Personalstruktur des Unternehmens die wichtigste Rolle.

Integration fordert die durchgängige Verknüpfung von Abteilungen sowie deren Systemen (CAD, CAM) und ermöglicht so die Realisierung rechnerintegrierter Verfahrensketten /82/. Integration heißt auch Akzeptanz von neuen Arbeitswerkzeugen und Methoden durch die Mitarbeiter.

Durchgängigkeit:

- Vernetzung von Rechnern in Konstruktion, Arbeitsplanung und Fertigung (Hardware),
- Zugriffe auf gemeinsame Datenbestände,
- Anpassung von Rechnermodellen an die Wirklichkeit.

Akzeptanz:

- Partizipation der Mitarbeiter bei Systemanpassung,
- Qualifikation durch Schulung und Weiterbildung,
- Motivation der Mitarbeiter.

Der Erfolg bei der Einführung von CAP-Werkzeugen kann durch eine systematische Vorgehensweise (siehe Bild 101) gesichert werden. Wesentlich ist, daß die Einführung unter einem ganzheitlichen Konzept und unter einer globalen Zielsetzung durchgeführt wird.

Wichtig bei der Systemintegration ist die klare Definition von Anforderungen und Schnittstellen, die sich aus der IST Analyse

und den darin ermittelten Schwachstellen ergeben. Je sorgfältiger und präziser verarbeitet wird, desto größer der Erfolg beim späteren Einsatz.

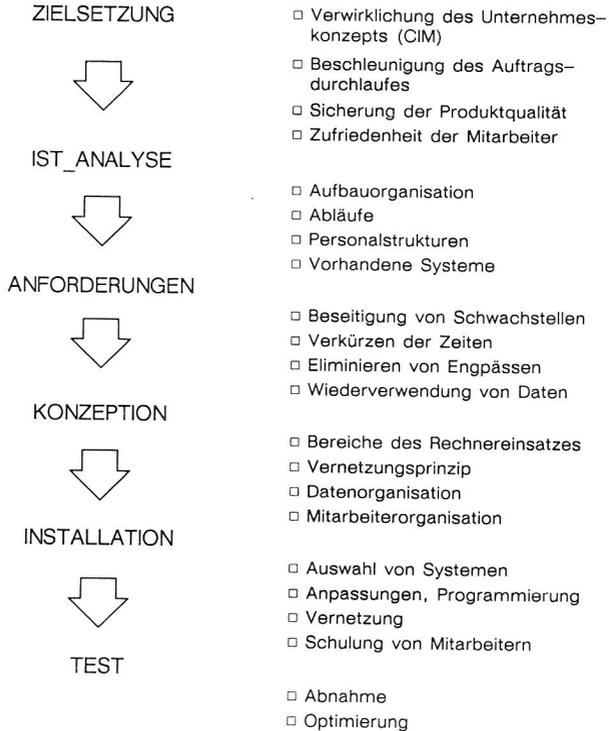


Bild 101: Vorgehensweise bei der Einführung von CAP-Werkzeugen

Da, wie bereits erläutert, Integration eine höchst individuelle Maßnahme ist, kann in den folgenden Punkten nur pauschal auf Integrationskonzepte eingegangen werden. Dazu wird in Kapitel 7.1 gezeigt, wie die technische Einbindung von CAP-Systemen in den Betrieb erfolgen kann und welche Anpassungen dabei vorgenommen werden müssen. Kapitel 7.2 beschreibt ergänzend die Maßnahmen zur Vorbereitung der Mitarbeiter auf die Systemeinführung.

7.1 Rechnerverbund und Fabrikkommunikation

Ganzheitliche CIM-Konzepte sehen die Verknüpfung aller Unternehmensbereiche über ein fabriktweites Kommunikationsnetzwerk vor. Das in Kapitel 6 vorgestellte System CARo wird auf 32 Bit Workstations entwickelt, die im Ring vernetzt sind. Dabei hat jeder Anwender die Möglichkeit, stets auf sämtliche Daten der anderen Rechner im Ring zurückzugreifen. Dieses Ringkonzept ist geeignet für den technischen Planungsbereich, da die vielfältigen und oft komplexen Informationsflüsse zwischen Konstruktion und Arbeitsplanung, wie sie auch im System CARo beschrieben werden, auf einheitlicher Hardware mit integrierter Datenhaltung sicher gewährleistet sind.

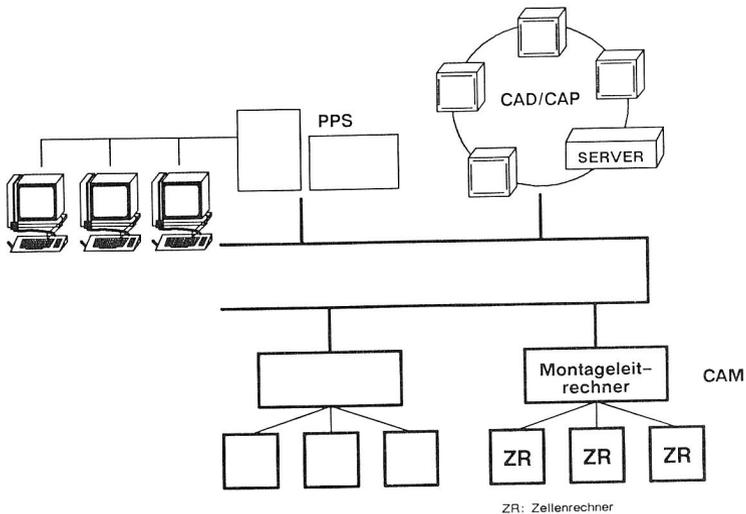


Bild 102: Integrierte Fabrikkommunikation

Die Rechner im Ringverbund sind an ein Fabriknetzwerk angekoppelt, wodurch der Planungs- mit dem Fertigungsbereich verbunden wird (siehe Bild 102). Steuerprogramme und andere Planungsunterlagen, wie Schaltpläne, Arbeitspläne, Prüfpläne können somit an ein Fertigungsleitsystem oder eine Fertigungsdatenbank übergeben werden. Von seiten der administrativen Auftragsabwicklung (PPS)

werden zum einen Auftragsdaten, wie Stückzahlen (Produktionsmenge) und Termine in den Fertigungsbereich übermittelt, zum anderen werden Zustandsdaten zur Auftragsverfolgung aus der Fertigung bzw. Montage übernommen. Die Zuordnung von Auftragsdaten und Steuerprogrammen erfolgt über die Kennzeichnung und Identifikation der zu fertigenden Produktvariante. So kann ein Montagerechner (anhand der Produktbezeichnung im Auftrag) Steuerprogramme und andere Fertigungsdaten in seiner Datenbasis finden und den Auftrag nach Menge, in den entsprechenden Montagezellen und mit den passenden Steuerprogrammen ausführen. Zur Optimierung der Planungsunterlagen müssen sowohl an die technische als auch an die administrative Auftragsabwicklung Daten aus dem Prozeß zurückgeliefert werden. Das PPS-System kann dadurch eine optimierte Terminierung und Kapazitätsauslastung sicherstellen. Im technischen Bereich können durch die Aufnahme von Prozeßdaten beispielsweise Anpassungen und Korrekturen der Roboterprogramme vorgenommen werden. Die Fabrikkommunikation wirkt damit als Regelkreis, in dem Planungsdaten zur Fertigung und Prozeßdaten als Stellgrößen zurück zur Planungsebene fließen.

Die in CARo erstellten Steuerprogramme werden nach der VDI-Richtlinie 2863 oder nach DIN 66215 in steuerungsneutralen IRDATA bzw. CLDATA-Formaten übertragen. Sie können so von unterschiedlichen Steuerungssystemen verstanden und verarbeitet werden. Die Übertragung der Programme an den Montagerechner erfolgt über einen SINEC_H1 Bus für den speziell die Schnittstelle RS 511 - auch MMS (Manufacturing Message Service for Bidirectional Transfer of Digitally Encoded Information)- implementiert wurde /84/.

Um Ungenauigkeiten abzugleichen, müssen alle Steuerungsprogramme in der realen Umgebung der Montagezelle getestet werden. Ungenauigkeiten treten insbesondere bei feinmotorischen Operationen der Bewegungssysteme auf. Sie entstammen den unterschiedlichen geometrischen Abmessungen und den abweichenden dynamischen Eigenschaften der bewegten Systeme.

Eine Möglichkeit, Fehlerquellen auszuschalten, besteht in der Ausführung von Kalibrierungsprogrammen zur Anpassung der geometrischen Informationen im Zellenmodell an die reale "Welt". Dazu kann nach folgendem Schema vorgegangen werden:

- Festlegen von Meßstellen beim Modellieren,
- Erstellen von Meßprogrammen (zum Anfahren der Meßpunkte mit Berührungssensoren),
- Übertragung der Meßprogramme an das Montagesystem,
- Aktualisierung der geometrischen Abmessungen im Modell (durch Rückgabe der gemessenen IST-Werte).

Durch intelligente bzw. sensorunterstützte Steuerungen kann u.U. vollständig auf exakte geometrische Daten in offline erstellten Bewegungsprogrammen verzichtet werden. Die tatsächlichen Anfahrpositionen können dann online durch Sensoren ermittelt und an das Bewegungsprogramm übertragen werden. In diesem Fall kann auf die Anpassung des Zellenmodells verzichtet werden, da das zu programmierende Montagesystem tolerant auf Abweichung reagiert. Beide Möglichkeiten -Kalibrierung bzw. online Anpassung sind derzeit Gegenstand der Forschung, so daß weder der einen noch der anderen Methode der Vorzug gegeben werden kann.

7.2 Personelle und organisatorische Integration

Der Innovationsprozeß bei der Integration rechnergestützter Werkzeuge und Methoden spielt sich nicht alleine auf technischer, sondern auch auf personalorganisatorischer Ebene ab. Beispiele aus der Vergangenheit, wie bei der Einführung von CAD, haben gezeigt, welche Probleme unerwartet aufgrund einer nicht angepaßten Personalpolitik auftreten können. Trotz der starken Förderung durch staatliche Programme in den 80iger Jahren konnte sich der Einsatz von CAD-Systemen in der mittelständischen Industrie nur schwer entfalten. Weder technische noch wirtschaftliche Gründe -bei 50% Förderung durch den Staat- konnten dafür gefunden werden. Die Ursachen lagen vielmehr in Unzulänglichkeiten wie:

- Vorhandene Qualifikationsstruktur im Unternehmen versus erforderlichen Qualifikationsbedarf,
- Abteilungsstrukturen (Abteilungsdenken) versus Kooperationsbedarf,
- Hierarchische Strukturen versus Kompetenzverteilung (Ängste vor Kompetenzverlust).

Bei der Einführung von neuen 'C'-Techniken muß heute die Personalpolitik als integraler Bestandteil und nicht wie bisher als Anhängsel einer CIM-Strategie berücksichtigt werden/85/. Dies bedeutet: Abstimmung alter Strukturen auf die neuen Erfordernisse und Ausgleich widerstrebender betrieblicher Ziele (Automatisierung versus Arbeitszufriedenheit!)

Um eine Verbesserung der Arbeitsorganisation und der Mitarbeiterqualifizierung zu erreichen, ist zunächst festzustellen, welche neuen Anforderungen und Belastungen durch den Einsatz von vernetzten Rechnersystemen im Unternehmen -speziell in der Montageplanung- auftreten. Es ist stets zu berücksichtigen, daß gezielte Veränderungen in bestimmten Abteilungen sich nicht allein dort auswirken, sondern auch Folgen für andere angrenzende Bereiche haben werden.

Die Bedingungen für die Montageplanung haben sich in den vergangenen Jahren durch den drückenden Einfluß des Weltmarktgeschehens wesentlich gewandelt. Kürzere Produktlebenszyklen und sinkende Stückzahlen erfordern den Einsatz von hochflexiblen Produktionssystemen einerseits und effizienten Planungswerkzeugen andererseits. Durch die Anwendung des Rechners können Planungsalternativen schneller und sicherer entwickelt und durch Simulation optimiert werden. Das Qualifikationsprofil des Planers wird dadurch entscheidend verändert. Diese Veränderungen rühren im wesentlichen von der unterschiedlichen Arbeitstechnik und Arbeitsgeschwindigkeit mit dem Rechner. Alte Aufgaben verschwinden (Zeichnungserstellung), neue Aufgaben treten hinzu (Bedienung der Rechnerhardware, Einsatz von Software).

Im folgenden werden die veränderten Arbeitsbedingungen beim Einsatz eines Planungs- und Programmierwerkzeuges in der Montagefeinplanung beschrieben.

a) Kenntnisse, Ausbildung und Erfahrung

Das Arbeiten mit Rechnern verlangt eine grundsätzlich andere Arbeitstechnik. Im Gegensatz zu einer individuellen Vorgehensweise beim Arbeiten mit Zeichenbrett, Papier und Bleistift verlangt jedes Rechnersystem eine disziplinierte Vorgehensweise. Auch wenn die Benutzerführung größte Freiheiten (z.B. durch benutzerinitiierten Dialog) zuläßt, hat sich der Planer an einige Regeln zu halten. Zum Beispiel verlangt der Rechner eine syntaktisch korrekte Eingabe, die zudem über ungewohnte Bedienelemente, wie Maus, Pen oder Keyboard erfolgen muß. Solche Konventionen haben bei vielen Anwendern unerwartet große Probleme hervorgerufen und sind deshalb in einer geeigneten Einlernphase besonders zu berücksichtigen.

b) Geschicklichkeit

Der Umgang mit der Eingabeperipherie erhöht in der Regel die Anforderungen an die Handfertigkeit in der Montageplanung nicht. Stärker als bisher ist jedoch die geistige Wendigkeit des Anwenders gefordert. Dieser muß in kürzester Zeit eine größere Informationsdichte verarbeiten und im Rhythmus des Dialogsystems reagieren können.

c) Belastung der Nerven und Sinne

Der Rechneinsatz wird eine erhöhte Aufmerksamkeit des Anwenders erfordern /86/. Durch die Verdichtung von Rechenprozessen, Reduzierung von Arbeitsschritten, paralleler Datenausgabe wird es für den Planer schwieriger, die Herleitung des Arbeitsergebnisses nachzuvollziehen. Allerdings bleibt ihm in Summe mehr Zeit, um

Alternativen zu entwickeln oder die Qualität des Arbeitsergebnisses zu prüfen. Dabei entlastet ihn der Rechner von Routinetätigkeiten, wie Berechnen von Taktzeiten, Erstellen von Stücklisten und Zeichnungen. Belastend wirkt sich die vorwiegend sitzende Tätigkeit und das Arbeiten an Sichtgeräten aus. Probleme beim Sehen werden durch die Arbeit mit 3D-Grafiken noch verstärkt. So ist das Größenverhältnis von Bildschirmmodellen zu realen Montagezellen etwa 1:100. Durch eingeschränkte Darstellungsmöglichkeiten -z.B. Drahtgrafik- wird das Arbeiten am Sichtgerät zusätzlich erschwert.

d) Verantwortung

Die Anforderungen aufgrund von Verantwortung beim Einsatz des Rechners als Planungswerkzeug werden sich nicht wesentlich verändern. Fachlich bleibt die Aufgabe des Anwenders gleich. U.U. kann die Verantwortung in einigen Punkten durch den Rechner reduziert werden -Verantwortung für die Richtigkeit von Berechnungen, für DIN-gerechte Zeichnungserstellung, für die Vollständigkeit von Planungsunterlagen. Andererseits wird mit teureren Arbeitsmitteln (Hardware) umgegangen und Disziplin bei Datenarchivierung und -sicherung gefordert.

e) Umgebungseinflüsse

Auch die Anforderungen durch Umgebungseinflüsse werden sich kaum ändern. Der Arbeitsplatz des Planers kann bei dezentralen Rechnerkonfigurationen in der gleichen Arbeitsumgebung bleiben wie bisher. Dazu sind allerdings Räume erforderlich, welche die starke Wärmeentwicklung der Arbeitsstationen abbauen können (Klimaanlage). Voraussetzung für gleichbleibende Qualität der Büroumgebung ist eine Minimierung der Geräuschentwicklung von Plattenlaufwerken oder Druckern. Es empfiehlt sich eine Isolierung von störenden Peripheriekomponenten (z.B. schallgeschützte Räume für Drucker und Plotter). Störende Einflüsse aufgrund von Vibrationen und ungünstigen Beleuchtungsverhältnissen können in der Regel mit arbeitsplatzgestaltenden Maßnahmen beseitigt werden.

Befragungen und Analysen haben ergeben, daß Akzeptanzprobleme beim Rechnereinsatz weniger aufgrund von erhöhten Anforderungen entstehen, sondern vielmehr aufgrund motivierender bzw. demotivierender Faktoren (siehe Bild 103).

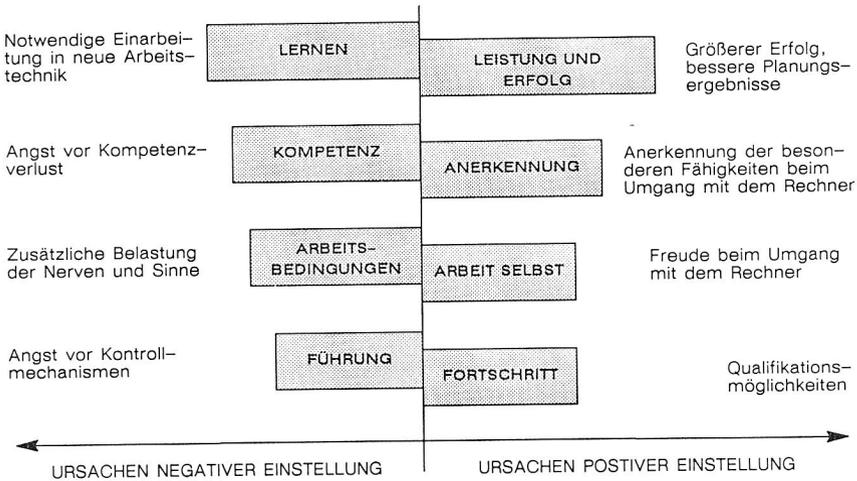


Bild 103: Motivations- und Konsistenzfaktoren beim Einsatz von rechnergestützten Planungswerkzeugen (CAP).

Besondere Probleme zeigen sich bei lernungsgewohnten, meist älteren Mitarbeitern. Diese sehen sich häufig nicht in der Lage, neue Arbeitstechniken anzunehmen. Es wird befürchtet, daß das über Jahre gesammelte Erfahrungswissen an Bedeutung gegenüber neuem Wissen über Rechnerwerkzeuge verliert und damit ein Verlust an Kompetenz und an besonderen Vorzügen bzw. Befugnissen entsteht. Eine Strategie zur Einführung von Rechnerwerkzeugen wird diese, vorwiegend emotionalen aber schwerwiegenden Faktoren berücksichtigen.

Kritisch ist es auch, die personelle Trennung von konventioneller und rechnergestützter Planung vorzunehmen. Dabei ist unter Umständen eine Zweiklassenbildung mit vielen Problemen vorprogrammiert.

Grundsätzlich sollte also die Einführung des Rechners von allen betroffenen Mitarbeitern getragen werden. Um dies zu gewährleisten, wird, wie in der Arbeitsgestaltung üblich, parallel in zwei Aktionen vorgegangen (siehe Bild 104).

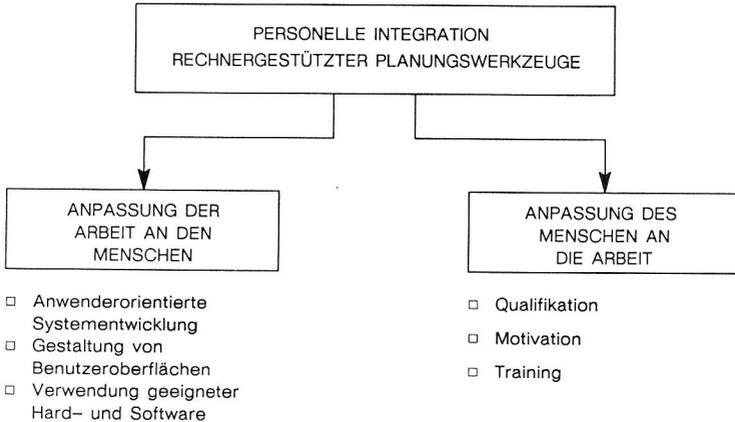


Bild 104: Mitarbeitergerechte Einführung rechnergestützter Werkzeuge

Bei einer Systemeinführung ist der größte Erfolg erfahrungsgemäß dann sichergestellt, wenn diese von Beginn an im Einvernehmen mit den betroffenen Abteilungen und dem Betriebsrat durchgeführt wird. Letztendlich muß jede Innovation von allen Beteiligten getragen werden.

Stufe 1:

Bereits im Rahmen der Zielsetzung sollte allen Mitarbeitern die Notwendigkeit zur Innovation verständlich gemacht werden. Erst nach der Erklärung zu einer gemeinschaftlichen Vorgehensweise können weitere Schritte unternommen werden.

Stufe 2:

Während der IST-Analyse tragen die Mitarbeiter dazu bei, Schwachstellen im konventionellen Arbeitsprozeß aufzudecken (Engpässe, Qualitätsmängel, etc.).

Stufe 3:

In der Definitionsphase wirken die Mitarbeiter mit an der Spezifikation der Anforderungen und prüfen am Ende das erstellte Pflichtenheft.

Stufe 4:

Das Konzept beinhaltet ein Ausbildungsprogramm und eine klare Beschreibung der neuen betrieblichen Organisationsstruktur. Dabei muß konkret auf die Rolle älterer und langjähriger Mitarbeiter, sowie auf die Problematik der Qualifizierung Lernungsgewohnter eingegangen werden. Neben dem Schulungsprogramm ist die Einbeziehung geeigneter pädagogischer Ausbilder erforderlich, die die Qualifizierung zielgruppenspezifisch umsetzen können. Für den dauerhaften Einsatz rechnergestützter Planungswerkzeuge ist zu überlegen, ob sich die Ausbildung über die Einlernphase hinaus als institutioneller Bestandteil des Unternehmens einrichten läßt. Die Organisationsstruktur im Konzept regelt die Kompetenzverteilung (Systembetreuer - Anwender) und berücksichtigt dabei vor allem das fachliche Erfahrungspotential langjähriger Mitarbeiter. Dabei sollte keine Spaltung zwischen konventioneller und rechnergestützter Planung erfolgen, sondern, wenn möglich, einer vertikalen Arbeitsteilung -Daten sammeln, konzipieren zu Dateneingaben, rechnergestützt ausarbeiten- der Vorzug gegeben werden.

Stufe 5:

Bei der Auswahl und Installation eines geeigneten Systems sind Auswahlkriterien und Gestaltungsalternativen den beteiligten Mitarbeitern und dem Betriebsrat transparent zu machen. Spezielle Anforderungen und Änderungswünsche der Mitarbeiter sollten bei der Systemanpassung berücksichtigt werden. Es ist zu überlegen, ob Vorabschulungen und die Beurteilung von Systemen zumindest teilweise von den späteren Anwendern selbst durchgeführt werden. Folgende Maßnahmen im Rahmen eines Qualifikationskonzepts können erfolgen durch:

- vorbereitende Seminare zum Stand der Technik,
- praktische Weiterbildung in CIM-Zentren,
- konkrete Schulung beim Systemanbieter,
- Systemanwendungen im Probetrieb,
- Erfahrungsaustausch in Anwendergruppen.

Stufe 6:

In der Testphase sind in jedem Fall Systemtests (Benchmarks) durch geeignete Mitarbeiter zu fahren. Erst so können Probleme beim praktischen Einsatz erkannt werden.

Diese Art der Vorgehensweise garantiert nicht nur eine funktionsfähige Systeminstallation, sondern sichert auch die Akzeptanz des Systems durch den Anwender.

8. Zusammenfassung und Ausblick

Sinkende Stückzahlen, kürzere Produktlebenszyklen, steigende Variantenvielfalt und die Notwendigkeit, schnell (Termine) und flexibel auf Markterfordernisse zu reagieren, setzen den Einsatz effektiver Planungswerkzeuge für die Produktion voraus. Besonders automatisierte oder rechnergeführte Fertigungs- und Montagesysteme bedürfen aufgrund ihrer hohen Investitionskosten einer sorgfältigen Planung und Optimierung. Der Trend, rechnergestützte Methoden und Verfahren in der Konstruktions- und Planungsebene einzusetzen, wird sich deshalb in den nächsten Jahren verstärken.

Am Beispiel der aufgabenorientierten Planung und Programmierung von Montagezellen läßt sich zeigen, wie sich Aufwand und Nutzen gegenüber herkömmlichen Methoden in der Planung qualitativ verhalten. Kürzere Planungs- und Programmierzeiten stehen hierbei größeren Datenmengen und verstärktem Hardwareeinsatz gegenüber.

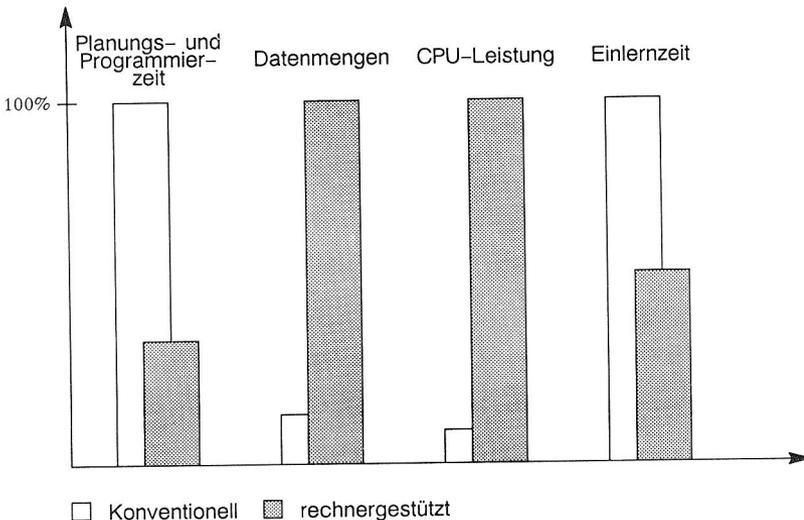


Bild 105: Vergleich von konventioneller und aufgabenorientierter rechnergestützter Planung und Programmierung

Die verkürzte Einlernphase läßt sich dadurch erklären, daß zur Programmierung von automatisierten Montageanlagen bisher Spezialisten mit Informatik- und Steuerungskennnissen erforderlich waren. Mit aufgabenorientierten Programmieretechniken kann das vorhandene Ingenieurwissen des Planers aufgrund gewohnter Arbeitstechniken effizient unterstützt werden. Datenformate (Codes) und steuerungstechnische Details brauchen den Bediener beim Erlernen der Systemfunktionen nicht zu interessieren.

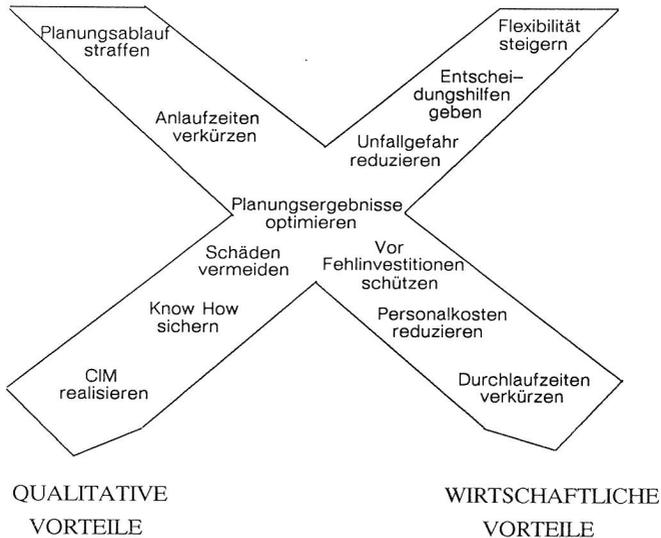


Bild 106: Qualitative und wirtschaftliche Vorteile rechnergestützter Montagefeinplanung

In Bild 106 werden die wirtschaftlichen und qualitativen Vorteile beim Einsatz von CAP-Werkzeugen in der Montagefeinplanung zusammengefaßt. Diese genannten Vorteile müssen allerdings durch einen, häufig nicht zu unterschätzenden Aufwand, erkaufte werden. Aufwand bedeutet in diesem Zusammenhang:

- Investitionen in Hard- und Software,
- hohe laufende Kosten (Wartungs- / Serviceverträge),
- Neueinstellung und/oder Schulung von qualifizierten

Mitarbeitern.

- Nebenkosten (Räumlichkeiten, Klimaanlage, usw.),
- Kosten in der Anlaufphase (Installations-, Anpassungsaufwand, Störungen von Abläufen).

Für kleine und mittlere Unternehmen stellen diese genannten Kosten meist unüberwindliche Hürden dar. Durch sinkende Hardwarepreise (siehe Bild 107) und durch das in einigen Jahren aus dem Hochschulbereich dringende qualifizierte Personal, zeichnet sich aber auch hier eine Wende ab. Studenten an fertigungstechnischen Lehrstühlen werden heute bereits intensiv auf den Einsatz moderner Planungswerkzeuge vorbereitet. So hat heute jeder Absolvent des Studiengangs Produktionstechnik Erfahrung im Umgang mit 'C-Techniken' gesammelt oder sich sogar an deren Entwicklung beteiligt.

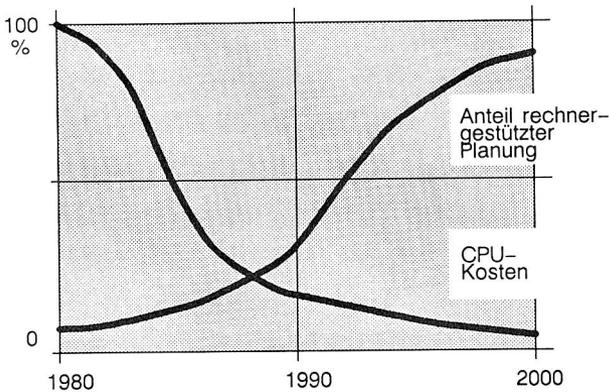


Bild 107: Entwicklungstendenzen der Hardwarekosten und der Rechnerdurchdringung in der Planung (nach /87/)

Dennoch kann eine weitere Durchdringung des Rechnereinsatzes in der Planung nur dann erfolgen, wenn die technischen Probleme dieser Werkzeuge beseitigt sind.

Planungswerkzeuge in der Montage sind derzeit immer noch Gegenstand von Forschung und Entwicklung. Probleme machen vor allem:

- zu einfache Modelle,
- große Datenmengen,
- lange Antwortzeiten,
- nicht planbare Störungen im Realbetrieb,
- eingeschränkte Computergraphik,
- mangelhafte Kommunikation.

Käufliche 'C-Systeme' bieten sehr einfache Datenmodelle an. Für die Programmierung von Robotern bedeutet dies z.B. starke Abweichungen zwischen berechnetem und realem Zeit- und Positionerverhalten. Durch eingeschränkte Kommunikation zwischen Planung und Betrieb gehen zudem viele Informationen verloren, die bereits einmal eingegeben worden sind (unzureichende Schnittstellen). Forschungslaboratorien der Industrie und der Hochschulen arbeiten intensiv an diesen Problemen.

Die vorliegende Arbeit liefert das Fundament für den Ausbau bestehender Werkzeuge und gibt Hinweise, wie diese effektiv integriert werden können.

10. Literaturverzeichnis:

- /1/ Feldmann, K.; Entwicklung und Einsatz rechnerintegrierter Produktionssysteme. ZWF-CIM Ausgabe 6/88, Carl Hanser Verlag München, S.290-295.
- /2/ Feldmann, K.; Rechnergestützte Montageplanung, Eisele, R.; Schulungsunterlagen CIM-Forum, CFA, Kleineidam, G.: SIEMENS - Moorenbrunn 1987,1988,1989.
- /3/ Kleineidam, G.: CAD-NC Kopplung, Seminarunterlagen zum zweiten und dritten CIM-Symposium, OTTI Regensburg 1988,1989.
- /4/ Levi, P.: Planen für autonome mobile Roboter, Habilitation, Springer Verlag Berlin, Heidelberg, New York, Tokyo, 1988.
- /5/ Wloka, D.W.: ROBSIM - A Robot Simulation System, 11th IMACS World Congress, University of Oslo, 1985.
- /6/ N.N.: PLACE, User Guide, McDonnell Douglas Corporation, Cypress California 1987.
- /7/ Yong, Y. F.; Offline Programming of Robots, in et al.: Handbook of Industrial Robots (Nof), Wiley & Sons, New York 1985, S.366-380.
- /8/ Zühlke, D.; Grafische Robotersimulation, Osterwinter, M.; Elektronik, Heft 10, 1985, S.153-157.
- /9/ Hemberger, A.: CAD für Industrieroboter, interne Dokumentation, FAPS Erlangen 1985.
- /10/ Kleineidam, G.: Offline Programmierung und Simulation von Industrierobotern, Stand der Technik, interne Dokumentation, FAPS Erlangen 1985.
- /11/ Lozano-Perez, T.: An Approach to Automatic Robot Brooks, R.A.: Programming, in Boyse, J. et al., Solid Modeling by Applications, Plenum Press, New York, 1984.
- /12/ Winston, P.H.: Artificial Intelligence, second edition, Addison-Wesley, London 1984.
- /13/ Mujitaba, S.; AL User's Manual, Artificial Intelligence Goldmann, R.: Laboratory, Stanford 1979.
- /14/ Liebermann, L.; AUTOPASS: An Automatic Programming Wesley, M.: System for Computer Controlled Mechanical Systems, IBM R&D J.4/77.

- /15/ Lozano-Perez, T.; LAMA: A Language for Automatic
Winston, P.H.; Mechanical Assembly, Proceedings of the
et al.: 5th IJCAI, Cambridge, MA 1977,
S.710-716.
- /16/ Popplestone, R.; An Interpreter for a Language for
Ambler, A.; Describing Assemblies, Artificial
Bellos, I.: Intelligence 14, int. Journal,
Amsterdam 1980, S.79-107.
- /17/ Weck, M.; ROBEX - ein Programmiersystem für
Eversheim, W.; numerisch gesteuerte Handhabungsgeräte,
Zühlke, D.: verbesserte Programmierung von Robotern
in der Montage, PFT-Entwicklungsnotiz,
KfK-Pft-E2, Karlsruhe, 1981, S.17-53.
- /18/ Geus, L. Konzepte für Robotersprachen: ein
Sprachentwurf, Dissertation, Univ.
Erlangen-Nürnberg, 1989.
- /19/ Blume, C.; Programming Languages for Industrial
Jakob, W.: Robots, Springer Verlag Berlin,
Heidelberg, New York, Tokyo, 1985.
- /20/ Feldmann, K.; Montageplanung, CIM Technologie
Eisele, R.; Transfer Seminar, Erlangen 1989.
Kleineidam, G.:
- /21/ Feldmann, K.; Verfahrenskette zur Planung und Pro-
Eisele, R.; grammierung von Montagezellen. ZWF-CIM
Kleineidam, G.: Ausgabe 9/87, Carl Hanser Verlag
München, S.521-527.
- /22/ Feldmann, K.; CAD für Montageplanung und montagege-
Kleineidam, G.: rechte Produktgestaltung, 7ter Deutscher
Montagekongreß Tagungsband, verlag
moderne industrie, Landsberg 1987,
S.90-100.
- /23/ Nof, S.Y.: Robot Ergonomics: Optimizing Robot Work,
in Handbook of Industrial Robots (Nof),
Wiley & Sons, New York 1985, S.549-604.
- /24/ Luczak, H.: Manuelle Montagesysteme, in Handbuch der
Fertigungstechnik, Band 5 (Spur), Carl
Hanser Verlag, München, Wien 1986,
S.460-482.
- /25/ N.N.: REFA Methodenlehre des Arbeitsstudiums,
Band 2: Datenermittlung, Carl Hanser
Verlag, München 1978.
- /26/ Lacis, K.; MTM-organisierte Montagetechnik,
7ter Deutscher Montagekongreß Tagungs-
band, verlag moderne industrie, Lands-
berg 1987, S.68-89.

- /27/ N.N.: REFA Methodenlehre des Arbeitsstudiums, Band 6: Arbeitsunterweisung, Carl Hanser Verlag, München 1978.
- /28/ Merz, F.: Intelligenz und Lernen, in Lehrbuch der Ergonomie (Schmidtke), Hanser Verlag, München, Wien 1981, S.74-88.
- /29/ Schmidt, R.F.; Thews, G.: Physiologie des Menschen, Springer Verlag Berlin, Heidelberg, New York 1976.
- /30/ Lindsay, P.H.; Norman, D.A.: Human Information Processing. An Introduction to Psychology, IFS New York, London 1972.
- /31/ Bubb, H.: Analyse der Systemdynamik, in Lehrbuch der Ergonomie (Schmidtke), Hanser Verlag, München, Wien 1981, S.286-339.
- /32/ Feldmann, K.; Kleineidam, G.: Produktionssystematik, Skript zur Vorlesung Band 2, Universität Erlangen - Nürnberg 1988.
- /33/ Scheller, J.; Sommer, E.: Hierarchisches Steuerungskonzept, für flexible Montagezellen, in atp 4/89, Oldenbourg Verlag, München 1989.
- /34/ Paul, R.P.: Robot Manipulators: Mathematics, Programming and Control, MIT Press, Cambridge, MA 1983.
- /35/ Vukobratovic, M.; Potkonjak, V.: Dynamics of Manipulation Robots, Springer Verlag, Berlin 1982.
- /36/ Brady, M.; Hollerbach, J.M.: Robot Motion: Planning and Control, MIT Press, Cambridge MA 1982.
- /37/ Wratil, P.: Speicherprogrammierbare Steuerungen in der Automatisierungstechnik, Vogel Buchverlag, Würzburg 1989.
- /38/ Zühlke, D.: Offline Programmierung numerisch gesteuerter Industrieroboter, VDI-Verlag, Düsseldorf 1983.
- /39/ N.N.: DIN 19239: Messen, Steuern, Regeln; Steuerungstechnik, Speicherprogrammierte Steuerungen, Programmierung, Deutscher Normausschuß, Beuth Verlag, Berlin 1983
- /40/ Rieger, K.H.: Rechnergestützte Projektierung der Hardware und Software von SPS, Springer Verlag, Berlin Heidelberg N.Y. 1985.
- /41/ Spur, G.; Krause, F.-L.: CAD - Technik, Hanser Verlag, München, Wien 1984.

- /42/ Kleineidam, G.; Eisele, R.: Neue Sprachregelung: Entwicklungsschritte auf dem Weg zur aufgabenorientierten Roboterprogrammierung (Teil 1), Zeitschrift moderne fertigung 9/88, verlag moderne industrie, Landsberg 1988, S. 47-50.
- /43/ Balzert, H.: Die Entwicklung von Software-Systemen, B.-I. Wissenschaftsverlag, Mannheim, Wien, Zürich 1982.
- /44/ Eisele, R.; Kleineidam, G.: Pflichtenheft für ein explizites offline Programmier- und Simulationssystem, interne Projektdokumentation, FAPS Erlangen, März 1986.
- /45/ Nievergelt, J.: Interactive Systems for Education - the new look of CAI, Proc. IFIP World Conf.: Computers in Education, North Holland, 1975, S. 465-471
- /46/ Nolting, F.-W.: Projektierung von Montagesystemen Dissertation, Carl Hanser Verlag, München, Wien 1989.
- /47/ N.N.: DIN 66234, Teil 8: Dialoggestaltung (Entwurf), Deutscher Normausschuß, Beuth Verlag, Berlin 1984.
- /48/ Bullinger, H.J.; et. al.: Software Ergonomie, Reihe Kontakt und Studium, Bd. 200, expert verlag, Sindelfingen 1986.
- /49/ Scholz, B.: CIM - Schnittstellen, Oldenbourg Verlag, München, Wien 1988.
- /50/ N.N.: IRL Sprachentwurf, Normenausschuß Maschinenbau im DIN, FB-IA, UA 96.2.2, Frankfurt 1989.
- /51/ Suppan - Borowka, J.; Simon, T.: MAP - Datenkommunikation in der automatisierten Fertigung, Datacom - Verlag, Pulheim 1986.
- /52/ Kleineidam, G.; Schellig, H.: Farbschattierte Rasterbilder und -bildfolgen aus RGB - Pixelinformationen, CAD/CAM - Zeitschrift für Computeranwendungen, Ausgabe 9/88, CAD/CAM-Verlag, München, S.70-75
- /53/ Paul, R.L.; Nof, S.Y.: Work Method Measurement: A Comparison between Robot and Human Task Performance, International Journal of Production Research, Vol.17 Nr.3,1979 S.277-303.

- /54/ Nof, S.Y.;
Lechtman, H.: Analysis of Industrial Robot Work by
the RTM Method, Industrial Engineering,
Journal, April 1982.
- /55/ Scholz, W.: Modell zur Datenbankgestützten Planung
automatisierter Montageanlagen,
Dissertation, Carl Hanser Verlag,
München, Wien 1989.
- /56/ Eigner, M.: Semantische Datenmodelle als Hilfsmittel
der Informationshandhabung in CAD-Systemen
und deren programmtechnische Realisierung
auf Kleinrechnern, Fortschritts-
bericht VDI-Z, Reihe 10, Nr. 9, 1980.
- /57/ Winston, P.H.;
Horn, B.K.: LISP, Addison-Wesley Verlag Deutschland,
Bonn 1984.
- /58/ Hemberger, A.: Innovationspotentiale in der
rechnerintegrierten Produktion durch
wissensbasierte Systeme, Dissertation,
Carl-Hanser-Verlag, München Wien 1988.
- /59/ Herzog, M.: Wissensbasierte Systeme und Mensch-
Computer-Kommunikation, Unterlagen zum
Lehrgang Software-Ergonomie, Technische
Akademie Esslingen, Esslingen 1988.
- /60/ Schmitt, A.A.: Dialogsysteme, Reihe Informatik Band 40
B.I. - Wissenschaftsverlag, Mannheim,
Wien, Zürich 1983.
- /61/ Martin, J.: Design of Man-Computer Dialogues,
Prentice-Hall, Englewood Cliffs, New
Jersey, 1972.
- /62/ Denavit, J.;
Hartenberg, R.S.: A Kinematic Notation for Lower-Pair
Mechanism Based on Matrices. ASME J.
J.Appl.Mech., Juni 1955, S.215-221.
- /63/ Geyer, G.;
Feldmann, K.: Planungsverfahren für Greifersysteme in
variantenreichen automatisierten
Montageprozessen, DFG Zwischenbericht,
SPP Neue Planungs- und Steuerungsver-
fahren in indirekten Produktionsberei-
chen, FAPS Erlangen 1989.
- /64/ Hollerbach, J.M.: Workshop on the Development of Mecha-
nical Hands, A.I. Memo 661, MIT
Cambridge MA 1982.
- /65/ N.N.: FMC - Technische Beschreibung, SIEMENS
AG, Erlangen 1988.
- /66/ Wedekind, H.: Datenbanksysteme I, Bd.I, 2te Auflage
B.I.-Wissenschaftsverlag, Mannheim 1981

- /67/ Pomberger, G.: Softwaretechnik und MODULA2, Carl Hanser Verlag, München, Wien 1984.
- /68/ Ranky, P.G.;
Ho, C.Y.: Robot Modelling: Control and Applications with Software, Springer Verlag, Berlin 1985.
- /69/ Pritschow, G.;
Koch, T.;
Bauder, M.: Automatisierte Erstellung der Rückwärtstransformation für Industrieroboter unter Anwendung eines optimierten iterativen Lösungsverfahrens, Robotersysteme 5, Springer Verlag, Berlin 1989, S.3-8.
- /70/ Wittenberg, J.;
Wolz, U.: Mesa Verde, ein Computerprogramm zur Simulation der nichtlinearen Dynamik von Vielkörpersystemen, Robotersysteme 1, Springer Verlag, Berlin 1985, S.7-18.
- /71/ Paul, R.P.: Modeling, Trajectory Calculation and Servoing of a Computer Controlled Arm, Ph.D. Dissertation, Stanford Univ. 1972.
- /72/ Desoyer, K.;
Kopacek, P.;
Troch, I.: Industrieroboter und Handhabungsgeräte, Oldenbourg Verlag, München, Wien 1985.
- /73/ Bronstein, I.: Taschenbuch der Mathematik, Verlag Harri Deutsch Thun, Frankfurt/Main 1969.
- /74/ Brendes, M.: Ein neuer Ansatz zur Lage- und Orientierungsbestimmung bekannter Körper aus dreidimensionalen Abstandsbildern, Dissertation, Braunschweig 1988
- /75/ Hoyer, H.: Verfahren zur automatischen Kollisionsvermeidung von Robotern im koordinierten Betrieb, Dissertation, Univ. Hagen 1984.
- /76/ Eversheim, W.;
Schütze, P.;
Luszk, G.: Kollisionskontrolle für offline erstellte Industrieroboterprogramme, VDI-Z 130, Nr.1, VDI-Verlag, Düsseldorf 1985, S.43-52.
- /77/ Flierl-
Schamberger, U.: Qualitätssicherung an einem Softwaremodul zur Berechnung von Roboterbewegungen, Studienarbeit, FAPS Erlangen 1988.
- /78/ Kleineidam, G.;
Schmid, J.: Integrierte Verfahrenskette zur Herstellung von Keramik-Tauchformen, Projektbericht, Erlangen, Selb 1989.

- /79/ Bullinger, H.-J.; GRIPS - An Approach to a Realtime Simulation of Human Arm Motion, in Proc. of 1st int. Conf. on Ergonomics of Advanced Manufacturing of Hybrid Automated Systems in Louisville (USA), Elsevier, Amsterdam 1988, S.599-606.
Lay, K.;
Menges, R.:
- /80/ Milberg, J.; MTM am Bildschirm, Zeitschrift montage 6/88, verlag moderne industrie, Landsberg 1988, S.12-42.
Pfrang, W.:
- /81/ Krampe, D.: Ergonomieuntersuchung an einem Werkzeug zur rechnergestützten Planung und Programmierung von Montagezellen, Studienarbeit, FAPS Erlangen 1989.
- /82/ Scheer, A.-W.: CIM - Computer Aided Manufacturing: Der computergesteuerte Industriebetrieb, 3. Auflage, Springer Verlag, Berlin 1988.
- /83/ Bullinger, H.-J.: CIM - Die Herausforderung der nächsten Jahre, in Produktionsforum'88: die CIM fähige Fabrik, Springer Verlag, Berlin u.a. 1988, S.9-44.
- /84/ Gora, W.: Kommunikation in der Fertigung - MAP Manufacturing Automation Protocols, Informatik Hauptseminar, Erlangen 1986.
- /85/ Feldmann, K.; Personalentwicklung und Qualifikation unter dem Einfluß von CIM, CIM-Technologie Transfer Seminar, Erlangen 1989.
Moldaschl, M.:
- /86/ Balzert, H.: Software-Ergonomie, Teubner Verlag, Stuttgart 1983.
- /87/ Feldmann, K.: Anlagenoptimierung durch rechnergestützte Planung, Simulation und Steuerung, in: Die CIM-Fabrik der 90iger Jahre, TÜV-Rheinland, Mannheim 1989, S.275-293.

Abkürzungen und Kurzzeichen

2D	zweidimensional
3D	dreidimensional
AI	Artificial Intelligence
AWP	Anwenderprogramm
BDE	Betriebsdaten Erfassung
BEM	Boundary Element Method
CAD	Computer Aided Design
CAD*I	CAD Interfaces (Standard)
CAP	Computer Aided Planing
CAM	Computer Aided Manufacturing
CAM-I	CAM International (Gremium)
CARo	Computer Aided Robotic (Systembezeichnung)
CAQ	Computer Aided Quality Assurance
CIM	Computer Integrated Manufacturing
CLDATA	Cutter Location Data (Norm)
CP	Continues Path
DAF	Datenaustausch Format (nach Siemens)
DB	Datenbank
DIN	Deutsche Industrie Norm
DNC	Direct Numeric Controlled
ESP	Experimental Solid Proposal
F	Frame
FAPS	Lehrstuhl für Fertigungsautomatisierung und Produktionssystematik, Erlangen
FEM	Finite Element Method
FhG	Fraunhofer Gesellschaft
FMC	Flexible Manufacturing Cell
GKS	Graphical Kernel System (Grafikstandard)
GMR	Graphics Metafile Resource (Fa. APOLLO)
GPR	Graphic Primitives Resource (Fa. APOLLO)
IGES	Initial Graphics Exchange Standard
IPK	Institut für Produktionsanlagen und Konstruktions- technik (FhG)
IR	Industrieroboter
IRDATA	Industrieroboter Datenschnittstelle (Normentwurf)

IRL	Industrial Robot Language (Sprachentwurf)
ISO	International Organisation for Standardisation
IWF	Institut für Werkzeugmaschinen und Fertigungstechnik, (TU) Berlin
IWB	Institut für Werkzeugmaschinen und Betriebswissenschaften, München
KfK	Kernforschungszentrum Karlsruhe
KI	Künstliche Intelligenz
MAP	Manufacturing Automation Protocols
MB	Megabyte
MDE	Maschinendaten Erfassung
MP	Multiple Point
MTM	Method Time Measurement
NC	Numeric Controlled
PDDI	Product Definition Data Interface
PC	Personal Computer
PG	Programmiergerät
PHIGS	Programmers Hierarchical Interactive Graphic System (Grafikstandard)
PPM	Produkt-Produktion-Modell
PPS	Produktion Planung Steuerung
PTP	Point To Point
RC	Robot Control
SET	Standard D'Exchange Specification
SMD	Surface Mounted Device
SPS	Speicherprogrammierbare Steuerung
SRCL	SIEMENS Robot Control Language
STEP	Standard for Exchange of Product Model Data
SvZ	System vorbestimmter Zeiten
TAP	Transfer and Archivierung Produkt-definierender Daten
TCP	Tool Center Point
VAL_II	Variable Assembly Language (Fa. UNIMATION)
VDI	Verein Deutscher Ingenieure
VDAFS	Flächenschnittstelle des Verbands Deutscher Automobilhersteller
WF	Work Factor
XBF-2	Experimental Boundary File

Formeln und Formelzeichen

Vereinbarungen:

$$\begin{array}{ll}
 s_i & := \sin \theta_i, & c_i & := \cos \theta_i, \\
 s_{ij} & := \sin (\theta_i + \theta_j), & c_{ij} & := \cos (\theta_i + \theta_j)
 \end{array}$$

$$\arctan2(a, b) = \begin{cases} \pi / 2 & \text{für } b = 0 \wedge a \geq 0 \\ -\pi / 2 & \text{für } b = 0 \wedge a < 0 \\ \arctan a/b & \text{für } b > 0 \\ \arctan a/b + \pi & \text{für } b < 0 \wedge a \geq 0 \\ \arctan a/b - \pi & \text{für } b < 0 \wedge a < 0 \end{cases}$$

Zeichen:

$\theta_i; \alpha_i$	kinematische Gelenkparameter [-]
$d_i; a_i$	kinematische Gelenkparameter [m]
$A_i; {}^aT_b$	Transformationsmatrizen
(x, y, z)	Koordinaten in einem Rechtssystem
$\underline{n}, \underline{o}, \underline{a}$	Einheitsvektoren eines Rechtskoordinatensystems
aa; hh; dd; verschw	Matrizenelemente
A; B; C; D; E	Rechengrößen [-]
$\underline{p}_i; \underline{x}_i$	Ortsvektoren im Umweltkoordinatensystem
v	absolute Geschwindigkeit [m/s], [°/s]
c	absolute Beschleunigung [m/s ²], [°/s ²]
T; t	Zeit [s]
Δs	Weginkrement [m]
\underline{O}	Orthonormalmatrix
\underline{m}	Ortsvektor des Kreismittelpunktes
γ	Segmentwinkel auf der Kreisbahn
u; v; w	Rechengrößen
\underline{h}_i	Ortsvektor zu Punkt auf der Kreisbahn
m	Zahl der Simulationsschritte in (5.69)
n; v; s; p	Anzahl von Zellenelementen in (5.70)

Lebenslauf

Name, Vorname: Kleineidam, Gerhard

Geburtsdatum: 15. August 1958

Familie: Frau Susanne, Sohn Roman, Tochter Ruth

Schulen:
1964 - 1968 Volksschulen in Lauenstein und Windsbach
1968 - 1977 Gymnasien in Kronach und Kulmbach
1977 Abitur (math.-nat.)

Bundeswehr:
1977 - 1979 Grundwehrdienst und Offiziersausbildung

Studium:
1979 - 1984 Maschinenbaustudium an der Technischen
Universität München,
Abschluß: Dipl.-Ing.

Berufliche Tätigkeit:
15.01.1985 - 30.04.1987 Angestellt als wissenschaftlicher
Mitarbeiter am Lehrstuhl für Ferti-
gungsautomatisierung und Produk-
tionssystematik an der Universität
Erlangen-Nürnberg.

01.05.1987 - 30.04.1990 Akademischer Rat am Lehrstuhl für Ferti-
gungsautomatisierung und Produk-
tionssystematik an der Universität
Erlangen-Nürnberg.
Oberingenieur und Leiter der Gruppe für
Rechnereinsatz in Konstruktion und
Planung.

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 Montagesysteme
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.

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 rechnerintegrierter Planungssysteme

183 Seiten. 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, 85 Bilder. 1990. Kartoniert.

Band 12

Gerhard Kleineidam

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.