

Olaf Schrödel

*Flexible
Werkstattsteuerung
mit objektorientierten
Softwarestrukturen*

Olaf Schrödel

*Flexible
Werkstattsteuerung
mit objektorientierten
Softwarestrukturen*

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 Universität Erlangen-Nürnberg

Tag der Einreichung:	04.11.1991
Tag der Promotion:	27.2.1992
Dekan:	Prof. Dr. rer. nat. H. Mughrabi
Berichterstatter:	Prof. Dr.-Ing. K. Feldmann
	Prof. Dr.-Ing. H. Wedekind

Die Deutsche Bibliothek - CIP-Einheitsaufnahme

Schrödel, Olaf:

Flexible Werkstattsteuerung mit objektorientierten
Softwarestrukturen / Olaf Schrödel. - München; Wien;
Hanser, 1992

(Fertigungstechnik - Erlangen; 30)

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

ISBN 3-446-17242-4

Dieses Werk ist urheberrechtlich geschützt.

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

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

© Carl Hanser Verlag München, Wien 1992

Herstellung: Copy Center 2000, Erlangen-Eltersdorf

Printed in Germany

Vorwort

Die vorliegende Dissertation entstand während meiner Tätigkeit als wissenschaftlicher Mitarbeiter am Lehrstuhl für Fertigungsautomatisierung und Produktionssystematik der Universität Erlangen – Nürnberg.

Herrn Professor Dr.–Ing. K. Feldmann, dem Leiter dieses Lehrstuhls am Institut für Fertigungstechnik, danke ich für die großzügige Förderung bei der Durchführung dieser Arbeit. Ich bedanke mich ebenso bei Herrn Professor Dr.–Ing. H. Wedekind für die Übernahme des Korreferats.

Weiterhin danke ich allen Kolleginnen und Kollegen, insbesondere jenen der Forschungsgruppe für Steuerungs– und Sensortechnik, die durch Ihre ständige Diskussionsbereitschaft zum Gelingen dieser Arbeit beigetragen haben. Ferner gilt mein Dank den zahlreichen Studenten, die mir mit unermüdlichem Engagement und Hilfsbereitschaft zur Seite standen.

So möchte ich an dieser Stelle auch die Gelegenheit nutzen all diejenigen zu motivieren, die diesen beschwerlichen Weg noch vor sich haben. Vor allem im wissenschaftlichen Spannungsfeld zwischen Informatik und Fertigungstechnik gestaltet sich das Arbeiten nicht immer leicht. Die interdisziplinäre Zusammenarbeit wird jedoch gerade im Bereich der Fertigungsautomatisierung eine wesentliche Rolle für zukünftige Entwicklungen spielen.

Der letzte, aber herzlichste Dank gilt allen meinen Freunden und Angehörigen – in erster Linie meinen Eltern und meiner Freundin Gaby. Sie alle sorgten für das harmonische Umfeld, welches für eine derartige Arbeit erforderlich ist.

Olaf Schrödel

Flexible Werkstattsteuerung mit objektorientierten Softwarestrukturen

INHALTSVERZEICHNIS

1	Einleitung und Motivation	1
2	Flexibilitätssteigerung in der Fertigung	3
2.1	Entwicklungsstand der rechnerintegrierten Werkstattsteuerung	3
2.1.1	Der Arbeitsplan als Eingangsgröße	3
2.1.2	Strukturen in der Fertigung	6
2.1.3	Dispositionsaufgaben der Werkstattsteuerung	10
2.2	Nutzung vorhandener Flexibilitätspotentiale	14
2.2.1	Definition der Flexibilität	14
2.2.2	Flexibilitätspotentiale	16
3	Softwaretechnische Aspekte zur rechnerintegrierten Werkstattsteuerung	18
3.1	Bewertung verschiedener Wissensrepräsentationsformen	18
3.1.1	Mathematische Logik	18
3.1.2	Regelsysteme	20
3.1.3	Constraints	21
3.1.4	Relationale Datenmodelle	22
3.1.5	Frames	23
3.1.6	Petrinetze	25
3.1.7	Semantische Netze	28
3.1.8	Zusammenfassende Bewertung	30
3.2	Programmierung	31
3.2.1	Verschiedene Stilarten	32
3.2.2	Objektorientierte Programmierung	33
4	Der Ablaufgraph – Innovative Komponente in der Arbeitsvorbereitung	35
4.1	Architekturmodell	35
4.2	Zeitliche Relationen von Arbeitsgängen	36
4.3	Darstellung durch Graphen	41
4.3.1	Grundlegende Definitionen	41
4.3.2	Vervollständigung des Ablaufgraphen	43
4.3.3	Bedingte und unbedingte Kanten	44
4.3.4	Berücksichtigung der Technologieflexibilität	47
4.4	Algorithmische Reihenfolgeermittlung	49
4.4.1	Breitensuche	50
4.4.2	Tiefensuche	50

5	Objektorientierte Softwarestrukturierung	55
5.1	Objektorientierter Entwurf	56
5.1.1	Objekte	56
5.1.2	Erweiterte Entwurfsmöglichkeiten	57
5.1.3	Abstrakte Datentypen	58
5.2	Klassifikation von Fertigungskomponenten	61
5.3	Aufbau einer Wissensbasis	63
5.3.1	Anforderungen zur funktionalen Werkstattsteuerung	63
5.3.2	Semantisches Netz zur Wissensrepräsentation	67
5.3.3	Objekte als abstrakte Datentypen	72
5.3.4	Realisierungsaspekte	75
5.3.5	Einbettung von Ablaufgraphen	76
5.4	Ausbau zu einem ganzheitlichen Konzept	77
5.4.1	Vorteile durch die Objektorientierung	77
5.4.2	Architekturprinzip eines Gesamtsystems	78
6	Selbstregelnde Werkstattsteuerung	81
6.1	Untersuchung mathematischer Methoden	83
6.1.1	Vollenumeration	83
6.1.2	Lineare Optimierung	83
6.1.3	Entscheidungsbaumverfahren	85
6.1.4	Heuristische Verfahren	88
6.1.5	Vorrangregeln	90
6.1.6	Resümee	91
6.2	Rahmenbedingungen	91
6.2.1	Bereitstellung notwendiger Informationen	91
6.2.2	Erweiterung des Arbeitsplans	92
6.2.3	Voraussetzungen für das Auftragsvolumen	93
6.3	Planungskomponente	94
6.3.1	Optimierungsziele	95
6.3.2	Planungseinheit	96
6.3.3	Planungsvorgehen	99
6.3.4	Zusätzliche Funktionalität der Planungskomponente	106
6.4	Steuerungskomponente	108
6.4.1	Notwendigkeit einer Steuerung	108
6.4.2	Funktionsumfang der Steuerungskomponente	109
6.5	Überwachungskomponente	115
6.5.1	Kommunikationsstruktur	115
6.5.2	Selbstregelung	120

7	Fertigungsbeispiel in einer Testumgebung	123
7.1	Produktspektrum	125
7.1.1	Getriebevarianten	125
7.1.2	Ritzelwelle, Abtriebswelle und Zahnrad	126
7.2	Flexibles Fertigungssystem	127
7.2.1	Zellenstruktur	127
7.2.2	Fertigungshilfsmittel	129
7.2.3	Arbeitspläne	129
7.3	Rechnerintegrierte Werkstattsteuerung	132
7.4	Anwendungsspezifische Konfiguration der Wissensbasis	135
7.4.1	Ablaufgraphen	135
7.4.2	Transportnetz	137
7.4.3	Systemlayout	138
7.5	Inferenzkomponenten – Ergebnisse und Bewertung	139
7.5.1	Auftragsspektrum 1	139
7.5.2	Auftragsspektrum 2	141
7.5.3	Auftragsspektrum 3	144
8	Zusammenfassung	149
9	Literaturverzeichnis	151

1 Einleitung und Motivation

Will ein Unternehmen den zukünftigen Anforderungen des Marktes gewachsen sein, so muß es zur Sicherung seiner Wettbewerbsfähigkeit rasch und flexibel auf die sich stetig ändernden Situationen reagieren können /1/. Die immer stärker werdende Konkurrenz sowohl auf nationalen als auch internationalen Märkten erfordert den optimalen Einsatz sämtlicher Ressourcen, um auch dem Anspruch einer ganzheitlichen Integration der informationstechnischen Anwendungsgebiete möglichst nahe zu kommen. Als Ausgangspunkt jedes wirtschaftlichen Handelns ist hierbei das Rationalprinzip zu sehen /2/.

Die Wettbewerbssituation, der ein Unternehmen ausgesetzt ist, wird im wesentlichen vom Kundenverhalten und der Konkurrenz bestimmt /3/. Doch nicht nur der Wandel vom Verkäufer- zum Käufermarkt, sondern vielmehr die Zahl der Anbieter, die sich durch den europäischen Binnenmarkt noch erhöhen wird, verlangt von den Unternehmen kürzere Produktlebenszyklen, höhere Variantenvielfalt und schnellere Lieferzeiten. Vornehmlich für den Mittelstand ist es notwendig, seine Flexibilität in der Fertigung zu steigern. Ausschlaggebend für eine zukunftsorientierte Fertigung wird deshalb eine geschickte datenorganisatorische Verknüpfung der Planung mit der technischen und dispositiven Steuerung sein /4/.

Durch die interdisziplinäre Verbindung von Fertigungstechnik und Informatik kann eine hervorragende Ausgangsbasis geschaffen werden, um die anfallenden Aufgaben zu bewältigen. Eine gewinnbringende Synergie wird jedoch erst dann erreicht, wenn zur Informationsverarbeitung geeignete Strukturen konzipiert und auf die technischen Gegebenheiten projiziert werden. Für den Einsatz rechnerintegrierter Systeme ist gerade mit zunehmender Komplexität der Problemstellung die Definition einer adäquaten Softwarestruktur unabdingbar, um der von Anwenderseite häufig gestellten Anforderung nach Flexibilität gerecht zu werden /5/.

Mit dem Begriff der Flexibilität werden nämlich im Bereich der Fertigungsautomatisierung, respektive der Werkstattsteuerung, eine Reihe von Eigenschaften assoziiert. Flexibilitätspotentiale, die mit den Automatisierungskomponenten in der Fertigung prinzipiell vorhanden sind, werden aber softwaretechnisch nur unzureichend oder gar nicht genutzt. Auch in der Arbeitsvorbereitung finden derartige Potentiale noch keinen Eingang. Existierende Softwaresysteme, seien es Produk-

tionsplanungs und –steuerungssysteme oder sogenannte Leitstände, berücksichtigen diese Freiheitsgrade nur in sehr geringem Maße. Zur Verbesserung des Zustands soll an dieser Stelle für den Bereich der Werkstattsteuerung eine softwaretechnisch gangbare Möglichkeit aufgezeigt werden.

So liegt das Ziel dieser Arbeit in der Konzeption eines rechnergestützten Informationssystems zur hochflexiblen Werkstattsteuerung, welche vornehmlich in der Einzel- und Kleinserienfertigung Anwendung finden soll. Aufgabe ist es, vorgegebene Fertigungsprogramme für kurzfristige Planungsaufgaben autonom unter bestmöglicher Nutzung vorhandener Freiheitsgrade abzuarbeiten. Mittels einer objektorientierten Softwarestruktur soll ein Modell der realen Fertigungsumwelt als Wissensbasis am Rechner erzeugt werden. Entsprechende Algorithmen sollen für ein geeignetes Instrumentarium zur "optimalen" Maschinenbelegung und zur Steuerung des Material- und Fertigungshilfsmittelflusses sorgen. Im Fertigungsverlauf wird eine Anpassung an die sich ändernden Situationen durch eine selbstregelnde Überwachungskomponente zu erzielen sein.

Im Verlauf der Arbeit wird dabei ausgehend vom Stand technischer Erkenntnisse ein Anforderungsprofil an das zu konzipierende System festgelegt. Basierend auf der Gegenüberstellung potentiell einsetzbarer Softwaremethoden und der Einführung des Ablaufgraphen als innovative Komponente in der Arbeitsvorbereitung ist die Definition einer Wissensbasis Gegenstand der Betrachtung. Ein Hauptaugenmerk wird auf die objektorientierte Klassifikation und Strukturierung gelegt. Ferner wird das eigentliche Informationssystem, welches die Komponenten zur Planung, Steuerung und Überwachung innerhalb der Werkstattsteuerung beinhaltet, mit den entsprechenden Inferenzmechanismen vorgestellt, um dieses sodann an einem beispielhaften Szenario zu validieren.

2 Flexibilitätssteigerung in der Fertigung

2.1 Entwicklungsstand der rechnerintegrierten Werkstattsteuerung

Aus den unterschiedlichsten Gründen sind Fertigungsbetriebe gezwungen, flexibel zu automatisieren. Mit dem Einsatz rechnerintegrierter Steuerungssysteme wird versucht die Qualität zu erhöhen, die Kosten zu senken und die Flexibilität zu steigern. Hierbei verstärkt sich der Trend dahingehend, kleine überschaubare Systeme einzusetzen /6/, was auch der Zielsetzung Rechnung trägt, eine schrittweise Einführung neuer Techniken zu ermöglichen.

2.1.1 Der Arbeitsplan als Eingangsgröße

Die wichtigste Komponente in der Arbeitsvorbereitung ist der Arbeitsplan. In ihm werden die einzelnen Schritte zur Herstellung eines Produkts (Rohmaterial, Einzelteile, Baugruppen usw.) beschrieben. Die Informationen aus dem Arbeitsplans gliedern sich dabei in:

- organisatorische Daten (Stückzahl, Ident-Nr., Termin, ...)
- materialspezifische Daten (Werkstoff, Abmessung, Gewicht, ...)
- arbeitsgangsabhängige Daten (Vorgangsbeschreibung, Kosten, ...)

Prinzipiell sind für die Teilefertigung und Montage die Struktur und der Inhalt der Arbeitspläne gleich zu gestalten, wobei Montageteile entsprechend zu kennzeichnen sind. Den Hauptanteil im Arbeitsplan nehmen die Arbeitsvorgänge ein. Für die weiteren Ausführungen wird daher ein Arbeitsvorgang wie folgt definiert:

*Der **Arbeitsvorgang** (kurz: Arbeitsgang, AG) bildet einen technologischen Abschnitt eines Arbeitsablaufs an **einem** Arbeitsplatz (bzw. Arbeitsort). Eine stärkere Detaillierung beschreibt den Teilarbeitsvorgang, der einen Bezug zu einem Werkstückmerkmal (Nut, Tasche, Fläche, ...) herstellt.*

Für variable Telespektren in kleinen Losgrößen wirkt sich die Qualität der Arbeitsplanung in der Fertigung am stärksten aus. Die Ergebnisse der Vorbereitungsphase bilden nämlich den Grundstock für eine wirtschaftliche Fertigung. Sie müssen in einer geeigneten Form bereitgestellt und an die Fertigung weitergeleitet werden. Im wesentlichen sind das Arbeitspläne und Steuerdaten. Zukünftig ist im Sinne **durchgängiger Verfahrensketten** mit weiteren Informationen, wie Reihenfolgebeziehungen zwischen Arbeitsvorgängen, zu rechnen. In Kapitel 4 wird darauf ausführlich eingegangen.

Neben diesen Reihenfolgebeziehungen müssen jedoch auch Alternativen bezüglich Geräte und Fertigungshilfsmittel in den Arbeitsplan integriert werden. Bild 1 zeigt die Aufgaben bei der Arbeitsplanerstellung nach Eversheim /7/, wobei es für eine flexible Fertigung wichtig ist, daß alternative Informationen schon in der Arbeitsvorbereitung bereitgestellt werden. Bei strategischen Änderungen im Arbeitsablauf aufgrund nicht zu erwartender Ereignisse kann dann auf diese Informationen zurückgegriffen werden /8/.

In der Teilefertigung werden momentan die einzelnen Arbeitsgänge aus dem Arbeitsplan streng sequentiell abgearbeitet. Freiheitsgrade bezüglich der Ablaufreihenfolge werden nicht aufgeführt und können somit auch nicht genutzt werden. Etwas anders verhält sich dagegen der Montagebereich. Hier werden die Montageablaufstrukturen durch die Darstellung von logischen und zeitlichen Folgen der Teilaufgaben in besserer Form repräsentiert. Ein Beispiel ist im Vorranggraph zu sehen, bei dem es sich um eine netzplanähnliche Darstellung handelt /9/.

Insgesamt bleibt jedoch festzustellen, daß Arbeitsplanstrukturen in der Praxis viel zu starr gestaltet sind. Technologiealternativen, Betriebsmittelressourcen und unterschiedliche Ablaufreihenfolgen müssen in der Arbeitsplanung Berücksichtigung finden. Rechnerunterstützte Systeme zur Arbeitsplanung werden in vermehrtem Maße gefragt sein.

Im Hochschulbereich wird unter diesen Aspekten bereits entwickelt. Am Institut für Werkzeugmaschinen und Fertigungstechnik der Technischen Universität Berlin zum Beispiel ist ein wissensbasiertes Planungssystem für flexible Fertigungssysteme Gegenstand der Forschungsaktivitäten. Hierbei werden Arbeitsvorgangsfolgen bestimmt und bleiben als planerische Freiheitsgrade erhalten. Sogar alternative Operationsfolgen gehen in die Betrachtung mit ein /8/.



Bild 1: Aufgaben bei der Arbeitsplanerstellung

Ein Programmsystem des Instituts für Werkzeugmaschinen und Betriebstechnik der Universität Karlsruhe ermöglicht unter Nutzung von CAD-Daten mittels einer rechnergestützten Demontagesimulation ebenfalls die automatische Generierung von Montagevorranggraphen /10/.

Das System AVOGEN zur Generierung von Arbeitsgangfolgen für rotationssymmetrische Werkstücke wurde am Institut für Fertigungstechnik und Spanende Werkzeugmaschinen der Universität Hannover entwickelt. Arbeitsgangfolgen werden mittels eines sogenannten Situationskalküls errechnet. Die reale Fertigung wird dazu in den elementaren Schritten simuliert. Ist ein Weg vom Ausgangs- in den Endzustand gefunden, so ist auch eine Arbeitsgangfolge ermittelt. Der Arbeitsplaner kann sich Reihenfolgevorschlüsse generieren lassen /11/. Ein neues, regelbasiertes Verfahren desselben Instituts erlaubt die automatische Generierung von Montagevorgangsfolgen /12/.

2.1.2 Strukturen in der Fertigung

Insbesondere im Fertigungsbereich (CAM-Bereich) werden einige fachspezifische Begriffe in der Literatur unterschiedlich ausgelegt. So soll im weiteren unter Fertigung sowohl Teilefertigung als auch Montage verstanden werden.

Auch die verschiedenen Organisationsformen, wie Zentrum, Zelle, System usw., sind in der Literatur nicht einheitlich definiert /13/, /14/, /15/, /16/, /17/, /18/, /19/. Ausschlaggebend für eine Abgrenzung können die funktionale Integration, die Anzahl der Geräte, der Fertigungstyp, das Werkstückspektrum oder der Produktionsbedarf sein.

Unter einem **Zentrum** oder einer **Station** ist daher im weiteren eine Maschine bzw. ein Gerät¹⁾ zu sehen, das die Bearbeitung unterschiedlicher Arbeitsgänge erlaubt. Der Gerätetyp ist von der Fertigungsaufgabe abhängig.

Eine **Zelle** besteht aus zwei oder mehreren Geräten, sowie ver- und entsorgenden Peripherieeinrichtungen und Lagersystemen. Sie erlaubt eine vollautomatische, bedienerunabhängige Bearbeitung vornehmlich in der Einzel- oder Kleinserienfertigung.

Die weitergehende Ausbaustufe stellt das **System** dar, welches flexibel gestaltete Arbeitsabläufe zulässt. Ersetzende beziehungsweise ergänzende Maschinen werden zur Bearbeitung gleicher oder ähnlicher Werkstücke in größtmöglicher Unabhängigkeit eingesetzt.

In Fertigungs- bzw. Montageeinsein wird eine nahezu vollständige Endproduktion aus vorgegebenen Materialien angestrebt. Die Trennlinie zwischen System und Insel ist dabei nicht exakt zu ziehen.

Obwohl das zu konzipierende Informationssystem zur Werkstattsteuerung an keine Organisationsform gebunden sein soll, rückt doch das flexible Fertigungssystem in den Mittelpunkt der Untersuchung, da hier das Flexibilitätspotential am stärksten ausgeprägt ist.

1) Unter dem Begriff Gerät sind alle Einrichtungen zur direkten und indirekten Fertigungszustandsänderung zu subsumieren. Hierzu zählen Werkzeugmaschinen, Handhabungseinrichtungen, Roboter, Montageeinrichtungen usw. Der Begriff wird synonym mit dem der Maschine verwendet.

*Ein **flexibles Fertigungssystem** (FFS) besteht aus einer Menge von Fertigungseinrichtungen, die über gemeinsame Transportsysteme so miteinander verknüpft sind, daß eine automatische Fertigung möglich ist und innerhalb eines durch die Geräte vorgegebenen Bereichs voneinander unabhängige Bearbeitungen durchführbar sind. Werden in diesem Fertigungssystem auch Montageoperationen durchgeführt, kann von einem hybriden System gesprochen werden.*

Bei der Anzahl bereits realisierter Fertigungssysteme wird für 1990 von knapp 1600 Einsatzfällen ausgegangen (Bild 2). Für die nächsten drei Jahre wird dann nochmals mit einer Verdoppelung dieser Zahl gerechnet. Prognosen für das jährliche Wachstum in Europa belaufen sich auf bis zu 20% /20/, wobei der europäische Markt gerade für den Maschinenbau ein erhebliches Potential bietet /21/.

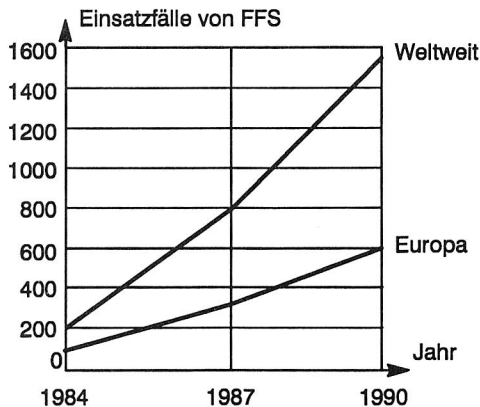


Bild 2: Einsatzfälle von flexiblen Fertigungssystemen /20/

In einer Untersuchung des Forschungsinstituts für Rationalisierung (FIR) der Rheinisch-Westfälischen Technischen Hochschule Aachen wurden die Einsatzbedingungen von 60 Systemen in 46 Unternehmen getestet. Dabei konnte festgestellt werden, daß rund 50 % der im Betrieb befindlichen Fertigungssysteme Kleinanlagen sind (2 oder 3 Maschinen). Innerhalb dieser flexiblen Fertigungssysteme spielen die Bearbeitungszentren eine zentrale Rolle, was auf ihre komplexen

Bearbeitungsmöglichkeiten zurückzuführen ist (Bohren, Fräsen, etc.). Die Integration von Drehmaschinen und Schleifmaschinen hingegen ist nur sehr gering /6/. Technologien wie Räumen oder Prägen werden in flexiblen Fertigungssystemen fast nicht berücksichtigt. Die meistverbreiteten Hilfsfunktionen sind das Waschen und Messen (Bild 3).

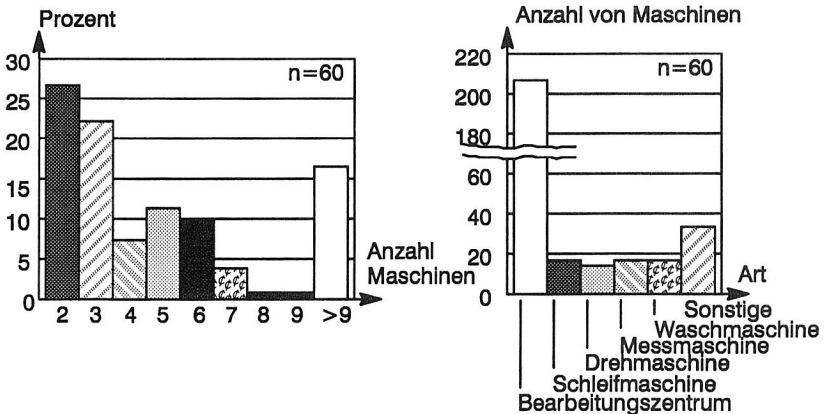


Bild 3: Analyse der Systemgröße und verwendeten Maschinen in FFS /6/

Die Analyse zeigt weiterhin, daß 53% aller Fertigungssysteme aus ersetzenden und 17% aus ergänzenden Maschinen bestehen. Bei Bearbeitungszentren werden trotz gleichen Maschinentyps jedoch nur bei 19% der Systeme die Maschinen ergänzend eingesetzt, wodurch Flexibilität verloren geht, da eine feste Zuordnung der Maschinen zu den Aufträgen erfolgt.

Bei den Transportsystemen wird zwischen einem Werkzeug- und Werkstücktransport unterschieden /22/. Diese können prinzipiell gemeinsam oder getrennt durchgeführt werden. Beide Varianten werden in der Praxis eingesetzt, wobei eine klare Favorisierung eines Prinzips nicht zu erkennen ist.

Fertigungshilfsmittel sind Werkzeuge, Greifer, Vorrichtungen, Spannmittel etc. Im erweiterten Sinne können ebenso Meß- und Prüfmittel dazu gezählt werden. Ihrer Verfügbarkeit kommt eine immer stärkere Bedeutung zu, da sie eine wesentliche Voraussetzung für den Beginn eines Bearbeitungsvorgangs im Fertigungsprozeß darstellen /23/. Puffer bilden in einem System ebenfalls eine wichtige Komponente

und sind deshalb integraler Bestandteil eines FFS. Eine durchschnittliche Größe für Werkzeuge zum Beispiel liegt bei 20 bis 30 Stück.

Der Grad der Montageautomatisierung dagegen ist im Vergleich zur Fertigung relativ niedrig. Der Grund dafür liegt in der vielfältigen Struktur der Montageaufgabe /24/, /25/. Auch hybride Systeme sind kaum vorhanden. Sie werden jedoch eine weitere Entwicklungsstufe auf dem Weg zu einem ganzheitlichen Konzept sein.

Zur Werkstattsteuerung sind jedoch nicht nur die Fertigungskomponenten von Bedeutung, sondern auch der Fertigungstyp. In Anlehnung an Warnecke /26/ werden im wesentlichen Einzel-, Serien- und Massenfertigung differenziert, welche in Bild 4 ihrer Steuerungsart gegenübergestellt sind.

TYP	KENNZEICHEN	STEUERUNG
Einzel- und Kleinserienfertigung	wenige Erzeugnisse langfristiger Produktionszyklus Fertigung nach Kundenwunsch	Werkstattsteuerung
Mittel- und Großserienfertigung	begrenzte Stückzahl Standardisierung der Erzeugnisse meist Auftragsproduktion	Kanban oder belastungsorientierte Steuerung
Massenfertigung	große Stückzahlen Gleichartigkeit der Produkte Lagerproduktion	Steuerung nach Fortschrittszahlen

Bild 4: Unterschiedliche Fertigungstypen

Innerhalb Einzel- und Kleinserienfertigung rückt die Werkstattfertigung stärker in den Mittelpunkt. In den früheren Jahren war diese Fertigung an einer räumlichen Anordnung der Maschinen gleicher Verrichtungsart orientiert. Aus wirtschaftlichen Gründen und infolge von technologischen Verbesserungen steht heutzutage die Werkstattfertigung für Komplettbearbeitungen mit dem Erfolg kürzerer Durchlaufzeiten in kleineren Losen /27/.

2.1.3 Dispositionsaufgaben der Werkstattsteuerung

Das traditionelle Einsatzgebiet der PPS (Produktionsplanung und –steuerung) ist die lang- und mittelfristige Durchlauf- und Kapazitätsterminierung. Die eingesetzten Methoden reichen hierbei vom klassischen Lösungsansatz MRP (Manufacturing Resources Planning) über das japanische KANBAN-Prinzip bis zur belastungsorientierten Auftragsfreigabe, um nur einige aufzuzählen /28/. Obwohl detaillierte Termine berechnet werden und die Planungsperiode bis auf einen Tag reduziert sein kann, ist das Planungsergebnis aufgrund der zu groben Datengrundlage nicht direkt als Steuergröße einsetzbar. Insbesondere die Planung der Auftrags- und Arbeitsgangreihenfolge in Anlehnung an eine erweiterte Ressourcenbetrachtung ist oft nicht gelöst. Bei unvorhergesehenen Störungen ist eine schnelle Reaktion nicht möglich.

Zwischen der PPS und der Fertigung ist die Werkstattsteuerung angesiedelt. Nach Weule /29/ ist sie wie folgt definiert:

*Die **Werkstattsteuerung** soll für einen kurzfristigen Planungszeitraum, zum Beispiel eine Schicht oder kürzer, aus einem vorgegebenen Fertigungsprogramm von n Aufträgen entsprechend den Arbeitsplänen nach einem vorgegebenem Optimierungsziel den Auftragsdurchlauf entscheiden.*

Diese Aufgaben löste bisher der Mensch durch seine Intelligenz und Intuition. Doch können weitreichende Konsequenzen aufgrund ihrer Komplexität nicht vollständig durchdacht werden, was eine Rechnerunterstützung sinnvoll erscheinen läßt.

Seit einiger Zeit werden vermehrt werkstattnahe Steuerungssysteme, sogenannte Leitstände, angeboten. Sie orientieren sich an der klassischen Plantafel und fungieren als zentrale Fertigungssteuerung. Ihr Einsatzschwerpunkt liegt im Maschinenbaubetrieb der Einzel- bis Kleinserienfertigung bei einer hohen Planungsfrequenz (Bild 5).

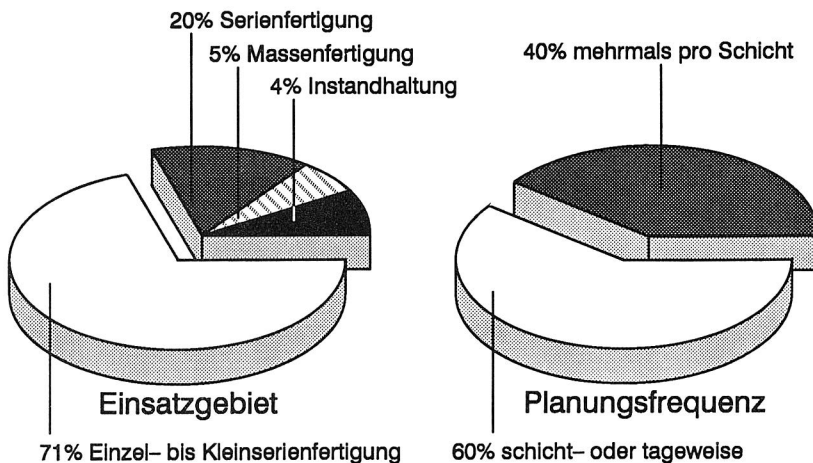


Bild 5: Einsatzgebiet und Planungsfrequenz elektronischer Leitstände /30/

Die Anforderungen an die Werkstattsteuerung differieren je nach Fertigungstyp und Organisationsform. Ihre Funktionalität ist so breit gefächert, daß sich die vorliegende Arbeit auf das Kernstück, das Leitsystem, konzentriert (Bild 6). Es läßt sich in die Funktionsblöcke der Planung, Steuerung und Überwachung unterteilen.

Innerhalb der einzelnen Blöcke kann wiederum eine Aufgabenverteilung vorgenommen werden. So umfaßt die Planung unter anderem die Auftragsverwaltung, Reihenfolgeplanung und Kapazitätsfeinplanung, während sich die Steuerung als Bereitstellung der benötigten Materialien und Fertigungshilfsmittel versteht. Die Überwachung sorgt für die zeitgerechte Durchsetzung des Auftragspektrums.

Prinzipiell kann ein Werkstattsteuerungssystem sowohl als autonomes Instrument als auch in Verbindung mit einem PPS-System eingesetzt werden. Über diese Schnittstellen kommen dann die Fertigungsprogramme und Planungsvorgaben, während die Maschinen- und Betriebsdaten nach oben weitergereicht werden.

Der Funktionsumfang der einzelnen Leitstände ist sehr unterschiedlich ausgeprägt. Indikatoren für die Mächtigkeit eines Systems sind der Wertebereich der Datenverwaltung und die Manipulationsmöglichkeiten. Die für einen Leitstand relevanten Daten müssen in einer Form veränderbar sein, die einen entsprechenden Dis-

positionsspielraum zur Verfügung stellt. Der Bogen kommerziell angebotener Lösungen läßt sich vom reinen Visualisierungsinstrument, welches sich auf die Übernahme von Auftrags-, Arbeitsplan- und Arbeitsgangdaten aus übergeordneten Bereichen zur Anzeige des Kapazitäts- und Materialbedarfs beschränkt, bis zum autark arbeitenden System spannen.

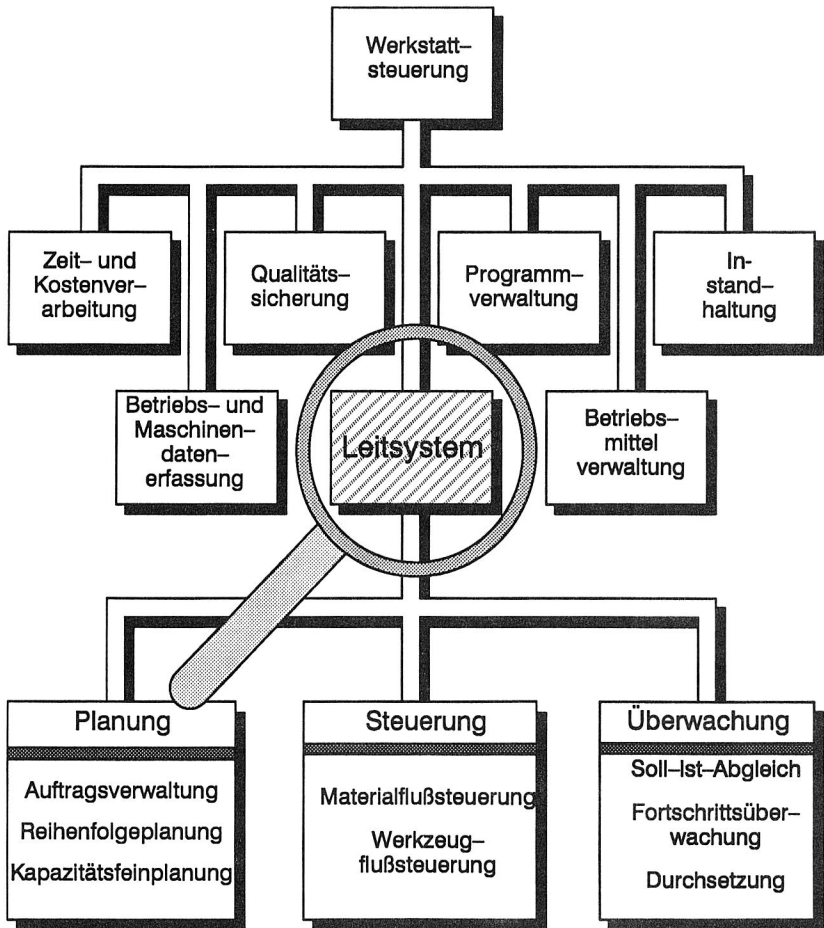


Bild 6: Aufgabenspektrum der Werkstattsteuerung

Die Verwaltung der Ressourcen, wie Arbeitsplätze, Werkzeuge oder Fertigungshilfsmittel, ist nicht vollständig ausgereift. Während die meisten Systeme Arbeitsplatzmodelle führen, ist eine datenmäßige Erfassung weiterer Betriebsmittel kaum zu finden /31/.

Die Belegungsplanung ist das Kernstück der Werkstattsteuerung und von Anwendung zu Anwendung unterschiedlich realisiert. Von der Methode (manuell, halbautomatisch, automatisch) über die verschiedenen Terminierungsarten (vorwärts, rückwärts, ...) bis hin zur Vorgehensweise (maschinen-, auftrags-, oder arbeitgangorientiert) werden alle möglichen Formen der Planung angeboten. Detaillierte Auskünfte hierüber bietet eine von Friedrichs und Gromotka /32/ veröffentlichte Marktübersicht.

Augenscheinlich ist, daß eine Planung unter bestimmten Zielsetzungen nur äußerst selten unterstützt wird. Eine solche Planung wird in Zukunft infolge höherer Flexibilitätsanforderungen eine größere Rolle spielen. Kurze Durchlaufzeiten oder minimale Rüstvorgänge als Zielkriterium rücken stärker in den Vordergrund. Strategische Vorgehensweisen könnten die Planung hierbei unterstützen. Die Einplanung nach Strategien erlaubt es, sich den ändernden Betriebsbedingungen schnell anzupassen. So können Aufträge mit kurzer Laufzeit bei einer SJF-Strategie (Shortest Job First) schneller durch das System geschleust werden als bei einer FIFO-Strategie (First In First Out), die eher einer ausgewogenen Auftragsbearbeitung nachkommt. Bild 7 zeigt exemplarisch die Ausprägung einer Plantafel, wie sie in den unterschiedlichsten Leitständen zu finden ist.

Leitstandssysteme können, wie bereits erwähnt, als eigenständige Einheiten betrieben werden. Doch gerade die Anbindung an andere rechnergestützte Systeme erhöht den Vorteil für den Einsatz im Unternehmen /33/. Bei den meisten Leitständen ist eine Verbindung zur PPS vorgesehen und auch realisierbar. Dies ist der Tatsache zuzuschreiben, daß in sehr vielen Unternehmen bereits PPS-Systeme existieren und durch das Werkstattsteuerungssystem eine präzisere Terminierung ermöglicht werden soll.

Die Integration in Richtung Fertigung ist diesbezüglich noch wenig entwickelt. Eine Kopplung "mit unten" beschränkt den Informationsfluß in den meisten Fällen auf Steueranweisungen in die eine und Betriebs- bzw. Maschinendaten in die andere Richtung. Diese Daten beinhalten Rück- oder Störmeldungen und werden in den

meisten Fällen an die PPS weitergeleitet. Eine selbständige Auswertung zur optimierenden Planung erfolgt nicht. Die Systeme sind als **rein steuernde Größen** zu verwenden und erlauben **keinen regelnden Eingriff**.

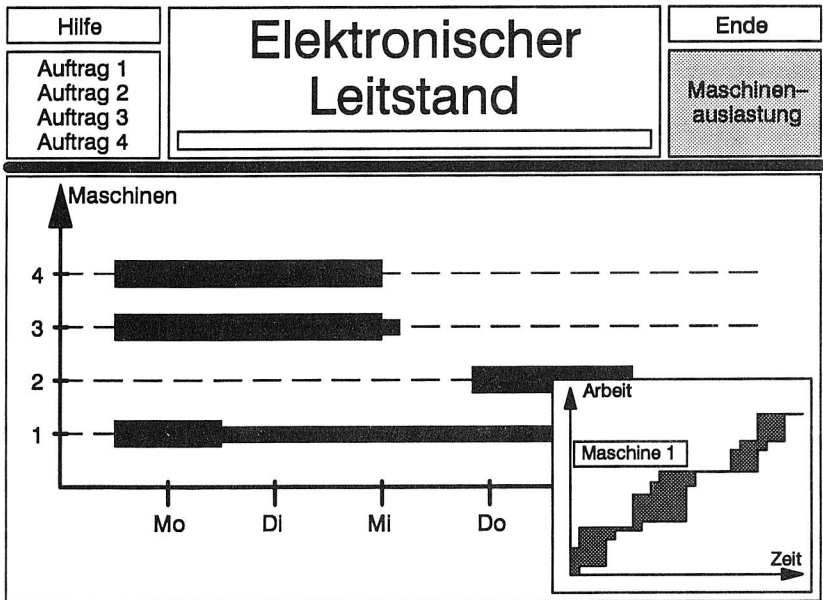


Bild 7: Plantafel eines elektronischen Leitstands

2.2 Nutzung vorhandener Flexibilitätspotentiale

2.2.1 Definition der Flexibilität

Auch der Begriff der Flexibilität findet nicht unbedingt eine einheitliche Verwendung, wie zahlreiche Veröffentlichungen belegen /17/, /34/, /35/, /36/. Von Fall zu Fall wird Flexibilität als Synonym für Begriffe wie Universalität, Reaktionsfähigkeit, Vielseitigkeit oder ähnlichem benutzt. In Anlehnung an die Definition von Kalde /17/ wird daher die Flexibilität wie folgt definiert:

Flexibilität ist die Fähigkeit, sich ändernden Situationen und Anforderungen anzupassen.

Bild 8 unterscheidet hierzu verschiedene Flexibilitäten in der Fertigung.

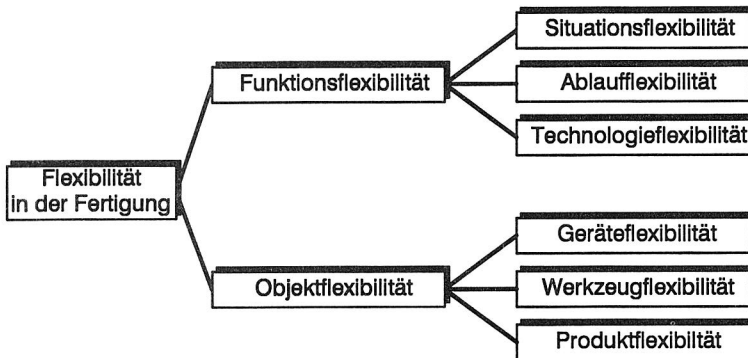


Bild 8: Flexibilität in der Fertigung

Für die weiteren Ausführungen wird eine Grobklassifizierung der Flexibilitätsarten nach Funktions- und Objektflexibilität verfolgt. Innerhalb dieser Klassifizierung kann nochmals in verschiedene Arten unterteilt werden, welche nachstehend kurz charakterisiert sind.

Die **Situationsflexibilität** beschreibt die Fähigkeit, auf die ständig wechselnden Gegebenheiten im Fertigungsverlauf reagieren zu können. Unvorhergesehene Ereignisse, wie zum Beispiel ein Geräteausfall, können Engpaßsituationen bewirken und müssen durch Strategieänderungen weitgehend kompensiert werden.

Eine Bearbeitung gleicher Aufträge auf unterschiedliche Weise kennzeichnet die **Ablaufflexibilität**. Dies umfaßt sowohl die Bearbeitung gleicher Werkstücke auf verschiedenen Maschinen als auch in unterschiedlicher Reihenfolge. Für sehr viele Bearbeitungen sind durchaus mehrere Arbeitsgangfolgen möglich.

Unter **Technologieflexibilität** wird die Fähigkeit verstanden, bei gegebenen technischen und organisatorischen Voraussetzungen alternative Arbeitsgänge durch-

zuführen. Zum Beispiel können bei entsprechenden Prämissen so unterschiedliche Technologieprozesse wie Zirkularfräsen, Planansenken und Plandrehen eingesetzt werden /8/.

Die **Geräteflexibilität** bietet nicht nur die Eigenschaft, das System durch Umrüsten (Rüstflexibilität) an andere Aufträge anzupassen, sondern auch gleiche Aufträge oder Arbeitsgänge auf verschiedene Maschinen zu verteilen.

Ähnlich hierzu ist die **Werkzeugflexibilität** zu sehen, bei der für die Bearbeitung alternative Werkzeuge oder Fertigungshilfsmittel genutzt werden können. Bei einem Werkzeugbruch muß daher nicht unbedingt der Arbeitsgang auf eine andere Maschine gelegt werden, sondern es kann auf ein anderes Werkzeug, falls vorhanden, zurückgegriffen werden.

Die **Produktflexibilität** kann sowohl bei der quantitativen (Mengenflexibilität) als auch bei der qualitativen (Variantenflexibilität) Produktion von Erzeugnissen einfließen. Zur wirtschaftlichen Fertigung muß ein System den Stückzahl-schwankungen bis hin zur Losgröße 1 anpaßbar sein. Verschiedene Varianten müssen ebenso bearbeitet werden können.

2.2.2 Flexibilitätspotentiale

Die technische Entwicklung der Rechenanlagen wird mit rasanter Geschwindigkeit vorangetrieben. Zur Steuerung der Fertigung werden heutzutage bereits Personal-computer eingesetzt, deren Leistungsfähigkeit an die eines Arbeitsplatzrechners (Workstation) reicht. Hinzu kommt, daß der Fortschritt im Geräte- und Maschinenbereich zu hochautomatisierten Komponenten tendiert.

Aus dieser Tatsache resultiert, daß eine Reihe von **Freiheitsgraden zur flexiblen Fertigung** existieren muß. Dem widerspricht jedoch die Wirtschaftlichkeit zahlreicher Fertigungssysteme. Der Grund dafür liegt sehr oft in einer Software, die die vorhandenen Freiheitsgrade in Form von Flexibilitätspotentialen nicht oder nur unzureichend nutzt.

Infolgedessen wird an dieser Stelle ein Anforderungskatalog zur Nutzung dieser Freiheitsgrade formuliert, um im weiteren zu zeigen, wie dieser durch die Gestaltung

geschickter Softwarestrukturen und intelligenter Algorithmen für die Fertigung erfüllt werden kann.

Anforderung 1:

In der Arbeitsvorbereitung sind innovative Darstellungsstrukturen notwendig, die bezüglich des Fertigungsverlaufs sowohl Ablauf- als auch Technologiealternativen erlauben.

Anforderung 2:

Alternative Arbeitsplanstrukturen, die eine erweiterte Ressourcenbetrachtung ermöglichen, sind unumgänglich. Die Berücksichtigung von ersetzenden oder ergänzenden Betriebsmitteln (Geräte, Werkzeuge, ...) ist dringend erforderlich.

Anforderung 3:

Hybride Systeme rücken stärker in den Vordergrund und dürfen einer Betrachtung nicht entzogen werden. Die Softwarestrukturen sind diesbezüglich universell zu gestalten.

Anforderung 4:

Bei vorgegebenen Rahmenbedingungen müssen die Werkstattsteuerungssysteme durch entsprechende Algorithmen in der Lage sein, für einen vorgegebenen Zeitraum einen hinsichtlich eines bestimmten Zielkriteriums optimalen Belegungsplan zu erzeugen.

Anforderung 5:

Die Werkstattsteuerung darf sich nicht nur auf die Planung beschränken, sondern muß vielmehr die Fertigung überwachen, um bei unvorhergesehenen Ereignissen **regelnd** eingreifen zu können. Zudem muß durch die Steuerungskomponente gewährleistet sein, daß alle benötigten Materialien und Betriebsmittel zum richtigen Zeitpunkt am richtigen Ort zur Verfügung stehen.

3 Softwaretechnische Aspekte zur rechnerintegrierten Werkstattsteuerung

Die Informatik bietet zur Bearbeitung der unterschiedlichsten Aufgaben eine sehr große Anzahl von Softwaremethoden. Deshalb ist es schon vor der Konzeptionsphase wichtig, verschiedene Methoden unter dem Aspekt der Eignung für eine konkrete Aufgabenstellung gegenüberzustellen und zu bewerten.

Im vorliegenden Fall ist ein rechnergestütztes Informationssystem zur Werkstattsteuerung zu konzipieren. Das vielfältige Wissen, welches zur Behandlung dieser Aufgabe erforderlich ist, ist daher entsprechend zu repräsentieren. Je besser dieses Wissen strukturiert, gespeichert und verwaltet werden kann, desto effizienter und leistungsfähiger kann sich das für das Gesamtsystem auswirken. Der Wissensrepräsentation kommt also eine erhebliche Bedeutung zu, weshalb im folgenden verschiedene Wissensrepräsentationsformen kurz vorgestellt werden, um sie dann zu bewerten.

3.1 Bewertung verschiedener Wissensrepräsentationsformen

Die Wissensrepräsentation befaßt sich mit der Darstellung von Wissen in einer für das verwendete System adäquaten Form. Explizit ist damit die Konstruktion eines Wissensbereichs verbunden, der für ein Anwendungsgebiet in einer entsprechenden Symbolik formalisiert wird. Eine Manipulation dieser Symbolstrukturen zur Wissenverarbeitung basiert auf geeigneten syntaktischen Operationen.

Als Teilgebiet der künstlichen Intelligenz wird die Wissensrepräsentation für zahlreiche Aufgabenstellungen eingesetzt. Besonders im Bereich der Sprachverarbeitung /37/ und Bildanalyse /38/ sind diese Einsatzfelder zu finden. Zur Gestaltung sind im wesentlichen die Grundsätze der hinreichenden Ausdruckstärke, der effizienten Darstellung und der uniformen Repräsentation zu beachten, denen ein Wissensrepräsentationsformalismus genügen sollte /39/.

3.1.1 Mathematische Logik

In der mathematischen Logik werden elementare Aussagen formuliert und durch logische Operatoren verknüpft. Durch Regeln können aus den gegebenen

Aussagen neue Aussagen abgeleitet werden. Dazu können auch sogenannte Beweisverfahren eingesetzt werden. Die Semantik der Logik ergibt sich durch die Zuordnung von Wahrheitswerten zu den Aussagen. Während in der klassischen Logik zwischen den beiden Wahrheitswerten "wahr" und "falsch" unterschieden wird, sind in der nicht-klassischen Logik zusätzliche Werte möglich. Als Beispiel hierfür sei die Fuzzy-Logik erwähnt /40/, /41/, in der jeder reelle Wert aus dem Intervall 0 bis 1 erlaubt ist. Die bekanntesten klassischen Logiken stellen die Aussagenlogik, bei der eine zweiwertige Logik auf Aussagen angewandt wird, und die Prädikatenlogik, wo diese Anwendung auf Prädikate vorgenommen wird, dar.

Die syntaktischen Elemente der Prädikatenlogik erster Ordnung sind Konstanten, Variablen, Funktionen, Prädikate, Terme und Atome /42/. Über Terme sollen Aussagen erzielt werden. Elementare logische Aussagen werden durch Atome getroffen, komplexere Aussagen mittels Junktoren und Quantoren:

<i>Junktoren:</i>	\neg Negation	<i>Quantoren:</i>	\exists Existenzquantor
	\wedge Konjunktion		\forall Allquantor
	\vee Disjunktion		
	\rightarrow Implikation		
	\leftrightarrow Äquivalenz		

Die Logik wird in der Grundform als Logik erster Ordnung verwendet. Der Problem-bereich wird in Formeln übersetzt. Der Unterschied zwischen der Aussagenlogik und der Prädikatenlogik liegt in der Existenz der Quantoren. Zusätzlich ist die Aussagenlogik entscheidbar, so daß für jede Aussage in endlicher Zeit festgestellt werden kann, ob sie wahr oder falsch ist. Für die Prädikatenlogik kann eine logisch richtige Schlußfolgerung nach einer endlichen Anzahl von Schritten festgestellt werden, das Gegenteil jedoch nicht. Es wird in diesem Zusammenhang auch von Semi-entscheidbarkeit gesprochen.

Mit den logischen Inferenzmechanismen lassen sich sehr mächtige Systeme, wie das Beispiel "Prolog" zeigt, konfigurieren. Weitere Vorteile liegen im strengen Formalismus, der Inkonsistenzen in der Darstellung des Problems beim Schlußfolgern aufdecken kann, und in der Automatisierbarkeit der Ableitung. Demgegenüber stehen die beschränkten Einsatzmöglichkeiten aufgrund einer kombinatorischen Explosion beim Problemlösen, welche die Akzeptanz der Logik als Wissens-repräsentationsformalismus stark einschränkt.

3.1.2 Regelsysteme

Diese Methode der Wissensrepräsentation basiert auf der Darstellung des Wissens in Form von wenn–dann–Regeln. Sie stellen einen Zusammenhang zwischen einem Situations– und einem Aktionsteil her:

wenn Prämisse dann Aktion

wobei die Prämisse die Bedingung zur Ausführung der Aktion bildet. Innerhalb der Aktion kann eine Anweisung stehen oder implikativ abgeleitet werden. Das gesamte System besteht aus einer Menge von Regeln, einer aktuellen Zustandsbeschreibung des Systems und Kontrollfunktionen, die auf die jeweils zutreffenden Regeln anwendbar sind /43/.

Beispiel einer Regel:

wenn der Arbeitsgang eines Werkstücks an der Maschine beendet ist
und weitere Werkstücke zur Bearbeitung anstehen,

dann nimm das Werkstück aus der Maschine
und fahre mit der Bearbeitung des nächsten Werkstücks fort

Regeln wurden mit als erstes zur Wissensrepräsentation für Expertensysteme herangezogen und sind auch heute in einer Vielzahl von Systemen anzutreffen. Die Vorteile liegen in einer expliziten Wissensdefinition und einer einfachen Modifizierbarkeit der Wissensbasis. Die Modularität und die Verständlichkeit derartiger Systeme erhöhen gerade bei umfangreichen Anwendungen die Akzeptanz beim Programmierer. Die Erweiterung bestehender Systeme wird durch die Gestaltung unabhängiger Regeln erleichtert. Dies unterstützt auch die Wartungsfreundlichkeit.

Diesen Vorteilen stehen jedoch auch Nachteile gegenüber. So kann aufgrund geringer Strukturierungsmöglichkeiten die Schwierigkeit bestehen, Wissen in Form von Regeln zu repräsentieren. Zudem ist eine Überprüfung der Verträglichkeit und Abgeschlossenheit der Regeln in vielen Fällen sehr aufwendig, wenn überhaupt möglich. Bei größeren Anwendungen kann es durch den Regelinterpretierer auch zu Effizienzproblemen kommen.

3.1.3 Constraints

Unter dem Begriff der Constraints sind explizite Abhängigkeiten zwischen Objekten beziehungsweise deren Eigenschaften zu verstehen. Bei ihrer Anwendung erfolgt eine Restriktion des Wertebereichs dieser Objekte.

Constraints können als mathematische Gleichungen angesehen werden, durch die ein Lösungsraum eingeschränkt wird. Sie repräsentieren eine n -stellige Relation, so daß in einem Constraint-Netz

- Kanten den Relationen und
- Knoten den Argumenten der Relationen entsprechen.

Die Vorgehensweise bei der Arbeit mit Constraint-Netzen liegt in der Vorgabe einer Variablenbelegung. Diese werden im Netz ausgewertet, bis sich keine weiteren Veränderungen ergeben, wobei die Erfüllung eines Constraints eine Menge von Wertetupeln bestimmt. Die Berechnung der unzulässigen Werte für Variablen wird Propagierung genannt /44/. In einem einfachen Beispiel sei ein Constraint-Netz für die Gleichung $u + v = x + y$ (exakt: $u + v = w1 = w2 = x + y$) gegeben. Die Variablen sind dunkel hinterlegt (Bild 9).

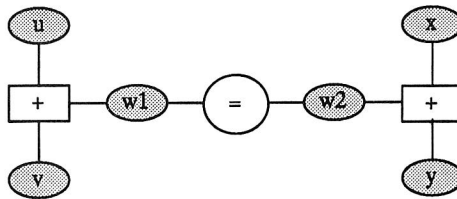


Bild 9: Constraint-Netz aus zwei Addierer- und einem Gleich-Constraint /44/

Constraints sind in der künstlichen Intelligenz weit verbreitet und eignen sich zur Darstellung von Beziehungen zwischen Objekten. Ihre Anwendung reicht vom Einsatz in Simulationssystemen bis zur Berechnung elektrischer Schaltungen /45/. Der strenge Formalismus ist ihr größter Vorteil, wogegen die Propagierung bei komplexen Problemstellungen lange Verarbeitungszeiten verursachen kann.

3.1.4 Relationale Datenmodelle

Relationale Datenmodelle stellen eine formale Methode zur Strukturierung von Informationen aus einem vorgegebenen Bereich dar. Sie stehen in enger Verwandtschaft mit der Logik und entsprechen den Erweiterungen der Prädikate. Grundlage bildet ein Modell, welches folgende Eigenschaften besitzt /46/:

- Einheitliche Repräsentation der Informationen
- Verborgene Verweisstrukturen für den Benutzer
- Definition von Operationen

Die Relation wird definiert zwischen n Attributen und deren Wertebereichen. Eine Darstellung erfolgt üblicherweise in einer 2-dimensionalen Tabelle, deren Spalten die Attribute wiedergeben. Der Aufbau der Datenstrukturen kann über verschiedene Normalformen geschehen /47/. Die Art der Darstellung bei Informationsanalysen wird sehr gut durch ein Entity-Relationship-Diagramm unterstützt (Bild 10). Aus dem Diagramm kann direkt eine Relationendarstellung abgeleitet werden.

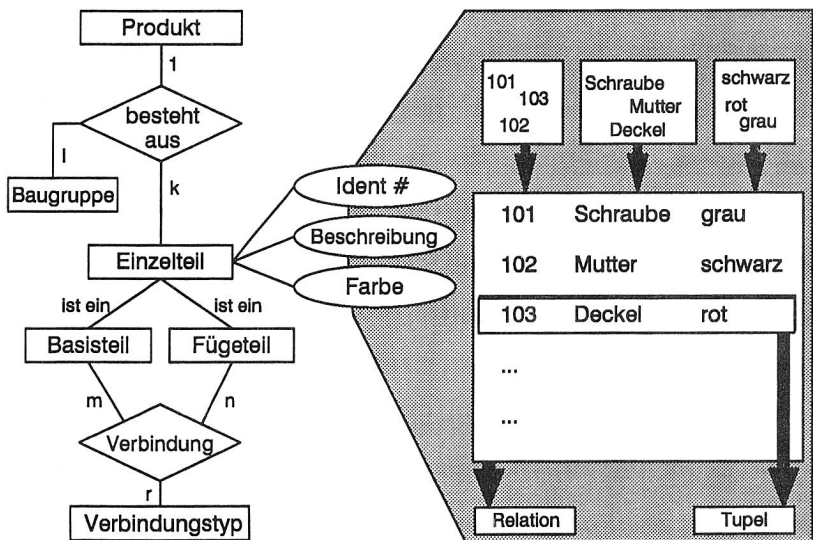


Bild 10: Beispiel eines Entity-Relationship-Diagramms

Die relationalen Datenmodelle besitzen zahlreiche Vorzüge. So zeichnet sich Ihre theoretische Basis durch Einfachheit und Verständlichkeit aus. Relationale Schemata gewähren in hohem Maße Datenunabhängigkeit und die darauf definierten Operationen sind sehr flexibel. Demgegenüber kann es jedoch gerade bei größeren Anwendungen zu Effizienzproblemen kommen /48/.

3.1.5 Frames

Die Entwicklung der Frames zur Wissensrepräsentation geht auf Minsky zurück /49/. Die Grundlage bildet die Annahme, daß das menschliche Gedächtnis eine Orientierung an Standardsituationen vornimmt und hierzu Rahmen bereitstellt. Wie die deutsche Übersetzung dieses Begriffes bereits andeutet, wird weniger ein strenger Formalismus verfolgt, als vielmehr ein allgemeiner **Rahmen** zur Beschreibung von Konzepten dargestellt. Das Rahmenkonzept vereint prozedurale und deklarative Aspekte der Wissensdarstellung. Die Modellierung von Objektklassen wird durch den dynamischen Charakter der Frames unterstützt.

Ein Frame besteht zunächst aus einer Menge von sogenannten Slots, welche durch Variablen belegt werden können. Diese sind wiederum feiner strukturierbar, und zwar in Facetten. Das Aufgabenprofil für Frames wird wie folgt umrissen /39/:

- Die konkrete Ausprägung eines Frames geschieht durch Variablenbelegung in den Slots und wird Instanz bezeichnet.
- Vorläufige Belegungen in Form von Standardwerten (Defaultwerten) entsprechen einer "Hypothese" und können jederzeit verändert werden. Generische Werte sind im Gegensatz dazu nicht veränderbar. Sie repräsentieren die Eigenschaften des Systems.
- Durch Bedingungen können Wertebereiche definiert werden, z.B. für den Slot "Achsenanzahl" wird ein Bereich zwischen 3 und 6 vorgesehen.
- Von einem Slot aus können auch Aktionen/Prozeduren gestartet werden. Dies entspricht der Vereinigung des deklarativen mit dem prozeduralen Teil. Die Überwachungsstelle der Bedingungen zur Ausführung solcher Aktionen wird Dämon genannt. Die wichtigsten

prozeduralen Zusätze sind die "Wenn–nötig"–Prozedur, die nur auf Anforderung in Kraft tritt, und die "Wenn–belegt"–Prozedur, die nur bei Wertzuweisungen durchgeführt wird.

Frame	Drehmaschine_W70										
<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td colspan="2" style="text-align: center;">Slot 1: Gerätetyp</td> </tr> <tr> <td style="width: 30%;">Facette 1</td> <td>Wert: CNC – Werkzeugmaschine</td> </tr> </table>		Slot 1: Gerätetyp		Facette 1	Wert: CNC – Werkzeugmaschine						
Slot 1: Gerätetyp											
Facette 1	Wert: CNC – Werkzeugmaschine										
<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td colspan="2" style="text-align: center;">Slot 2: Maximale–Drehzahl</td> </tr> <tr> <td style="width: 30%;">Facette 1</td> <td>Wert: —</td> </tr> <tr> <td>Facette 2</td> <td>Standard: 1000</td> </tr> </table>		Slot 2: Maximale–Drehzahl		Facette 1	Wert: —	Facette 2	Standard: 1000				
Slot 2: Maximale–Drehzahl											
Facette 1	Wert: —										
Facette 2	Standard: 1000										
<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td colspan="2" style="text-align: center;">Slot 3: Schnittgeschwindigkeit</td> </tr> <tr> <td style="width: 30%;">Facette 1</td> <td>Wert: —</td> </tr> <tr> <td>Facette 2</td> <td>Standard: —</td> </tr> <tr> <td>Facette 3</td> <td>Ermittlung: Ermittle über Maximale–Drehzahl</td> </tr> </table>		Slot 3: Schnittgeschwindigkeit		Facette 1	Wert: —	Facette 2	Standard: —	Facette 3	Ermittlung: Ermittle über Maximale–Drehzahl		
Slot 3: Schnittgeschwindigkeit											
Facette 1	Wert: —										
Facette 2	Standard: —										
Facette 3	Ermittlung: Ermittle über Maximale–Drehzahl										
<div style="text-align: center;">...</div> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td colspan="2" style="text-align: center;">Slot n</td> </tr> <tr> <td style="width: 30%;">Facette 1</td> <td></td> </tr> <tr> <td>Facette 2</td> <td></td> </tr> <tr> <td>...</td> <td></td> </tr> <tr> <td>Facette n</td> <td></td> </tr> </table>		Slot n		Facette 1		Facette 2		...		Facette n	
Slot n											
Facette 1											
Facette 2											
...											
Facette n											

Bild 11: Beispielframe

Das Beispielframe in Bild 11 veranschaulicht die Funktionsweise. In den Slots befinden sich die Facetten *Wert*, *Standard* und *Ermittlung* als eine mögliche Konfigurationsform. Weitere Facetten sind angedeutet. Durch die Belegung einer Variablen werden die Fakten widerspiegelt. Ist eine Facette *Wert* nicht belegt, so wird die Information aus *Standard* verwendet. Sind beide unbelegt, so setzt ein Vererbungsmechanismus ein, der über einen Slot geregelt wird.

Eine prägnante Eigenschaft der semantischen Rahmen sind die Zugriffsmöglichkeiten. So kann in obigem Beispiel unter der Facette *Ermittlung* eine Berechnungsmöglichkeit stehen, die den entsprechenden Wert berechnet. Der prozedurale Teil grenzt diese Form der Repräsentation von Wissen gegenüber den anderen deutlich ab. Ungünstig dagegen erweisen sich die Möglichkeiten der Strukturierung, um das gesamte Wissen in die Verarbeitung mit einzubeziehen.

3.1.6 Petrinetze

Ein gerichtetes Petrinetz ist gegeben, wenn für das Tripel $N = (S, T, F)$ gilt /50/:

- (1) $S \cap T = \emptyset$
- (2) $S \cup T \neq \emptyset$
- (3) $F \subseteq (S \times T) \cup (T \times S)$
- (4) $\text{domain}(F) \cup \text{codomain}(F) = S \cup T$,

wobei S die Menge der Zustände (Bedingungen, Stellen oder Kanäle),
 T die Menge der Aktionen (Ereignisse, Transitionen oder Instanzen) und
 F die Relation, deren Vorbereich als domain und der Nachbereich als
codomain bezeichnet werden, darstellen.

Die Theorie der Petrinetze versucht Elemente eines Systems und ihr Zusammenwirken durch Zustände und Aktionen zu illustrieren. Petrinetze sind Graphen, die aus Knoten und Kanten bestehen. Dynamische Abläufe können in Petrinetzen durch Belegung der Stellen mit Marken, sogenannte Token, erreicht werden. Eine Transition kann dann eintreten, wenn alle Stellen, die als Vorbedingung fungieren, erfüllt sind. Nach Eintreten der Transition dreht sich die Stellenbelegung um. Man spricht in diesem Zusammenhang auch vom Schalten oder Feuern einer Transition, wie Bild 12 zu entnehmen ist /51/.

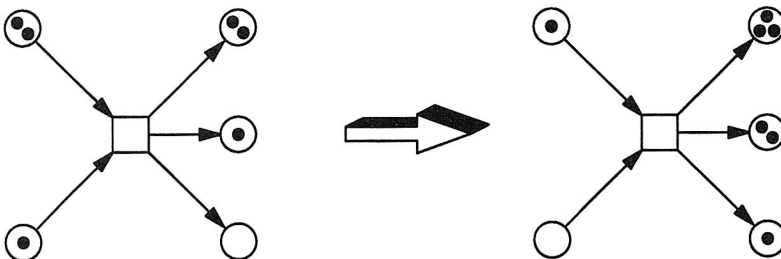


Bild 12: Schalten einer Transition in einem Petrinetz /51/

Als Beschreibungsform finden Petrinetze einen breiten Einsatz. Ursprünglich wurden diese Netze als graphische Sprache zur Darstellung, Entwicklung und Analyse von Systemen und deren Informationsflüssen verwendet. In die Entwurfs-, Planungs- und Analysephasen von Fertigungssystemen haben sie ebenso Eingang gefunden wie in die Steuerungskonzeption von Fertigungseinrichtungen /52/, /53/. Möhrle zeigt in seiner Arbeit sogar den durchgängigen Einsatz dieser Methode in allen Phasen eines Automatisierungskonzepts auf /54/.

Von den hier vorgestellten Formalismen unterscheiden sich die Petrinetze durch die gleichartige Behandlung von aktiven und passiven Komponenten. Sie unterstützen die Beschreibung von sequentiellen und parallelen Prozessen durch eine sehr anschauliche Darstellungsweise. Auch eine interaktive Erstellung solcher Netze am Rechner ist möglich, um den Entwickler zu unterstützen und die Einsatzmöglichkeiten zu erhöhen. Zudem können sie streng mathematisch formal verwendet werden, was sich in den Spezifikationsphasen als vorteilhaft erweist.

Erschwerend für den Einsatz ist die Tatsache, daß derartige Netze schnell einen relativ großen Umfang annehmen und zudem keine textuelle Darstellung existiert. Die Marken in den Petrinetzen spiegeln nur Zustandswerte wider und stellen keine beweglichen Teile dar. Marken können im Gegensatz zu physikalischen Komponenten verschmelzen oder sich teilen. Eine teilweise zu starke Verallgemeinerung läßt Petrinetze für manche theoretischen Untersuchungen ausscheiden /55/. Wesentlich gegen eine Verwendung für die vorliegende Aufgabenstellung erscheint jedoch die zu große Komplexität der Netze. Das Beispiel in Bild 13 verdeutlicht diesen Aspekt.

Gegeben sei dabei eine Fertigungsaufgabe, in der ein Werkstück zur Bearbeitung von zwei Arbeitsgängen an zwei unterschiedlichen Maschinen durch ein System geschleust wird. Das Werkstück wird zuerst für den Arbeitsgang 1 an der Maschine 1 bearbeitet, um anschließend für den Arbeitsgang 2 zur Maschine 2 transportiert zu werden. Koordiniert werden die Bereitstellungs-, Rüst-, Transport- und Bearbeitungsoperationen. Obwohl nur ein Werkstück betrachtet wird und keine Flexibilität berücksichtigt sind, erwächst sehr schnell ein komplexes Netz. Unter den hier verlangten Anforderungen würden die Petrinetze eine Größe erreichen, die sich nur äußerst schwer handhaben läßt, wodurch sie als Wissensrepräsentationsform ausscheiden.

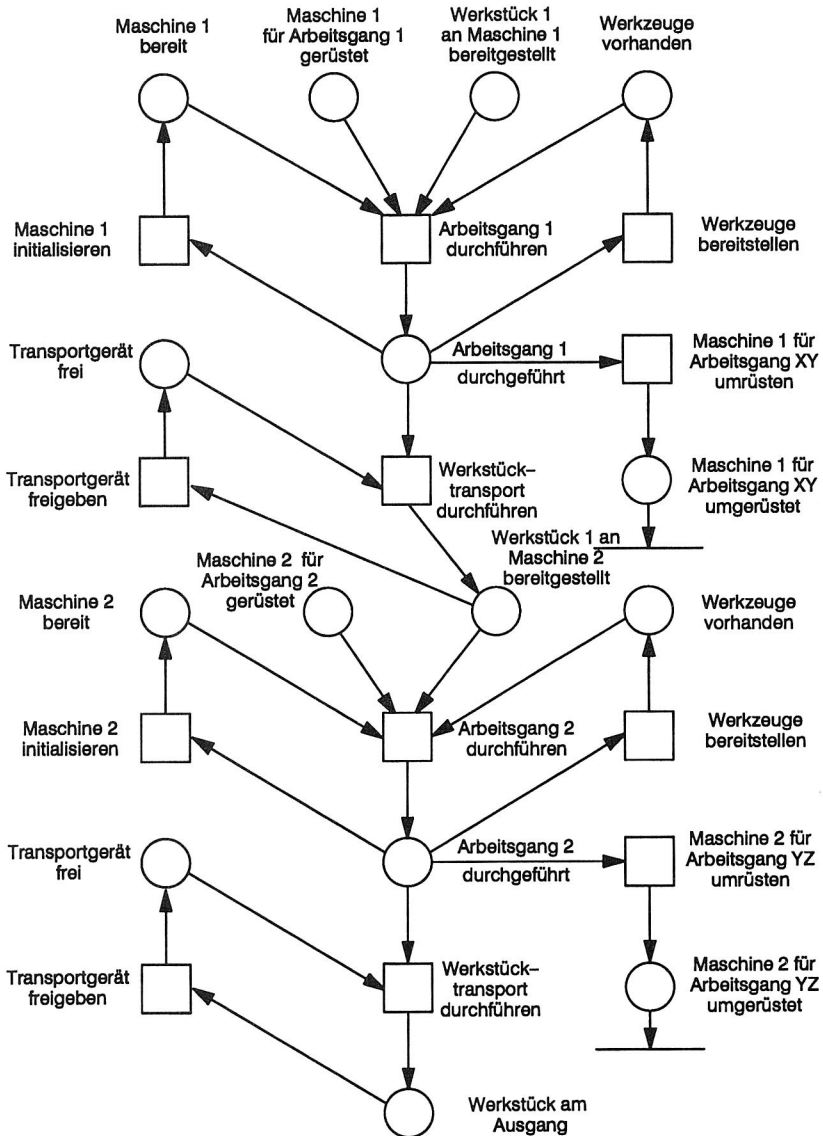


Bild 13: Petrinetz zur Darstellung einer Fertigungsaufgabe

3.1.7 Semantische Netze

Semantische Netze – gleichbedeutend wird auch von assoziativen Netzen gesprochen – wurden in ihrer ursprünglichen Form zur Darstellung der Semantik natürlicher Sprache verwendet /39/. Sie werden schon seit längerem zur Wissensrepräsentation eingesetzt. In ihrer einfachsten Form bestehen sie aus gerichteten beschrifteten Graphen mit

- Knoten, die Objekte oder Ereignisse symbolisieren, und
- Kanten, die Beziehungen zwischen diesen herstellen.

Somit repräsentieren die Knoten semantische Einheiten, während die Kanten den Relationen entsprechen /56/. Eine Unterscheidung bei Knoten kann zwischen einer intensionalen und extensionalen Beschreibung der Objekte getroffen werden. Die intensionale Beschreibung kennzeichnet ein Modell beziehungsweise einen Prototypen eines Objekts und wird **Konzept** genannt, während die Extension das konkrete, individuelle Objekt beschreibt und **Instanz** heißt. Weitere wesentliche Bestandteile eines semantischen Netzes stellen die Attribute, welche die realen Eigenschaften eines Systems beschreiben, die strukturellen Beziehungen und die Prozeduren dar /38/. Häufig verwendete Grundbeziehungen, die in annähernd jedem Netz vorkommen, sind:

- **Generalisierung**
Diese Beziehung wird in der Literatur häufig als IS_A-Beziehung bezeichnet. Entlang der Struktur wird von einem spezielleren auf ein allgemeineres Konzept verwiesen. Vererbungen von Attributen erfolgt in umgekehrter Richtung.
- **Aggregation**
Das deutsche Äquivalent zur PART_OF-Beziehung spaltet Objekte in detailliertere Komponenten auf. Dadurch wird eine Hierarchiebildung unterstützt und eine Betrachtung auf gleicher Abstraktionsebene erlaubt.
- **Individualisierung**
Auch unter der Bezeichnung INSTANCE_OF bekannt, verbindet sie die Instanz mit dem Konzept. Ein Konzept darf dabei mehrere Instanzen besitzen, eine Instanz jedoch nur ein Konzept.

Zu diesen Beziehungen existieren auch ihre Inversen. Andere Strukturrelationen, die für eine adäquate Beschreibung eines Problemfeldes erforderlich sein können und jeweils entsprechend definiert werden, bilden zusätzliche Komponenten eines Netzes. Ein strenger Formalismus zur Handhabung solcher Netze existiert leider nicht. Eine rein syntaktische Definition wäre auch völlig ungenügend. Vielmehr muß die Semantik definiert werden, damit die Knoten, Kanten und Prozeduren interpretiert werden können.

Ein System, welches durch ein semantisches Netz beschrieben wird, sollte in Analogie zur allgemeinen Struktur eines Bildanalysesystems von Niemann /57/ im wesentlichen vier Komponenten umfassen (Bild 14).

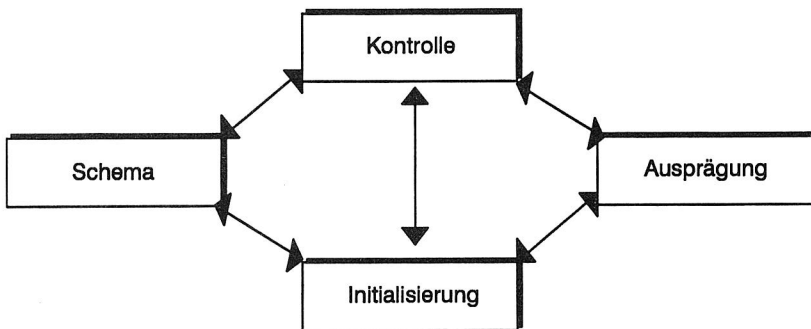


Bild 14: Prinzipielle Systemstruktur semantischer Netze

Durch die Initialisierung wird das System in einen Anfangszustand versetzt. Das Schema beinhaltet das Wissen zur Abbildung des realen Betrachtungsraums und enthält das semantische Netz, welches sich in prozedurale und deklarative Teile gliedern läßt. Der Ablauf wird durch die Kontrolle geregelt, während die Ausprägung das konkrete System widerspiegelt.

Der große Vorteil der semantischen Netze liegt in ihrer graphischen Anschaulichkeit. Sie kommen dem menschlichen Vorstellungsvermögen von den aufgezählten Methoden am besten entgegen. Zudem bieten sie eine hierarchische Strukturierung des Wissensbereichs und günstige Zugriffs- und Vererbungsmechanismen. Auch der Mangel an präzisen Formalismen schwächt diese Vorteile nur gering ab. Mit den

Möglichkeiten der Manipulation an Netzen (Hinzufügen, Löschen, Ändern, etc.) sind solche Netze zur Wissensrepräsentation optimal einsetzbar. Ein exemplarisches Netz zeigt Bild 15.

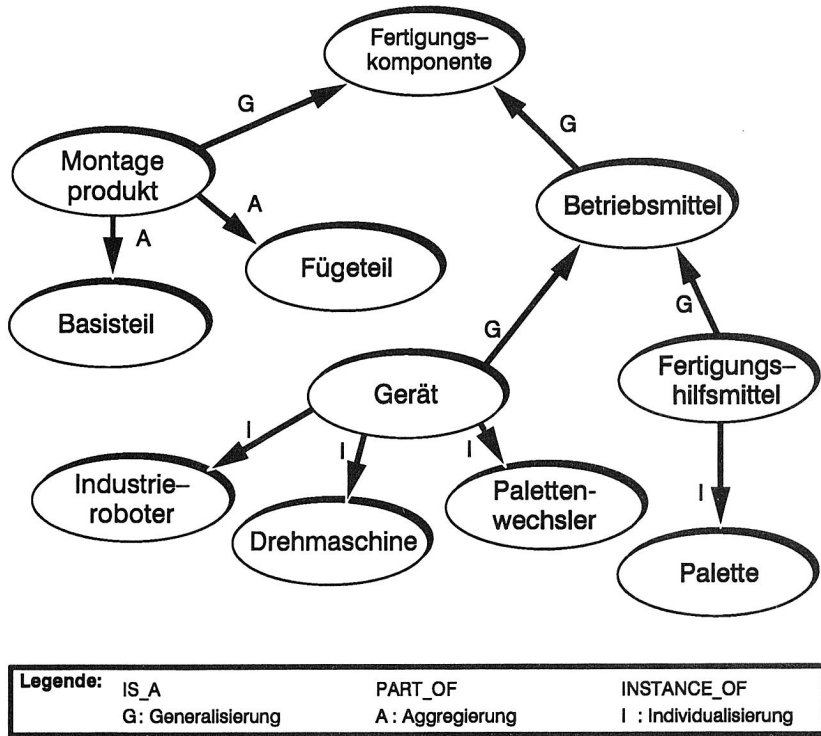


Bild 15: Ein exemplarisches semantisches Netz

3.1.8 Zusammenfassende Bewertung

In den vorhergehenden Kapiteln wurden unterschiedliche Wissensrepräsentationsformalismen vorgestellt. Gleichzeitig wurden ihre Vor- und Nachteile aufgeführt mit dem Ziel, die zur Darstellung des umfangreichen Wissens, welches für eine rechnergestützte Werkstattsteuerung erforderlich ist, geeignetsten Form zu finden.

Streng mathematisch formal kann mit den Petrinetzen gearbeitet werden. In ihrer Anwendung werden sie hauptsächlich als reines Spezifikationswerkzeug eingesetzt. Bei umfangreicheren Aufgaben – wie in diesem Fall – werden die Netze allerdings sehr komplex und verlieren dadurch an Transparenz.

Gegen eine Verwendung von Regelsystemen spricht die Eigenschaft der zu geringen Strukturierungsmöglichkeit, die den Systemen innewohnt. Auch die Frames zeigen sich diesbezüglich nicht geeignet, trotz ihrer starken Verbreitung /58/. Einen Mehrkomfort bieten zwar relationale Datenmodelle, hingegen kann es aber hier zu Effizienzproblemen während der Programmaufzeit kommen, was für eine Werkstattsteuerung nicht in Kauf genommen werden kann, da die Steuerung aufgrund ständig wechselnder Situationen in der Lage sein muß, "schnell" zu reagieren (Kapitel 6.3).

Durch ihren strengen Formalismus in der Entwurfsphase zeichnen sich besonders die mathematische Logik und die Constraints aus. Beide Formen beinhalten jedoch ebenso das Problem der zu langen Programmaufzeiten beim Ableiten beziehungsweise bei der Propagierung.

Um diese Nachteile zu vermeiden, werden für das Informationssystem zur Werkstattsteuerung von den vorgestellten Methoden die semantischen Netzen verwendet. Trotz ihres Mangels an präzisen Formalismen bieten sie schon in der Entwurfsphase mit den Möglichkeiten der Abstraktion (Generalisierung) und der Dekomposition (Aggregation) zudem fundamentale Relationen für eine objektorientierte Strukturierung. Nicht zuletzt ist aufgrund ihrer graphischen Anschaulichkeit für den Ingenieur die größte Akzeptanz von allen Methoden zu erwarten.

Für die Werkstattsteuerung werden die objektorientierten Softwarestrukturen innerhalb eines semantischen Netzes in Kapitel 5 ausführlich behandelt. Vorerst müssen jedoch grundlegende Begriffe für die objektorientierte Programmierung erklärt werden.

3.2 Programmierung

Stoyan setzt sich in seinem Buch über Programmiermethoden der künstlichen Intelligenz unter anderem mit dem Programmierstil und seiner Verwendung

auseinander /59/. Hierunter wird die Vorstellung von einem Verarbeitungsmodell assoziiert, welches durch Programmiersprachen unterstützt aber auch behindert werden kann. In Anlehnung an diese Gedanken sollen hier die wichtigsten Stilarten ohne Anspruch auf Vollständigkeit kurz erläutert werden, um anschließend die Merkmale der objektorientierten Programmierung davon abzugrenzen.

3.2.1 Verschiedene Stilarten

Konventioneller Programmierstil

Diese Art umschreibt die klassische Methode des Programmierens. Das Programm enthält unterschiedliche Datenstrukturen und Algorithmen zu deren Verwendung. Der Programmierer teilt dem Rechner genau mit, wie dieser die Variablen zu manipulieren hat.

Beispiel: C

Funktionsorientierter Programmierstil

Die Vorgehensweise ist hierbei, wie der Name bereits ausdrückt, an den Funktionen orientiert. Die Hauptmittel stellen die Verschachtelung der Terme und die bedingten Ausdrücke dar. Dadurch wird eine Neudefinition von Funktionen und deren Verfügbarkeit ermöglicht. Funktionen müssen verschiedene Typen zurückliefern können und wiederum Funktionen als Argumente erlauben /59/.

Beispiel: Lisp

Logikorientierter Programmierstil

Das Programm beinhaltet die Rekonstruktion von begrifflichen Zusammenhängen mittels Sprachen der formalen Logik. Somit stellt es eine Menge von Formeln dar und erhält erst durch die entsprechende Interpretation eine Bedeutung. Wie ein Ziel zu erreichen ist, wird nicht spezifiziert.

Beispiel: Prolog

3.2.2 Objektorientierte Programmierung

Die objektorientierte Programmierung basiert auf der Vorstellung von einer Welt miteinander kommunizierender Objekte. Jedes "Ding" aus dieser Modellwelt entspricht einem Objekt und kommuniziert mit den anderen Objekten über Nachrichten. Das Verhalten des Systems wird durch die Reaktion der Objekte auf diese Nachrichten abgebildet. Die Objekte sind verschiedenen Klassen zuzuordnen, zwischen denen eine verwandtschaftliche Beziehung existiert. Sie selbst bestehen aus **Attributen** und **Methoden** (Bild 16).

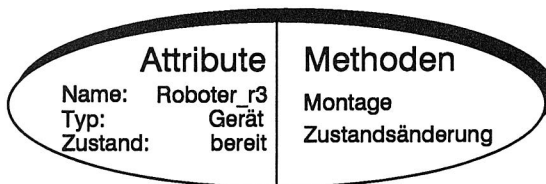


Bild 16: Bestandteile eines Objektes

Die Attribute der Objekte beschreiben deren Eigenschaften. Ihre Werte stellen die konkreten Ausprägungen dar, die ein Objekt kennzeichnen /60/. Die Methoden sind Prozeduren, die das Verhalten auf bestimmte Arten von Nachrichten widerspiegeln. Die möglichen Reaktionen eines Objekts bei Eintreffen einer Nachricht sind im wesentlichen durch Bild 17 aufgezeigt.

Die Methoden definieren demnach Aktionen, welche eine gewisse Funktionalität zur Verfügung stellen. Durch eine nach außen hin sichtbare Schnittstelle wird der Zugriff auf diese Attribute geregelt. Mißbräuchliche Datenzugriffe werden vermieden, wodurch sowohl die Programmerstellung als auch die Pflege erleichtert wird. Den Anwender interessiert nicht mehr wie etwas geschieht, sondern was. Änderungen im Implementierungsteil können leicht vorgenommen werden, ohne daß die Schnittstelle verändert werden muß. In diesem Zusammenhang wird auch von **Datenkapselung** gesprochen. Die Transparenz solcher Systeme wird gesteigert.

Die **Klassenbildung** stellt ein weiteres wesentliches Element der Objektorientierung dar. Eine Klasse repräsentiert das Konzept eines Objekts. Mittels der Instanzenbildung wird eine Projektion auf ein spezielles System konstruiert.

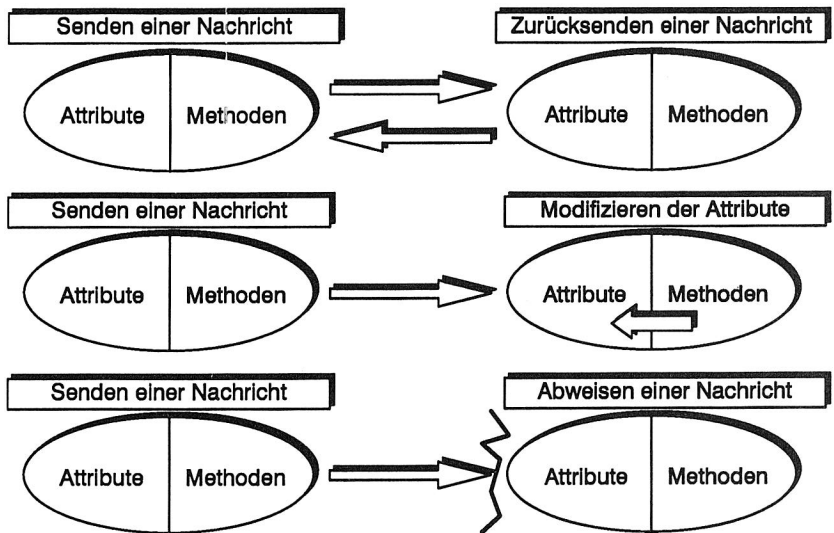


Bild 17: Unterschiedliche Reaktionen auf das Eintreffen einer Nachricht

Ferner unterstützen die Klassen das **Vererbungsprinzip**. Bei diesem Prinzip, das nach Wedekind /61/ mit der Technik der Objektorientierung untrennbar verknüpft ist, entsteht eine neue Klasse, indem Attribute und Methoden einer bestehenden Klasse übernommen und neue Objektkomponenten hinzugefügt werden. Der neuen Klasse (Unterklasse) stehen die Methoden der Oberklasse zur Verfügung.

Bei Übernahme von Methoden aus der Oberklasse können diese eventuell modifiziert werden. Dadurch werden verschiedene Implementierungen für eine Methode möglich, was unterschiedliches Verhalten für unterschiedliche Klassen zur Folge hat, obwohl die Schnittstelle dieselbe geblieben ist. Bei diesem Verhalten wird auch von **Polymorphismus** gesprochen.

Die Merkmale der objektorientierten Programmierung bieten zahlreiche Vorteile, welche die Wiederverwendbarkeit der Software erhöhen und die Portabilität steigern. Ein in diesem Stil konzipiertes System ist zudem leichter zu erweitern als herkömmliche Architekturen. Die Denkweise des Programmierers wird sich zu Anfang verändern müssen, da viel mehr Gedankengut in die Strukturierung und Konzeption der Objekte zu legen ist, wobei sich dieser Aufwand bei der Realisierung auszahlen wird.

4 Der Ablaufgraph – Innovative Komponente in der Arbeitsvorbereitung

4.1 Architekturmodell

Zur Herstellung eines Erzeugnisses ist während des Arbeitsablaufs eine Reihe von Einzelschritten notwendig. Diese, auch als Operationen bezeichnet, entsprechen den Arbeitsvorgängen und bilden in ihrer Gesamtheit den Fertigungsprozeß des Produkts. Für eine detailliertere Sichtweise bietet es sich aufgrund der teilweise komplexen Erzeugnisstrukturen an, die globalen Zusammenhänge in einem Architekturmodell darzustellen. Ein solches Architekturmodell strukturiert die zur Fertigung benötigten Informationsinhalte und ermöglicht unterschiedliche Betrachtungsweisen. Bild 18 zeigt das in dieser Arbeit verwendete Modell.

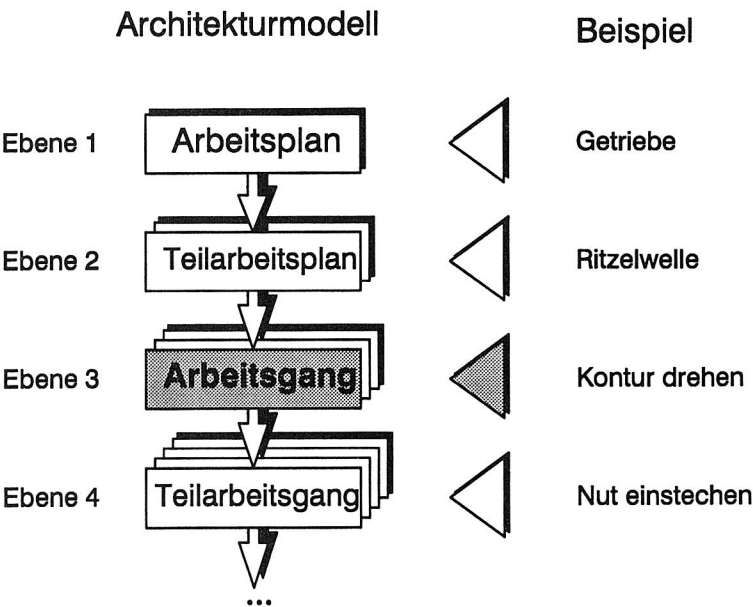


Bild 18: Architekturmodell

Die n Elemente einer Ebene i setzen sich aus den m Elementen der tieferliegenden Ebene $i + 1$ zusammen, mit $i, m, n \geq 1$ und $m \geq n$. Ebene 1 beschreibt das fertige Erzeugnis durch den Gesamtarbeitsplan. Es besteht aus Einzelteilen oder Baugruppen, welche in Ebene 2 den Teilarbeitsplänen entsprechen. Diese wiederum sind in Arbeitsgänge gegliedert, die in Ebene 3 angesiedelt sind. Eine noch feinere Strukturierung erfolgt über den Teilarbeitsgang (Ebene 4). Er wird durch das Modell zugelassen, soll aber im weiteren Verlauf nicht berücksichtigt werden. Ebene 3 kennzeichnet das relevante Niveau für das Informationssystem zur Werkstattsteuerung in dieser Arbeit.

Allgemein kann behauptet werden, daß derartige Informationssysteme auf den Elementen einer Ebene definiert werden sollen. Durch eine Modularisierung und wohldefinierte Schnittstellen wird eine klare Softwarestruktur unterstützt und ein kontrollierter Zugriff auf die einzelnen Einheiten gewährleistet.

4.2 Zeitliche Relationen von Arbeitsgängen

In der Wissensrepräsentation existieren zahlreiche Ansätze für die Behandlung der Zeit (Zeitachsen, formale Modelle, Zustandsräume, u.a.). Um die zeitlichen Bezüge zwischen Arbeitsgängen zu erfassen, werden ihnen hier Zeitintervalle zugeordnet, welche der Dauer des jeweiligen Arbeitsgangs entsprechen. Zwischen diesen Intervallen können nun verschiedene Relationen definiert werden. Basis dafür bildet eine Grundeinteilung für die Ausführung zweier Arbeitsgänge in sequentiell und parallel:

Arbeitsgang A wird vor Arbeitsgang B ausgeführt: $A > B$ (sequentiell)

Arbeitsgang A und B werden gleichzeitig ausgeführt: $A = B$ (parallel)

Welche genauen zeitlichen Betrachtungen nach Richter /39/ unter dieser Einteilung zu verstehen sind, detailliert Bild 19.

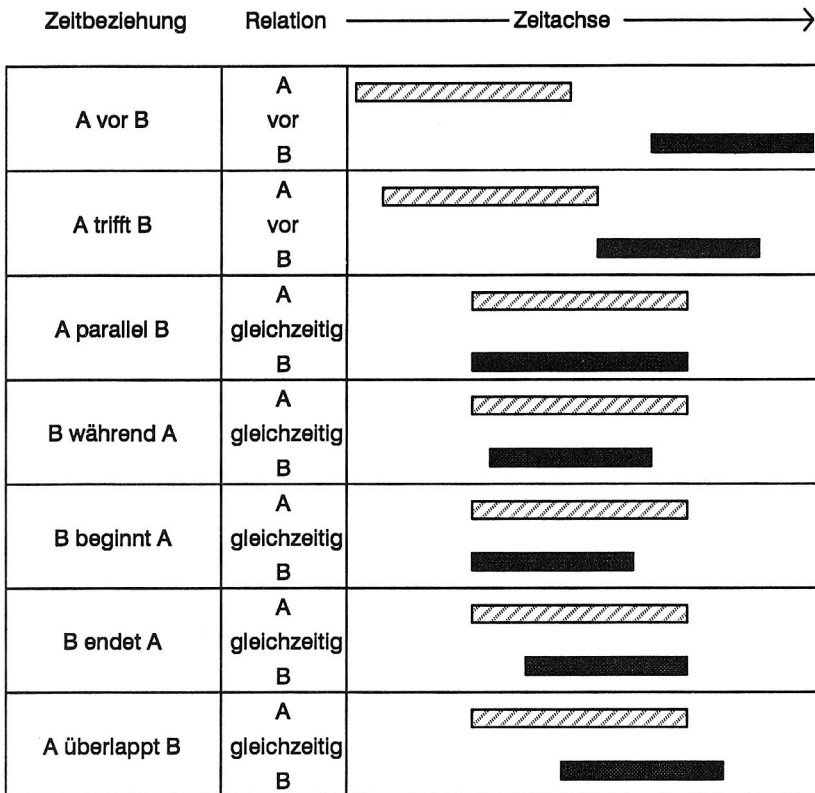


Bild 19: Detaillierte Zeitbeziehungen zwischen zwei Arbeitsgängen

Die wichtigsten Eigenschaften der Vor-Relation sind Ihre Irreflexivität, Transitivität und Asymmetrie:

$\forall A, B, C$ aus der Menge der Arbeitsgänge gilt:

Irreflexivität: $\neg(A > A)$

Transitivität: $A > B \wedge B > C \rightarrow A > C$

Asymmetrie: $A > B \rightarrow \neg(B > A)$

Weiterhin ist festzustellen, daß die Gleichzeitig-Relation aus der Vor-Relation konstruierbar ist:

$$A = B \leftrightarrow \neg(A > B) \wedge \neg(B > A)$$

Mit den inversen Relationen lassen sich somit für zwei Arbeitsgänge vier mögliche Ausführungsmöglichkeiten ableiten:

$$\begin{array}{ll} A > B & B > A \\ A = B & B = A \end{array}$$

Bei drei Arbeitsgängen sind dies bereits 24 Möglichkeiten:

$$\begin{array}{llll} A > B > C & A > C > B & B > A > C & B > C > A \\ C > A > B & C > B > A & A > B = C & A > C = B \\ B > A = C & B > C = A & C > A = B & C > B = A \\ A = B > C & B = A > C & B = C > A & C = B > A \\ A = C > B & C = A > B & A = B = C & A = C = B \\ B = A = C & B = C = A & C = A = B & C = B = A \end{array}$$

Die Anzahl aller Ausführungsmöglichkeiten, also **ohne Berücksichtigung der technologisch vorgegebenen Restriktionen**, errechnet sich für n Arbeitsgänge nach folgender Formel:

$$n! \cdot 2^{n-1}$$

Dies ergibt für

4 Arbeitsgänge	192	Möglichkeiten,
5 Arbeitsgänge	1920	Möglichkeiten,
6 Arbeitsgänge	23040	Möglichkeiten,
usw.		

Obwohl sich schon für kleine n eine große Anzahl von Möglichkeiten ergibt, wird der kombinatorischen Explosion durch konstruktionstechnische und ablauflogistische Gegebenheiten aus der Praxis stark Einhalt geboten. So bestehen im normalen Fertigungsablauf zwischen den einzelnen Arbeitsgängen Reihenfolgebeziehungen, die eingehalten werden müssen.

Wie bereits in Kapitel 2.1.1 ausgeführt, werden in der Arbeitsvorbereitung alternative Ablauffolgen momentan noch nicht berücksichtigt und somit auch nicht genutzt. Fischer versucht in seiner Arbeit /35/ mittels Fertigungsstrukturgraphen eine Möglichkeit aufzuzeigen, wie in der Teilefertigung dieses Manko behoben werden könnte. Aus dem Montagebereich ist zwar der Vorranggraph bekannt, jedoch lassen sich mit ihm auch nicht alle Möglichkeiten darstellen, wie folgendes Beispiel aus den Abhandlungen von Kalde /17/ beweist:

Gegeben sei eine Welle, auf die ein Zahnrad und zwei Lager zu fügen sind, wobei das Zahnrad spätestens nach einem der Lager montiert werden muß. Für diese Aufgabenstellung kann der Ablauf in drei Arbeitsgänge aufgeteilt werden:

- A: Montage Zahnrad
- B: Montage Lager
- C: Montage Lager

Als möglich erweisen sich folgende Reihenfolgen, welche nicht in einem Vorranggraphen dargestellt werden können.:

$A > B > C$
 $B > A > C$
 $A > C > B$
 $C > A > B$

Konsequenterweise wird, auch im Sinne einer ganzheitlichen Betrachtung von Teilefertigung und Montage, nach Strukturen verlangt, mit denen alle Ablaufalternativen für die Fertigung eines Produktes darstellbar sind. Deshalb werden basierend auf der Grundeinteilung (Vor- und Gleichzeitig-Relation) sieben Zeitrelationen definiert (Bild 20), die eine Reihenfolgebeziehung zwischen zwei Arbeitsgängen vorschreiben und nachfolgend beschriebener Semantik entsprechen.

- I) $A > B$
Arbeitsgang A muß vor B ausgeführt werden.
- II) $A < B \leftrightarrow \neg(A > B) \wedge \neg(A = B)$
Arbeitsgang B muß vor A ausgeführt werden.

- III) $A <> B \leftrightarrow (A > B) \vee (B > A)$
Arbeitsgang A kann vor B ausgeführt werden oder umgekehrt.
- IV) $A <=> B \leftrightarrow (A > B) \vee (B > A) \vee (A = B)$
Reihenfolge völlig beliebig
- V) $A <= B \leftrightarrow (B > A) \vee (A = B)$
Arbeitsgang A darf nicht vor B ausgeführt werden.
- VI) $A >= B \leftrightarrow (A > B) \vee (A = B)$
Arbeitsgang B darf nicht vor A ausgeführt werden.
- VII) $A = B \leftrightarrow \neg(A > B) \wedge \neg(B > A)$
Arbeitsgang A und B müssen gleichzeitig ausgeführt werden

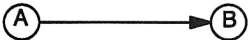
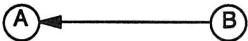
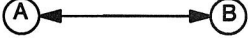
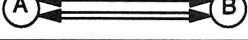
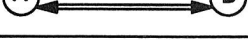
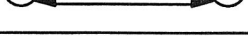
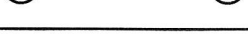
Zeitrelation	Notation	Graphische Darstellung
A vor B	$A > B$	
B vor A	$A < B$	
A und B sequentiell	$A <> B$	
A und B beliebig	$A <=> B$	
A nicht vor B	$A = < B$	
B nicht vor A	$A >= B$	
A parallel B	$A = B$	

Bild 20: Zeitrelationen zur Darstellung von Ablaufalternativen

Mit dieser Einteilung wird eine vollständige Bereitstellung möglicher Zeitrelationen geschaffen. Sie sind direkt in praktischen Anwendungen einsetzbar, wie die Ausführungen in Kapitel 7 zeigen.

4.3 Darstellung durch Graphen

4.3.1 Grundlegende Definitionen

In vielen Anwendungsfällen gelten Graphen aufgrund ihrer Anschaulichkeit als sehr beliebtes Darstellungsmittel. Sie bestehen aus Knoten und Kanten, sind mathematisch beschrieben als Tupel $G = (V, B)$ durch

- eine Menge V von Knoten und
- einer Relation $B \subset V \times V$,

sowie einer Inzidenzabbildung. Die Kardinalität $|V|$ der Knotenmenge wird als Ordnung des Graphen bezeichnet /62/.

Der **ABLAUFGRAPH** wird nun definiert als Graph A , dessen

- (1) Knotenmenge die Menge K^1 aller operativen Einheiten eines Arbeits- oder Teilarbeitsplans (Arbeitsgangmenge) und dessen
- (2) Relation durch $R = \{ >, <, <>, <=>, <=, >=, = \}$

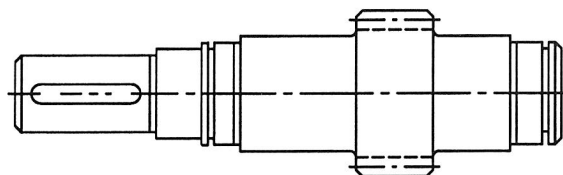
gegeben ist. Die Ordnung dieses Graphen ist endlich. Isolierte Teilgraphen dürfen nicht existieren.

Der Ablaufgraph beschreibt somit zeitliche Beziehungen zwischen den Arbeitsgängen. Jeweils zwei Knoten sind durch eine Ablaufreihenfolge beschrieben. Aus der Gesamtheit der Arbeitsgänge ergeben sich alle möglichen Reihenfolgen zur Fertigung eines Produkts, einer Baugruppe oder eines Werkstücks.

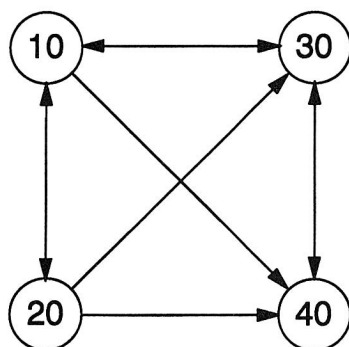
Am Beispiel der Fertigung einer Ritzelwelle für ein 1-stufiges Getriebe soll der Sachverhalt illustriert werden (Bild 21). Der Teilarbeitsplan enthält vier Arbeitsgänge, deren Ausführung mehrere Möglichkeiten zulassen. So kann der Arbeitsgang 10 vor 20 oder danach durchgeführt werden. Ebenso verhält es sich mit den Arbeitsgängen 30 und 40, wodurch insgesamt folgende Reihenfolgen erlaubt sind:

1) Ein Knoten kann einen oder mehrere Arbeitsgänge umfassen

- 1.) $10 > 20 > 30 > 40$
- 2.) $10 > 20 > 40 > 30$
- 3.) $20 > 10 > 30 > 40$
- 4.) $20 > 10 > 40 > 30$
- 5.) $20 > 30 > 10 > 40$



Arbeitsgänge:



10 Kontur drehen rechts

20 Kontur drehen links

30 Paßfedernut fräsen

40 Verzahnung wälzfräsen

Bild 21: Ablaufgraph zur Fertigung einer Ritzelwelle

Die verschiedenen Kantentypen des Graphen bestehen aus den sequentiellen und parallelen Grundrelationen. Der Sequenz entspricht ein einfacher Pfeil. Die gleichzeitige Bearbeitung wird durch einen Doppelpfeil repräsentiert. Um eine gültige Reihenfolge zu ermitteln, muß nur ein Pfad entlang den Pfeilrichtungen durch den Graphen gefunden werden, auf dem alle Knoten genau einmal durchlaufen werden, ohne die bestehenden Rahmenbedingungen zu verletzen. Dies ist dann der Fall, wenn zum Beispiel ein Knoten passiert wird, in den eine einzelne sequentielle Kante (Vor-Beziehung) von einem bis dahin noch nicht durchlaufenen Knoten einläuft.

4.3.2 Vervollständigung des Ablaufgraphen

Um die zulässigen Reihenfolgen aus dem Ablaufgraphen automatisch ermitteln zu können, wird in dieser Arbeit ein vollständiger Ablaufgraph vorausgesetzt. Bei n Knoten müssen also $(n * (n - 1))/2$ Beziehungen existieren. Bei zunehmender Knotenzahl steigt also auch die Anzahl der Beziehungen. Um zur Erstellung des Ablaufgraphen diesen Aufwand zu minimieren, kann aufgrund transitiver Eigenschaften bestimmter Relationen für zwei gegebene Beziehungen bei drei Knoten die resultierende dritte Beziehung algorithmisch ermittelt werden. Seien für die Arbeitsgänge A, B, und C die Beziehungen A vor B und B vor C, ist es direkt einleuchtend, daß A vor C abzuleiten ist.

In der Regel stehen zur Bestimmung einer Beziehung mehrere Möglichkeiten zur Verfügung. In einem Graphen mit n Knoten, bei dem bis auf eine alle Beziehungen festliegen, kann zur Ermittlung dieser letzten Kante auf $n - 2$ Möglichkeiten zurückgegriffen werden. Aus diesen wird nun die restriktivste Beziehung ausgewählt. Dabei muß sich der Planer bewußt sein, daß eine nicht so restriktive Kante ermittelt werden kann, wie vielleicht technologisch erlaubt ist.

Wird vom Planer ein nicht vollständiger Ablaufgraph vorgegeben, um die restlichen Kanten rechnerunterstützt ermitteln zu lassen, so können Planungsfehler in Form von unzulässigen Kanten aufgedeckt werden.

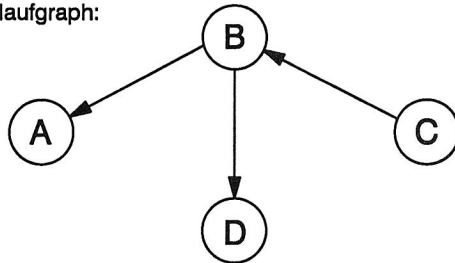
Das Beispiel in Bild 22 verdeutlicht die Vorgehensweise beim Vervollständigen. In den vorliegenden unvollständigen Graphen sind die Beziehungen A–C, A–D und C–D einzutragen. Im ersten Schritt folgt aus B vor A und B vor D die Beziehung A *beliebig* D. Ebenso läßt sich aus C vor B und B vor D C vor D folgern. Als weitergehenden Schritt können für die A–C Beziehung aus

- | | |
|-------------------------------------|---------------------|
| 1.) A <i>beliebig</i> D und C vor D | A <i>beliebig</i> C |
| 2.) B vor A und C vor B | C vor A |

abgeleitet werden.

Da die zweite Möglichkeit in der ersten enthalten ist, aber nicht umgekehrt, schränkt diese die erste ein und wird daher ausgewählt. Sie verkörpert in diesem Fall die restriktivere Beziehung.

Unvollständiger Ablaufgraph:



Vervollständigter Ablaufgraph:

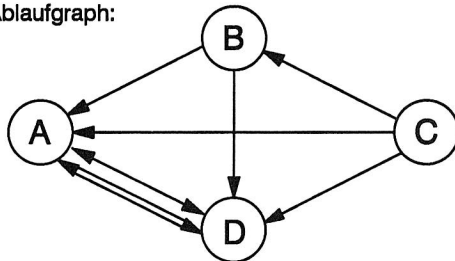


Bild 22: Vervollständigung eines Ablaufgraphen

4.3.3 Bedingte und unbedingte Kanten

Obwohl nun mit dem Ablaufgraphen ein sehr mächtiges Werkzeug vorhanden ist, um nahezu alle fertigungstechnisch möglichen Bearbeitungsfolgen darzustellen, reicht der Graph in dieser Form zu einer adäquaten Repräsentation in Hinblick auf Ausnahmefälle noch nicht aus. Das Montagebeispiel einer Welle in Kapitel 4.2 zeigt, daß alle erlaubten Reihenfolgen mit den bisherigen Mitteln noch nicht exakt dargestellt werden können.

Um dem zu begegnen, ist es erlaubt, den Kanten Bedingungen zuzuordnen, falls dies erforderlich ist. Die Bedingungen machen eine Aussage über das Eintreffen eines Ereignisses. Dies ist vor allem dann notwendig, wenn Zusammenhänge von mehr als zwei Arbeitsgängen betrachtet werden sollen. So darf in dem Montagebeispiel zur Ausführung der drei Arbeitsgänge der Arbeitsgang A nicht zuletzt ausgeführt werden. Den Kanten zwischen B und C kann dann die Bedingung

zugewiesen werden, daß zuerst der Arbeitsgang A ausgeführt sein muß, bevor die Arbeitsgänge B und C direkt nacheinander durchgeführt werden. Algorithmisch gesehen bedeutet das, daß bei der Wegsuche durch den Graphen die bedingten Kanten überprüft werden müssen. Ist die Bedingung erfüllt, kann fortgefahren werden, ansonsten erweist sich der momentan eingeschlagene Weg als falsch (Bild 23).

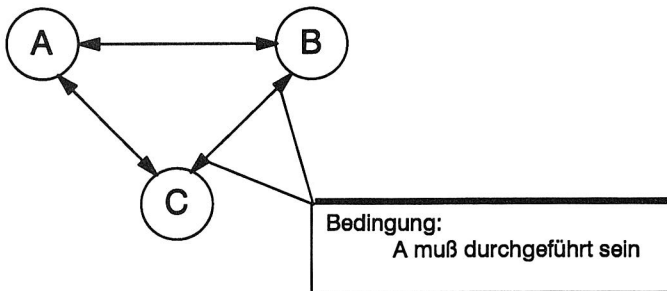


Bild 23: Bedingungen in einem Ablaufgraphen

Zur syntaktischen Notation sei folgendes vereinbart:

- Eine Kante in einem Ablaufgraph, die eine Bedingung trägt, wird bedingte Kante genannt.
- Kanten ohne Bedingung heißen unbedingte Kanten.
- Als Bedingungen sind logische Aussagen zugelassen, wobei die Konstanten den Arbeitsgängen entsprechen und mittels Junktoren verknüpft sind. Als Junktoren sind die Negation, Konjunktion, Disjunktion und Implikation zugelassen. Die Bedingungen können auf ihren Wahrheitswert hin ausgewertet werden.

Beispiele:

- A
- $A \rightarrow B \wedge C$
- $\neg B$
- $B \vee C$

Im Ablaufgraphen können nun Zusammenhänge zwischen beliebig vielen Arbeitsgängen beschrieben werden. Es ist gewährleistet, daß aus der Menge der theoretischen Reihenfolgemöglichkeiten **jede** Untermenge darstellbar ist. Mit dem Ablaufgraphen inklusive seiner Kantenbeschriftung lassen sich individuell für jede Anwendung die entsprechenden Fertigungsreihenfolgen **ohne Einschränkung** erzeugen.

Das Beispiel der Ritzelwelle aus Kapitel 4.3.1 soll dies nochmals unterstreichen. Auf die gefertigte Welle werden jeweils zwei Wälzlager und Paßscheiben montiert. Unter der Prämisse, daß das Lager vor der Scheibe zu fügen ist, spielt die weitere Reihenfolge der Operationen keine wesentliche Rolle. Einen Ablaufgraphen dazu zeigt Bild 24.

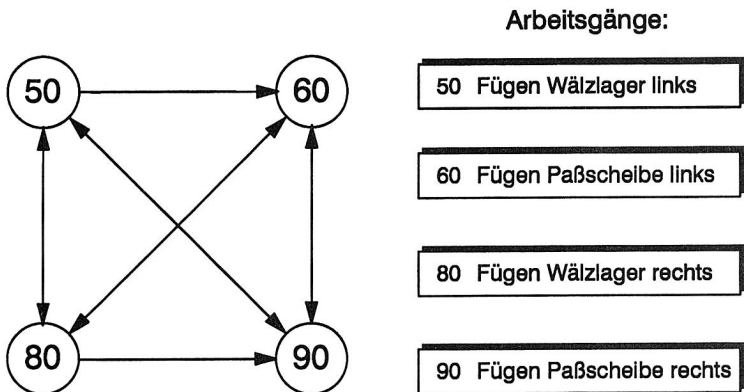


Bild 24: Ablaufgraph für Montageaufgaben an der Ritzelwelle

Für die Ablaufreihenfolgen läßt sich folgendes ermitteln:

- 1.) 50 > 60 > 80 > 90
- 2.) 50 > 80 > 60 > 90
- 3.) 50 > 80 > 90 > 60
- 4.) 80 > 90 > 50 > 60
- 5.) 80 > 50 > 60 > 90
- 6.) 80 > 50 > 90 > 60

Nun kann es sein, daß aus montage technischer Sicht die Möglichkeiten 2, 3, 5 und 6 nicht sinnvoll sind, da vielleicht nur aus einer Richtung gefügt werden kann. Durch das Einbringen von Bedingungen in obigen Ablaufgraphen kann dies jedoch verhindert werden (Bild 25). Als Resultat bleiben die Varianten 1 und 4.

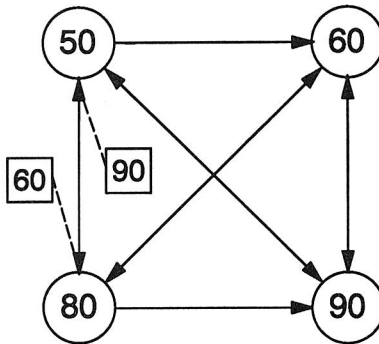


Bild 25: Ablaufgraph mit bedingten Kanten

4.3.4 Berücksichtigung der Technologieflexibilität

Unter der Technologieflexibilität ist die Fähigkeit zu verstehen, alternative Fertigungsoperationen durchzuführen (Kapitel 2.2.1). So kann eine Menge von n Arbeitsgängen durch m Arbeitsgänge ersetzt werden, wenn diese denselben Fertigungszustand am Erzeugnis herstellen. Wenn auch diese Fähigkeit eher als Ausnahme zu betrachten ist, muß sie doch in den Darstellungsstrukturen der Arbeitsvorbereitung Eingang finden. Gerade unter dem betriebswirtschaftlichen Aspekt kann diese Alternative eine nicht unerhebliche Rolle spielen.

Mit dem Ablaufgraphen besteht die Möglichkeit, diese alternativen Fertigungstechnologien zu berücksichtigen. Alternative Arbeitsgänge sind im Ablaufgraphen durch Rechtecke gekennzeichnet, welche in Parzellen aufgeteilt sind. Die einzelnen Parzellen enthalten die Operation(en), die zueinander ersetzend sind.

Für das Beispiel der Welle aus diesem Kapitel wäre es denkbar, daß die Verzahnung wälzgefräst oder wälzgestoßen und wälzgeschliffen wird, wenn in einem System entsprechende Maschinen vorhanden sind. Dadurch würde auch der Flexibilitätsgrad der Fertigung steigen. Der Ablaufgraph für die Herstellung der Ritzelwelle inklusive Technologiealternative ist in Bild 26 zu sehen.

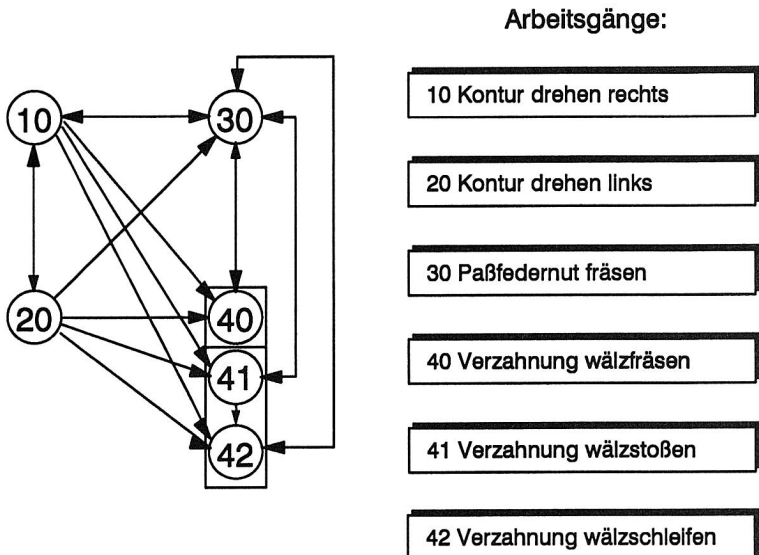


Bild 26: Ablaufgraph unter Berücksichtigung der Technologieflexibilität

Aus den fünf potentiellen Folgen ergeben sich nunmehr zwölf verschiedene Ablaufreihenfolgen:

10 > 20 > 30 > 40	10 > 20 > 30 > 41 > 42	10 > 20 > 41 > 30 > 42
10 > 20 > 40 > 30	10 > 20 > 41 > 42 > 30	20 > 10 > 30 > 41 > 42
20 > 10 > 30 > 40	20 > 10 > 41 > 30 > 42	20 > 10 > 41 > 42 > 30
20 > 10 > 40 > 30	20 > 30 > 10 > 40	20 > 30 > 10 > 41 > 42

Mit der Darstellungsmethode des Ablaufgraphen wird dem Arbeitsplaner ein Werkzeug an die Hand gegeben, um so viele **Freiheitsgrade unterschiedlicher Fertigung** datentechnisch zur Verfügung zu stellen, wie auch vorhanden sind. Starre Arbeitsplanstrukturen werden durch das zusätzliche Hilfsmittel in Form des Ablaufgraphen aufgebrochen. Flexibilitätspotentiale können im CAP-Bereich bereitgestellt und in die Fertigung weitergereicht werden. Die Durchlaufzeiten verkürzen sich durch die Nutzung zeitlich parallel ablaufender Arbeitsgänge. Alternative Ablaufreihenfolge verkleinern die Ausfallzeiten und erhöhen die Maschinenkapazitäten. Somit kann der Ablaufgraph sehr gut in der Arbeitsvorbereitung eingesetzt werden.

4.4 Algorithmische Reihenfolgeermittlung

Für ein Auftragsspektrum aus dem Fertigungsprogramm ist es die hier definierte Zielsetzung, bei gegebenen Ablaufgraphen **alle** zugelassenen Reihenfolgen zur Fertigung eines oder mehrerer Erzeugnisse zu ermitteln, um in einem daran anschließenden Planungslauf eine optimale Maschinenbelegung zu finden. Dadurch liegt ein Suchproblem vor, bei dem aus einem Graphen alle Pfade durch diesen zu finden sind.

Für derartige Probleme existiert ein sehr großes Repertoire an Suchprozeduren. Die bekanntesten Vertreter nach Winston /63/ sind:

- Breitensuche (breadth-first search)
- Tiefensuche (depth-first search)
- Strahlsuche (beam search)
- Bestensuche (best-first search)
- Bergsteigen (hill climbing)

Aus dieser Auswahl sollen nun die beiden ersten Problemlösungsmöglichkeiten gegenübergestellt werden, da nur diese ohne Zwischenbewertungen arbeiten können, was sich im ersten Ansatz für den Anwendungsfall als notwendig erweist.

Um zu einer Lösung zu gelangen, erscheint es zweckmäßig, eine Art Buchführung zu unterhalten. So können Pfade vermieden werden, die in einen endlosen Kreislauf geraten. Ist dies gewährleistet, sind Graphen äquivalent zu Bäumen /63/. In ihnen tauchen die Reihenfolgen als Äste von der Wurzel bis zum Blatt auf. Anhand der Generierungart dieser Bäume unterscheiden sich die Breiten- und die Tiefensuche.

4.4.1 Breitensuche

Bei der Erzeugung der Bäume wird stufenweise vorgegangen. Für jede Ebene werden alle Knoten ermittelt, bevor zur nächsten Ebene vorgestoßen wird. Um zu einer Lösung zu kommen, muß der komplette Baum erzeugt und im Speicher gehalten werden. Der Algorithmus arbeitet also zuerst in die Breite und benötigt daher für den rechnerinternen Aufbau des Lösungsraums viel Platz. Dies benachteiligt die Breitensuche gegenüber der Tiefensuche bei der Auswahl eines geeigneten Algorithmus. Die Arbeitsweise sei nochmals in Bild 27 vergegenwärtigt.

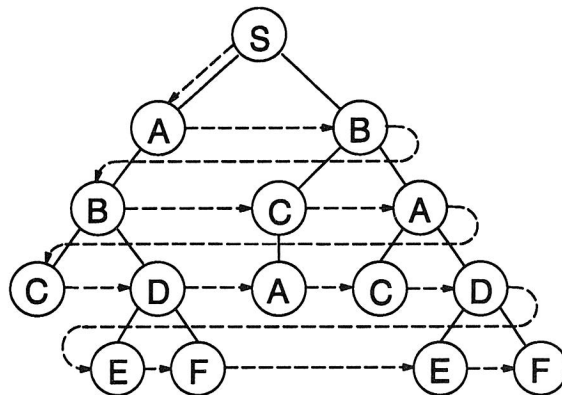


Bild 27: Breitensuche

4.4.2 Tiefensuche

Bei der Tiefensuche wird davon ausgegangen, daß möglichst schnell eine Lösung zu finden ist. Dabei wird die eingeschlagene Richtung von der Wurzel des Baumes

zu seinen Blättern unter Vernachlässigung alternativer Knoten derselben Ebene strikt fortgesetzt. So wird erst ein Ast des Baumes erzeugt, bevor mit der Generierung weiterer fortgefahren wird (Bild 28).

Man spricht in diesem Zusammenhang auch von einer sequentiellen Organisation der Knotenanalyse im Vergleich zur parallelen Organisation der Breitensuche /64/. Die speicherplatzsparende Arbeitsweise erklärt sich dadurch, daß nach dem Auffinden eines Lösungspfad es der dafür verwendete Platz freigegeben werden kann.

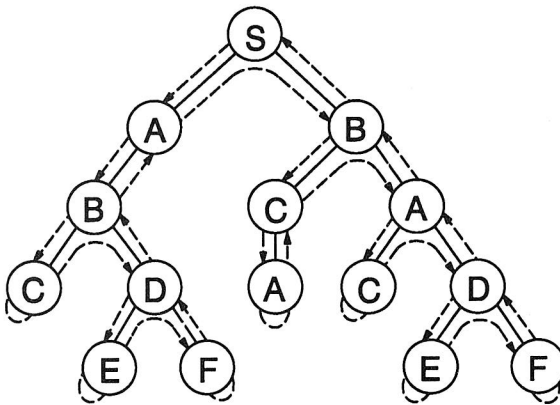


Bild 28: Tiefensuche

Aus den oben erwähnten Gründen wurde zur Implementierung der Reihenfolgeermittlung ein Algorithmus realisiert, der das Prinzip der Tiefensuche verfolgt. Der Algorithmus arbeitet im wesentlichen auf zwei nach einer Kellerverwaltung (LIFO-Strategie) organisierten Datenstrukturen, die mit VATER und SOHN bezeichnet werden. Während der VATER den zu einem bestimmten Zeitpunkt durchlaufenen Teilpfad enthält, befinden sich in SOHN der nächste zu untersuchende Knoten /65/.

Die einzelnen Schritte des Algorithmus werden nachfolgend skizziert. Um die Arbeitsweise transparenter zu gestalten, wird die Reihenfolgeermittlung eines Beispielgraphen aus Bild 29 in Bild 30 veranschaulicht. Abschließend ist der Ergebnisbaum in Bild 31 zu sehen.

- 1) Alle zulässigen Startknoten nach SOHN
- 2) Erstes Element von SOHN nach VATER
- 3) Alle noch nicht durchlaufenen Nachfolger des ersten Elements von VATER, für die Reihenfolgebeziehungen nicht verletzt werden, mit Verzeigerung zum ersten Element von VATER nach SOHN
- 4) Falls Nachfolger in 3) gefunden und Lösung noch nicht komplett, gehe zu 2) ansonsten zu 5)
- 5) Falls Lösung komplett,
wiederhole
 Ausgabe durch Zeigerrückverfolgung und
 Löschen des ersten Elements von SOHN
 solange bis SOHN leer oder
 Zeiger des ersten Elements von SOHN nicht zum ersten Element von VATER führt
- 6) Falls SOHN leer, VATER löschen und terminieren
- 7) Wiederhole
 Löschen des ersten Elements von VATER
 solange bis VATER leer oder
 Zeiger des ersten Elements von SOHN zum ersten Element von VATER zeigt
- 8) Gehe zu 2)

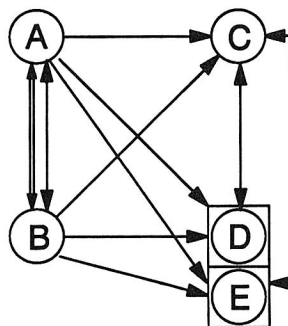


Bild 29: Beispiel-Ablaufgraph

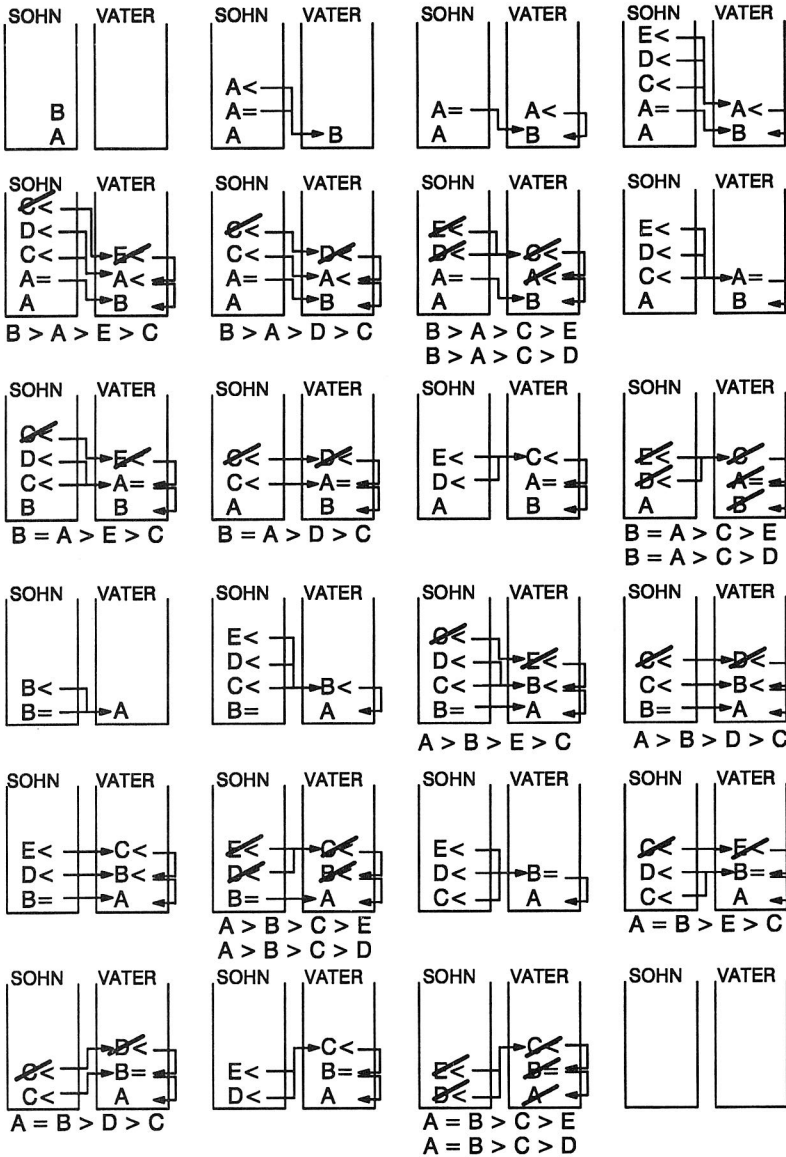
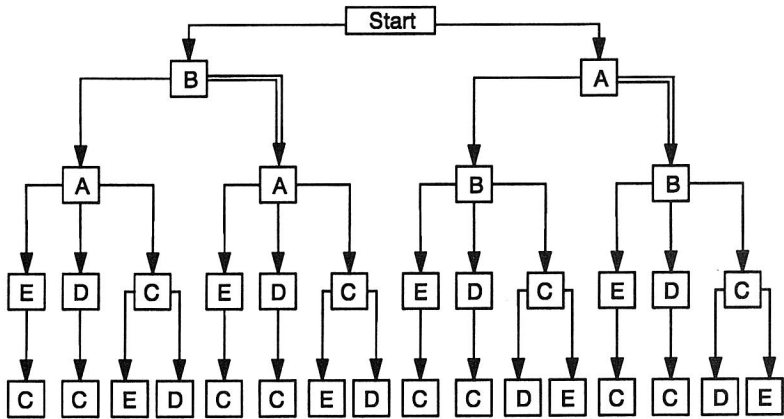


Bild 30: Arbeitsweise des Algorithmus zur Tiefsuche



Ablaufreihenfolgen:

B > A > E > C	A > B > E > C
B > A > D > C	A > B > D > C
B > A > C > E	A > B > C > E
B > A > C > D	A > B > C > D
B = A > E > C	A = B > E > C
B = A > D > C	A = B > D > C
B = A > C > E	A = B > C > E
B = A > C > D	A = B > C > D

Bild 31: Ablaufreihenfolgen des Beispielgraphen

5 Objektorientierte Softwarestrukturierung

Der Begriff der Objektorientierung ist in der letzten Zeit für die automatisierte Informationsverarbeitung verstärkt ins Blickfeld gerückt. Schlagworte wie "Objektorientierte Programmierung", "Objektorientierte Systeme" oder "Objektorientierte Strukturen" finden sich immer häufiger in der Literatur. Folgerichtig drängt sich die Frage auf, ob hier nicht nur ein zugkräftiger Terminus für etwas verwendet wird, das sich im Grunde genommen von den bisherigen Entwicklungen im Softwarebereich auch nicht unterscheidet.

Obwohl festzustellen bleibt, daß durch die Objektorientierung "das Rad auch nicht neu erfunden wird", kann jedoch die Softwareentwicklung reformiert werden. Es muß nicht gleich von einer Revolution gesprochen werden, wie es Meyer /66/ in seinem Buch über objektorientierte Softwarekonstruktion tut, doch verschiedene Tendenzen weisen unweigerlich auf Einflußfaktoren neuer Denkansätze hin. Der abstrakte Datentyp als eine Komponente innerhalb eines Systems mit definierten Schnittstellen nach außen und vor dem Benutzer verborgener Implementierung rückt den Aspekt der wiederverwendbaren Software stärker in den Mittelpunkt. Obgleich momentan die Wiederverwendbarkeit und Austauschbarkeit von Softwarekomponenten eher die Ausnahme darstellen, kann ein wichtiger Schritt gemacht werden, die Zeit der Unikat-Software zu beenden und die Softwarekosten zu senken.

Allein die Verwendung einer objektorientierten Sprache, wie zum Beispiel Eiffel, Smalltalk oder C++, impliziert aber noch lange nicht, daß objektorientiert programmiert wird. Vielmehr wird ein Vorgehen propagiert, welches in den Entwurfsphasen der Softwareerstellung Eingang finden muß. Das Hauptaugenmerk verlagert sich in das Design der Softwarestrukturen. Der Programmierer muß sich stärker als bisher Gedanken machen, wie er sein Konzept strukturiert, um es mit den ihm zur Verfügung stehenden Mitteln umzusetzen. Durch das "einfach mal herunterprogrammieren" werden die Fähigkeiten, welche durch die Objektorientierung geboten werden, nicht genutzt, weil eine schrittweise Verfeinerung auf der Basis von Objekten und deren Eigenschaften (Vererbung, Polymorphismus, etc.) nicht möglich ist. Der Mechanismus der Abstraktion verlangt vielmehr eine sorgfältige Strukturierung und Klassifizierung des zu betrachtenden Problemfeldes.

5.1 Objektorientierter Entwurf

5.1.1 Objekte

Ausgehend von der klassischen Philosophie gliedert Wedekind /67/ die Objekte grundlegend in Dinge und Handlungen. Bei der Verwendung des Terminus "Objekt" wird dadurch jedoch seine Universalität in Zweifel gestellt ("Handlung ist ein Objekt"?). Deshalb schlägt Wedekind vor, im Zusammenhang mit Objekten von Entitäten (engl. entity) zu sprechen. Fundamental ist ebenso die Unterscheidung zwischen einem Objektschema und seiner Ausprägung.

Für einen Entwurf nach objektorientierten Gesichtspunkten wird der Begriff des Objekts weiter präzisiert. Nach Booch /68/ ist eine Objektausprägung eine Entität, die wie folgt definiert ist. Sie

- ist durch seine *Attribute* und *Methoden* charakterisiert.
- hat einen *Zustand*.
- besitzt einen *Namen*.
- entspricht einer *Instanz* einer Klasse von Objekten.
- kann begrenzt "*sehen*" und "*gesehen*" werden.
- ist durch die *Spezifikation* und die *Implementierung* gegeben

Die *Attribute* eines Objekts repräsentieren die Eigenschaften und nehmen die Daten auf. *Methoden* (Aktionen)¹⁾ dienen zur Manipulation der Daten. Ein Methodenaufruf wird durch das Versenden von Nachrichten an entsprechende Objekte realisiert. Dabei ist nur das Ergebnis von Wichtigkeit, die Bewerkstelligung bleibt dem jeweiligen Objekt beziehungsweise seiner Realisierung überlassen.

Der *Zustand* eines Objekts ist durch den Wert seiner Attribute gegeben, wodurch es existiert. Die Werte können modifiziert werden, was eine dynamische Zustandsände-

1) Die Begriffe Methode und Aktion werden im folgenden synonym verwendet.

rung herbeiführt. Ebenso können Objekte erzeugt und vernichtet werden. Dies grenzt den objektorientierten Stil vom konventionellen ab, da Funktionen oder Prozeduren im allgemeinen keine Zustände besitzen.

Da ein Objekt etwas einzigartiges ist, muß es auch eindeutig zu identifizieren sein. Es wird durch seinen *Namen* bezeichnet. Mehrfache Namensvergebung kann zwar erlaubt sein, erscheint aber nicht sinnvoll.

Gleichartige Objekte können in der objektorientierten Programmierung zu einer Gruppe von Objekten zusammengefaßt werden. Die Gruppe enthält dieselben Attribute und wird als Klasse bezeichnet. Ein konkretes Objekt aus dieser Klasse wird *Instanz* genannt und verfügt über die Attributwerte, die nur für das Objekt gültig sind. Die Klasse kann somit als eine Art Schablone für die verschiedenen Ausprägungen der einzelnen Instanzen angesehen werden. Sie beschreibt die gemeinsamen Eigenschaften der Objekte aus dieser Klasse.

Die *Sichtweise* eines Objekts sollte ebenso begrenzt werden wie die Sicht auf ein Objekt. Dadurch werden die Manipulationsmöglichkeiten gegenüber den Objekten beziehungsweise für die Objekte eingeschränkt. Als Vorteil resultiert daraus eine kontrollierbare und transparente Softwarestruktur.

Durch die Trennung der *Spezifikation* von der *Implementierung* wird das Geheimnisprinzip (Information hiding) verfolgt. Jedes Objekt besteht aus diesen Teilen und eröffnet unterschiedliche Beschreibungsweisen. Der Implementierungsteil enthält die einzelnen Realisierungen, die für den Systementwurf nicht relevant sind. Hierfür reicht es aus, den Spezifikationsteil zu kennen. In ihm werden die Schnittstellen dargestellt. Das WAS (Spezifikation) ist vom WIE (Implementierung) separiert.

5.1.2 Erweiterte Entwurfsmöglichkeiten

Nach Jalote /69/ können gerade bei komplexen Systeme erweiterte Möglichkeiten für den objektorientierten Entwurf eingesetzt werden. Der Entwurfsprozeß wird in drei Phasen eingeteilt, die sequentiell zu durchlaufen sind:

- Grundentwurf
- Funktionale Verfeinerung
- Objektverfeinerung

In der ersten Phase soll mit einer informellen Beschreibung der Transformationsfunktion auf hohem Abstraktionsniveau begonnen werden. Diese Funktion kennzeichnet die Aufgabe des Systems. In Folge sind die Objekte inklusive ihrer Attribute festzulegen. Dazu werden die Methoden zugeordnet. Als Ergebnis bleibt eine Menge von Objekten und Methoden übrig, die in den weiteren Phasen feiner zerlegt werden müssen.

Die Ergebnisse der vorangegangenen Phase werden in der funktionale Verfeinerung verarbeitet. Sie stellt einen iterativen Prozeß dar. Für zu verfeinernde Methoden werden Lösungen skizziert und daraus wiederum Objekte und Methoden abgeleitet. Diese sind solange weiter zu verfeinern bis keine Methoden für eine Verfeinerung mehr auftreten. Das Ergebnis bilden alle Objekte, die zur Spezifikation notwendig sind. Die Methoden müssen jedoch noch nicht unbedingt vollständig sein.

In der letzten Phase sollen Objekte und Methoden gefunden werden, die von den bereits ermittelten Objekten zur Realisierung benötigt werden, da sich die Objekte durchaus noch auf hohem Abstraktionsniveau befinden können. Neu hinzukommende Objekte dienen nur der Implementierung der Methoden bereits existierender Objekte und sind für das globale System nicht relevant. Die Verfeinerung kann stufenweise erfolgen und wird solange fortgesetzt, bis die Methoden direkt mit den vorhandenen Mitteln umgesetzt werden können. Der Aufbau von Bibliotheken kann dabei wertvolle Unterstützung bieten.

Eine Transformationsfunktion innerhalb der Werkstattsteuerung kann zum Beispiel eine Planungsfunktion für die Maschinenbelegung sein. Die Objekte sind dann in den Geräten, Werkzeugen, Hilfsmitteln und Materialien zu sehen, welche es zu disponieren gilt. Sie bilden die elementaren Objekte für die Spezifikationsphase. Zur Realisierung kann es dann noch hilfreich sein, weitere Objekte abzuleiten, die für die Schnittstelle nach außen nicht sichtbar sind.

5.1.3 Abstrakte Datentypen

Unter Abstraktion ist prinzipiell eine Verallgemeinerung eines Sachverhalts zu verstehen. Das dabei verfolgte Ziel besteht darin, sich von vernachlässigbaren Details zu trennen, um sich auf das Wesentliche zu konzentrieren [70]. Der Blick auf komplexe Zusammenhänge reduziert sich auf die wichtigen Teile. Unwesentliches kann verborgen bleiben und spielt für das globale Verständnis keine Rolle.

Dieses Prinzip findet in sehr vielen Bereichen des Lebens Eingang. Auch bei der Entwicklung von Software kann es in den verschiedenen Phasen des Entwurfs wertvolle Unterstützung bieten. In diesem Zusammenhang taucht der Begriff des abstrakten Datentyps (ADT) auf, der durch folgende Definition in enger Anlehnung an /71/ beschrieben ist.

*Ein **abstrakter Datentyp** kennzeichnet eine Klasse von Datenobjekten. Die Klasse wird ausschließlich und vollständig charakterisiert durch das äußere Verhalten der Objekte. Dieses Verhalten wird mit einer Menge von Methoden (Aktionen) abgebildet, die für Instanzen dieser Klasse erlaubt und erklärt sind. Die Methoden bieten die einzige Möglichkeit, auf die Realisierung des Typs zuzugreifen. Dem Benutzer des Datentyps bleibt verborgen, wie das Objekt repräsentiert ist und wie die Methoden intern wirken.*

Als wichtigste Eigenschaft eines abstrakten Datentyps gilt die Trennung der Spezifikation von der Implementierung. Hinzu kommen noch die Vererbung und die generische Parametrisierbarkeit, die konkrete Instanzen aus den vorhandenen Klassen erzeugt, wodurch der Bezug zu einer realen Anwendung hergestellt wird.

Der größte Nutzen dieser Datentypen liegt im Strukturierungsprinzip. So ist der abstrakte Datentyp kein festgelegter Datentyp im herkömmlichen Sinne (z.B. Integer, Real), sondern kann ganz individuell für das jeweilige Anwendungsgebiet definiert werden. Durch das Dualitätsprinzip von Attributen und darauf arbeitenden Methoden stellt er eine **Erweiterung** der Elementartypen üblicher Programmiersprachen im Sinne der Objektorientierung dar /72/ (Bild 32).

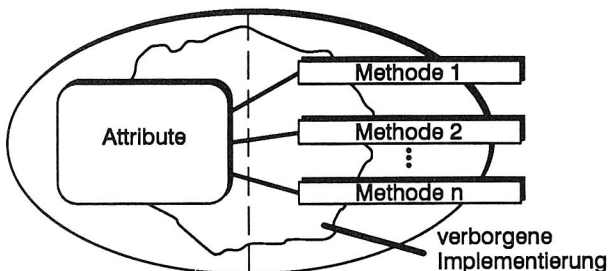


Bild 32: Abstrakter Datentyp

Der Vorteil einer objektorientierten Strukturierung gegenüber konventionellen Strukturen ist dabei offensichtlich. Anstatt Funktionen wie

```
fahren (roboternummer, quelle, ziel)
greifen (roboternummer)
rotieren (roboternummer, winkel)
```

zu realisieren, die zum Beispiel für alle im System vorhandenen Roboter verwendet werden, besitzt jeder Roboter eine eigene Methode in der Art

```
roboter_x (aktion, ...)
```

wobei sich hinter der *aktion* die auf den Roboter anzuwendende Operation (fahren, greifen, rotieren) verbirgt. Spezielle Details, wie zum Beispiel welche Fahrweise ein Roboter besitzt (kartesisch oder skalar), werden in der Implementierung verborgen. Dies müßte jedoch bei der konventionellen Darstellung noch explizit als Parameter mitaufgenommen werden:

```
fahren (roboternummer, robotertyp, quelle, ziel)
```

Für etwaige Änderungen am *robotertyp* müßten dann alle Stellen im Programmsystem korrigiert werden, an denen diese Funktion auftritt. Im objektorientierten Fall beschränken sich diese Änderungen auf ein Modul (*roboter_x*), vorausgesetzt die Schnittstelle bleibt gleich.

Für einen objektorientierten Entwurf kann der abstrakte Datentyp somit als geeignetes Strukturierungsmittel eingesetzt werden. Gerade bei hoher Komplexität kann durch eine Zerlegung des Gesamtsystems in kleine überschaubare Einheiten die Abstraktion in den Entwurfsphasen eingesetzt werden (Bild 33).

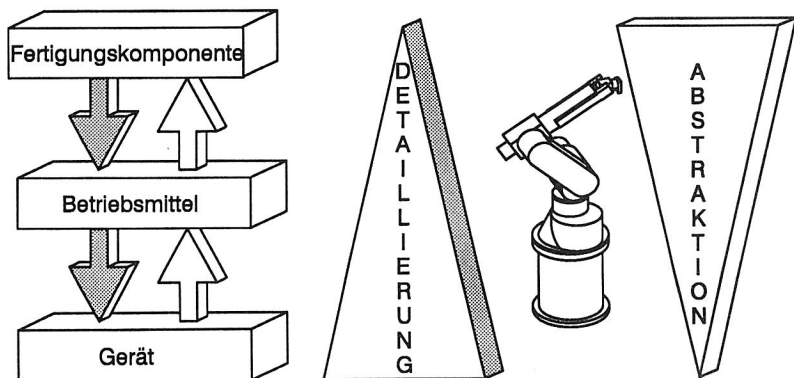


Bild 33: Abstraktion durch objektorientierten Entwurf

5.2 Klassifikation von Fertigungskomponenten

Unter dem zentralen Blickwinkel der Objektorientierung erscheint also die "Welt" als eine Menge von Objekten. Diese Objekte besitzen unterschiedliche Charakteristika, wodurch sie sich voneinander abgrenzen. Auch in der Projektion auf den vorliegenden Untersuchungsraum, nämlich einem konkreten Ausschnitt aus der Fertigung in Form eines Zentrums, einer Zelle oder eines Systems, ändert sich an diesem Sachverhalt nichts. Die Geräte, Werkzeuge, Hilfsmittel und das Material sind physikalische Objekte (Fertigungskomponenten), welche es datentechnisch entsprechend zu repräsentieren gilt (Bild 34).

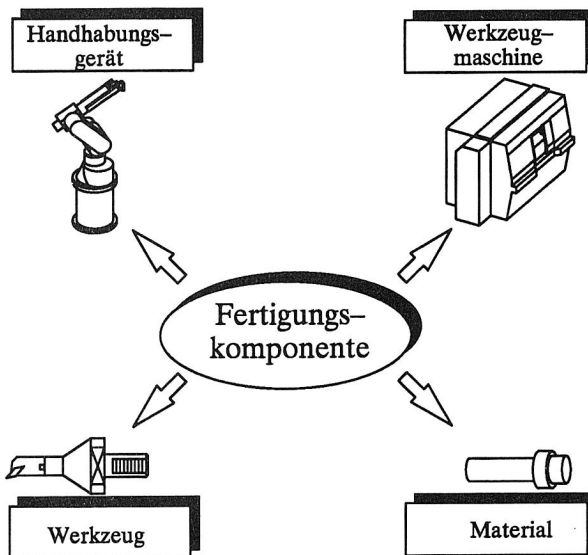


Bild 34: Objekte aus der Fertigung

Zur **Klassifikation** der Fertigungskomponenten können verschiedene Gesichtspunkte in Erwägung gezogen werden. Möglich ist eine Unterteilung in mobil bzw. immobil, aktiv bzw. passiv oder ähnlichem. Für die vorliegende Aufgabenstellung erscheint jedoch eine **funktionale** Aufteilung sinnvoll.

In einem ersten Schritt wird daher zwischen Material und Betriebsmitteln differenziert. Material wird in das System eingebracht und verläßt es nach der Bearbeitung wieder. Betriebsmittel bleiben bei vorhandener Funktionstüchtigkeit im System.

Eine weitere Aufspaltung unterteilt die Betriebsmittel in diejenigen, die in der Lage sind gesteuert zu werden, also über eine physikalische Leitung Signale oder Daten empfangen, und den übrigen Betriebsmitteln.

Erstere werden unter dem Sammelbegriff Geräte subsumiert. Sie können von einer übergeordneten Instanz, zum Beispiel einem Zellen- oder Leitrechner, mittels ihrer Steuerung Befehle empfangen und diese ausführen. Sie gelten als aktive Betriebsmittel. Die übrigen Betriebsmittel sind passiv einzustufen und werden unter dem Namen Fertigungshilfsmittel geführt. Folgende Aufstellung und Bild 35 verdeutlichen diesen Zusammenhang nochmals.

Material

"Sammelbegriff für Rohstoffe, Werkstoffe, Halbzeuge, Hilfsstoffe, Betriebsstoffe, Teile und Gruppen, die zur Fertigung eines Erzeugnisses erforderlich sind" /73/.

Betriebsmittel

"Anlagen, Geräte und Einrichtungen, die zur betrieblichen Leistungserstellung dienen"/73/.

Geräte

Einrichtungen mit steuerbarer Einheit. Sie dienen meistens zur direkten oder indirekten Form- beziehungsweise Fertigungszustandsänderung des Materials. Doch auch datenverarbeitende Anlagen, wie zum Beispiel Zellenrechner, können hierunter eingeordnet werden.

Fertigungshilfsmittel (FHM)

Gegenstände, die unmittelbar oder mittelbar zur Fertigung benötigt werden.

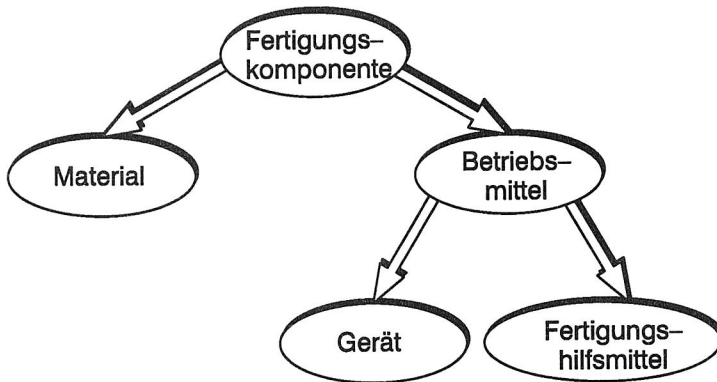


Bild 35: Klassifikation von Systemkomponenten

Ohne Anspruch auf Vollständigkeit seien einige Beispiele für die jeweiligen Klassen genannt:

Geräte: Werkzeugmaschinen, Montageeinrichtungen, Roboter, Handhabungseinrichtungen, Transportsysteme, Sensoren, ...

Fertigungshilfsmittel: Werkzeuge, Greifer, Paletten, Vorrichtungen, Magazine, Puffer, ...

Material: Werkstücke, Baugruppen, ...

5.3 Aufbau einer Wissensbasis

5.3.1 Anforderungen zur funktionalen Werkstattsteuerung

Das Anforderungsprofil der Werkstattsteuerung resultiert aus ihrer Lage im Gesamtsystem /74/. Zwischen der PPS und der Fertigung zeigt sie sich vor allem im Bereich der Planung, Steuerung und Überwachung für kurzfristige Aufgaben verantwortlich (Kapitel NO TAG). Da dieses Aufgabenspektrum in der Literatur nicht einheitlich gesehen wird und der Werkstattsteuerung in vielen Fällen auch unterschiedliche Bedeutung zukommt, werden die Inhalte, so wie sie im Rahmen dieser Arbeit verstanden werden, zum besseren Verständnis kurz dargelegt.

Planung

Die Vorgaben, die aus der Produktionsplanung und –steuerung oder einer übergeordneten Instanz kommen, erweisen sich meist als sehr grob terminiert. Die Planungsergebnisse können teilweise nicht direkt umgesetzt werden, so daß es einer Feinplanung bedarf. Hierbei gilt es einen Auftragsvorrat entsprechend seinen Randbedingungen, wie Termin, Menge, etc., so für die Fertigung vorzubereiten, daß eine Zuteilung der Arbeitsabschnitte auf die einzelnen Arbeitsplätze möglich ist. Im einzelnen ergeben sich folgende Aufgabenpunkte:

- Auftragsverwaltung
- Verfügbarkeitsprüfung
- Kapazitätsfeinplanung
- Reihenfolge- und Maschinenbelegungsplanung

Aus einem Pool zu verwaltender Aufträge wird nach der Verfügbarkeitsprüfung, die sowohl die Geräte als auch die Fertigungshilfsmittel und das Material berücksichtigt, eine Feinplanung für einen kurzfristigen Zeitraum durchgeführt. Dabei werden die Operationen (Arbeitsabschnitte), die einen oder mehrere Arbeitsgänge umfassen können, so auf die Maschinen verteilt, daß bestimmte Optimierungskriterien wie zum Beispiel hohe Auslastung oder minimale Durchlaufzeiten möglichst gut erreicht werden. Als Ergebnis der Planung wird der Maschinenbelegungsplan ausgegeben.

Steuerung

Die Ausgabe der Planung dient der Steuerung als Eingabe. Im Maschinenbelegungsplan ist festgelegt, wann und wo eine Operation durchgeführt wird. Nicht bestimmt ist, wie das Material und auch die Fertigungshilfsmittel (Werkzeuge, Spannvorrichtungen, u.a.) zu den jeweiligen Fertigungsstätten gelangen. Durch eine Iststandsaufnahme wird daher festgestellt, welche Objekte sich noch nicht am richtigen Ort befinden. Die notwendigen Transport- und Handhabungsoperationen werden initiiert und daraus resultierende Zeitverschiebungen im Belegungsplan können korrigiert werden.

Nach der Steuerungsphase liegt ein kompletter Anlagenfahrplan für das System vor, der sowohl die primären als auch die sekundären Operationen enthält. Unter den primären Operationen sind die Operationen zu verstehen, die direkt zur Wertschöpfung des Erzeugnisses beitragen, wie zum Beispiel drehen, bohren, montieren u.a. Die sekundären Operationen umfassen im wesentlichen die Transport- und Bereitstellungsoperationen. Das Aufgabenfeld der Steuerung hat demnach folgendes Aussehen:

- Bereitstellung von Fertigungshilfsmitteln und Material
- Erweiterung des Maschinenbelegungsplans

Überwachung

Der Anlagenfahrplan aus der Steuerungsphase kann nun umgesetzt werden. Unter der Voraussetzung, daß die ablauffähigen Programme, welche die Durchführung der jeweiligen Operationen gewährleisten, bei Bedarf in die entsprechenden Steuerungen (DNC, RC, SPS) geladen und gestartet werden können, bewirkt die Überwachung die Durchsetzung des Plans. Während des Fertigungsprozesses werden dazu Maschinendaten aufgenommen. Zur Regelung findet ein Soll-Ist-Abgleich auf zeitlicher Basis statt. Sind Abweichungen festzustellen, die aufgrund irgendwelcher Störungen auftreten, muß entschieden werden, welche Reaktionen zu treffen sind. Je nach Größe des Toleranzspielraums und der auftretenden Störung kann versucht werden, durch Anpassung an die gegebene Situation die Störung zu kompensieren oder eine neue Planung anzustoßen. Diese Stufenreaktion ermöglicht ein feinfühliges Agieren auf unterschiedlichste Störungen. Resultierend ergibt sich für die Überwachung:

- Durchsetzung des Anlagenfahrplans
- Fertigungsfortschrittsüberwachung
- Regelung des Fertigungsprozesses

Ziel ist somit die Abbildung der wesentlichen Funktionalitäten eines Leitsystems zur Werkstattsteuerung (Planung, Steuerung und Überwachung), welche in einem **selbstregelnden System** bereitgestellt werden, um die Schwachstellen herkömmlicher Werkstattsteuerungen aufzuheben (Bild 36).

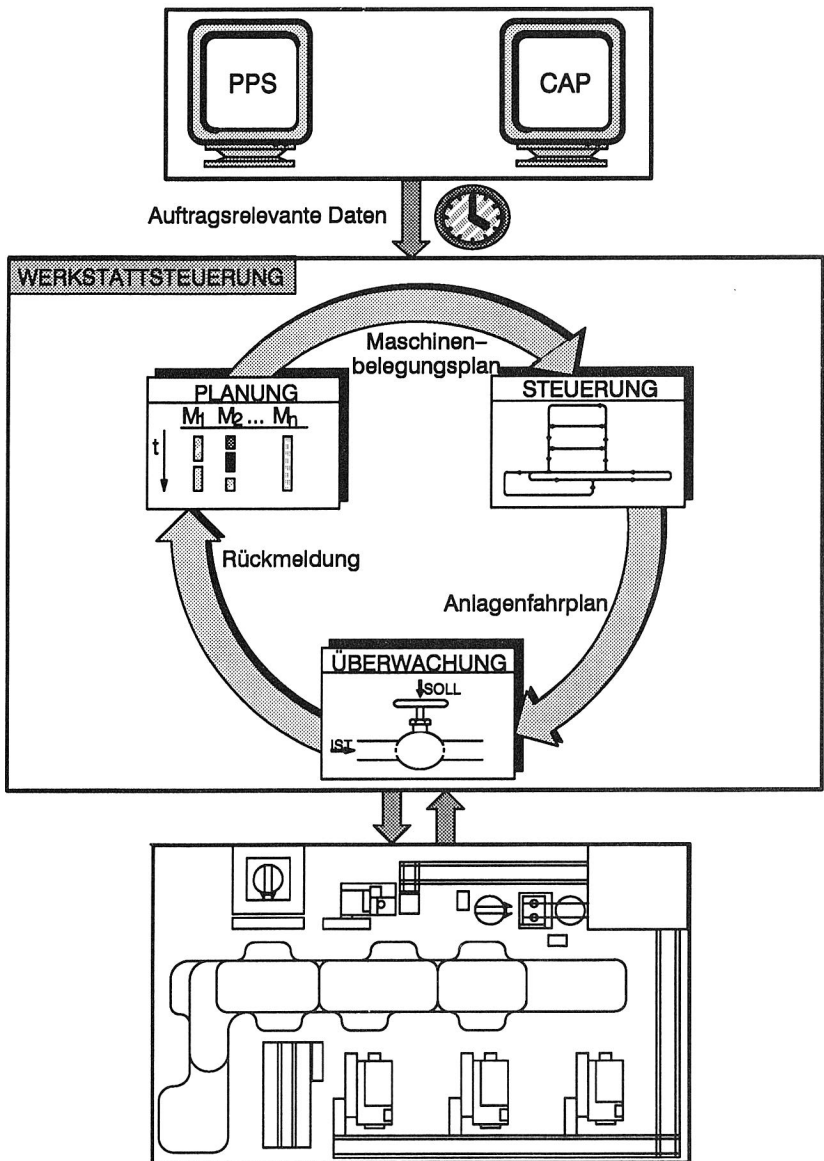


Bild 36: Regelkreis im Werkstattsteuerungssystem

5.3.2 Semantisches Netz zur Wissensrepräsentation

Das Informationssystem zur Werkstattsteuerung wird als Wissensbasis realisiert, die die zur Lösung der Aufgabenstellung benötigten Elemente zur Verfügung stellt. Wissenbasierte Systeme enthalten neben der Wissensbasis eine Inferenzkomponente /75/. Die Basis verfügt dabei unter anderem über Objektschemata, die konkreten Ausprägungen entstehender Situationen und die Möglichkeit der Speicherung von Zwischenergebnissen. Zur Verarbeitung des Wissens dient die Inferenzkomponente, die problemspezifisch arbeitet (Bild 37).

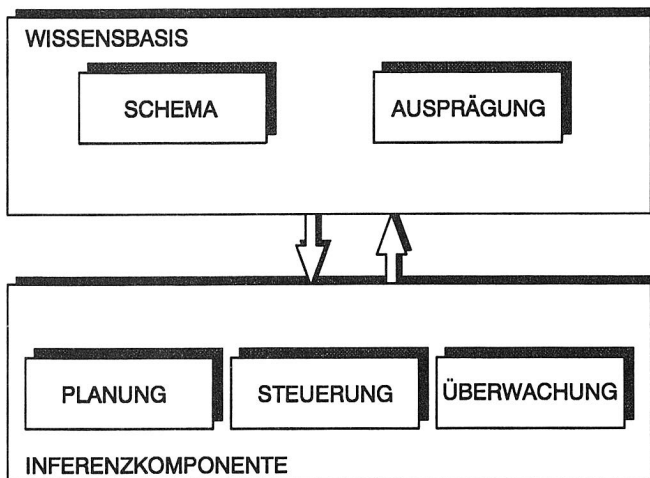


Bild 37: Struktur der Wissensbasis zur Werkstattsteuerung

Der Aufbau, die Struktur und der Inhalt der Wissensbasis zeigen sich im wesentlichen für die Güte eines informationstechnischen Systems verantwortlich. Der Form der Repräsentation kommt daher eine sehr gewichtige Rolle zu. Die Ausführungen in Kapitel 3 haben gezeigt, daß semantische/assoziative Netze ein hervorragendes Mittel zur Wissensrepräsentation sind. Deshalb bieten die Netze auf objektorientierter Basis mittels der abstrakten Datentypen ein ausgezeichnetes Feld zur Definition dieser Wissensbasis.

Die Beziehungen, die in den meisten semantischen Netzen auftauchen, sind:

- Generalisierung (IS_A) —————→
- Individualisierung (INSTANCE_OF) -----→
- Aggregierung (PART_OF) - - - - -→

Auf Grundlage der **komponentenorientierten Klassifikation / Modellierung** aus Kapitel 5.2 kann nun ein semantisches Netz generiert werden. Die einzelnen Objekte werden durch Knoten repräsentiert und ihre Beziehung zueinander in den Kanten widerspiegelt. Die Generalisierung erlaubt eine Abstraktion, die Individualisierung entspricht der Instanzierung. Für die Funktionalität der Planung, Steuerung und Überwachung ist dabei eine Abstraktion über drei Hierarchieebenen (Fertigungskomponente – Betriebsmittel – Gerät/Fertigungshilfsmittel) ausreichend. Eine tiefergehende Abstraktion (wie zum Beispiel Gerät – Werkzeugmaschine – Fräsmaschine) ist durchaus erlaubt, jedoch für die vorliegende Aufgabenstellung nicht unbedingt erforderlich. Ein kleines Beispiel zeigt Bild 38.

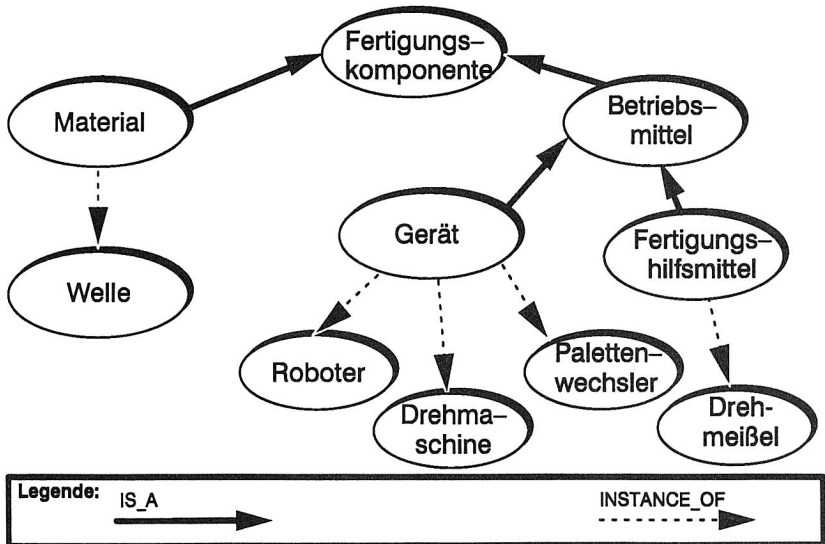


Bild 38: Fertigungskomponenten im semantischen Netz

Zur Planung wäre diese Grundstruktur im Netz bereits ausreichend. Jedoch ist die Steuerung zur Generierung der Fahrpläne darauf angewiesen zu wissen, wie eine sekundäre Operation (Transport- oder Handhabungsoperation) zu organisieren ist. Das läßt eine Erweiterung des Netzes um einen zusätzlichen Objekttyp sinnvoll erscheinen.

Sekundäre Operationen erfolgen in der Regel von einem Ort (Quelle) zu einem anderen Ort (Ziel). Unabhängig von der Semantik kann allgemein von **Ortseinheiten bzw. Orten** gesprochen werden. Sie spiegeln die physikalischen Gegebenheiten aus der Fertigung wider. Für eine Fertigungseinrichtung ist demnach eine logische Zerlegung in relevante Ortseinheiten erforderlich. Relevant zeigen sich die Ortseinheiten, die die Steuerung zur Organisation des Fertigungshilfsmittel- und Materialflusses benötigt. Eine Werkzeugmaschine zum Beispiel kann als ein einzelner Ort angesehen oder weiter detailliert werden – ein Ort an dem das Werkstück eingespannt wird und einer für das Werkzeugsystem. Der Fahrkurs eines fahrerlosen Transportsystems kann auch in Orte eingeteilt werden. Er setzt sich aus den einzelnen Haltebahnhöfen zusammen, woraus sich unter Berücksichtigung der Weichen die zur Fahrzeugführung notwendigen Ortseinheiten ableiten lassen.

Jedes System besitzt somit eine ganz individuelle Ortsstruktur, deren Detaillierungsgrad beim Entwurf der Wissensbasis festgelegt wird und der Steuerung als Grundlage ihrer Berechnungen dient. Je höher dabei der Detaillierungsgrad liegt, desto besser können die Flexibilitätspotentiale genutzt werden.

Datentechnisch stellt ein Ort ein eigenständiges Objekt dar. Seine Zugehörigkeit zu anderen Objekten wie zum Beispiel zu Geräten kann mittels der Aggregation (Part_of-Beziehung) ausgedrückt werden.

Mit dieser **Teil-/Ganze-Beziehung** steht eine weitere wichtige Relation zur objektorientierten Strukturierung zur Verfügung. Neben der **Generalisierung**, die sich für die Vererbung verantwortlich zeigt, unterstützt sie eine detaillierte Betrachtungsweise komplexer Objekte.

Im Beispielszenarium aus Bild 38 liegt eine Drehzelle bestehend aus Drehmaschine, Handhabungsroboter und Palettenwechselsystem vor. Der Einfachheit halber wird nur ein Werkstück (zu drehende Welle) und ein Werkzeug (Drehmeißel) betrachtet. Wellen, die ein fahrloses Transportsystem (FTS) anliefert, können auf dem

Palettenwechsler zwischengepuffert werden. Ein Portalroboter bringt sie anschließend in die Drehmaschine ein. Die Drehmaschine wird in Bild 39 logisch in die beiden Orte "Spannfutter" und "Werkzeugrevolver" zerlegt. Eine weitere kapazitätsorientierte Unterteilung bietet sich auch für das vorgelagerte Palettenwechselsystem an.

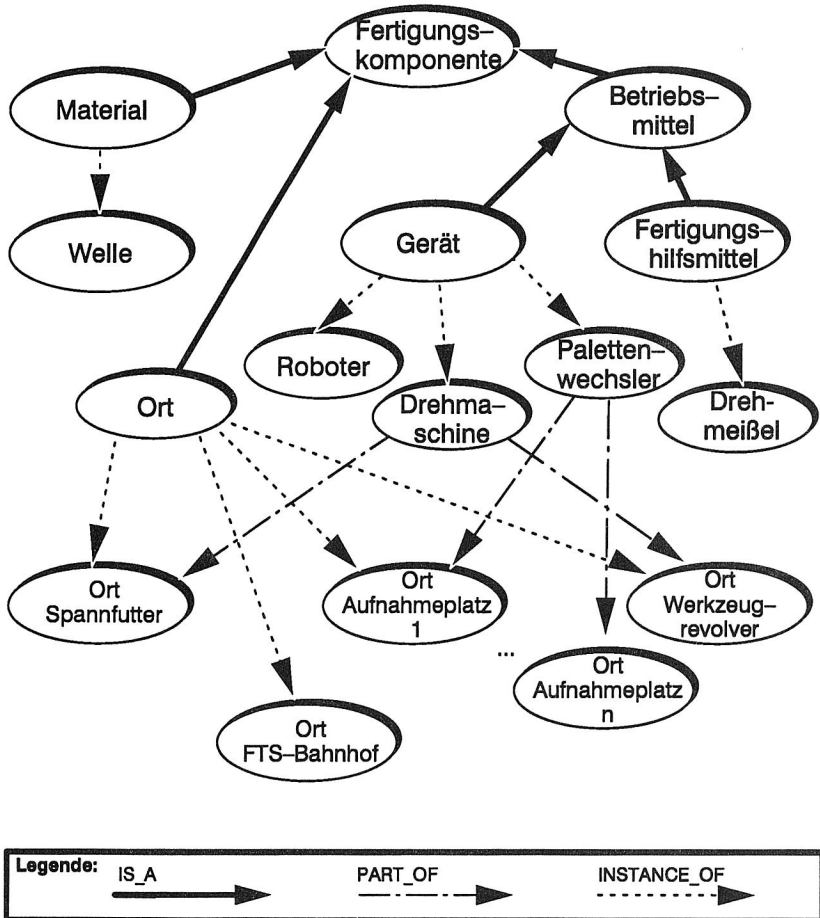


Bild 39: Semantisches Netz für eine beispielhafte Fertigungszelle

Mit den vorhandenen Mitteln gelingt es, Fertigungseinrichtungen jeglicher Art unabhängig von der Organisationsform adäquat in der Wissensbasis statisch abzubilden. Die Systemkomponenten sind bezüglich ihrer Beziehungen durch die Generalisierung und Individualisierung charakterisiert, während die Konfiguration des Systems und Ablauflogistik mittels der Orte beschrieben wird.

Eine Fertigungseinrichtung unterliegt jedoch einer gewissen Dynamik, die auch Berücksichtigung finden muß. Fertigungshilfsmittel und Material werden im Verlauf des Fertigungsprozesses durch das System geschleust. Auch Geräte müssen nicht unbedingt ortsinvariant sein, wie der Einsatz eines mobilen Roboters zur automatischen Werkstückversorgung am Institut für Werkzeugmaschinen und Betriebswissenschaften der Technischen Universität München zeigt /76/,/77/. Die Einführung des Konzepts der Orte trägt auch dieser Forderung Rechnung.

So wird mit der **Plazierung** (AT) eine zusätzliche semantische Beziehung definiert. Von den mobilen Objekten gibt diese Relation Auskunft darüber, an welchem Ort sich dieses befindet. Wird eine Ortsänderung vollzogen, wird auch die Relation aktualisiert, so daß zu jedem Zeitpunkt festgestellt werden kann, wo die Objekte im System platziert sind. Das wiederum ermöglicht der Steuerung, auf der Basis der Maschinenbelegungsplanung einen Soll-Ist-Abgleich der Aufenthaltsorte von Objekten durchzuführen, um einen Bedarf an sekundären Operationen für den Anlagenfahrplan abzuleiten.

Nach Lorenzen /78/ existieren zwar 216 Lokalpräpositionen – sie entstehen aus den Grundpräpositionen ("unten", "oben"), ("von", "zu"), ("innen", "außen"), ("links", "rechts") und ("hinten", "vorne") durch Kombination –, für die dispositiven Aufgaben der Werkstattsteuerung jedoch reicht die Verwendung der undifferenzierten Präposition "an der Stelle" völlig aus (AT-Relation). Eine Behandlung der exakten Positionen einzelner Komponenten (Koordinaten und Lage) erfolgt erst in der operativen Ebene durch die Maschinensteuerungen.

Unterschiedliche Möglichkeiten der Routenwahl können durch die Relation der **Orientierung** (TO) im semantischen Netz integriert werden. Sie bewirkt eine Veränderung der Lokalpräposition AT. Die Relation gibt außerdem Auskunft darüber, zwischen welchen Orten bestimmte Objekte in welcher Menge transferiert werden können, welche Objekte dazu notwendig sind und wie lange dies dauert. Ihre Semantik bleibt im Netz verborgen (Bild 40).



Bild 40: Semantik der TO-Relation

Durch die Einführung der Orte als eigenständige Objekte¹⁾ ist es möglich, ein Netz, das nur von den Orten und der Orientierungsrelation aufgespannt wird, als Teilnetz der gesamten Wissensbasis herauszulösen. Es gibt bei korrekter Konfiguration alle in einem System vorhandenen Routen wieder, die zum Transport von Objekten möglich sind und ist Voraussetzung für die Steuerungskomponente (Kapitel 6.4).

Bild 41 zeigt eine Momentaufnahme aus obigem Beispiel. Die Welle ist im Spannfutter der Drehmaschine eingespannt, der Drehmeißel steckt im Werkzeugrevolver (AT-Relationen). Wird weiteres Material zu Bearbeitung angeliefert, so können die Werkstücke über die Aufnahmeplätze des Palettenwechslers in die Maschine eingebracht und bereits bearbeitete Wellen entsorgt werden (TO-Relationen).

5.3.3 Objekte als abstrakte Datentypen

Bei einem Systementwurf muß sorgfältig überlegt werden, welche Objekte überhaupt notwendig sind. Außerdem ist es sinnvoll, das System durch eine richtige Zuordnung der Objekte zur entsprechenden Hierachiestufe erweiterbar zu gestalten, wie zum Beispiel beim Ausbau eines Prototyps zu einem Komplettsystem durch Modifikation oder Hinzufügen neuer Objekte.

Für die Objekte selbst sind neben den Methoden seine Attribute wesentliche Merkmale. Welche Attribute ein Objekt besitzt, hängt vom Verwendungszweck ab. So können Attribute für unterschiedliche Applikationen (Planung, Steuerung und Überwachung) gleichermaßen relevant sein. Auch bei der Methodenauswahl zur Manipulation der Attributwerte (Daten) muß auf eine hinreichende Definition derselben geachtet werden.

¹⁾ Eigenprädikative Lösung im Gegensatz zur apprädikativen Lösung

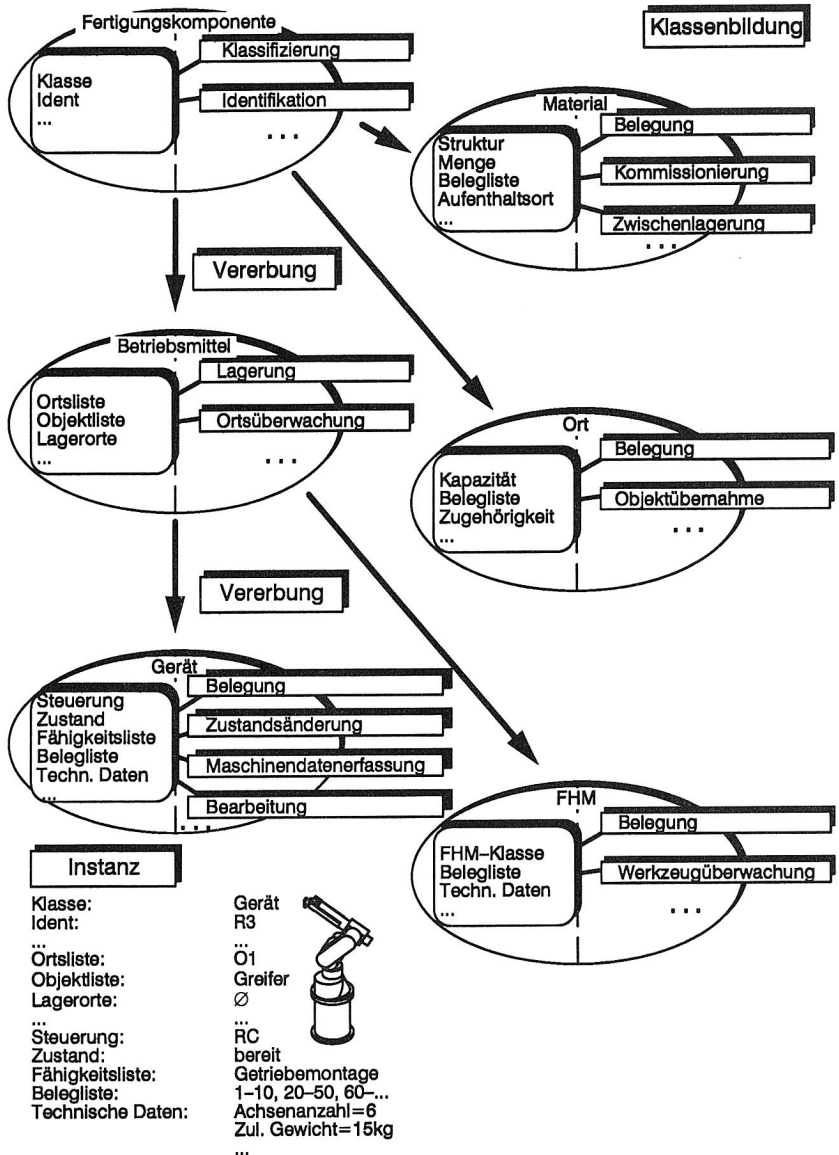


Bild 42: Abstrakte Datentypen zur Werkstattsteuerung

5.3.4 Realisierungsaspekte

Rapid Prototyping

Die Erstellung von Prototypen leistet gerade in technischen Forschungsgebieten beim Entwurf komplexer Systeme hilfreiche Unterstützung. Zu Versuchszwecken eingesetzt, kann damit eine rasche Überprüfung der Funktionalität erzielt werden, wenn das Ausmaß des Gesamtsystems so umfangreich ist, daß die prinzipielle Wirkungsweise mittels eines Prototypen getestet werden muß. Der Prototyp selbst stellt dabei ein erstes funktionsfähiges Produkt dar. Verlaufen die Tests zufriedenstellend, so kann der Prototyp zur Marktreife gebracht werden /82/.

Beim Rapid Prototyping wird versucht, möglichst schnell zu einem Prototypen zu gelangen. Dies kann gelingen, wenn die informelle Problembeschreibung direkt in ein Labormodell umgesetzt werden kann. Die Klassenbildung leistet dabei wertvolle Dienste. Durch vordefinierte Klassen können mittels Zusammensetzung neue Klassen abgeleitet werden. Soll für diese Wissensbasis zum Beispiel eine Klasse für Spannvorrichtungen kreiert werden, so kann sie direkt aus der Klasse der Betriebsmittel entstehen. Die Attribute werden automatisch vererbt und neue können problemlos angefügt werden. Es braucht kein kompletter Datentyp neu definiert werden.

Für die prototypische Realisierung der vorliegende Aufgabenstellung standen neben einem Rechner mit einem Intel 80386 Prozessor, 25MHz Taktfrequenz, 80387 Koprozessor und 8 MByte Hauptspeicher als Hardwarebasis noch folgende Softwaretools zur Verfügung:

UNIX

Als Betriebssystem kam Unix™ System V Release 3.2 zum Einsatz. Dieses System liegt eingebettet im Open Desktop™ von SCOT™, mit OSF/Motif™, Ingres 386™ und anderen Softwarefunktionalitäten.

Die multi-tasking-Fähigkeit und die enorme Verbreitung gerade im Hochschulbereich waren wichtige Voraussetzungen bei der Auswahl dieses Betriebssystems /83/, /84/. Der entscheidende Vorteil liegt jedoch in der Unabhängigkeit von herstel-

lereligen Systemen und speziellen Rechnerarchitekturen. Zudem sind verstärkte Anstrengungen zur Standardisierung im Gange /85/. Obwohl dieses Betriebssystem vornehmlich zur Softwareentwicklung konzipiert wurde, wird es nach und nach auch im Bereich der Fertigungsautomatisierung eingesetzt.

C++

Zur Implementierung der Wissensbasis wurde die problemorientierte Programmiersprache C++ verwendet. Dabei handelt es sich um eine Obermenge der Programmiersprache C, die um objektorientierte Merkmale erweitert wurde. Dazu zählen insbesondere das Klassenkonzept mit den Vererbungsmöglichkeiten, die Mechanismen zur Überladung, virtuelle Funktionen, das dynamische Binden, u.a. /86/. Ihre Knappheit, Vielseitigkeit und Maschinennähe zeichnen diese Sprache besonders aus /87/.

5.3.5 Einbettung von Ablaufgraphen

In Kapitel 4 wurde der Ablaufgraph als innovative Komponente in der Arbeitsvorbereitung vorgestellt. Zur Einbettung der Ablaufgraphen in die Wissensbasis bietet das semantische Netz als Wissensrepräsentationsform auch hier durch ein Drei-Ebenen-Modell sehr gute Möglichkeiten.

Die erste Ebene ist in der Arbeitsvorbereitung anzusiedeln. Sie umfaßt die Arbeitspläne und Ablaufgraphen mit funktionaler Zuordnung zur zweiten Ebene. Diese bildet das semantische Netz zur Werkstattsteuerung. In der dritten Ebene befinden sich die einzelnen Systemkomponenten, also Geräte, Fertigungshilfsmittel, u.a. Diese Ebene ist die sogenannte operative Ebene und enthält die physikalischen Ausprägungen der datentechnischen Strukturen aus den darüberliegenden Schichten (Bild 43).

Auch im Sinne **durchgängiger Verfahrensketten** kann eine derartige Schichtung wertvolle Dienste leisten. Das Gesamtkonzept kann als integraler Bestandteil in unterschiedliche Anwendungen Eingang finden.

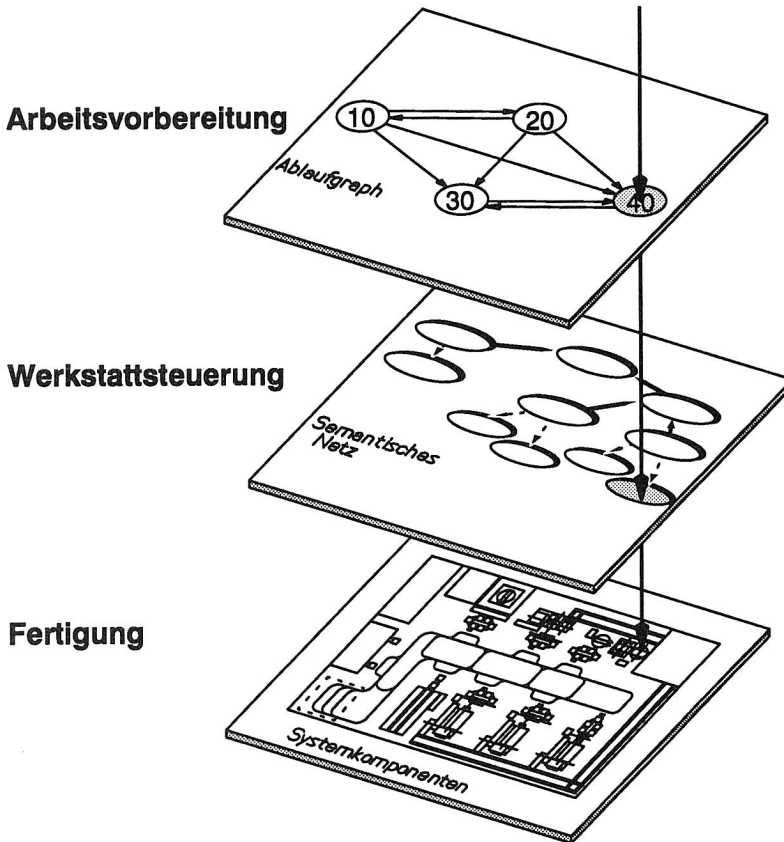


Bild 43: Drei-Ebenen-Modell

5.4 Ausbau zu einem ganzheitlichen Konzept

5.4.1 Vorteile durch die Objektorientierung

Unzählige Unternehmen stehen vor der Aufgabe, Programmaterial für komplexe Aufgabenstellungen implementieren zu müssen. In dieses Material werden erhebliche Investitionen getätigt, wodurch der Wunsch nach optimaler Nutzung der Software steigt. Ein Schritt in diese Richtung liegt in einer wiederverwendbaren Softwarekonzeption.

Es muß also Software geschrieben werden, die in unterschiedlichen Umgebungen ohne "größeren Anpassungsaufwand" einsetzbar ist. Ziel ist es nicht, große Softwaresysteme zu entwickeln, um diese dann durch unzählige Parameter und Stellschrauben dem jeweiligen Anwendungsgebiet anzupassen, sondern eher ein System aus kleinen, überschaubaren Einheiten zu konfigurieren, das dann von Anwendungsfall zu Anwendungsfall modelliert wird. Die Objektorientierung unterstützt mit ihren Möglichkeiten eine derartige Konfiguration.

Eine optimale Struktur ergibt sich, wenn die einzelnen Einheiten größtmöglichen inneren Zusammenhang und kleinstmögliche äußere Bindung besitzen. Das Geheimnisprinzip fordert in sich abgeschlossene Einheiten. Die Methoden stellen gleichzeitig die Schnittstelle zwischen den Softwarekomponenten dar, die diese zum Beispiel in Form von Funktionen aufrufen. Somit wird erreicht, daß eine Datenmanipulation nur über diese Funktionen möglich ist, was folgende Vorteile einbringt:

- kontrollierbarer Datenzugriff
- erleichterte Fehlersuche
- transparente Software

Da bei der Softwareentwicklung im allgemeinen eine projektbezogene Arbeitsweise eingeschlagen wird, ist es unabdingbar, daß dem gesamten Projektteam die Vorgehensweise bekannt ist und nach entsprechenden Gestaltungsgrundsätzen und Richtlinien entwickelt wird. Dem Programmierer müssen die Schnittstellen bekannt sein. Ferner sollte er auch nur diese Schnittstellen benutzen und sich in keinem Fall eigene definieren, die nicht den Richtlinien entsprechen. Nur durch konsequente Einhaltung der Richtlinien wird gewährleistet, daß das Gesamtkonzept auch funktioniert. Der damit verbundene Erfolg schlägt sich in der Kostenreduktion bei der Softwareerstellung und der Softwarepflege nieder.

5.4.2 Architekturprinzip eines Gesamtsystems

Funktionskomponenten, die auch zum Umfang einer Werkstattsteuerung zählen, sind neben der Planung, Steuerung und Überwachung zum Beispiel:

- NC/RC-Programmsystem
- Werkzeugmanagement

- Lagermanagement und Kommissionierung
- Datenerfassung, –auswertung und –verdichtung
- Instandhaltung u.a.

Bei konventionellen Softwarearchitekturen, wie sie in zahlreichen Softwareprojekten realisiert sind, wird die Funktionalität zwar in einzelne Module gepackt, jedoch bestehen zwischen ihnen große Abhängigkeiten. Um diese Korrelationen zu vermeiden, können im Sinne der Objektorientierung funktionale Einheiten in eigenständige Prozesse gepackt werden. Diese Prozesse agieren auf einer gemeinsamen Wissensbasis, die die Methoden zur Manipulation der Daten bereitstellt (Bild 44).

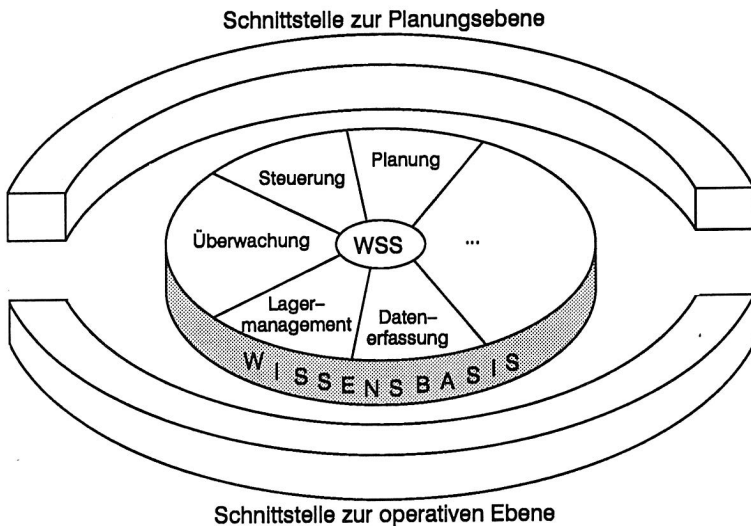


Bild 44: Objektorientiertes Architekturprinzip

Durch die Aufteilung der Funktionalität auf eigenständige und unabhängige Prozesse kann eine Verteilung dieser ermöglicht werden. Es ist nicht zwingend erforderlich, daß sich alle Prozesse zur Werkstattsteuerung auf demselben Rechner befinden. Vielmehr können sie zum Beispiel zur Performanzsteigerung auf verschiedene Rechensysteme verteilt werden.

Als übergeordnete Instanz kann die eigentliche Werkstattsteuerung auch als eigenständiger Prozeß realisiert werden und auf die anderen Prozesse zugreifen. Eine Organisation nach dem Client–Server–Modell wäre denkbar. Bei diesem Modell übergibt ein Auftraggeber (Client) einem Auftragnehmer (Server) einen Auftrag. Nach Beendigung des Auftrags wird das Ergebnis dem Auftraggeber mitgeteilt. In vorliegendem Fall wäre in der Werkstattsteuerung der Client zu sehen. Von ihr aus würden die einzelnen Einheiten (Planung, Steuerung, etc.), die als Server zu realisieren wären, angestoßen werden. Die Konfiguration würde auch ein paralleles Arbeiten der Server ermöglichen (Bild 45).

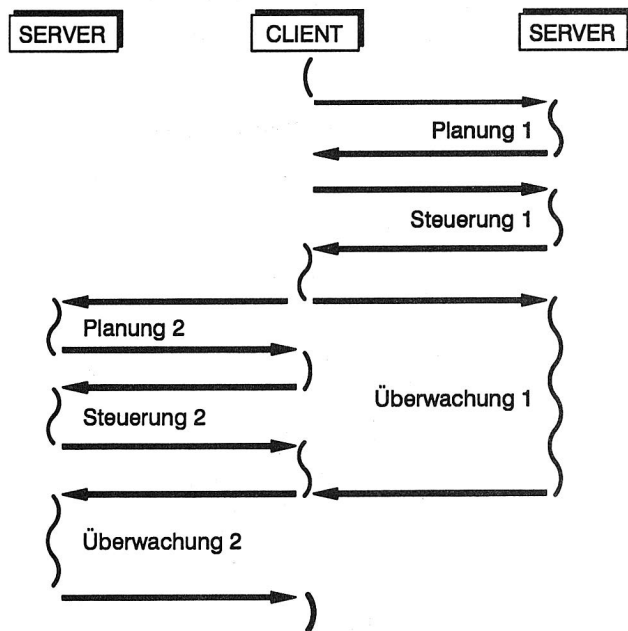


Bild 45: Werkstattsteuerung als Client–Server–Modell

Die Vorteile dieser Architektur sind:

- hohe Portabilität infolge der Zentralisierung von Schnittstellen
- Modularisierung des Systems und bessere Kombinierbarkeit
- offene Architektur

6 Selbstregelnde Werkstattsteuerung

Die Definition einer Wissensbasis allein reicht zur Lösung einer Aufgabenstellung nicht aus. Vielmehr ist es die Aufgabe der Inferenzkomponente, durch entsprechende Algorithmen das Wissen aus der Basis geschickt zu nutzen und so zu verarbeiten, daß die zugrundeliegende Problematik gelöst werden kann.

Es sollen Mechanismen zur Werkstattsteuerung vorgestellt werden, die auf der objektorientierten Wissensbasis aus Kapitel 5 fundierend in intelligente Strategien zur Erfüllung der Anforderungen münden. Ferner wird ein Lösungsvorschlag erarbeitet, der die bisherigen Schwachstellen in der Werkstattsteuerung behebt und ein Bindeglied zwischen der Planungs- und der operativen Ebene herstellt.

Besonders für die **Einzel- und Kleinserienfertigung** im Bereich der **kurzfristigen** Planung sind die bisher vorhandenen Lösungen nicht ausreichend. Die Existenz **konkreter** Planungsergebnisse ist von enormer Wichtigkeit, um die vorhandenen Ressourcen zeitlich genau einsetzen zu können (Bild 46).

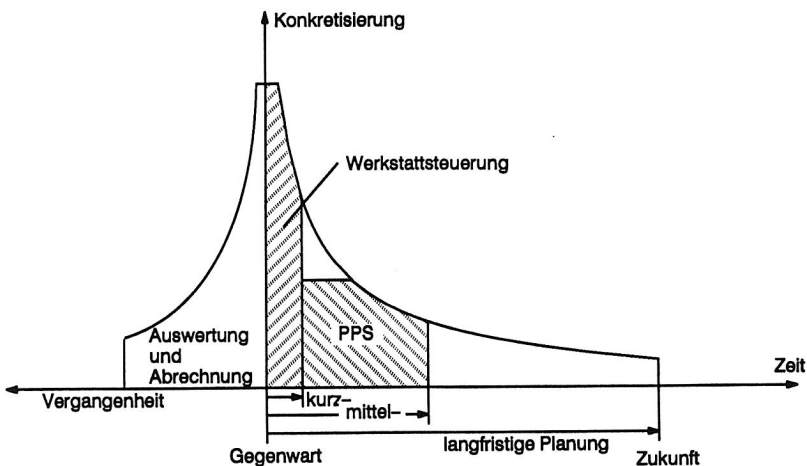


Bild 46: Konkrete Planungsergebnisse aus der Werkstattsteuerung /88/

Aufgrund der bekannten Mängel (Kapitel 2.1.3) lassen sich für die Konzeption eines Werkstattsteuersystems folgende Anforderungen ableiten (Bild 47):

- Adaption an unterschiedliche Systemumgebungen in der Fertigung
- Situationsbedingte Laufzeitregulierung (zeitkritisch – zeitunkritisch)
- Zieloptimierung und Prioritätenberücksichtigung
- Steuerung des Fertigungshilfsmittel- und Materialflusses
- Selbstregelung bei unvorhergesehenen Ereignissen mit von außen einstellbaren Größen

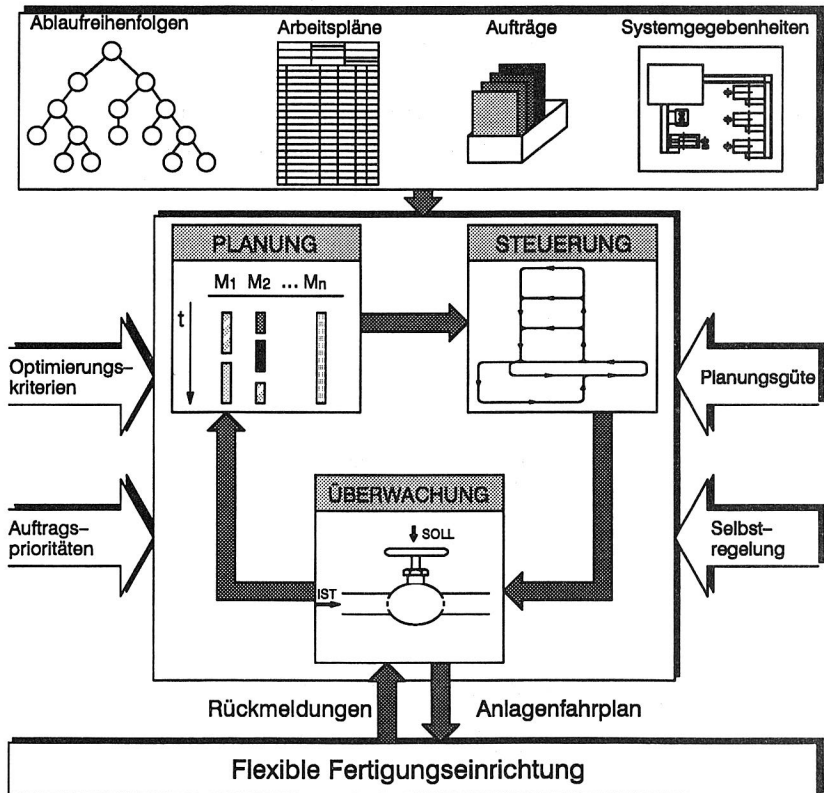


Bild 47: Konzeption der Werkstattsteuerung

6.1 Untersuchung mathematischer Methoden

Die Anwendung mathematischer Methoden zur optimalen Entscheidung wird hinlänglich als Operations Research (Optimalplanung) bezeichnet. Diese Planungsmethoden sind vornehmlich für die betriebliche Praxis von Bedeutung, so daß eine nähere Betrachtung erachtenswert erscheint. Es werden einige Methoden vorgestellt und bezüglich ihrer Eignung bewertet, sie innerhalb der Inferenzkomponenten einzusetzen.

6.1.1 Vollenumeration

Im ersten Ansatz kann davon ausgegangen werden, daß alle möglichen Lösungen für eine Problemstellung berechnet werden, um dann die beste zu ermitteln. Unmittelbar einsichtig ist, daß bei steigender Lösungszahl auch der Aufwand zur Berechnung (Enumeration) wächst. Daher kann nur bei kleinen, abgegrenzten Lösungsräumen eine Vollenumeration sinnvoll sein. Dies ist hauptsächlich dann der Fall, wenn eine Lösungsfindung für ein einmalig auftretendes Problem erforderlich ist. Hier stünde dann auch der Nutzen in Relation zum Aufwand.

Für den Fall der Maschinenbelegung kann deshalb die Vollenumeration nicht in Erwägung gezogen werden, da sich im dynamischen Verlauf des Fertigungsprozesses die Situationslage ständig ändert. Der Planungsaufwand für eine Belegung könnte im ungünstigsten Fall fakultativ wachsen. Dies ist gegeben, wenn keine Restriktionen vorliegen, die die Berechnung beschränken, also die Planung alle Varianten durchspielen muß. Bei zehn Aufträgen/Arbeitsgängen wären 10! (3628800) verschiedene Einplanungsvorgänge durchzuführen, ohne Geräte- oder Fertigungshilfsmittelalternativen zu berücksichtigen.

6.1.2 Lineare Optimierung

Bei der linearen Optimierung wird versucht, ein Minimum oder Maximum einer linearen Funktion mit endlich vielen Variablen zu ermitteln. Die Variablen müssen zudem endlich viele Nebenbedingungen erfüllen, die als Gleichungen oder Ungleichungen formuliert werden. Sie werden auch als Restriktionen bezeichnet.

Für eine Zielfunktion der Gestalt

$$Q(x) = p_1x_1 + p_2x_2 + p_3x_3 + \dots + p_nx_n$$

sind die Werte der n reellen Variablen $x_1, x_2, x_3, \dots, x_n$ so zu bestimmen, daß die Funktion ihren Extremwert erreicht, wobei alle Punkte zu betrachten sind, deren Koordinaten die Nebenbedingungen

$$a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + \dots + a_{1n}x_n = < b_1$$

$$a_{21}x_1 + a_{22}x_2 + a_{23}x_3 + \dots + a_{2n}x_n = < b_2$$

...

$$a_{m1}x_1 + a_{m2}x_2 + a_{m3}x_3 + \dots + a_{mn}x_n = < b_m$$

$$\text{mit } x_1 \geq 0, x_2 \geq 0, \dots, x_n \geq 0 \text{ (Nichtnegativitätsbedingungen)}$$

erfüllen. Die Koeffizienten können dabei reelle Zahlen sein /89/.

Das wohl bekannteste Lösungsverfahren in der linearen Optimierung stellt die sogenannte Simplex-Methode dar. Zur Lösung von ganzzahligen linearen Aufgaben sind ebenso verschiedene Verfahren entworfen worden wie zum Beispiel das Schnitverfahren oder das Verzweigungsverfahren.

Eine Zielfunktionen bei der Maschinenbelegung könnte

- Minimiere Durchlaufzeit,
- Maximiere Termintreue oder
- Minimiere Fertigungshilfsmittelbedarf

sein, wobei die Nebenbedingungen die Kapazitäten der jeweiligen Maschinen beschränken.

Die Lösungsverfahren der linearen Optimierung erweisen sich in Bezug auf vorliegende Problemstellung (Werkstattsteuerung) als nicht geeignet. Ausschlaggebend sind die zu langen Rechenzeiten aufgrund der NP-Vollständigkeit (nichtdeterministisch polynomial zeitleistvollständig) der Aufgabe. Die Rechenzeit wächst exponentiell mit dem Umfang des Problems. Sinnvolle praktische Anwendungen sind nur bei sehr begrenzten Aufgabenstellungen möglich.

6.1.3 Entscheidungsbaumverfahren

Die begrenzte Enumeration, die dynamische Planungsrechnung und das Branch-and-Bound-Verfahren zählen zu den bekanntesten Vertretern der Entscheidungsbaumverfahren. Auch sie sind in der Lage, das Optimum zu ermitteln.

Alle derartigen Verfahren arbeiten nach dem Enumerationsprinzip. Während dieses Prozesses wird sich von Lösungen oder Teillösungen getrennt, von denen abzusehen ist, daß sie nicht zum Optimum führen. Hier liegt auch der wesentliche Unterschied zur Vollenumeration. Um sich von Lösungen oder Teillösungen zu trennen, werden Vergleiche durchgeführt. Die dynamische Planungsrechnung vergleicht nur die parallel aufgebauten Teillösungen, wohingegen beim Branch-and-Bound auch zwischen anderen Teillösungen verglichen wird. Bei der begrenzten Enumeration werden die Teillösungen nur mit der Enumerationsgrenze verglichen /91/.

Eine Trennlinie zwischen der begrenzten Enumeration, der dynamischen Planungsrechnung und des Branch-and-Bound ist aufgrund der Organisation der Enumerierung zu ziehen. Während bei der begrenzten Enumeration streng sequentiell vorgegangen wird, arbeitet die dynamische Planungsrechnung streng parallel. Eine Mischung zwischen paralleler und sequentieller Organisation liegt beim Branch-and-Bound vor.

So gesehen bildet das Branch-and-Bound-Verfahren eine Verallgemeinerung der beiden anderen Vorgehensweisen. Sie sind als extreme Vertreter desselben zu betrachten und bedürfen deshalb keiner näheren Untersuchung.

Die Grundidee beim Branch-and-Bound ist es, die endliche Menge von zulässigen Lösungen so aufzuspalten, daß in einer oder einigen Teilmengen nur noch eine Lösung enthalten ist. Eine Darstellung kann mittels eines Baums geschehen. Die Wurzel beinhaltet alle Lösungen und die Knoten beherbergen eine Teilmenge dieser. Desweiteren müssen eine Verzweigungsvorschrift (Branch), eine Abschätzung (Bound), eine Auswahlvorschrift und ein Abbruchkriterium definiert werden, welche individuell dem Problemfeld angepaßt sind /90/.

In der Verzweigungsvorschrift wird festgelegt, wie der Baum aufzuspalten ist, wobei sukzessive die möglichen Lösungen berechnet werden. Die Abschätzung soll

feststellen, welche Teilmenge der Zielfunktion am besten entspricht. Für das weitere Verzweigen ist die Auswahlvorschrift verantwortlich. Die Grundlage dafür kann die Abschätzung bilden. Bei Erreichen des Optimums sollte das Abbruchkriterium dafür sorgen, daß das Verfahren terminiert.

Der schlechteste Fall tritt dann ein, wenn zur Ermittlung des Optimums alle möglichen Lösungen durchgespielt werden müssen. Obwohl dies in sehr vielen Fällen nicht zutrifft, kann es im voraus nicht festgestellt werden. Dieses Charakteristikum disqualifiziert das Verfahren für einen praktischen Einsatz innerhalb der Werkstattsteuerung, da die Planungslaufzeit als abschätzbare Größe vorliegen muß und dies bei heterogenen Auftragspektren nicht gewährleistet ist. Ein abschließendes Beispiel zur Maschinenbelegung, das /90/ entnommen ist, soll die Arbeitsweise des Verfahrens jedoch nochmals verdeutlichen:

Gegeben sind zwei Maschinen (M,N) und ein Auftragsvolumen über fünf zu fertigende Produkte. Jedes Werkstück muß dazu zuerst auf der Maschine M bearbeitet werden, um dann die Maschine N zu durchlaufen. Nachfolgende Tabelle zeigt die Bearbeitungszeit inklusive ihrer Nebenzeiten pro Werkstück und Maschine.

Werkstück	Maschine M	Maschine N
1	3	2
2	6	5
3	2	3
4	4	9
5	8	3

Die Zielfunktion ist die minimale Gesamtzeit zur Fertigung aller fünf Produkte. Verzweigt wird nach allen Möglichkeiten, ein Werkstück als nächstes zu fertigen. Für die Abschätzung wird die Stillstandszeit von N verwendet, indem bereits feststehende Stillstandszeiten berechnet werden. Diese Schranke erscheint zweckmäßig, da die kleinste Stillstandszeit von N dem Optimum gleichbedeutend ist. Die Auswahlvorschrift ist durch die Schranke bereits vorgegeben. Bei gleichen Schranken wird zuerst der tiefere Ast des Baums gewählt (fortgeschrittenere Fertigung) und anschließend der frühere Zeitpunkt, an dem Maschine M wieder frei wird (Bild 48).

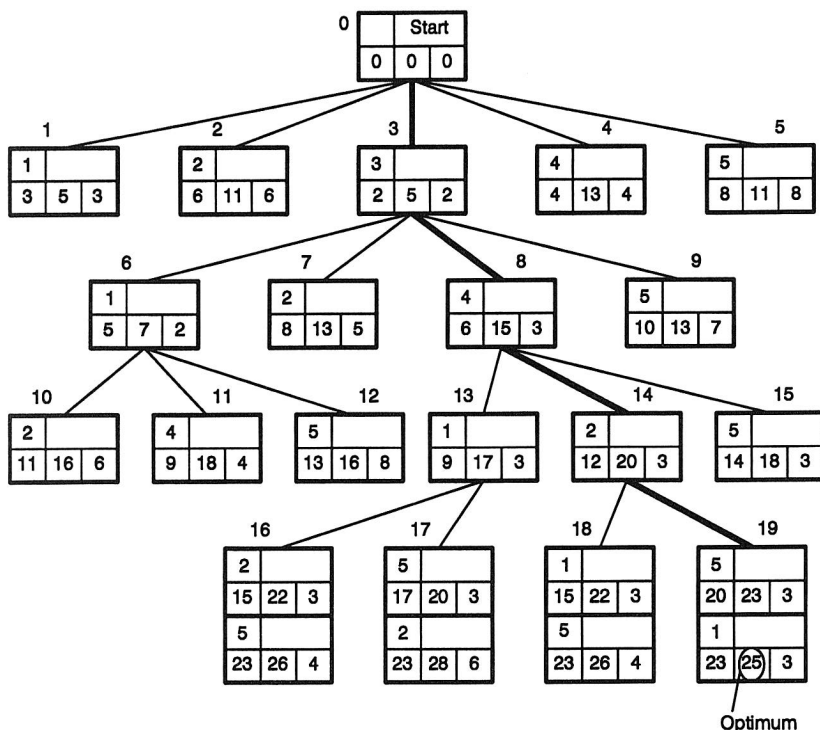


Bild 48: Branch-and-Bound-Verfahren am Beispiel einer Maschinenbelegung

Zur Erläuterung der einzelnen Schritte sei der Baum bis Knoten 12 erzeugt. Die Nummern über den einzelnen Kästchen spiegeln dabei die Entstehungsreihenfolge wider. Der aktuelle Knoten muß mit den noch verzweigbaren Knoten verglichen werden, um sicher zu gehen, daß ein Optimum erreicht wird. In diesem Fall wird "zurückgelaufen", da die Schranke in Knoten 10, 11 und 12 größer ist als in Knoten 8, der dann seinerseits weiter verzweigt wird. Soll die Eindeutigkeit des Optimums nachgewiesen werden, müssen noch die Knoten 1 und 15 verzweigt werden.

6.1.4 Heuristische Verfahren

Der teilweise hohe Rechen- und Zeitaufwand der bisherigen Verfahren schränkt ihre Einsatzmöglichkeiten für die Praxis erheblich ein. Ist dieser Aufwand nicht mehr zu rechtfertigen, kann auf sogenannte heuristische Verfahren zurückgegriffen werden. Dabei wird nach bestimmten Regeln, die hinsichtlich der Problemlage sinnvoll erscheinen, versucht, die Zielfunktion zu lösen. Es kann nicht gewährleistet werden, daß das Optimum erreicht wird. Vielmehr genügt die Ermittlung eines sogenannten Suboptimums, was mit geringem Aufwand zu bewerkstelligen ist. (Die Beschreibung der heuristischen Verfahren ist im wesentlichen dem Buch "Operations Research" von Müller-Merbach /91/ entnommen.)

Die Auswahl, ein heuristisches Verfahren oder ein Entscheidungsbaumverfahren einzusetzen, wird von der jeweiligen Anwendung bestimmt. Ist eine exakte Lösung erforderlich, liegt die Optimumsfindung in einem vertretbaren Rahmen und liefern die heuristischen Verfahren sehr schlechte Ergebnisse, dann sind die Entscheidungsbaumverfahren den Heuristiken vorzuziehen, im anderen Fall umgekehrt.

Heuristische Verfahren werden in der Regel auf ein bestimmtes Problem zugeschnitten. Ihre Anzahl ist daher immens. Dennoch können sie in zwei Gruppen geteilt werden, den Eröffnungsverfahren und den suboptimierenden Iterationsverfahren. Sie unterscheiden sich in ihrer Vorgehensweise.

Die Eröffnungsverfahren arbeiten auf eine bestimmte Lösung zu, welche dann nicht mehr verbessert werden kann. Die suboptimierenden Lösungsverfahren versuchen eine schrittweise Verbesserung der Lösung zu erzielen. Mit der Kombination verschiedener Verfahren kann eine bessere Lösung erzielt werden.

Als Beispiel eines Eröffnungsverfahrens sei das Verfahren des besten Nachfolgers auf das Problem des Handlungsreisenden angewandt. Hierbei ist eine Rundreise durch verschiedene Orte zu bestimmen, so daß jeder Ort genau einmal durchlaufen wird und beginnend vom Ausgangsort eine optimale Wegstrecke bezüglich einer Bewertung (Kosten, Weglänge, etc.) zurück zu diesem gefunden wird.

Das Verfahren des besten Nachfolgers geht von einem beliebigen Ort aus und legt den nachfolgenden Ort so fest, daß eine minimale Strecke im Bezug zur Gewichtung zurückgelegt wird. Es wird solange fortgefahren, bis eine komplette Strecke ermittelt ist.

Für das Beispiel in Bild 49 ergibt das Verfahren über alle möglichen Startknoten folgende Lösungen:

A - C - D - B - F - E - A	Summengewicht: 62 ¹⁾
B - D - C - A - E - F - B	Summengewicht: 62
C - A - D - B - F - E - C	Summengewicht: 64
D - C - A - E - F - B - D	Summengewicht: 62
E - C - A - D - B - F - E	Summengewicht: 64
F - D - C - A - E - B - F	Summengewicht: 65

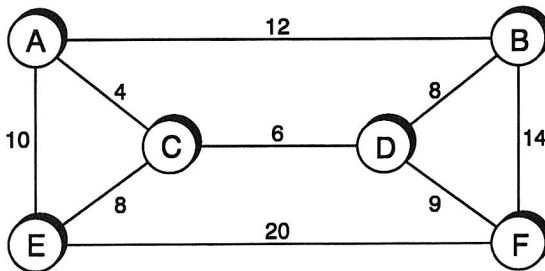


Bild 49: Ortnetz mit gewichteten Strecken

Das Verfahren führt in der Regel zu relativ schlechten Ergebnissen. Dies liegt daran, daß der Freiheitsgrad der Auswahl im Laufe des Verfahrens abnimmt. Dafür hält sich der Rechenaufwand in Grenzen. Bei n Orten müssen $0,5 \cdot (n^2 - n)$ Möglichkeiten berechnet werden. Im Vergleich zu $(n - 1)!$ möglichen Lösungen ist diese Zahl gerade bei großem n eher gering.

Zusammenfassend kann zur Beurteilung der heuristischen Verfahren festgestellt werden, daß sie sehr oft nicht zum Optimum führen. Schwierig erweist sich auch die Kombination mehrerer Zielfunktionen. Doch gerade diese beiden Eigenschaften (Möglichkeit der Optimumsfindung und Definition mehrerer Zielfunktionen) stellen wesentliche Eigenschaften für die Konzeption des Werkstattsteuerungssystems in dieser Arbeit dar.

1) Optimum = 59

6.1.5 Vorrangregeln

Eingesetzt werden auch Verfahren, bei denen nach sogenannten Vorrangregeln vorgegangen wird. Die Auswahl eines Lösungsschritts wird eben durch diese Regeln festgelegt. Sie liefern in bezug auf ein Optimierungsziel gute Lösungen, versagen jedoch bei mehrschichtiger Zielvorlage.

Aufgrund ihrer Beliebtheit sollen die wichtigsten Regeln für eine Auftragsauswahl zur Maschinenbelegung vorgestellt werden. Dabei ist für einen Planungsschritt der Auftrag auszuwählen, der der Regel entspricht /92/. Eine Optimierung ist dann nur relativ zur Vorrangregel und nicht zur Gesamtlösungsmenge zu sehen.

First-In-First-Out

Der zuerst ankommende Auftrag wird zuerst eingeplant.

Last-In-First-Out

Der zuletzt ankommende Auftrag wird zuerst eingeplant.

Longest-Processing-Time

Der Auftrag mit der längsten Bearbeitungsdauer wird zuerst eingeplant.

Shortest-Processing-Time

Der Auftrag mit der kürzesten Bearbeitungsdauer wird zuerst eingeplant.

Nearest-End

Der Auftrag mit dem frühesten Fertigstellungstermin wird zuerst eingeplant.

Earliest-Possible-Beginning

Der Auftrag mit dem frühest möglichen Starttermin wird zuerst eingeplant.

Greatest-Operation-Number

Der Auftrag mit den meisten Restoperationen wird zuerst eingeplant.

Slack

Der Auftrag mit der geringsten Differenz zwischen Fertigstellungstermin und benötigter Bearbeitungsdauer wird zuerst eingeplant.

Zudem ist zu erwähnen, daß diese Regeln auch miteinander zu kombinieren sind. Eine neue Regel kann aus dem Verhältnis der Schlupfzeit (Slack) zur Anzahl der Restoperationen gebildet werden.

6.1.6 Resümee

Wie bereits in der Bewertung der einzelnen Methoden ausgeführt, bleibt festzustellen, daß die bisherigen Methoden nicht direkt zur Planung, Steuerung oder Überwachung verwendbar sind. Einerseits ist die zu lange beziehungsweise nicht immer abschätzbare Laufzeit für eine praktische Anwendung im Bereich der Werkstattsteuerung nicht akzeptabel (Vollenumeration, lineare Optimierung, Branch-and-Bound), andererseits ist eine gleichbleibende Güte der Ergebnisse bei heterogenen Auftragspektren nicht gewährleistet (heuristische Verfahren, Vorrangregeln). Für diese Aufgabe müssen daher neue Konzeptionen entworfen werden, wobei auf die Grundgedanken der vorgestellten Methoden zurückgegriffen werden kann.

6.2 Rahmenbedingungen

6.2.1 Bereitstellung notwendiger Informationen

Die Voraussetzung für eine flexible Fertigung ist die Fähigkeit, vorhandene Flexibilitätspotentiale zu nutzen (Kapitel 2.2). Objektflexibilitäten sind mit der Fertigungseinrichtung mehr oder weniger gegeben, denn die Geräte- oder Werkzeugflexibilität wird von den zur Verfügung stehenden Systemkomponenten bestimmt.

Diese Fähigkeit muß jedoch in der Arbeitsvorbereitung bereitgestellt werden, um sie auch informationstechnisch verwenden zu können.

Gerätealternativen

In einer flexiblen Fertigungseinrichtung können zur Bearbeitung eines Arbeitsganges unterschiedliche Geräte eingesetzt werden. Zum Beispiel werden in einem System zur Herstellung von Drehteilen mehrere Drehmaschinen vorhanden sein, da der Anteil der Drehbearbeitung im Vergleich zu den anderen Technologien überwiegt und Engpässe vermieden werden sollen.

Fertigungshilfsmittelalternativen

Die Bearbeitung eines Arbeitsganges erfordert zudem den Einsatz von Fertigungshilfsmitteln. Dies können Werkzeuge, Greifer, Vorrichtungen, Spannmittel und andere sein. In Analogie zu oben können auch hier pro Arbeitsgang unterschiedliche Fertigungshilfsmittel eingesetzt werden. Das kann beispielsweise für Werkzeuge entweder die Auswahl eines anderen Werkzeugtyps sein oder eines alternativen Werkzeugs des gleichen Typs. Im ersten Fall kann durch die Verwendung eines anderen Werkzeugs eventuell die Bearbeitungszeit eines Arbeitsganges vermindert werden, da sich unterschiedliche Werkzeugtypen auf die Dauer der Bearbeitung auswirken. Als Beispiel seien eine Hartmetall- und Keramik-Wendeschneidplatte genannt. Im zweiten Fall kann bei Werkzeugbruch oder ähnlichen Vorfällen auf ein redundantes Werkzeug zurückgegriffen werden.

Derartige Informationen müssen im Arbeitsplan festgehalten werden. Dazu werden die existierenden Arbeitsplanstrukturen erweitert, ohne auf das vorhandene Grundgerüst zu verzichten. Bei auftretenden Störungen während des Fertigungsprozesses besteht dann die Möglichkeit, daß darauf zurückgegriffen und eine flexible Reaktion in die Wege geleitet wird; andernfalls gehen wichtige Flexibilitätspotentiale verloren.

6.2.2 Erweiterung des Arbeitsplans

Finden die Geräte- und Fertigungshilfsmittelalternativen ebenso Berücksichtigung im Arbeitsplan wie die Technologiealternativen, so kann die Arbeitsplanstruktur die

Gestalt annehmen, wie sie in Bild 50 in quasiformaler Darstellung illustriert ist und im weiteren auch verwendet wird. Die Arbeitsganginformationen werden dabei in den Knoten des Ablaufgraphen gespeichert, wodurch sich der Ablaufgraph und der Arbeitsplan ausgezeichnet ergänzen.

Ident-Nr.:		Datum:		ARBEITSPLAN	
		Bearbeiter:			
Auftrags-Nr.:		Stückzahl:		Termin:	
Werkstoff:		Abmessungen:		Rohgewicht:	
				Fertiggewicht:	
Nr.	Beschreibung	Gerätegruppe	Fertigungs- mittel	Rüstzeit	Haupt- zeit
10	AG-10	M ₁	FHM-Satz ₁₁	trZ	thZ
			FHM-Satz ₁₂	trZ	thZ
		M ₂	FHM-Satz ₂₁	trZ	thZ
20	AG-20	M ₃	FHM-Satz ₃₁	trZ	thZ
30	AG-30	M ₁	FHM-Satz ₁₂	trZ	thZ
31	AG-31 (Alternative zu 30)	M ₄	FHM-Satz ₄₁	trZ	thZ
40	AG-40	M ₅ M ₆	FHM-Satz ₅₁	trZ	thZ
			FHM-Satz ₅₂	trZ	thZ

Bild 50: Neue Arbeitsplanstruktur

6.2.3 Voraussetzungen für das Auftragsvolumen

Für das Werkstattsteuerungskonzept im Rahmen dieser Arbeit müssen zudem verschiedene Voraussetzungen für das Auftragsvolumen erfüllt sein.

Auftragspool

Vom Auftragspool muß sichergestellt sein, daß er auch innerhalb der zeitlichen Rahmenbedingungen bearbeitet werden kann. Möglich ist eine Planung über einen unbegrenzten Zeitraum, sinnvoll erscheint aber eine zeitliche Limitierung auf

maximal eine Schicht beziehungsweise einen Tag (Kapitel 2.1.3), da die Planung einen Idealzustand in der Fertigung annimmt und keine unvorhergesehenen Störungen berücksichtigt. Die Planung kann auf die jeweilige Situation abgestimmt werden und behält länger ihre Gültigkeit. Außerdem kann durch einen kleiner dimensionierten Pool die Rechenzeit der einzelnen Inferenzkomponenten in akzeptablen Grenzen gehalten werden.

Aufträge

Der Auftragspool beinhaltet eine Menge von Aufträgen. Ein Auftrag darf nur die Fertigung identischer Erzeugnisse umfassen. Varianten eines Produktes werden in verschiedenen Aufträgen behandelt. Wenngleich zwischen einer los- und einer mixweisen Fertigung nicht unterschieden wird, sind nachfolgende Konzeptionen eher für eine Mixfertigung bei kleinen Serien ausgelegt. Zudem besteht die Möglichkeit, den Aufträgen Prioritäten zuzuordnen. Dadurch können auch Eilaufträge schnell durch das System geschleust werden.

Globale Verfügbarkeitsprüfung

Von den Aufträgen muß durch eine Verfügbarkeitsprüfung gewährleistet sein, daß die nötigen Fertigungshilfsmittel und Materialien im Bearbeitungszeitraum vorhanden sind. Es muß nicht abgeklärt werden, zu welchem Zeitpunkt sie an welchem Ort sein müssen, sondern vielmehr, daß sie prinzipiell zur Verfügung stehen. Das umfaßt auch nicht die eigentliche Bereitstellung, was Aufgabe der Steuerung ist, sondern schließt nur das Vorhandensein der Objekte in ausreichender Menge ein. Im Auftragspool dürfen sich demnach nur Aufträge befinden, die durch die Verfügbarkeitsprüfung gelaufen sind.

6.3 Planungskomponente

Der Planung obliegen eine Reihe von Aufgaben, die die Voraussetzungen für eine wirtschaftliche Fertigung bilden sollen. Als Ergebnis eines Planungslaufs entsteht ein Maschinenbelegungsplan für ein bestimmtes Zeitintervall. Die Güte dieses Plans wirkt sich dabei auch in finanziell für das Unternehmen aus.

6.3.1 Optimierungsziele

Die Anforderungen des Marktes, wie zum Beispiel hohe Termintreue oder preisgünstige Produkte, widersprechen mitunter denen der Unternehmer /93/. Diese verlangen eine gute Auslastung der Maschinen oder kurze Durchlaufzeiten /94/. Für die Fertigung ergibt sich daher ein Zielkonflikt /95/, den es zu lösen gilt.

Um aber diesen Konflikt in den Griff zu bekommen, muß eine Planung, die nur in eine Richtung verläuft, vermieden werden. Dadurch kann nämlich eine Verschlechterung in eine andere Zielrichtung verursacht werden, da die einzelnen Ziele zueinander in Korrelation stehen. Zwar kann durch mehrere Planungsläufe eine Optimierung zur Lösung der anderen Ziele durchgeführt werden, aber die einzelnen Pläne müßten gegenübergestellt und bewertet werden. Im Endeffekt wäre eine Entscheidung für den besten Plan auch eine Entscheidung zugunsten einer Zielrichtung, ohne die anderen zu berücksichtigen. Ebenso wäre eine Vorgehensweise denkbar, die zuerst nach dem meistverfolgten Ziel optimiert, um dann innerhalb festgesteckter Rahmenbedingungen (z.B. Termineckdaten) zu versuchen, noch in andere Richtungen weiterzuoptimieren. Die Abhängigkeiten der Zielsetzungen bedingen jedoch einen zu engen Spielraum, um auch hier einer ganzheitlichen Optimierung zu genügen.

Daher wird eine Vorgehensweise deklariert, die versucht allen Zielgebungen gerecht zu werden. Dies kann nur gelingen, wenn eine Gesamtbetrachtung der Zielfunktionen angestrebt wird. Da die Planung aus verschiedenen Reihenfolgemöglichkeiten und deren Alternativen auswählen kann, entstehen zahlreiche Einplanungsvarianten, von denen die beste zu ermitteln ist. Sie stellt dann den optimalen Maschinenbelegungsplan dar.

Im vorliegenden Fall ermöglicht die Planungskomponente eine Optimierung nach folgenden Zielfunktionen:

- minimale Bearbeitungszeiten
- niedrige Kosten
- hohe Auslastung bzw. wenig Stillstand
- kurze Durchlaufzeiten
- geringe Rüstzeiten

Unter der Prämisse, daß der Auftragspool, der aus einer höheren Ebene stammt, so terminiert ist, daß eine zeitgerechte Fertigung möglich ist, wird die Termintreue nicht mehr als explizites Ziel aufgeführt.

Die einzelnen Optimierungsziele können vom Planer mit unterschiedlichen Gewichten versehen werden. Diese geben Auskunft darüber, wie stark ein Ziel verfolgt werden soll. Sie entsprechen einem ganzzahligen Wert aus dem Intervall von null bis neun. Eine Gewichtung mit null kommt einer Nichtbeachtung gleich, während eine Neun als stärkster Faktor in die Bewertung eingeht.

Bewertet werden dabei die unterschiedlichen Ergebnisvarianten aus der Planungsphase. Dazu multiplizieren sich die Zeit- und Kostenwerte der Planungseinheiten (Kapitel 6.3.2) mit dem Gewicht, wobei die Kosten wiederum mittels einer Nivellierung an die Zeiten angeglichen werden müssen. Die Summe der einzelnen Bewertungsfaktoren bildet einen Gesamtwert, der als Güte der Planung zu verstehen ist. Der Plan mit der besten Güte stellt gleichzeitig den optimalen Maschinenbelegungsplan dar. Er bildet ein Optimum bezüglich aller Zielfunktionen entsprechend ihren Gewichten.

Wie bereits oben erwähnt, können sich bei der Optimierung die einzelnen Zielfunktionen gegenseitig beeinflussen. Dieses Problem ist grundsätzlicher Natur, jedoch kann durch die Möglichkeit der Gewichtsverteilung die Planung individuell auf die Anforderungen aus der Fertigung auch im Zusammenhang mit den unternehmerischen Zielen und den Marktbedingungen abgestimmt werden. Der Planer muß sich also dieser Problematik bewußt sein, besitzt aber, und das ist der entscheidende Vorteil, für seinen Fertigungsbereich ein adaptives Planungswerkzeug.

6.3.2 Planungseinheit

In der operativen Ebene finden unterschiedliche Operationen statt. Dabei wird durch eine Grobklassifizierung in primäre und sekundäre Operationen eine Aufteilung hinsichtlich der Funktionalität getroffen. Die sekundären Operationen sind Gegenstand der Untersuchungen für das Steuerungskonzept, welches die erforderlichen Bereitstellungsoperationen von Material und Fertigungshilfsmitteln bewerkstelligt. Sie werden an anderer Stelle von Interesse sein.

Die primären Operationen – sie tragen zur Wertschöpfung des Erzeugnisses bei – können noch feiner klassifiziert werden. Je nachdem an welchem Objekt eine Zustandsänderung bewirkt wird, kann zwischen mittel- und unmittelbaren Operationen unterschieden werden.

Unmittelbar primäre Operationen

Sie führen zu einer Zustandsänderung am Material. Darunter sind sowohl alle direkten Bearbeitungsschritte als auch fertigungsbegleitende Tätigkeiten zu verstehen. Beispiele können sein:

- Werkstück drehen
- Baugruppe fügen
- Werkstück waschen
- Werkstück auf-, um- oder abspannen

Mittelbar primäre Operationen

Sie führen zu einer Zustandsänderung am Gerät oder am Fertigungshilfsmittel. Beispiele können sein:

- Maschine rüsten
- Werkzeug einstellen
- Werkzeug vermessen

Da bei der Planung eine Verteilung des Auftragsvorrats auf die einzelnen Geräte vorgenommen wird, bedarf es einer entsprechenden Dispositionsgröße:

*Eine **Planungseinheit** stellt einen Ausschnitt aus dem Fertigungsverlauf eines Erzeugnisses dar. Sie umfaßt einen oder mehrere Arbeitsgänge, die an einer Fertigungskapazität bearbeitet werden können, und entspricht einem Knoten aus dem Ablaufgraphen.*

Neben den unmittelbar primären Operationen können auch mittelbar primäre Operationen zeitliche Berücksichtigung finden. Aus der Definition lassen sich nämlich für eine Planungseinheit verschiedene Phasen ableiten. Es kann zwischen einer Vorbereitung, einer Bearbeitung und einer Nachbereitung unterschieden werden. Die eigentliche Bearbeitung setzt sich aus den unmittelbar primären Operationen zusammen, während die mittelbar primären Operationen die vor- und nachbereitenden Operationen bilden. Sie kommen dann zum Tragen, wenn keine hauptzeitparallele Durchführung möglich ist.

Bei aufeinanderfolgenden Planungseinheiten kann aber auch eine Vor- oder Nachbereitung entfallen, wenn sich zum Beispiel die Maschinen im richtigen Rüstzustand befinden. Die Bearbeitungsphase selbst kann in ihrer Minimalkonfiguration auch nur einen Arbeitsgang beinhalten. Aufspann-, Umspann- oder Abspannoperationen werden in der Regel in einer eigenen Planungseinheit untergebracht. Eine Ausnahme liegt dann vor, wenn diese Operationen an derselben Fertigungskapazität durchgeführt werden können wie die Arbeitsgänge aus der Planungseinheit (z.B. Umspannen eines Drehteils). Die einzelnen Phasen veranschaulicht Bild 51.

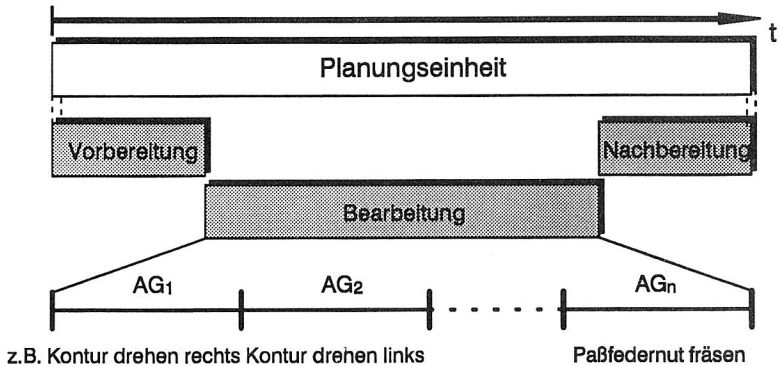


Bild 51: Phasen der Bearbeitung einer Planungseinheit

Mittelbar primäre Operationen, die sich auf Fertigungshilfsmittel beziehen, können in einem eigenen Management behandelt werden. So kann zum Beispiel durch ein Werkzeugmanagement eine Werkzeugeinstellung initiiert werden. Die Integration in das System erfolgt dann über einen separaten Auftrag, der im Auftragspool bereitgestellt wird.

6.3.3 Planungsvorgehen

Die Reihenfolgeermittlung, wie sie in Kapitel 4.4.2 beschrieben ist, liest die notwendigen Arbeitsganginformationen ein und legt ihr Ergebnis in einer baumartigen Datenstruktur ab. Jeder Ast stellt eine mögliche Bearbeitungsreihenfolge dar. Sowohl die **Geräte- als auch die Fertigungshilfsmittelalternativen** sind in dieser Struktur eingebunden. Ein Knoten entspricht dabei einer Planungseinheit (Bild 52).

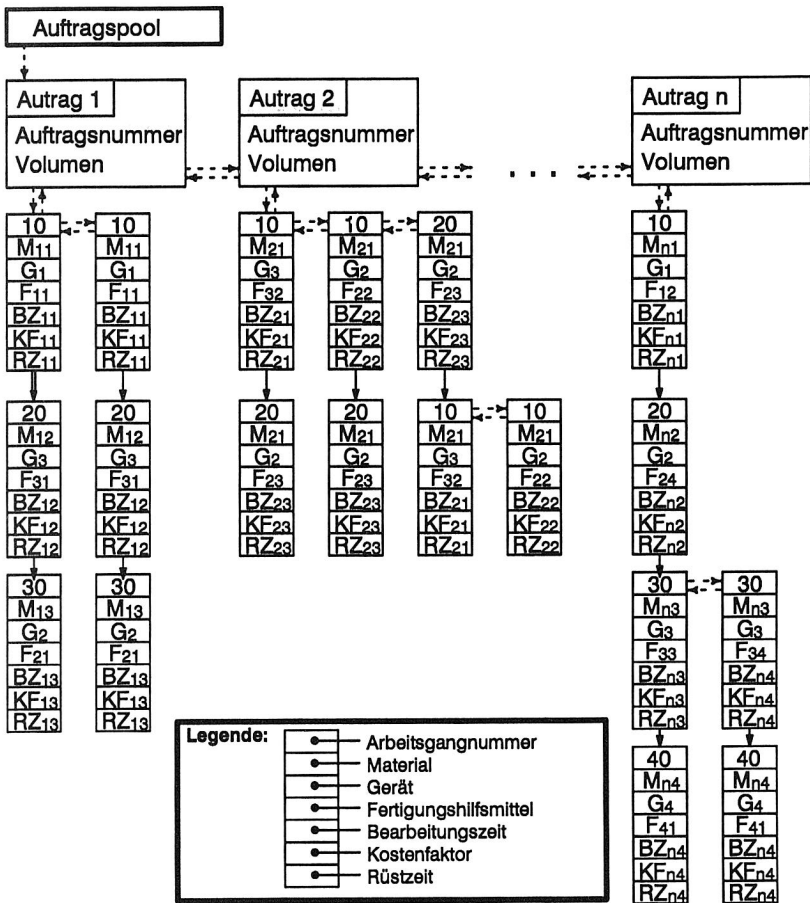


Bild 52: Ablaufreihenfolgestruktur unter Berücksichtigung der Alternativen

Gegeben ist also eine diskrete Optimierungsaufgabe. Die Variablen nehmen nur endlich viele Werte an. Gesucht ist ein Kombinationsverfahren, das die verschiedenen Aufträge so einplant, daß ein Optimum gefunden wird, ohne die Randbedingungen zu verletzen.

Für das Planungsvorgehen muß aus den verschiedenen Reihenfolgemöglichkeiten für jedes Erzeugnis die Reihenfolge ausgewählt werden, die sich mit den Reihenfolgen aller Aufträge aus dem Pool am besten kombinieren läßt, um zu einem Gesamtoptimum zu kommen. Dabei führt eine Planung, welche aus jedem Ablaufgraph nur die bestbewertete Reihenfolge heraussucht, mit sehr hoher Wahrscheinlichkeit nicht zum Optimum. Dies kann nur dann mit Sicherheit ermittelt werden, wenn jede Einplanungsmöglichkeit aus dem Ergebnisbaum auch mit jeder anderen kombiniert und bewertet wird.

Ein kleines Beispiel soll die beschriebene Ablaufweise besser illustrieren. Es liegen zwei Aufträge für die Planung an, die jeweils zwei Planungseinheiten umfassen. Reihenfolgebeschränkungen aus dem Ablaufgraphen existieren nicht. In einem ersten Schritt stehen vier Planungseinheiten zur Auswahl. Dazu können im zweiten Schritt jeweils drei weitere hinzukommen. Beim dritten Einplanungsvorgang bleiben dann noch jeweils zwei Möglichkeiten übrig. Insgesamt entstehen 24 Planungsvarianten.

Ausschlaggebend für die Optimumsfindung ist, daß wirklich alle denkbaren Einplanungsmöglichkeiten berücksichtigt werden, da für einen anstehenden Planungsvorgang von entscheidender Bedeutung ist, welche Einheiten bereits eingeplant sind. Optimal ist diesbezüglich nur eine vollständige Permutation/Enumeration, jedoch kann diese Vorgehensweise bei großem Auftragsvolumen sehr komplex werden (Kapitel 6.1.1).

Es soll jedoch auf der einen Seite die Planung in der Lage sein, ein wirkliches Optimum zu ermitteln, auf der anderen Seite soll aber auch die Laufzeit des Algorithmus für einen Einsatz in der Praxis akzeptabel bleiben. Um beiden Ansprüchen Genüge zu leisten, werden mit der **Planungsauswahl** und der **Planungsdistanz** zwei Stellgrößen eingeführt, die eine individuelle Planung ermöglichen.

Die beiden Schranken gewährleisten, daß die Planungslaufzeit den vorliegenden Gegebenheiten angepaßt werden kann, also auch für eine schnelle Umplanung, die

aufgrund von Störungen notwendig werden kann, einzusetzen ist. Zudem wird die Möglichkeit einer Optimalplanung nicht ausgeschlossen, wenn entsprechende Zeiträume für einen Planungslauf vorhanden sind (z.B. während der Nacht).

Planungsauswahl

*Die **Planungsauswahl** stellt einen ganzzahligen Wert dar, der Auskunft darüber gibt, wieviele Reihenfolgemöglichkeiten aus dem Ergebnisbaum eines Ablaufgraphen für die Planung berücksichtigt werden. Diese sind bezüglich einer Bewertung optimal.*

Durch eine Reihenfolgebewertung können aus allen vorhandenen Reihenfolgen die besten ermittelt werden. Dies geschieht nach folgenden Kriterien:

- minimale Bearbeitungszeit
- minimale Kosten
- minimale Rüstzeit

Dabei werden von der Planung die jeweiligen Werte aus den Arbeitsganginformationen entnommen und mit einem Gewichtungsfaktor, der analog dem der Optimierungsziele zu verwenden ist, multipliziert. Die Summe der einzelnen Faktoren bildet gleichzeitig die Grundlage zur Entscheidung, welche Reihenfolgen Berücksichtigung finden. Die Rüstzeit ist als Absolutwert anzusehen, da erst während der Planung festgestellt werden kann, ob überhaupt gerüstet werden muß.

Planungsdistanz

*Die **Planungsdistanz** ist die Größe, die festlegt, wie weit der Planungsalgorithmus zur Einplanung einer Planungseinheit vorausschauen soll.*

Die Planungsdistanz kann vom Wert 1 bis zur maximalen Suchtiefe (Summe aller zu betrachtenden Planungseinheiten) jeden ganzzahligen Zwischenwert annehmen. Zur Einplanung einer Einheit werden alle Kombinationsmöglichkeiten mit der Länge

gleich der Planungsdistanz ermittelt, simulativ eingeplant und bewertet. Anschließend wird von der bestbewerteten Kombination die erste Planungseinheit fest eingeplant. Das Verfahren wird nun iterativ solange fortgeführt, bis alle Einheiten verplant sind. Unterschreitet dabei die Anzahl der noch zur Planung anstehenden Einheiten die Planungsdistanz, wird die beste Kombination komplett eingeplant. Die Wirkungsweise der beiden Größen ist in Bild 53 zu sehen.

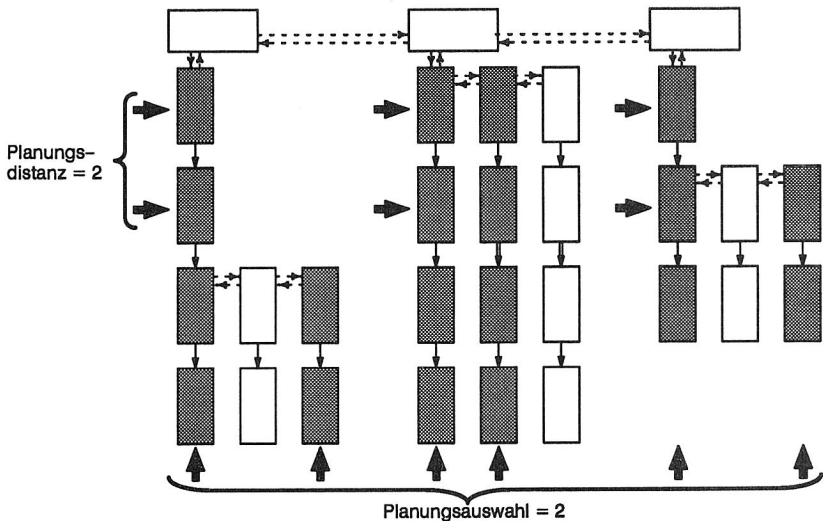


Bild 53: Wirkungsweise der Planungsauswahl und -distanz

In einer Variation des Algorithmus kann nicht nur eine Planungseinheit eingeplant werden, sondern über die gesamte Planungsdistanz. Der Vorteil hierbei ist in einer kürzeren Laufzeit zu sehen. Dem stehen etwas schlechtere Planungsergebnisse gegenüber.

Testläufe haben ergeben, daß der Algorithmus auch schon bei kleinen Schranken einen optimalen Plan ermittelt. Außerdem konnte festgestellt werden, daß Ergebnisse, die nicht das Optimum trafen, doch zumindest sehr nahe an dieses heranreichten. Selbst bei einer Planungsdistanz gleich eins werden noch gute Maschinenbelegungspläne erzeugt.

Die algorithmische Arbeitsweise ist durch das Struktogramm in Bild 54 nochmals verdeutlicht.

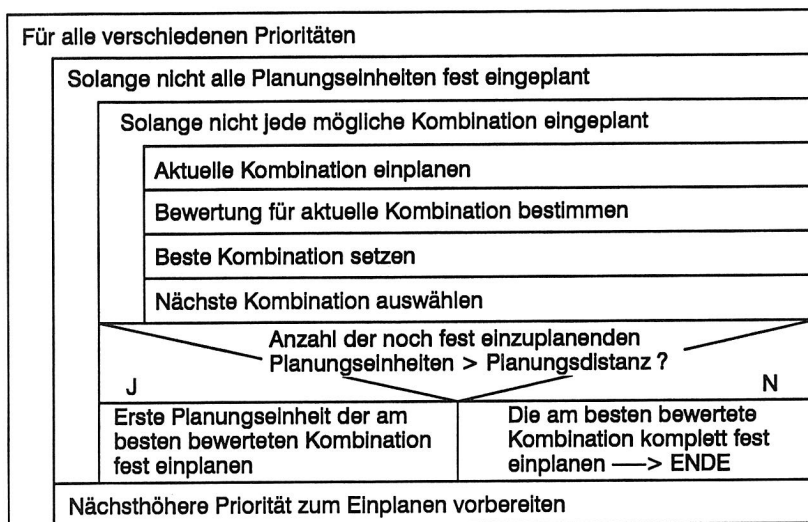


Bild 54: Planungsalgorithmus /96/

Die Begriffe der Planungsauswahl und der Planungsdistanz werden im folgenden an einem Beispiel verdeutlicht:

Im Auftragspool liegen zwei Aufträge, die jeweils drei Planungseinheiten umfassen. Aus dem Ablaufgraphen und den Arbeitsganginformationen lassen sich die auftragsspezifischen Reihenfolgемöglichkeiten ermitteln. Mit diesen wird eine Reihenfolgebewertung unter folgenden Gewichtungsfaktoren durchgeführt:

Gewicht für die Bearbeitungszeit: 5

Gewicht für die Kosten: 4

Gewicht für die Rüstzeit: 1

Bei einer Planungsauswahl von 2 bleiben von den ursprünglich 7 Möglichkeiten 4 übrig, welche in Bild 55 dunkel hinterlegt sind.

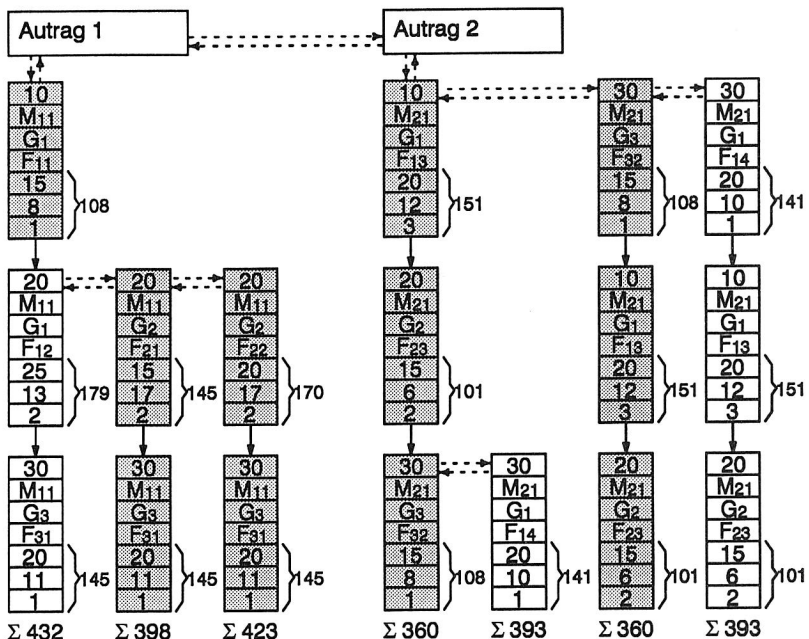


Bild 55: Bewertete Reihenfolgen

Für die Planung ist eine Planungsdistanz = 2 vorgesehen. Die Optimierungsziele besitzen folgende Gewichtung:

Gewicht für die Bearbeitungszeit: 2

Gewicht für die Kosten: 0

Gewicht für die Stillstandszeit: 5

Gewicht für die Durchlaufzeit: 3

Gewicht für die Rüstzeit: 0

Zu beachten ist, daß diese Bewertungsfaktoren mit den vorherigen nichts zu tun haben. Hier wird eine Bewertung für die Maschinenbelegung vorgenommen, während im anderen Fall nur die besten Auftragsreihenfolgen festgelegt wurden.

Zu Anfang werden alle möglichen Kombinationen der Länge 2 ermittelt und mit obigen Gewichten bewertet. Für die beste Kombination wird die erste Planungseinheit fest eingeplant. Von den restlichen Einheiten werden dann wieder alle Kombinationen errechnet und bewertet. Danach wird die zweite Planungseinheit eingeplant und so weiter. Bei gleichem Bewertungsfaktor wird die zuerst ermittelte Möglichkeit weiterverwendet.

Mögliche Kombinationen für den ersten Schritt:

(Planungseinheit/Auftrag)

10/1 – 20/1

10/1 – 20/1

10/1 – 10/2

10/1 – 30/2

10/2 – 10/1

10/2 – 20/2

30/2 – 10/1

30/2 – 10/2

Die Kombination 10/1 – 30/2 ist diejenige mit der besten Bewertung. Die Planungseinheit 10 des Auftrags 1 wird also fest eingeplant. Mittels der vorher erwähnten Planungsvariante können auch beide Planungseinheiten fest eingeplant werden.

Mögliche Kombinationen für den zweiten Schritt:

(Planungseinheit/Auftrag)

20/1 – 30/1

20/1 – 10/2

20/1 – 30/2

usw.

Im zweiten Schritt hat die Kombination 20/1 – 30/2 die beste Bewertung. Zur Planungseinheit 10 von Auftrag 1 wird jetzt 20 eingeplant. Weiterführend wird aus der Kombination 30/2 – 10/1 die Planungseinheit 30 des zweiten Auftrags aufgrund ihrer guten Bewertung eingeplant (3.Schritt).

Das Verfahren wird solange fortgesetzt, bis alle Planungseinheiten fest eingeplant sind. Im fünften Schritt entspricht die Anzahl der einzuplanenden Einheiten der Planungsdistanz, wodurch die Zweierkombination als Ganzes eingeplant werden kann. Der resultierende Maschinenbelegungsplan ist in Bild 56 visualisiert.

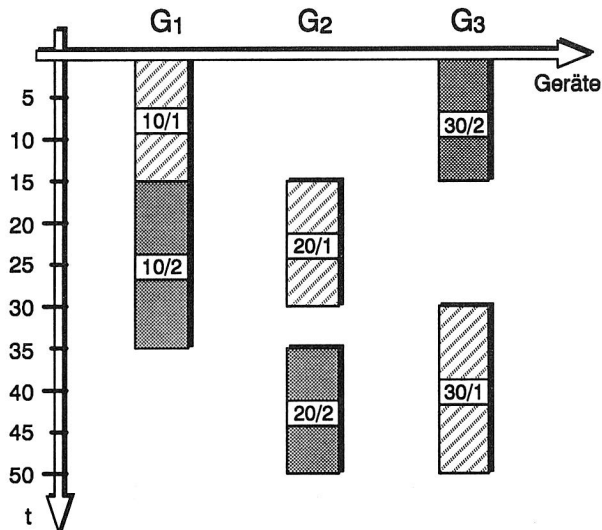


Bild 56: Maschinenbelegungsplan zum Planungsbeispiel

6.3.4 Zusätzliche Funktionalität der Planungskomponente

Neben dem rein kombinatorischen Vorgehen beinhaltet die Planung noch weitere Funktionalitäten:

Lokale Verfügbarkeitsprüfung

Nach der globalen Verfügbarkeitsprüfung, die für den Planungszeitraum abklärt, ob die benötigten Objekte in ausreichender Menge vorhanden sind, kann bei einem einzelnen Planungsschritt eine detailliertere Prüfung vollzogen werden. Es wird sichergestellt, daß genau für die Dauer der Bearbeitung das Material, die Geräte und

die Fertigungshilfsmittel zur Verfügung stehen. Eine mehrfache Belegung wird nicht zugelassen. Störungen im Fertigungsverlauf, die aufgrund fehlender Betriebsmittel oder Maschinenengpässen verursacht werden, sind dadurch von vornherein ausgeschlossen.

Werkskalender

Für Geräte kann das betrachtete Planungsintervall, das in einzelne Zeitscheiben aufgeteilt ist, genauer spezifiziert werden. Analog der Funktionalität eines Werkskalenders können Zeiträume definiert werden, in denen Maschinenkapazität vorhanden ist oder nicht. So können Schichtmodelle eingebracht und/oder Wartungszeiten der Maschinen berücksichtigt werden.

Überlappende Planung

Nach einem Planungslauf wird der Belegungsplan zur Steuerung weitergereicht. In der Regel ist es dabei so, daß der nächste Planungslauf, der den zeitlich folgenden Auftragspool zu disponieren hat, den Beginn seines Planungsintervalls auf das Ende des vorhergehenden setzt. Nun beenden in den meisten Fällen die Geräte ihre Tätigkeit nicht genau zu dem Zeitpunkt, an dem das Intervall endet, sondern früher oder später. Daraus folgt, daß jeder Belegungsplan ein für sich spezifisches Zeitprofil besitzt.

Eine Planung, die einen Folgeplan zu dem Zeitpunkt starten läßt, zu dem der letzte Arbeitsgang des Vorgängers endet, verschenkt vorhandene Kapazität und senkt die Wirtschaftlichkeit in der Fertigung. Sinnvoll ist es daher, zwei aufeinanderfolgende Planungen überlappen zu lassen. Das bedeutet, daß bereits Arbeiten aus dem zweiten Pool angefangen werden, während der erste noch bearbeitet wird.

Durch den Einsatz der Wissensbasis kann zu jedem Zeitpunkt festgestellt werden, welche Situation in der Fertigung vorliegt. Da der Belegungsplan des vorherigen Pools während der Planung für einen laufenden Auftragsvorrat noch existiert, wird aus dem Zeitprofil genau der Startzeitpunkt ermittelt, wann eine überlappende Bearbeitung beginnen kann. Eine "verzahnte" Fertigung ist dadurch ermöglicht.

6.4 Steuerungskomponente

6.4.1 Notwendigkeit einer Steuerung

Ob der Material- und der Fertigungshilfsmittelfluß gemeinsam oder getrennt organisiert werden, hängt hauptsächlich davon ab, wie die Systemkomponenten in der Fertigungseinrichtung konfiguriert sind. Es wird ein Konzept vorgestellt, welches die Steuerungsaufgaben **unabhängig von der Struktur des Fertigungssystems** durchführen kann. Die Steuerung ist als Flußsteuerung zu betrachten, welche Objekte jeder Klassifikationsart, sofern sie mobil sind, von einer Quelle zu einem Ziel bringt. Dadurch lassen sich sämtliche Transport- und Handhabungsoperationen betrachten, ohne sich durch irgendwelche Rahmenbedingungen beschränken zu müssen.

Die Steuerungskomponente ist dabei im Gesamtsystem für einen optionalen Einsatz konzipiert. So könnte mit dem Maschinenbelegungsplan, wie er aus der Planung kommt, bereits die Fertigung durchgeführt werden. Die notwendigen Bereitstellungsoptionen müßten dann aber von einem anderen Teilsystem der Werkstattsteuerung übernommen werden. Wird jedoch die Steuerung angestoßen, so ist es ihre Aufgabe, alle sekundären Operationen zu ermitteln, die für eine Bereitstellung der benötigten Fertigungshilfsmittel und Materialien erforderlich sind.

Die Notwendigkeit einer Steuerung wird durch die Tatsache gestärkt, daß von 100% Belegungszeit eines flexiblen Fertigungssystems nur ein effektiver Nutzungsgrad von durchschnittlich 81,3% verbleibt. Ein sehr großer Anteil des Verlustes wird neben den technischen Ausfallzeiten, wie zum Beispiel Maschinenausfall, durch organisatorische Störungen verursacht – bis zu maximal 24% /97/ (Bild 57).

Diese organisatorischen Störungen werden hauptsächlich durch fehlende Werkstücke, Werkzeuge oder Vorrichtungen verursacht. Sie gilt es zu minimieren, indem eine **rechtzeitige Bereitstellung der Objekte** gewährleistet wird.

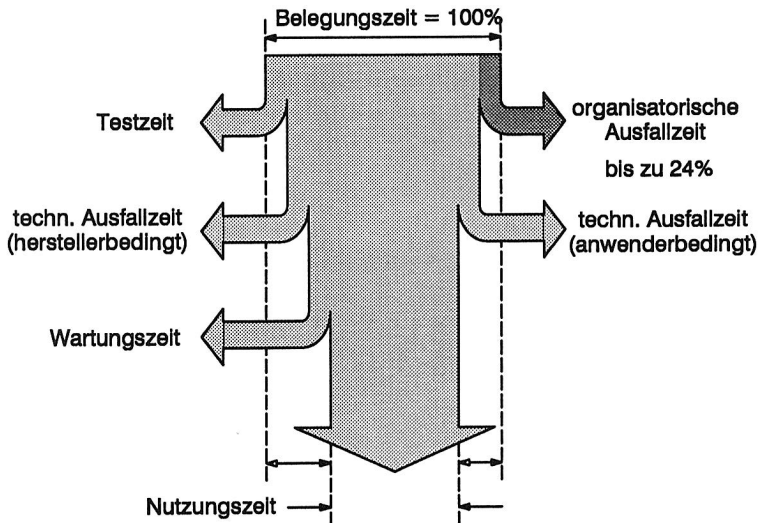


Bild 57: Nutzungsgradverminderung bei flexiblen Fertigungssystemen /97/

6.4.2 Funktionsumfang der Steuerungskomponente

Ausgangspunkt ist der Maschinenbelegungsplan. Durch ihn kann festgestellt werden, wo ein Bedarf an Objekten herrscht. Die Steuerungskomponente ermittelt diesen Bedarf, generiert die daraus notwendigen sekundären Operationen und fügt sie in den Belegungsplan ein.

Bedarfsermittlung für sekundäre Operationen

Die Bedarfsermittlung fand bereits bei der Konzeption der Wissensbasis Berücksichtigung. Eine Fertigungseinrichtung wird für die Wissensbasis in physikalisch relevante Ortseinheiten zerlegt (Kapitel 5.3.2). Relevant erscheinen die Orte, die entweder Quelle oder Ziel einer Steuerungsoperation sein können.

In den Arbeitsganginformationen ist festgehalten, welche Objekte (In der Regel Material und Fertigungshilfsmittel) zur Bearbeitung benötigt werden und an welchem Ort sie sich befinden müssen. Durch die Platzierungs-Beziehung kann im

semantischen Netz der Wissensbasis festgestellt werden, an welchem Ort sich ein Objekt zu einem beliebigen Zeitpunkt aufhält. Mit diesen beiden Informationen ist die Steuerung in der Lage, für jede Planungseinheit abzuklären, ob alle Objekte am richtigen Ort sind. Ist dies nicht gegeben, muß herausgefunden werden, wo sich das Objekt befindet und wie es an den entsprechenden Ort gelangen kann (Bild 58).

Planungseinheit

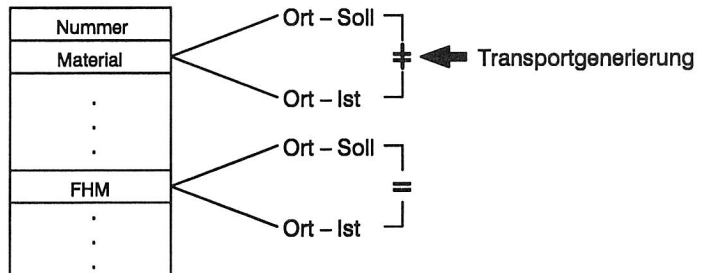


Bild 58: Bedarf an Transportoperationen

Für die Realisierung der Bedarfsermittlung kann im wesentlichen zwischen zwei Prinzipien unterschieden werden, einem Schiebe-Prinzip (Push) und einem Zieh-Prinzip (Pull). Der wichtigste Unterschied dabei ist der Zeitpunkt, an dem die Bedarfsermittlung angestoßen wird.

Beim **Push-Prinzip** wird nach Ablauf einer Planungseinheit festgestellt, wo die Objekte als nächstes benötigt werden. Eine Generierung der entsprechenden Operationen wird sofort eingeleitet. Die Objekte werden sozusagen weitergeschoben.

Dagegen wird beim **Pull-Prinzip**, welches die Grundlage für diese Realisierung bildet, die Bereitstellungsoperation erst direkt vor der Bearbeitung generiert, so daß die benötigten Objekte gerade rechtzeitig ankommen. Die primäre Operation "zieht" quasi ihre Objekte heran. Ausgangspunkt der Überlegungen ist die Annahme, daß eine sekundäre Operation möglichst spät initiiert werden sollte. Das hat den Vorteil, daß bereits frühzeitig angestoßene Transportoperationen, die aufgrund später eintretender Störungen überflüssig geworden sind, vermieden werden können. Somit wird ein **Just In Time** – Steuerungsprinzip verfolgt.

Generierung der Operationen

Aus der gesamten Wissensbasis können die Menge aller Orte herausgelöst werden. Es entsteht so ein Teilnetz, welches zur Generierung der Transportoperationen dient. Mittels der Orientierungs-Beziehung soll nun ein Weg in diesem Netz gefunden werden, auf dem sich das Objekt von seinem aktuellen Aufenthaltsort zu seinem Ziel transportieren läßt. Die Wegsuche gestaltet sich derart, daß der jeweils kürzeste Weg zwischen der Quelle und dem Ziel gesucht wird, wobei bei mehreren Alternativen eine geringere Modifikation des Maschinenbelegungsplans prior behandelt wird. Die Länge eines Weges bezieht sich dabei auf die Dauer der Ausführung.

Durch die Einführung der Orte als Knoten im semantischen Netz kann das Problem der Wegsuche auf das des kürzesten Wegs in einem Graphen reduziert werden. Für die Berechnung eignet sich am besten das Verfahren nach Dijkstra, das im wesentlichen in drei Schritten charakterisiert ist /91/:

- 1) Wahl des Ausgangsorts als Basisort
- 2) Berechnung der gesamten Weglänge zu allen Nachbarorten des Basisorts, die noch nicht Basisort waren. Ist der Weg kürzer als der beste bisher bekannte Weg, wird der bisherige Weg, andernfalls der neue Weg gelöscht.
- 3) Wahl desjenigen Orts als Basisort, der unter den Orten, die noch nicht Basisort gewesen sind, dem Ausgangsort am nächsten ist. Ist dieser Ort der Zielort, dann terminiert das Verfahren, ansonsten zu 2)

An einem Beispiel /91/ sei das Verfahren verdeutlicht. Im Graphen aus Bild 59 soll der kürzeste Weg von Knoten 1 nach Knoten 3 gefunden werden. Die Zahlen an den Kanten spiegeln dabei die Distanz wider.

Der Algorithmus beginnt also bei Knoten 1 und berechnet für alle direkten Wege zu den Nachbarknoten die Entfernung:

Strecke:	1 – 2	Entfernung:	8
Strecke:	1 – 3	Entfernung:	12
Strecke:	1 – 4	Entfernung:	2
Strecke:	1 – 5	Entfernung:	5

Der nächste Knoten ist in diesem Fall der Knoten 4. Er wird als Basis weiterverwendet. Von da an werden jetzt alle noch nicht durchlaufenen Nachbarknoten berechnet:

Strecke: 1 – 4 – 2 Entfernung: 11

Strecke: 1 – 4 – 5 Entfernung: 4

Der Weg 1 – 4 – 2 ist weiter als der direkte Weg und daher nicht weiter interessant. Dagegen wird entlang von 1 – 4 – 5 gegenüber dem direkten Weg Strecke eingespart, wodurch Knoten 5 zum Basisknoten wird. In Fortführung des Verfahrens ergeben sich dann:

Strecke: 1 – 4 – 5 – 2 Entfernung: 11

Strecke: 1 – 4 – 5 – 3 Entfernung: 8

Die Strecke 1 – 4 – 5 – 3 ist kürzer als die direkte Verbindung und kein unbehandelter Knoten ist dem Ausgangsknoten näher als der Zielknoten. Die Lösung ist somit gefunden.

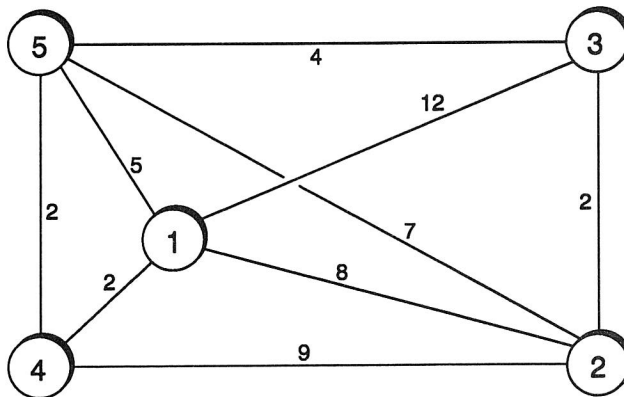


Bild 59: Beispielgraph für das Dijkstra – Verfahren

Der Lösungsweg kann sich demnach aus einzelnen Teilstrecken zusammensetzen. Zu beachten ist, daß alle Transportoperationen, die im Netz ablaufen, zu

synchronisieren sind. Ebenso kann es sein, daß unterschiedliche Objekte aus physikalischen Gegebenheiten auf verschiedenen Wegen ihr Ziel erreichen müssen.

Die Orientierung gibt zusätzlich zum Quell- und Zielort an, welche Objekte auf diesem Weg überhaupt transportiert werden können, in welcher Menge und welche Hilfsmittel benötigt werden. Die Steuerung kann somit gewährleisten, daß der Transport korrekt durchgeführt wird (Bild 60). Ist eine Route momentan nicht gangbar, weil zum Beispiel ein Zwischenort belegt und eine Alternativroute nicht existent ist, so muß eine Operation generiert werden, die durch einen Ausweichtransport den besetzten Ort räumt. Als Prämisse wird deshalb ein ausreichend großes Netz verlangt. Dies ist gegeben, wenn für jedes Transportmittel ein exklusiver Ausweichort existiert. Die Ausweichtransporte können dann problemlos generiert werden, ohne Verklemmungen zu verursachen.

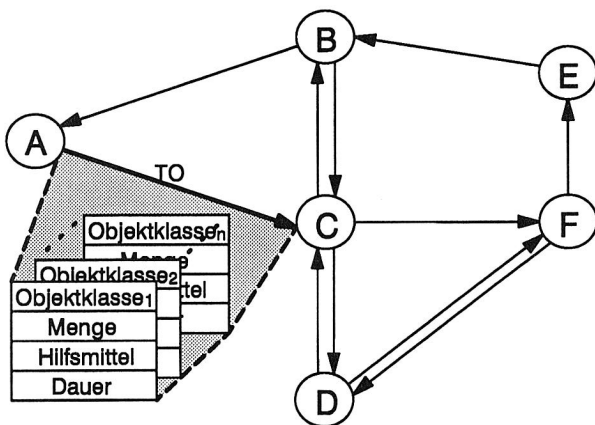


Bild 60: Transportnetz mit Orientierungs-Beziehungen

Einfügen der sekundären Operationen

Nachdem die Routen und somit auch die Zeiten festgelegt sind, können in einer abschließenden Phase die sekundären Operationen in den Maschinenbelegungsplan eingefügt werden. Im Belegungsplan sind die Planungseinheiten

aber so terminiert, daß Transport- oder Handhabungsoperationen noch nicht berücksichtigt werden.

Ist für die Planungseinheiten durch die Steuerung ein Bedarf an sekundären Operationen ermittelt worden, wird die Zeit, die für einen Transport benötigt wird, mit der Zeitspanne verglichen, die zur Verfügung steht, um die Objekte rechtzeitig beizubringen. Ist die Transportzeit größer als die vorhandene Zeitspanne, müssen die abhängigen Einheiten verschoben werden. Hier ist darauf zu achten, daß alle Nachfolgeneinheiten auf diesem Gerät und die davon abhängige Einheiten auch verschoben werden (Bild 61).

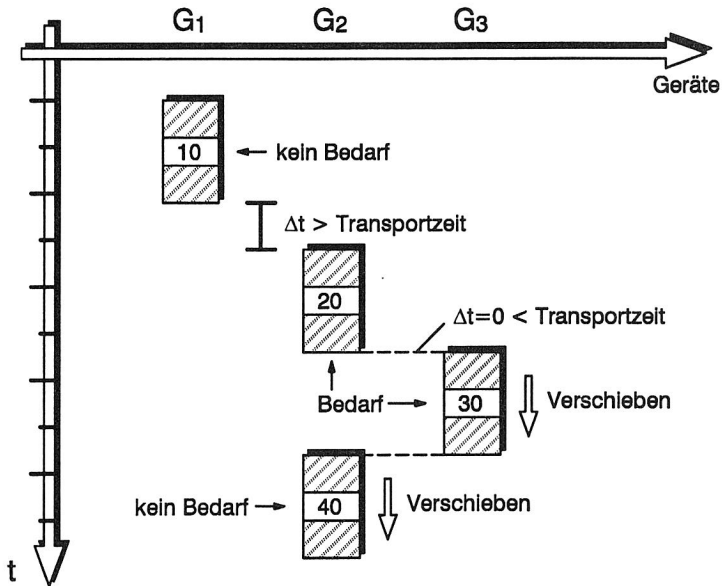


Bild 61: Einfügen der sekundären Operationen in den Belegungsplan

Die sekundären Operationen werden in den durch Verschiebungen modifizierten Maschinenbelegungsplan eingetragen und wie Planungseinheiten behandelt. Der Maschinenbelegungsplan selbst kann als **Anlagenfahrplan** angesehen werden.

6.5 Überwachungskomponente

Die Ergebnisse aus der Planung und Steuerung können nun in der Fertigung umgesetzt werden. Für die Überwachungskomponente, die diese Umsetzung bewerkstelligt, spielt es keine Rolle, ob eine primäre oder sekundäre Operation eingeplant ist. Sie "sieht" nur Operationen, deren Ausführungen zum richtigen Zeitpunkt an der richtigen Gerätesteuerung gewährleistet sein müssen. Ihr kommt somit eine Art Schedulerfunktion zu.

Die derzeit verfügbare Software, die für solche Anforderungen in flexiblen Fertigungseinrichtungen eingesetzt wird, ist ausnahmslos speziell auf das System zugeschnitten. Für zukünftige Aufgaben wird eine Konzeption zu wählen sein, die folgende Voraussetzungen erfüllt /98/:

- leichte Adaptierbarkeit
- möglichst hohe Hardwareunabhängigkeit
- einfache Schnittstellen
- gute Kommunikationsstruktur

6.5.1 Kommunikationsstruktur

Bei einer Kommunikationsstruktur für derartige Anwendungen kann im wesentlichen zwischen einer hierarchischen und einer zentralen Struktur unterschieden werden. Die Überwachung im Leitsystem dieses Werkstattsteuerungskonzepts wird beiden Modellen gerecht.

So kann die Kommunikation vom Überwachungsrechner aus direkt mit der Gerätesteuerung erfolgen oder mit einem Rechner, der seinerseits wieder einen bestimmten Systemausschnitt, wie zum Beispiel eine Zelle oder eine Station, überwacht. Dadurch können beliebige Hierarchiestrukturen berücksichtigt werden.

Der Datentransfer kann über sogenannte Telegramme erfolgen, die standardisiert sein sollten. Der Inhalt muß vom Kommunikationspartner zu interpretieren sein. Beispiele können sein:

- Anstoß einer Bearbeitung,
- Rückmeldung,
- Statusmeldung u.a.

Als Kommunikationspartner gelten Systemkomponenten oder Rechenanlagen, mit denen die Kommunikation über eine definierte Schnittstelle möglich ist. Auf der Überwachungsseite wird zudem ein eigenständiger Prozeß für jeden Kommunikationspartner kreiert, über den die Koordination läuft. Dadurch wird eine Entkopplung von der operativen Ebene des eigentlichen Fertigungssystems erreicht (Bild 62).

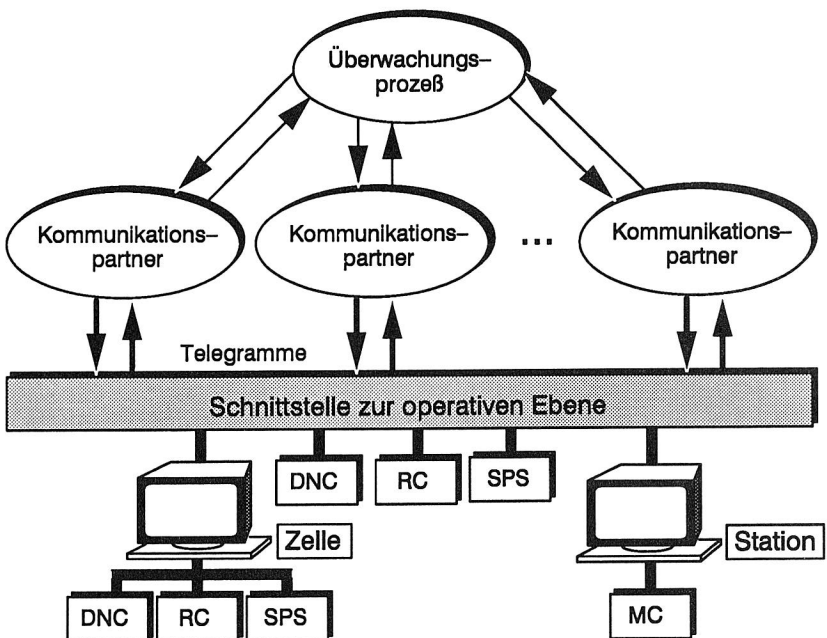


Bild 62: Kommunikationsstruktur in der Überwachung

Welcher Kommunikationsmechanismus zwischen der Überwachung und dem Partner explizit eingesetzt wird, hängt hauptsächlich von den gestellten Anforderungen ab. Der auszuwählende Mechanismus muß folgenden Ablauf verwirklichen können:

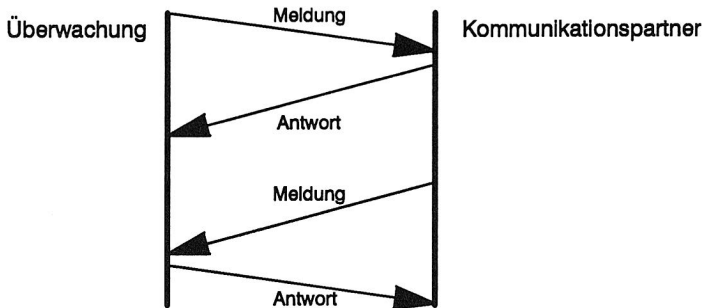


Bild 63: Kommunikationsmechanismus in der Überwachung

Zur Koordination stellt die Überwachung durch den Belegungsplan fest, daß eine Operation zur Bearbeitung ansteht. Dazu wird der Plan in einem bestimmten Takt durchsucht. Dem Kommunikationspartner werden dann die nötigen Informationen übergeben. Er quittiert die Übermittlung. Anhand der Planungseinheit muß der Partner in der Lage sein, das richtige Steuerprogramm anzustoßen. Während der Ausführung oder danach sendet der Partner den Zustand an den Überwachungsprozeß, der ihn seinerseits quittiert. Somit ergeben sich für diese Anwendung im wesentlichen folgende Nachrichteninhalte:

- Kreieren eines Partnerprozesses (und Rückmeldung)
- Ausführung einer Planungseinheit (und Rückmeldung)
- Ausführungsmeldung (und Rückmeldung)
- Fehlermeldung bzw. Störungsmeldung (und Rückmeldung)
- Terminieren eines Partnerprozesses (und Rückmeldung)

Die getrennte Realisierung von <Meldung> und <Rückmeldung> ist durch den Kommunikationsmechanismus zwischen den Prozessen begründet. Dieser wird

durch das zugrundegelegte Betriebssystem bestimmt, wobei es sich um das multi-tasking-fähige Unix™ handelt.

Obwohl Unix™ keine Echtzeitfähigkeit besitzt, reicht die Quasi-Parallelität des Betriebssystems für diese Anforderungen aus. Zeitkritische Kommunikation wie zum Beispiel Interrupts oder Alarmmeldungen können "weiter unten" abgefangen werden. Die Zeitscheibentechnik, nach der die Prozesse der Verarbeitungseinheit des Rechners zugeteilt werden und die von außen nicht manipulierbar ist, zeigt sich hinreichend geeignet für die Koordinationsaufgabe der Überwachung, da die Verzögerungszeiten, die beim Absenden eines Auftrags (Starten einer Planungseinheit) entstehen können, im Vergleich zur Laufzeit dieses Auftrags verschwindend gering sind.

Unter Unix™ stehen eine Reihe von Mechanismen für die Prozeßkommunikation zur Verfügung, von denen die wichtigsten kurz aufgeführt sind:

Signale

Sie stellen ein asynchrones Ereignis dar und bewirken einen Anstoß oder eine Unterbrechung eines Prozesses. Der Prozeß hat die Möglichkeit, ein Signal explizit abzufangen. Ihr Hauptanwendungsgebiet ist in der Prozeßsynchronisation zu sehen.

Semaphore

Unter Semaphoren werden Zustandsvariablen verstanden, dessen Werte von den Prozessen abgefragt werden können. Gleichzeitiger Zugriff auf mehrere Semaphoren ist ebenso gestattet wie das Warten darauf, daß eine Semaphore einen bestimmten Wert annimmt. Auch sie dienen hauptsächlich der Prozeßsynchronisation.

Pipes

Pipes sind als gemeinsamer unidirektionaler Puffer zwischen Prozessen anzusehen. Voraussetzung ist der Aufbau der Pipe mittels eines Systemaufrufs durch den Vaterprozess und Erzeugung zweier Sohnprozesse, die die entsprechenden Dateien erben. Auf diesen Dateien können dann Operationen ausgeführt werden.

Mit dem Ausbau zu sogenannten Named Pipes kann die Existenz eines Vaterprozesses umgangen werden.

Gemeinsamer Speicher (Shared Memory)

Dieses Konzept umfaßt die Nutzung eines gemeinsamen Speicherbereichs durch mehrere Prozesse. Der Nachrichteninhalt der Kommunikation wird temporär gehalten. Die Zugriffe können durch Semaphoren oder Signale synchronisiert werden.

Nachrichten (Messages)

Hierbei werden die Nachrichten an sogenannte Nachrichtenspeicher (Message Queues) gesendet. Von dort können sie dann abgeholt werden. Die Nachrichten können nach einem bestimmten Typ differenziert werden.

In Bild 64 ist die Geschwindigkeit und ein wesentliches Merkmal der einzelnen Mechanismen aufgeführt.

Mechanismus	Geschwindigkeit	Merkmale
Signale	sehr hoch	Nutzinformation 1 bit
Semaphore	sehr hoch	Nutzinformation 1 bit
Pipes	mittel	systemabhängige Nutzinformation
Shared Memory	sehr hoch	keine Synchronisation
Messages	mittel	Typdifferenzierung der Nachricht möglich

Bild 64: Prozeßkommunikationsmöglichkeiten unter Unix™

Für die Koordinierungsaufgaben der Überwachung wurde das Konzept der Message Queues verwendet. Als Operationen sind das Senden und Empfangen von Nachrichten zu nennen. Das Warten auf eine Nachricht ist durch zyklisches Nachfragen in der Message-Queue realisiert. Der Überwachungsprozeß und jeder Kommunikationspartner besitzen dabei eine eigene Queue.

6.5.2 Selbstregelung

Die Arbeitsweise der Werkstattsteuerung, wie sie in dieser Arbeit dargestellt und realisiert wurde, ermöglicht es, einen Auftragspool einzuplanen (Planung), neben den primären Operationen auch die Transport- und Handhabungsfunktionen für das Material und die Fertigungshilfsmittel zu berücksichtigen (Steuerung) und den erweiterten Maschinenbelegungsplan in der Fertigung umzusetzen (Überwachung). Somit ist ein optimaler Ablauf mittels Rechneinsatz zur automatisierten Planung und Koordinierung gewährleistet.

Während der Fertigung kann es natürlich vorkommen, und die Praxis beweist dies, daß aufgrund von Störungen der "optimale" Plan nicht umgesetzt werden kann. Störungen treten im Fertigungsverlauf immer wieder auf und können verschiedenste Ursachen haben:

- Geräteausfall,
- Ausfall von anderen technischen Einrichtungen,
- Werkzeugbruch,
- usw.

So unterschiedlich die Störungsursache sein kann, so unterschiedlich ist auch die Störungsdauer. Dies hat dann Auswirkungen auf den umzusetzenden Plan. Es treten Verzögerungen auf, und geplante Termine können eventuell nicht mehr eingehalten werden.

Störungen müssen also durch die Überwachungskomponente behandelt werden. Beim Eintreten einer Störung kann je nach Situationslage verschieden reagiert werden.

Abfangen kurzer Störungen

Prinzipiell kann davon ausgegangen werden, daß ein einmal berechneter Plan infolge seiner guten Qualität solange wie möglich beizubehalten ist. Zeitlich kurze Störungen werden daher durch stetige Anpassung des aktuellen Plans kompensiert.

Erreicht also eine Störungsmeldung die Überwachung, so verschiebt diese die Operationen, die durch den Störungsverursacher beeinträchtigt werden. De facto sind das bei einem Geräteausfall alle Operationen, die ab diesem Zeitpunkt für das Gerät eingeplant sind. Zudem werden auch alle davon abhängigen Operationen, wie zum Beispiel Nachfolgearbeitsgänge, verschoben. Dieser Vorgang wird solange fortgeführt, bis die Störung behoben ist oder eine Zeitschranke überschritten wird. Diese Zeitschranke kann systemabhängig definiert werden (Bild 65). Sie sollte so gewählt sein, daß kurze Störungen auch behoben werden können.

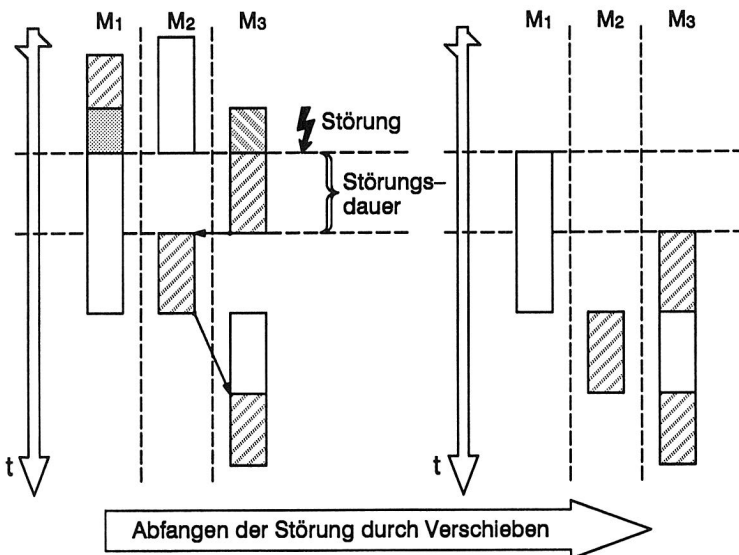


Bild 65: Abfangen kurzer Störungen

Realisiert ist diese Funktionalität in der Form, daß die Überwachung in einem bestimmten Takt bei dem Kommunikationspartner, bei dem die Störung aufgetreten ist, nachfragt, ob eine Behebung durchgeführt wurde (Soll – Ist – Abgleich).

Abfangen langer Störungen

Kann eine Störung innerhalb der Schranke nicht behoben werden, wird ein neuer Planungslauf angestoßen. Mittels der Planungsauswahl und der –distanz kann dabei ganz individuell auf zeitlichen Anforderungen aus der Fertigung eingegangen werden. Es wird versucht, so schnell wie möglich einen neuen Plan zu erzeugen, da lange Ausfallzeiten zu vermeiden sind (Bild 66).

Diese Mechanismen zur Störungsbehebung bieten im Informationssystem zur **Werkstattsteuerung** die Möglichkeit der **Selbstregelung**. Auf veränderte Situationen in der Fertigung kann adäquat reagiert werden. Durch einen Regelkreis ist gesichert, daß auf jede Veränderung die richtige Reaktion folgt.

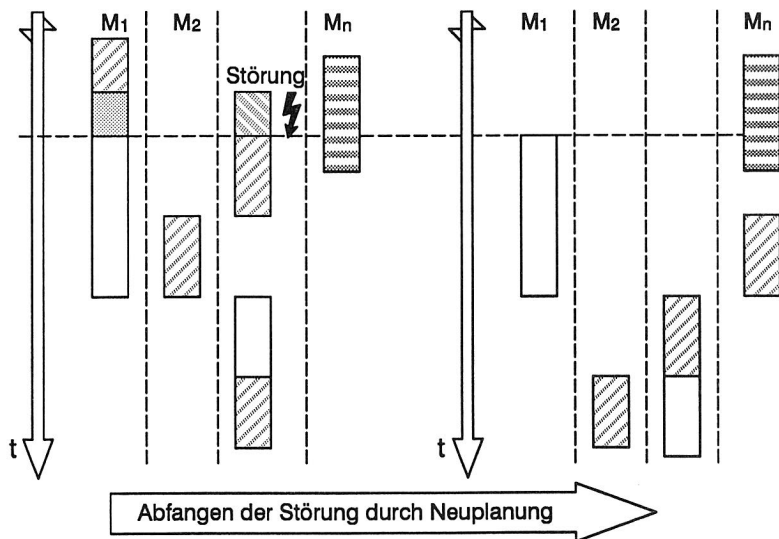


Bild 66: Abfangen langer Störungen

7 Fertigungsbeispiel in einer Testumgebung

Umfangreiche Softwareprojekte können ebenso wie andere Projekte in einzelne Phasen gegliedert werden. In diesem Zusammenhang wird auch von einem Softwarelebenszyklus gesprochen, der sich wie folgt unterteilen läßt /99/:

- *Analyse:* Bestimmung der Funktionalitäten
- *Spezifikation:* Anforderungsdefinition
- *Entwurf:* Festlegung der Realisierung
- *Implementierung:* Übertragung in eine Programmiersprache
- *Test:* Sicherstellung der geforderten Anforderungen

Die Testphase bildet dabei einen wesentlichen Teil des Ganzen. Der Softwaretest muß den Beweis erbringen, daß der gewählte Ansatz den gewünschten Erfolg herbeiführt. Zudem können während der Testphase Fehler in der Software aufgedeckt und beseitigt werden.

Für Softwarekonzepte, die größere Systeme zum Gegenstand der Betrachtung haben, wie es hier mit flexiblen Fertigungseinrichtungen der Fall ist, erweist sich das Testen als äußerst schwierig. In realen Systemen kann die Fertigung für ein solches Vorhaben nicht unterbrochen werden. Doch auch Laborumgebungen sind für den Aufbau derartiger Systeme, in dem die Funktionalität der Werkstattsteuerung auch ausgeschöpft werden kann, nicht geeignet.

Um aber nicht auf die Testphase verzichten zu müssen, wurde für diese Arbeit als Konsequenz ein Modell am Rechner geschaffen. Mit diesem Modell kann die reale Systemumgebung abgebildet werden. Es besteht aus den Komponenten zur Werkstattsteuerung, wie sie in den vorhergehenden Kapiteln beschrieben wurden. Auf der Seite der Überwachung wurde die Schnittstelle zur operativen Ebene simuliert.

Zur Validierung der Werkstattsteuerung wurde beispielhaft ein flexibles Fertigungssystem konzipiert. In diesem werden für eine Getriebebaureihe die

verschiedenen Wellen und Zahnräder gefertigt. Zusätzlich erfolgt die Montage der Zahnräder inklusive den Wälzlagern, Paßscheiben und Wellensicherungsringen. Zur Vorfertigung wird vereinfachend angenommen, daß die Verzahnungen nach dem Wälzfräsen fertig sind. Ein Zahnkantenverrunden wird nicht berücksichtigt.

Somit werden in dem flexiblen Fertigungssystem die Wellen zur Montage in das Getriebegehäuse vorbereitet. In anderen Systemen, die nicht näher untersucht wurden, werden die Gehäuse und Abschlußdeckel gefertigt, bevor die Baugruppen und Einzelteile komplett montiert werden.

Die Getriebeherstellung kann im Gefüge des Gesamtunternehmens hierarchisch geschichtet sein. Eine Softwarekonzeption, wie sie hier vorgestellt wurde, unterstützt eine derartige Struktur (Bild 67).

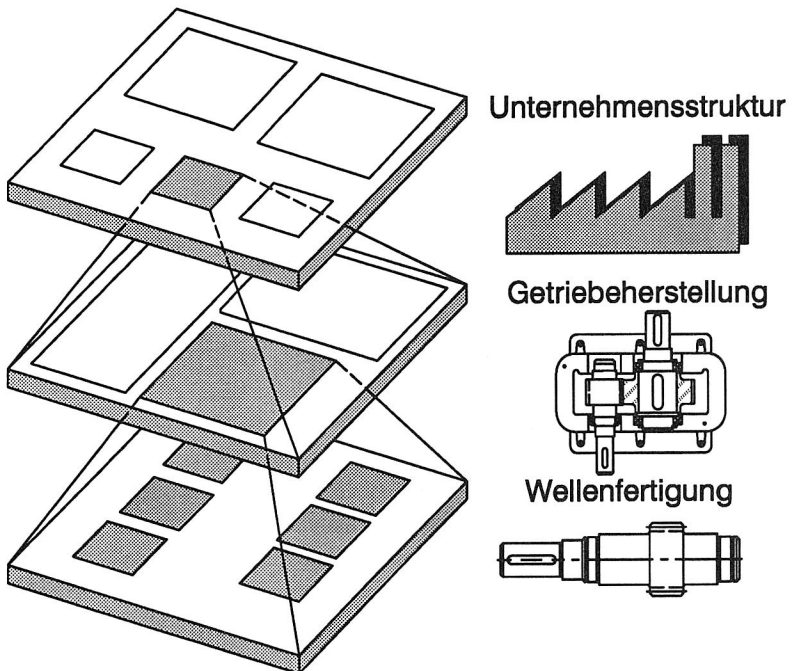


Bild 67: Beispielhafte Struktur in einem Unternehmen

7.1 Produktspektrum

7.1.1 Getriebevarianten

Für das 1-stufige Getriebe werden zwei Varianten, die sich im übertragbaren Drehmoment unterscheiden, ins Auge gefaßt. Die kleine Baureihe besitzt ein Eingangsdrehmoment von 45 Nm, die große von 60 Nm. Zudem existieren zu jeder Drehmoment-Variante noch unterschiedliche Übersetzungsverhältnisse:

- 45 Nm (1:2, 1:3, 2:3)
- 60 Nm (1:2, 1:3, 2:3)

Eine Getriebevariante ist in Bild 68 zu sehen.

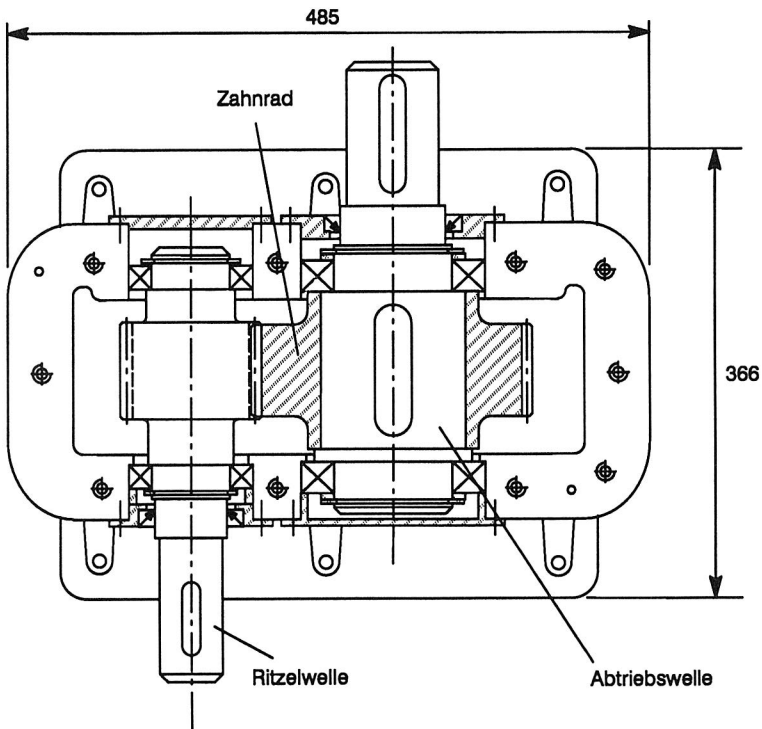
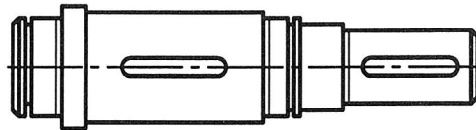


Bild 68: 1-stufiges Getriebe, Variante 60 Nm – Übersetzung 1:2

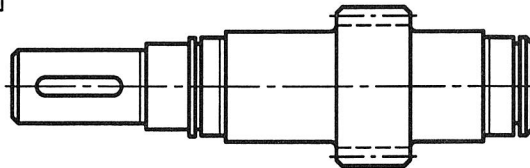
7.1.2 Ritzelwelle, Abtriebswelle und Zahnrad

In dem zu betrachtenden Fertigungssystem werden also die Ritzelwelle, die Abtriebswelle und das Zahnrad für das Getriebe vorgefertigt und montiert (Bild 69).

Abtriebswelle



Ritzelwelle



Zahnrad

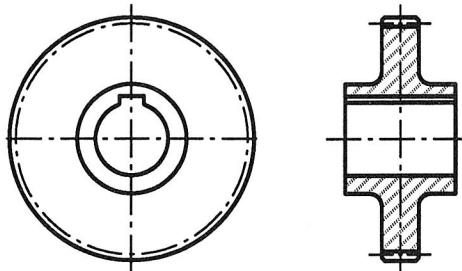


Bild 69: Abtriebswelle, Ritzelwelle und Zahnrad

Die Ritzelwelle ist eine mehrstufig abgesetzte Welle, deren Gesamtlänge, Wellendurchmesser und Breite drehmomentabhängig sind, während die Maße der Verzahnung drehmoment- und Übersetzungsabhängig sind. Über die Paßfederver-

bindung wird das Drehmoment eingeleitet und über die Verzahnung übertragen. Als Werkstoff wird 16 MnCr 5 aufgrund seiner guten Zerspanbarkeit und niedrigen Werkstoffkosten eingesetzt.

Auch die Abtriebswelle ist eine mehrfach abgestufte Welle. Sie besitzt jedoch im Gegensatz zur Ritzwelle keine Übersetzungsabhängigen Maße. Wegen der niedrigen Belastung wird der Baustahl St 60 als Werkstoff verwendet. Die Drehmomentübertragung zwischen Zahnrad und Welle erfolgt über die Paßfederverbindung Z, die Übertragung am Abtrieb über A. Das Zahnrad besteht ebenfalls aus St 60.

7.2 Flexibles Fertigungssystem

7.2.1 Zellenstruktur

Das flexible Fertigungssystem¹⁾ läßt sich in einzelne Zellen unterteilen:

- DZ: Drehzellen 1–3, bestehend aus einer Drehmaschine mit lokalem Werkzeugspeicher für maximal 14 Werkzeuge im Revolver, einem Portalroboter und einem Palettenwechselsystem mit 6 Aufnahmeplätzen
- RZ: Räumzelle, bestehend aus einem Räumzentrum mit Drehmagazin für 6 Räumnadeln und Räumnadelzuführeinrichtung, einem 6–Achsen–Industrieroboter als Handhabungssystem und einem Palettenwechselsystem mit 6 Aufnahmeplätzen
- WFZ: Wälzfräszelle, bestehend aus einem Wälzfräszentrum mit lokalem Speicher für 4 Wälzfräser und 3 Fräsdornaufnahmen, einem Portalroboter und einem Palettenwechselsystem mit 6 Aufnahmeplätzen
- MZ: Montagezelle, bestehend aus einem 6–Achsen–Industrieroboter, einem Palettenwechselsystem mit 6 Aufnahmeplätzen

¹⁾ Bei der Konzeption des flexiblen Fertigungssystems und der Auswahl des Produktspektrums inklusive der fertigungstechnischen Zeichnungen wurde der Verfasser von cand.–Ing. J. Knoblach unterstützt.

Die einzelnen Maschinen werden über die Transportwagensysteme mit Werkzeugen aus der zentralen Werkzeuginstandhaltung versorgt. Die Werkstücke können an den beiden Beladeplätzen ins System eingebracht werden. Zusätzliche Montageobjekte (Läger, Sicherungsringe und Paßscheiben) werden direkt an der Montagezelle aus dem Kommissioniersystem eingeschleust. Außerdem können noch nicht fertig bearbeitete Teile in einem Lagersystem zwischengepuffert werden (Bild 70).

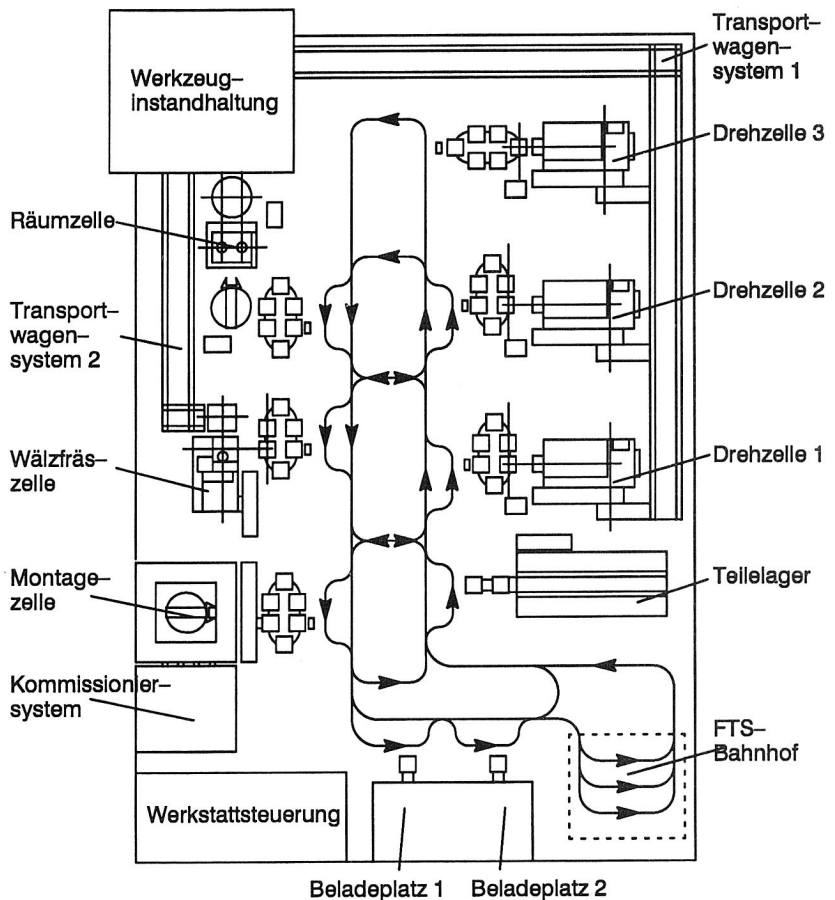


Bild 70: Layout des flexiblen Fertigungssystems

7.2.2 Fertigungshilfsmittel

Eine vollständige datentechnische Behandlung der Werkzeuge und anderer Fertigungshilfsmittel kann in einem separaten Teilbereich der Werkstattsteuerung, wie zum Beispiel einer Betriebsmittelverwaltung, erfolgen. Hier sollen die relevanten Werkzeuge nur aufgezählt werden, da sie auch in den Arbeitsplänen Eingang finden. Es werden folgende Werkzeuge benötigt:

HM1: Universal Hartmetall–Wendeschneidplatte für 16 MnCr 5

HM2: Universal Hartmetall–Wendeschneidplatte für St 60

HM3: Hartmetall–Innendrehmeißel für St 60

SK1: Universal Keramik–Wendeschneidplatte für 16 MnCr 5

SK2: Universal Keramik–Wendeschneidplatte für St 60

SK3: Keramik–Innendrehmeißel für St 60

E1: Abstech–Drehmeißel Breite 2.15 mm

E2: Abstech–Drehmeißel Breite 3.15 mm

F1: Freistich–Formwerkzeug DIN 509 F 0,6 * 0,3

F2: Freistich–Formwerkzeug DIN 509 F 1 * 0,4

N1: HM–Fingerfräser d = 14 mm, z = 4

N2: HM–Fingerfräser d = 16 mm, z = 4

N3: HM–Fingerfräser d = 20 mm, z = 4

N4: HM–Fingerfräser d = 22 mm, z = 6

N5: HM–Fingerfräser d = 28 mm, z = 6

N6: HM–Fingerfräser d = 32 mm, z = 6

R1: Räumnadel für Nabe d = 110 mm, Paßfeder b = 28 mm

R2: Räumnadel für Nabe d = 120 mm, Paßfeder b = 32 mm

W1: Wälzfräser für Modul m = 4 mm

W2: Wälzfräser für Modul m = 5 mm

G1: Greifer für Wälzlager

G2: Greifer für Paßscheiben

G3: Greifer für Wellensicherungsringe

G4: Greifer für Zahnrad und Paßfeder

7.2.3 Arbeitspläne

Stellvertretend für alle anderen sollen die Arbeitspläne für das Zahnrad und die Wellen in der Drehmomentsausprägung von 45 Nm (Übersetzungsverhältnis 1:2)

exemplarisch dargestellt sein. Sie enthalten die in Kapitel 6.2.2 geforderten Geräte- und Fertigungshilfsmittelalternativen. Die Angabe der Zeiten (Rüst- und Hauptzeit) erfolgt in Sekunden (Bild 71 bis Bild 73).

Benennung: Ritzelwelle 45 Nm 1:2		Datum:		ARBEITSPLAN	
		Bearbeiter:			
Auftrags-Nr.:		Stückzahl:		Termin:	
Werkstoff: 16 Mn Cr 5		Abmessungen: Ø100 x 340		Rohgewicht: 20.8 kg	
				Fertiggewicht:	
Nr.	Beschreibung	Gerätegruppe	Fertigungs- mittel	Rüstzeit	Haupt- zeit
10	Kontur drehen (rechts)	DZ1 DZ2 DZ3		15	
	– mit Hartmetall		HM1,E1,F1		121
	– mit Schneidkeramik		SK1,E1,F1		97
20	Kontur drehen (links)	DZ1 DZ2 DZ3		15	
	– mit Hartmetall		HM1,E1,F1		609
	– mit Schneidkeramik		SK1,E1,F1		417
30	Paßfedernut M fräsen	DZ1 DZ2	N1	5	21
40	Verzahnung wälzfräsen	WFZ	W1	10	340
50	Wälzlager links fügen	MZ	G1	15	62
60	Paßscheibe links fügen	MZ	G2	15	58
70	Sicherungsring links fügen	MZ	G3	15	60
80	Wälzlager rechts fügen	MZ	G1	15	62
90	Paßscheibe rechts fügen	MZ	G2	15	58
100	Sicherungsring rechts fügen	MZ	G3	15	60

Bild 71: Arbeitsplan der Ritzelwelle

Benennung: Abtriebswelle 45 Nm		Datum:		ARBEITSPLAN	
		Bearbeiter:			
Auftrags-Nr.:		Stückzahl:		Termin:	
Werkstoff: St 60		Abmessungen: Ø110 x 341		Rohgewicht: 25.3 kg	
				Fertiggewicht:	
Nr.	Beschreibung	Gerätegruppe	Fertigungs- mittel	Rüstzeit	Haupt- zeit
10	Kontur drehen (links)	DZ1 DZ2 DZ3		15	
	– mit Hartmetall		HM2,F2,E2		105
	– mit Schneidkeramik		SK2,F2,E2		81
20	Kontur drehen (rechts)	DZ1 DZ2 DZ3		15	
	– mit Hartmetall		HM2,F2,E2		669
	– mit Schneidkeramik		SK2,F2,E2		484
30	Paßfedernut A fräsen	DZ1 DZ2	N3	5	36
40	Paßfedernut Z fräsen	DZ1 DZ2	N5	5	44

Benennung: Montage Abtriebswelle		Datum:		ARBEITSPLAN	
		Bearbeiter:			
Auftrags-Nr.:		Stückzahl:		Termin:	
Werkstoff:		Abmessungen:		Rohgewicht:	
				Fertiggewicht:	
Nr.	Beschreibung	Gerätegruppe	Fertigungs- mittel	Rüstzeit	Haupt- zeit
10	Fügen Wälzlager links	MZ	G1	15	62
20	Fügen Paßscheibe links	MZ	G2	15	58
30	Fügen Sicherungsring li.	MZ	G3	15	60
40	Fügen Zahnrad mit Paßf.	MZ	G4	15	90
50	Fügen Wälzlager rechts	MZ	G1	15	62
60	Fügen Paßfeder rechts	MZ	G2	15	58
70	Fügen Sicherungsring re.	MZ	G3	15	60

Bild 72: Arbeitspläne zur Fertigung und Montage der Abtriebswelle

Benennung: Zahnrad 45 Nm 1:2		Datum:		ARBEITSPLAN	
		Bearbeiter:			
Auftrags-Nr.:		Stückzahl:		Termin:	
Werkstoff: St 60		Abmessungen: Ø200 x Ø90 x 132		Rohgewicht: 25.8 kg	
				Fertiggewicht:	
Nr.	Beschreibung	Gerätegruppe	Fertigungs- mittel	Rüstzeit	Haupt- zeit
10	Kontur drehen (rechts)	DZ1 DZ2 DZ3		5	
	– mit Hartmetall		HM2		203
	– mit Schneidkeramik		SK2		133
20	Kontur drehen (links)	DZ1 DZ2 DZ3		5	
	– mit Hartmetall		HM2		128
	– mit Schneidkeramik		SK2		82
30	Innenloch ausdrehen	DZ1 DZ2 DZ3		5	
	– mit Hartmetall		HM3		60
	– mit Schneidkeramik		SK3		38
40	Verzahnung wälzfräsen	WFZ	W1	10	440
50	Paßfedernut räumen	RZ	R1,R2	20	180

Bild 73: Arbeitsplan des Zahnrad

7.3 Rechnerintegrierte Werkstattsteuerung

Da das flexible Fertigungssystem in einzelne Zellen aufgeteilt ist, bietet sich für eine informationstechnische Betrachtung eine Zweiteilung an. Es wird dabei zwischen einer Zellen- und einer Leitebene unterschieden. Die Objektklassifikation, wie sie in Kapitel 5.3.2 definiert wurde, ermöglicht dies. Bild 74 zeigt die Softwarestruktur für das vorliegende flexible Fertigungssystem.

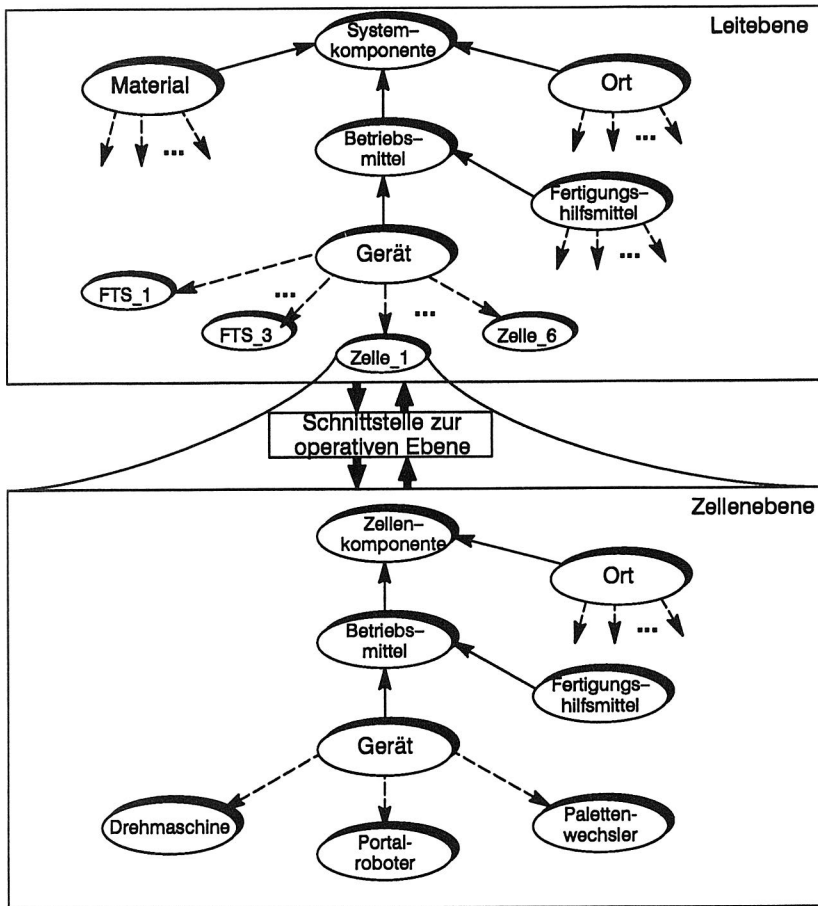


Bild 74: Hierarchische Softwarestruktur des flexiblen Fertigungssystems

Auf Zellenebene ist der Zellenrechner für die gesamte Koordination verantwortlich. Im wesentlichen sind das:

- Ablaufsteuerung und Überwachung der Geräte in der Zelle
- Bereitstellung der erforderlichen Programme (NC, RC, etc.)
- Kommunikation mit dem Leitrechner

Die objektorientierte Softwarestruktur kann direkt zur Steuerung der Zellen eingesetzt werden. Als Wissensbasis fungiert dann ein semantisches Netz, welches die jeweilige Zelle beschreibt. Für die Werkstattsteuerung wiederum regelt der Leitreechner den übergeordneten Ablauf. Ihm erscheinen die Zellen als autonome Einheiten, die einen Auftrag zur Bearbeitung erhalten und über die Ausführung Bericht erstatten. Die Bewerksstellung bleibt dem Zellenrechner überlassen.

So wurde im Rahmen dieser Arbeit der Prototyp eines Softwaresystems zur rechnerintegrierten Werkstattsteuerung realisiert¹⁾. Dem Benutzer steht dabei ein interaktives Werkzeug zur Verfügung, mit dem zwei Zielsetzungen verfolgt werden. Zum einen kann die Wissensbasis **individuell für jede Fertigungseinrichtung** konfiguriert werden. Es werden Editoren zur Generierung der Ablaufgraphen, der Transportnetze und des Systemlayouts mit entsprechenden Ein- und Ausgabemechanismen bereitgestellt. Zum anderen können die Inferenzkomponenten zur eigentlichen Werkstattsteuerung (Planung, Steuerung und Überwachung) aktiviert werden. Dazu laufen die jeweiligen Algorithmen ab, wie sie in den vorgangenen Kapiteln spezifiziert wurden. Das Menüfenster hierzu ist in Bild 75 zu sehen.

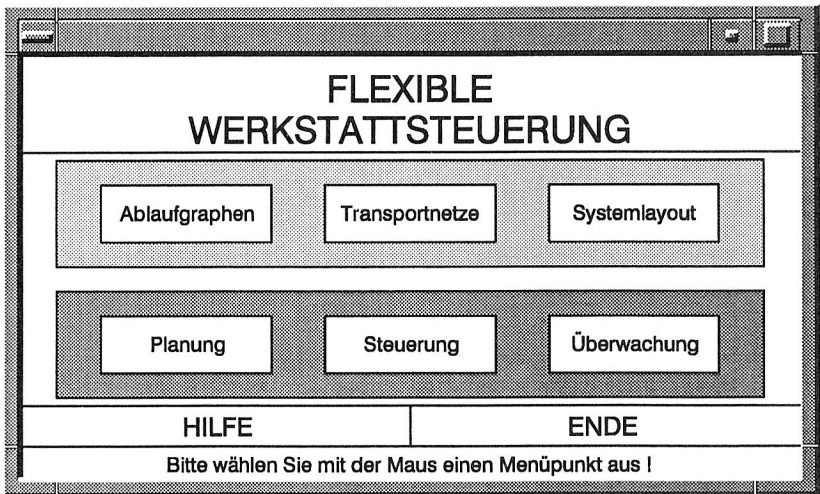


Bild 75: Menüfenster zur flexiblen Werkstattsteuerung

¹⁾ Bei den Implementierungsarbeiten wurde der Verfasser von den Herrn Dipl.- Inf. J. Rohde /100/ und Herrn cand. - inf. C. Seggewies unterstützt

7.4 Anwendungsspezifische Konfiguration der Wissensbasis

7.4.1 Ablaufgraphen

Im Ablaufgraphen besteht die Möglichkeit, mehrere Arbeitsgänge zu einer Planungseinheit zusammenzufassen. So kann zum Beispiel erreicht werden, daß Arbeitsgänge, die nacheinander auf derselben Maschine gefertigt werden sollen, sinnvollerweise als eine Einheit betrachtet und als solche auch verplant werden. Für die Ritzelwelle ist der Ablaufgraph in Bild 76 illustriert, wobei sich hinter den Planungseinheiten folgende Arbeitsgänge aus dem Arbeitsplan verbergen:

10:	Arbeitsgang 10:	Ritzelwelle: Kontur drehen (rechts)
	Arbeitsgang 20:	Ritzelwelle: Kontur drehen (links)
	Arbeitsgang 30:	Ritzelwelle: Paßfedernut fräsen
20:	Arbeitsgang 40:	Ritzelwelle: Verzahnung wälzfräsen
30:	Arbeitsgang 50 – 100:	Ritzelwelle: Montage

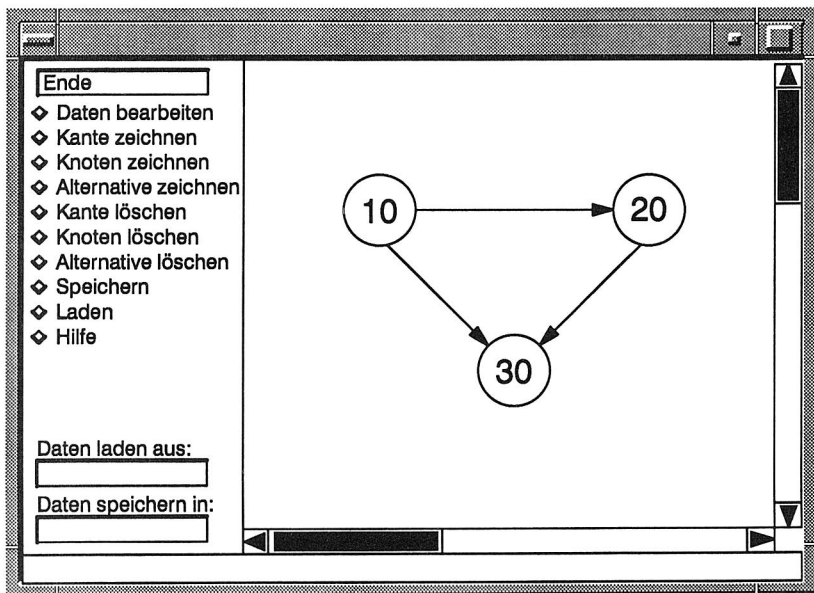


Bild 76: Menüfenster zum Ablaufgraphen der Ritzelwelle

Für die Abtriebswelle ist der Ablaufgraph in Bild 77 wiedergegeben. Die Planungseinheiten setzen sich aus den nachstehend aufgeführten Arbeitsgängen zusammen, welche den drei unterschiedlichen Arbeitsplänen entnommen sind:

- | | | |
|-----|----------------------|------------------------------------|
| 10: | Arbeitsgang 10: | Abtriebswelle: drehen (rechts) |
| | Arbeitsgang 20: | Abtriebswelle: drehen (links) |
| | Arbeitsgang 30 – 40: | Abtriebswelle: Paßfedernut fräsen |
| 20: | Arbeitsgang 10: | Zahnrad: Kontur drehen (rechts) |
| | Arbeitsgang 20: | Zahnrad: Kontur drehen (links) |
| | Arbeitsgang 30: | Zahnrad: Innenloch drehen |
| 30: | Arbeitsgang 40: | Zahnrad: Verzahnung wälzfräsen |
| 40: | Arbeitsgang 50: | Zahnrad: Paßfedernut räumen |
| 50: | Arbeitsgang 10 – 70: | Abtriebswelle und Zahnrad: Montage |

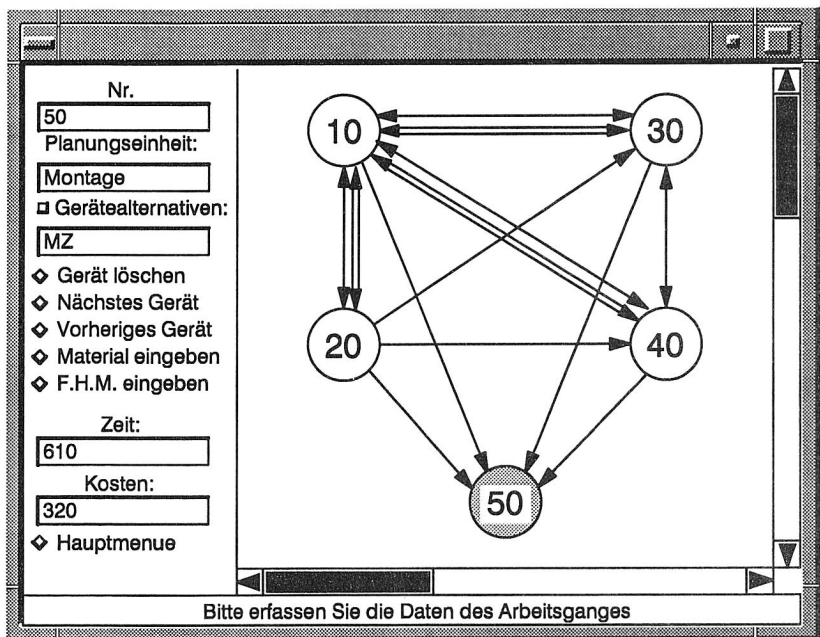


Bild 77: Menüfenster zum Ablaufgraphen der Abtriebswelle

7.4.2 Transportnetz

Der Material- und Werkzeugfluß ist im FFS getrennt. Während für die Werkzeuge ein Transportwagensystem zur Verfügung steht, werden die Werkstücke mittels fahrerloser Transportsysteme (FTS) zu den Zellen gebracht (Bild 78).

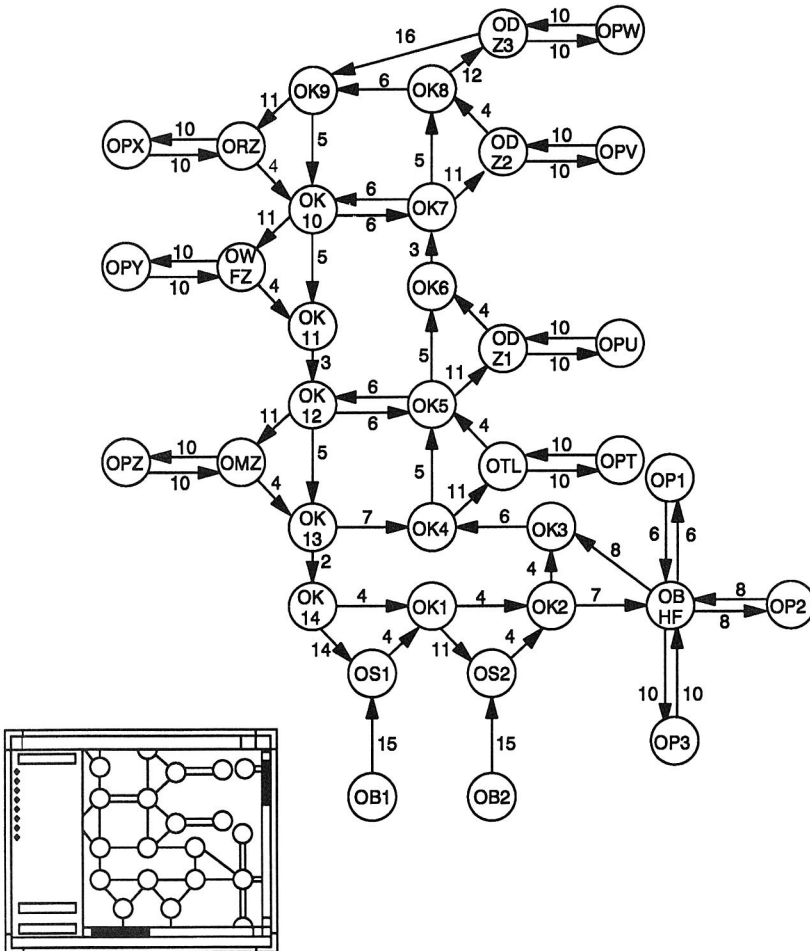


Bild 78: Transportnetz für den Materialfluß

Die für das Transportnetz relevanten Orte ergeben sich direkt aus den elementaren Fahrstrecken. Die Kantenbeschriftung gibt die jeweilige Ausführungsdauer der Operation an. Zusätzliche Informationen, die zur Routenwahl notwendig sind, wie zum Beispiel benötigte Hilfsmittel oder mögliche Transportgeräte, werden mit dem Editor zur Initialisierung der Wissensbasis eingegeben.

7.4.3 Systemlayout

Alle Geräte, welche der Überwachungsprozeß verwalten soll, müssen der Wissensbasis bekanntgemacht werden. Hierzu steht ebenfalls ein Editor zur Verfügung, der standardmäßig unterschiedliche Gerätetypen am Bildschirm erzeugen kann.

Gleichzeitig kann damit auch das Layout der jeweiligen Fertigungseinrichtung generiert werden. Während die Überwachung läuft, besteht somit die Möglichkeit einer **Online-Prozeßüberwachung**. Die Piktogramme erscheinen je nach Zustand des Gerätes (aktiv, inaktiv, gestört) in unterschiedlichen Farben. Auf den verschiedenen Stationen werden die aktuellen Operationen angezeigt (Bild 79).

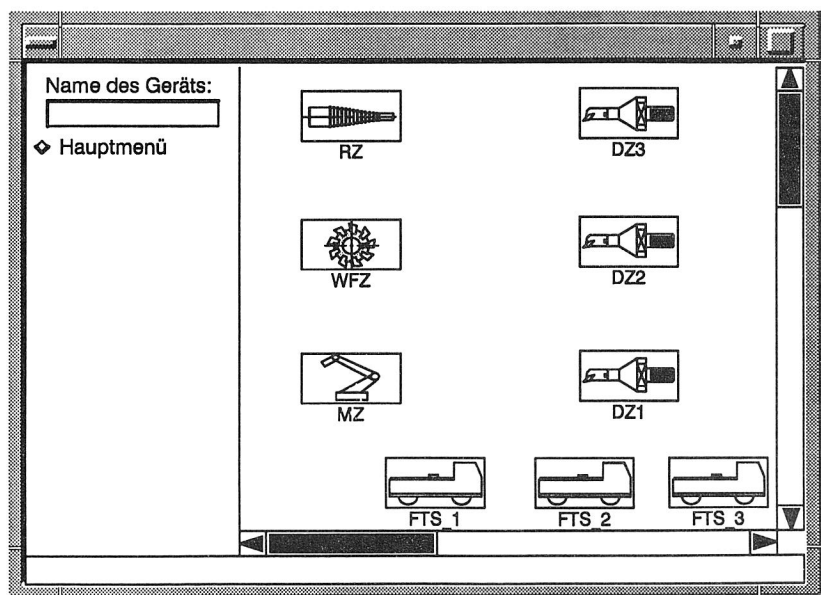


Bild 79: Menüfenster zum Systemlayout

7.5 Inferenzkomponenten – Ergebnisse und Bewertung

Mit dem vorhandenen Produktspektrum wurden zahlreiche Tests hinsichtlich der Funktionalität der Werkstattsteuerung durchgeführt. Aus diesem Spektrum kann hier jedoch nur eine begrenzte Auswahl vorgestellt werden. Somit soll anhand von drei unterschiedlichen Beispielen die Arbeitsweise illustriert und verifiziert werden.

7.5.1 Auftragsspektrum 1

Zur Disposition stehen folgende Aufträge aus dem Auftragspool:

Nr.1:	Ritzelwelle 45 Nm (1:3)	Losgröße: 1	Priorität: 1
Nr.2:	Ritzelwelle 60 Nm (1:3)	Losgröße: 1	Priorität: 3
Nr.3:	Abtriebswelle 45 Nm (1:3)	Losgröße: 1	Priorität: 3
Nr.4:	Abtriebswelle 60 Nm (1:3)	Losgröße: 1	Priorität: 3

Die Optimierungskriterien sind mit folgenden Gewichten versehen:

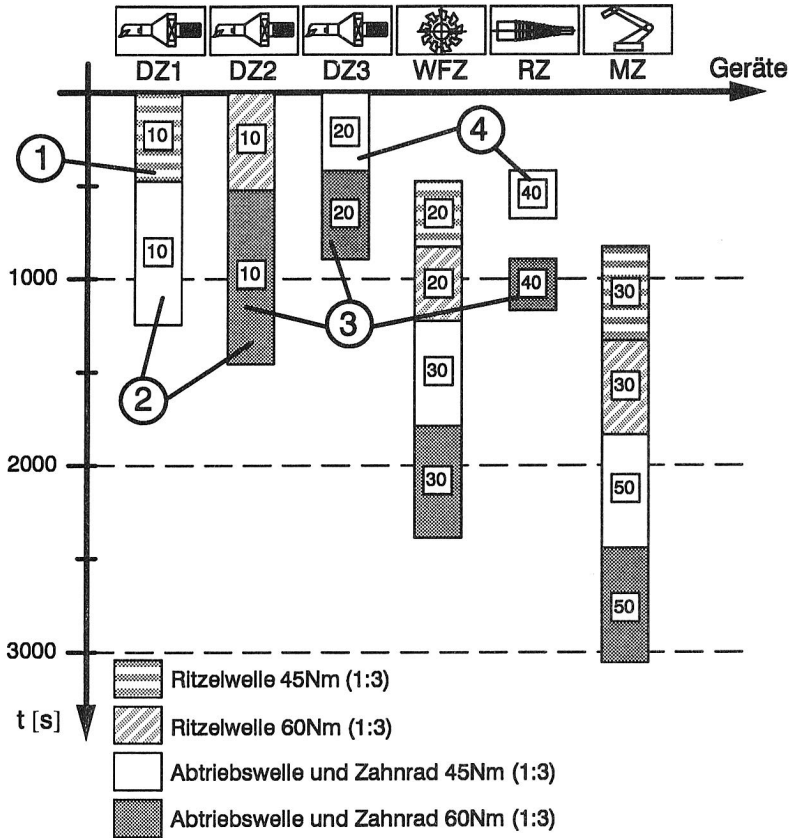
minimale Bearbeitungszeiten:	2
niedrige Kosten:	0
kurze Durchlaufzeiten:	3
hohe Auslastung bzw. wenig Stillstand:	5
geringe Rüstzeiten:	0

Die Planung erfolgt gemäß unten stehender Stellschrauben:

Planungsdistanz:	4
Planungsauswahl:	18 (=Maximum)

Der daraus resultierende Belegungsplan ist in Bild 80 dargestellt. Dabel sind die Fähigkeiten des Softwaresystems zu erkennen:

- 1) Berücksichtigung der Priorität:
Der Auftrag mit der niedrigsten Priorität wird zuerst eingeplant.
- 2) Nutzung der Maschinenalternativen:
Die Gerätealternativen aus dem Arbeitsplan werden verwendet.
- 3) Parallele Einplanung von Arbeitsgängen:
Eine streng sequentielle Einplanung ist nicht zwingend
- 4) Alternative Ablaufreihenfolgen
Die Reihenfolgebeziehungen aus dem Ablaufgraphen werden genutzt.



- ① Berücksichtigung der Priorität
- ② Nutzung der Maschinenalternativen
- ③ Parallele Einplanung von Arbeitsgängen
- ④ Alternative Ablaufreihenfolge

Bild 80: Maschinenbelegungsplan zu Auftragspektrum 1

7.5.2 Auftragsspektrum 2

Zur Disposition stehen folgende Aufträge aus dem Auftragspool:

Nr.1:	Ritzelwelle 45 Nm (1:2)	Losgröße: 1	Priorität: 1
Nr.2:	Ritzelwelle 45 Nm (2:3)	Losgröße: 1	Priorität: 2
Nr.3:	Ritzelwelle 60 Nm (1:2)	Losgröße: 1	Priorität: 3
Nr.4:	Ritzelwelle 60 Nm (2:3)	Losgröße: 1	Priorität: 3
Nr.5:	Abtriebswelle 60 Nm (1:2)	Losgröße: 1	Priorität: 3
Nr.6:	Abtriebswelle 60 Nm (2:3)	Losgröße: 1	Priorität: 3

Die Optimierungskriterien sind mit folgenden Gewichten versehen:

minimale Bearbeitungszeiten:	0
niedrige Kosten:	0
kurze Durchlaufzeiten:	9
hohe Auslastung bzw. wenig Stillstand:	0
geringe Rüstzeiten:	0

Die Planung erfolgt gemäß unten stehender Stellschrauben:

- A) Planungsdistanz: 1
Planungsauswahl: 2
- B) Planungsdistanz: 2
Planungsauswahl: 18

Die beiden Belegungspläne sind in Bild 81 und Bild 82 gegenübergestellt. Bemerkenswert ist dabei die Tatsache, daß schon bei sehr niedrigen Planungsdistanzen sehr gute Ergebnisse erzielt werden. Plan A) ist unwesentlich schlechter als Plan B), der seinerseits ein Optimum hinsichtlich der Optimierungskriterien darstellt. Da die Planungsparameter bei A) sehr niedrig sind, wirkt sich das auch auf die Laufzeit des Algorithmus aus. Diese ist nämlich bei einer derartigen Parametereinstellung so kurz, daß die Planungskomponente im Falle einer unvorhergesehenen Störung unmittelbar zur Umplanung eingesetzt werden kann, wenn dies aufgrund der Situation in der Fertigung erforderlich ist. Die noch nicht bearbeiteten Arbeitsgänge werden dann in den Pool zurückgelegt und ein neuer Planungslauf mit reduziertem Auftragsspektrum wird angestoßen.

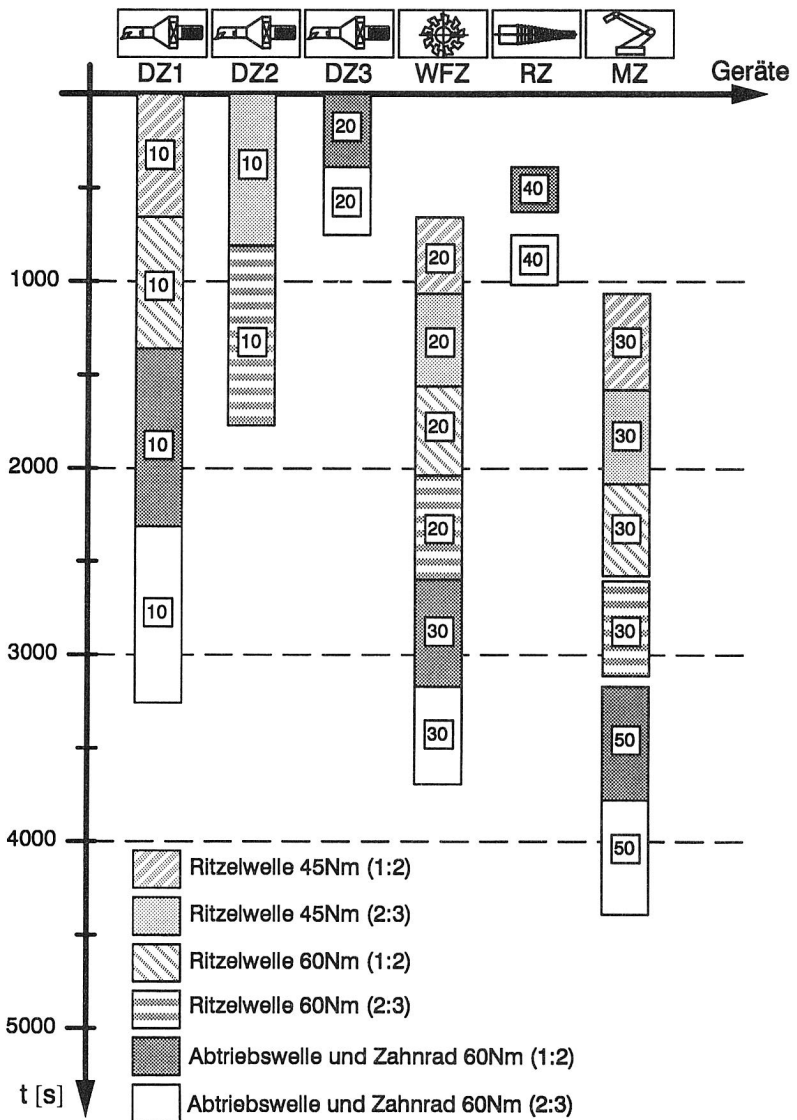


Bild 81: Maschinenbelegungsplan zu Auftragspektrum 2 A)

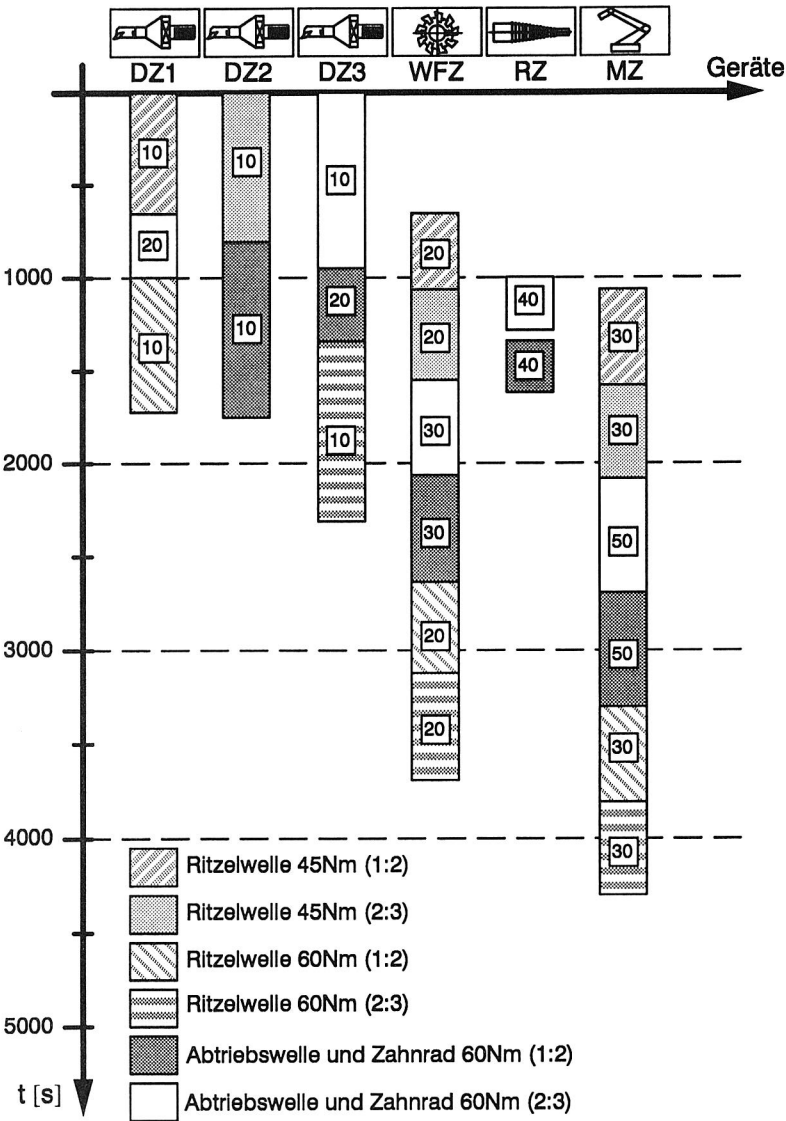


Bild 82: Maschinenbelegungsplan zu Auftragspektrum 2 B)

7.5.3 Auftragspektrum 3

Zur Disposition stehen folgende Aufträge aus dem Auftragspool:

Nr.1: Abtriebswelle 45 Nm (1:2) Losgröße: 3 Priorität: 1

Die Optimierungskriterien sind mit folgenden Gewichten versehen:

minimale Bearbeitungszeiten:	3
niedrige Kosten:	0
kurze Durchlaufzeiten:	3
hohe Auslastung bzw. wenig Stillstand:	3
geringe Rüstzeiten:	0

Der Belegungsplan, den die Steuerung als Eingabe erhält, ist in Bild 83 zu sehen.

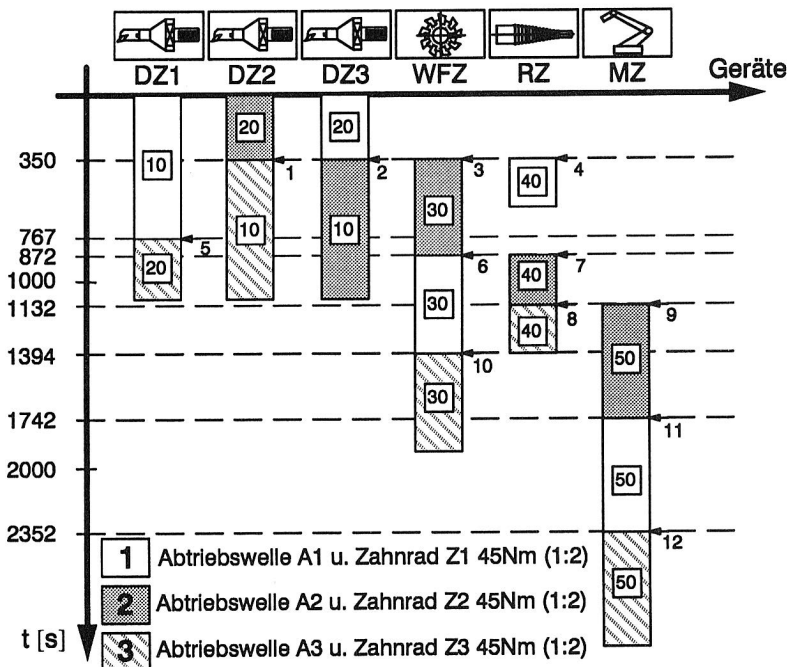


Bild 83: Maschinenbelegungsplan zu Beispiel 3

Wie aus Bild 83 direkt entnommen werden kann, ist die Dauer der Planungseinheiten länger als die effektive Bearbeitungszeit der einzelnen Arbeitsgänge. Der Grund dafür liegt in der Planung auf Leitebene. Die Leitebene sieht die Zelle "nur" als dispositive Einheit. Deshalb müssen die mittleren Transportzeiten, die zum Beispiel zur Beschickung der Maschinen anfallen, berücksichtigt werden. Sie werden zur Bearbeitungszeit hinzuaddiert.

Unter der Voraussetzung, daß sich das Material A1, Z2 und Z1 zu Beginn bereits am richtigen Ort befindet, nämlich in den Palettenwechselsystemen vor den jeweiligen Maschinen (Orte OPU, OPV und OPW aus dem Transportnetz)¹⁾, sind in den Belegungsplan sequentiell die Punkte 1–12 eingezeichnet, an denen die Steuerung einen Bedarf ermittelt. Schrittweise wird nun für jeden dieser Punkte die nötige Transportoperation generiert und in den Plan eingetragen:

Schritt 1: – Bedarfsermittlung:

A3:	Ist:	OB1
	Soll:	OPV

– Generierung der Operation:

Sequenz 1:	Holen eines Transportmittels (kürzester Weg)	
FTS1:	Von:	ODZ1
	Nach:	OS1
	Dauer:	42 ZE
Sequenz 2:	Material transportieren (kürzester Weg)	
A3:	Von:	OB1
	Nach:	OPV
	Dauer:	67 ZE

– Einplanung der Operation:

FTS1:	Beginn:	241
	Ende:	349

Bemerkung:

- Keine Verschiebungen im Maschinenbelegungsplan notwendig.
- Der Ort ODZ2 wird freigeräumt (Ausweichen von FTS2)
- Die Kanten von OB1 nach OS1 und ODZ2 nach OPV repräsentieren die Belade- und Entladeaktion (FTS wartet auf dem Ort davor). Solche Kanten gelten hier als statisch.

¹⁾ Weitere Ortsbelegungen:

A2 an OB2
FTS1 an ODZ1

A3 an OB1
FTS2 an ODZ2

Z3 an OB2
FTS3 an ODZ3

Schritt 2: – Bedarfsermittlung:

A2: Ist: OB2
 Soll: OPW

– Generierung der Operation:

Sequenz 1: Holen eines Transportmittels (kürzester Weg)

FTS3: Von: ODZ3
 Nach: OS2
 Dauer: 51 ZE

Sequenz 2: Material transportieren (kürzester Weg)

A2: Von: OB2
 Nach: OPW
 Dauer: 69 ZE

– Einplanung der Operation:

FTS3: Beginn: 230
 Ende: 349

Bemerkung:

- Keine Verschiebungen im Maschinenbelegungsplan notwendig.
- Kollisionsüberwachung der Fahrtrouten bei der Wegermittlung.

Schritt 3: – Bedarfsermittlung:

Z2: Ist: OPV
 Soll: OPY

– Generierung der Operation:

Sequenz 1: Holen eines Transportmittels (kürzester Weg)

FTS1: Von: ODZ2
 Nach: ODZ2
 Dauer: 0 ZE

Sequenz 2: Material transportieren (kürzester Weg)

Z2: Von: OPV
 Nach: OPY
 Dauer: 46 ZE

– Einplanung der Operation:

FTS1: Beginn: 350
 Ende: 395

Bemerkung:

- Verschiebungen notwendig:
Planungseinheit 30/2 und alle auf dieser Maschine direkt folgenden Planungseinheiten werden verschoben, also 30/1 und 30/3.
Alle von 30/2 abhängigen Einheiten und deren Nachfolger werden verschoben: 40/2 und 40/3, sowie 50/2, 50/1 und 50/3.

Schritt 4 – 12: Analog

Zu erwähnen bleibt noch, daß für die Montageoperationen sowohl die Welle als auch das Zahnrad Just in Time bereitgestellt werden, so daß für den Transport eine Synchronisation der beiden Fahrzeuge erforderlich ist, da pro FTS nur ein Teil transportiert werden kann. Der Anlagenfahrplan ist in Bild 84 zu sehen, wobei die Belegung der einzelnen Fahrzeuge aus Gründen der Übersichtlichkeit nicht explizit aufgeführt sind.

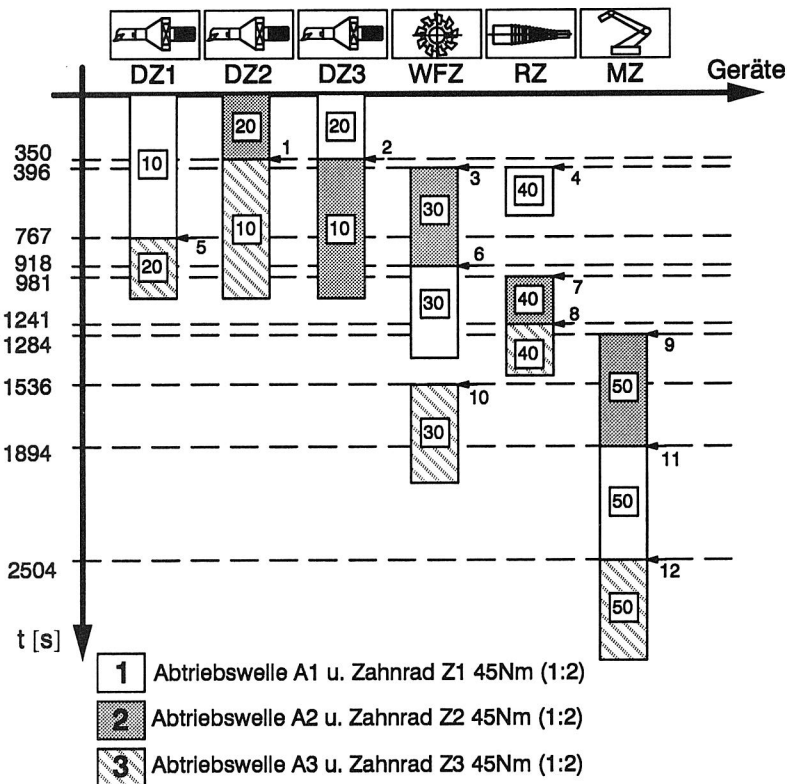


Bild 84: Anlagenfahrplan zu Auftragsspektrum 3

Wie aus der Gegenüberstellung von Bild 83 und Bild 84 zu erkennen ist, kann die zeitliche Verschiebung aller Arbeitsgänge aus dem Maschinenbelegungsplan so minimal wie nur möglich gehalten werden. Das bedeutet, daß nur dann Verschiebungen erforderlich sind, wenn von der Planungskomponente zwei direkt aufeinanderfolgende Arbeitsgänge mit demselben Material auf zwei unterschiedlichen Maschinen disponiert wurden. Die Steuerung ermittelt also bei vorgegebenem Belegungsplan das Minimum bzw. Optimum an zeitlichen Verschiebungen, welches möglich ist.

Abschließend kann festgehalten werden, daß die Tests, welche mit den unterschiedlichen Auftragspektren durchgeführt wurden, gezeigt haben, daß sich die Konzeption der Planungs- und Steuerungskomponente in dieser Form bewährt hat. Die Planung kann nicht nur zur eigentlichen Maschinenbelegungsplanung eingesetzt werden, sondern auch als Umplanungskomponente. In beiden Fällen können hinsichtlich der Planungsgüte sehr gute Ergebnisse erzielt werden. Die Steuerung ermöglicht auf der Basis dieser Ergebnisse die Flußsteuerung für die Fertigungshilfsmittel und das Material (Werkstücke und Baugruppen). Die Maschinenbelegung wird dabei nur insofern beeinflusst, als dies unbedingt erforderlich ist. Alternative Wege durch die Fertigungseinrichtung sind ebenso realisiert, wie die Synchronisation mehrerer Transportmittel (Kollisionsüberwachung). Ferner erlaubt die objektorientierte Softwarestruktur die Abbildung beliebiger Fertigungsstrukturen unabhängig von der Organisationsform.

Mit der Überwachungskomponente existiert zudem eine Komponente, die einen regelnden Eingriff in den Fertigungsverlauf erlaubt und es so ermöglicht, auf die sich ändernden Situation reagieren zu können.

8 Zusammenfassung

Die Anforderungen, die im Bereich der automatisierten Fertigung an ein Unternehmen gestellt werden, steigen aus den unterschiedlichsten Gründen immer weiter. Das verlangt vor allem eine zunehmende Integration der verwendeten Produktionsmittel. Mit der Einführung von flexiblen Fertigungseinrichtungen kann ein wichtiger Schritt dahin getan werden.

Während die Flexibilität in derartigen Systemen bereits einen beachtlichen Stand erreicht hat, steht die Flexibilität der auf Steuerungs- und Leitebene eingesetzten Softwarekomponenten im krassen Gegensatz dazu /101/. Um dieses Manko zu beseitigen, wurden in den letzten Jahren verstärkt werkstattnahe Steuerungssysteme entwickelt. Sie versuchen die Lücke zwischen der Produktionsplanung und -steuerung auf der einen und der Fertigung auf der anderen Seite zu schließen. Diese Leitstände sind jedoch von der Konzeption her nur darauf ausgelegt, den Meister in der Werkstatt zu unterstützen. Eine vollständige Überwachung wird ebenso vermißt wie zeitkonkrete Planungsergebnisse oder eine ausgereifte Ressourcenbereitstellung. Dies aber sind Voraussetzungen für eine optimale Durchsetzung in der Fertigung.

Vor diesem Hintergrund war es die Zielsetzung der vorliegenden Arbeit, ein Werkstattsteuerungssystem, welches diese Schwachstellen beseitigt, zu konzipieren, prototypisch zu realisieren und an einem realen Szenarium zu validieren. In Kapitel 2 wurde deshalb eine Anforderungsliste zur Steigerung der Flexibilität aufgestellt, die es zu erfüllen galt.

So wurde mit dem **Ablaufgraphen** eine innovative Darstellungsstruktur definiert, die bezüglich des Fertigungsverlaufs sowohl Ablauf- als auch Technologiealternativen erlaubt. Erweiterte Arbeitsplanstrukturen, die eine umfangreiche Ressourcenaufbereitung ermöglichen, können in den Ablaufgraphen integriert werden und stehen der Werkstattsteuerung zur Verfügung. Informationen über ersetzende oder ergänzende Betriebsmittel (Geräte, Werkzeuge, ...) werden bereitgestellt.

Darauf aufbauend wurde die **objektorientierte Softwarestruktur** in ihrer Konzeption so ausgelegt, daß sie dem Anspruch nach Universalität gerecht wird. Mittels der komponentenorientierten Klassifikation und der Realisierung durch ein seman-

tisches Netz können Fertigungseinrichtungen gleich welcher Organisationsform (Zelle, System, ...), auch über unterschiedliche Hierarchiestufen hinweg, in einer Wissensbasis abgebildet werden. Auf eine Trennung zwischen den Bereichen der Teilefertigung und der Montage wird verzichtet, womit hybride Fertigungssysteme in die Betrachtung mit eingeschlossen sind.

Beim Entwurf der Algorithmen zur eigentlichen Werkstattsteuerung wurde auf die Belange der Praxis geachtet. Durch die **Planung** kann für einen vorgegebenen Zeitraum hinsichtlich eines bestimmten Zielkriteriums ein optimaler Belegungsplan erzeugt werden. Mit den Stellschrauben in Form der Planungsabstand und der Planungsauswahl ist ein Instrument gegeben, das den zeitkritischen Anforderungen für etwaigen Umplanungsvorgängen bei unvorhersehbaren Ereignissen (z.B. Störungen) genügt. Weiterhin organisiert die **Steuerung** auf der Basis des Maschinenbelegungsplans die erforderlichen Bereitstellungsoperationen, um die zeitgerechte Durchsetzung der Aufträge zu gewährleisten. In der **Überwachung** ist zudem eine situationsadaptive Regelung realisiert. Die Prämisse kleiner Auftragspektren kann durch eine höhere Planungsfrequenz erfüllt werden. Dies trägt dann auch der Forderung nach konkreten Planungsergebnissen Rechnung.

Abschließend bleibt für den Ausblick festzustellen, daß dieses Softwaresystem nicht nur zur arbeitgangbezogenen Werkstattfeinsteuerung für den Bereich der Einzel- und Kleinserienfertigung fähig ist, sondern auch als Simulationsinstrument zur Schwachstellenanalyse in flexiblen Fertigungseinrichtungen Einsatz finden könnte.

9 Literaturverzeichnis

- /1/ Feldmann, K. und Schrödel, O.:
Möglichkeiten und Chancen der rechnergestützten Produktion für den Mittelstand
Zur Fachtagung: Rechnergestützte Produktion – Chance und Herausforderung für den Mittelstand (Internationalen Fachmesse für Metallbearbeitung), Juni 1990
- /2/ Spur, G.:
Rationalisierung zeitbestimmter Arbeitsprozesse
In: Wettbewerbsfaktor Zeit in Produktionsunternehmen, S.95–112
Berlin Heidelberg: Springer Verlag, 1991
- /3/ Milberg, J.:
Wettbewerbsfaktor Zeit in Produktionsunternehmen
In: Wettbewerbsfaktor Zeit in Produktionsunternehmen, S.13–32
Berlin Heidelberg: Springer Verlag, 1991
- /4/ Feldmann K.:
Entwicklung und Einsatz rechnerintegrierter Produktionssysteme
In: Zwf 6, (1988), Nr. 6, S.290–295
- /5/ Feldmann, K.:
Fabrik der Zukunft: Strukturen und Ziele
In: Siemens-Zeitschrift Special, Forschung und Entwicklung, Herbst 1990
S. 29–33
- /6/ Hirt, K. et. al.:
Einsatzbedingungen von flexiblen Fertigungssystemen
In: VDI-Z 133 (1991), Nr. 1, S. 41–44
- /7/ Eversheim, W.:
Organisation in der Produktionstechnik
Band 3: Arbeitsvorbereitung
Düsseldorf: VDI-Verlag GmbH, 1989

- /8/ Krause, F.-L. und Altmann, C.
Arbeitsplanung alternativer Prozesse für flexible Fertigungssysteme
In: Zwf 84 (1989), Nr. 5, S. 228–232
- /9/ Bullinger, H.J.:
Systematische Montageplanung
München, Wien: Carl Hanser Verlag, 1986
- /10/ Weule, H. und Friedmann, T.:
Rechnergestützte Produktanalyse in der Montageplanung
In: VDI-Z 129 (1987), Nr. 12, S. 59–63
- /11/ Anders, N. et. al.:
AVOGEN – wissensbasierte Generierung von Arbeitsvorgangsfolgen
In: VDI-Z 132 (1990), Nr. 4, S. 92–52
- /12/ Tönshoff, K. et. al.:
Wissensbasierte Generierung von Montagevorgangsfolgen
In: VDI-Z 133 (1991), Nr. 14, S. 48–54
- /13/ Geitner, U.W.:
CIM-Handbuch
Braunschweig, Wiesbaden: Vieweg, 1987
- /14/ Scheer, A.-W.:
CIM-Der computergesteuerte Industriebetrieb
Berlin, Heidelberg: Springer Verlag, 1987
- /15/ Rembold, U., et. al.:
CAM-Handbuch
Berlin, Heidelberg: Springer-Verlag, 1990
- /16/ Tuffentsammer, K.:
Flexibles Fertigungssystem
Weinheim: VCH Verlagsgesellschaft mbH, 1988

- /17/ Kalde, M.:
Methodik zur Festlegung der Flexibilität in der Montage
Dissertation an der Rheinisch-Westfälischen Technischen Hochschule
Aachen, 1987
- /18/ Groha, A.:
Universelles Zellenrechnerkonzept für flexible Fertigungssysteme
Dissertation am Institut für Werkzeugmaschinen und Betriebstechnik der
Technischen Universität München
Berlin Heidelberg: Springer Verlag, 1988
- /19/ Schulz, J.:
Auslegung computerintegrierter Arbeitsplanung für flexible Fertigungs-
systeme
Dissertation an der Rheinisch-Westfälischen Technischen Hochschule
Aachen, 1987
- /20/ Schönheit, M., et. al.:
Flexible Fertigung
In: VDI-Z 132 (1990), Nr. 10, S. 92–109
- /21/ AWK Aachener Werkzeugmaschinen-Kolloquium (Hrsg.)
Autorenkollektiv (Weck, M. et. al.)
Wettbewerbsfaktor Produktionstechnik
Düsseldorf: VDI-Verlag GmbH, 1990
- /22/ Shah, R.:
Flexible Fertigungssystem in Europa: Erfahrungen der Anwender
In: VDI-Z 133 (1991), Nr. 6, S.16–30
- /23/ Reichel, H.:
Optimierung der Werkzeugbereitstellung durch rechnergestützte Ar-
beitsfolgenbestimmung
Dissertation am Lehrstuhl für Fertigungsautomatisierung und Produk-
tionssystematik der Friedrich – Alexander Universität Erlangen – Nürnberg
München, Wien: Carl Hanser Verlag, 1991

- /24/ Feldmann, K.:
Rechnergestützte Planung und Steuerung von automatisierten Montageanlagen
In: Siemens-Zeitschrift (1990), Nr. 5, S. 4–8
- /25/ Scholz, W.:
Modell zur datenbankgestützten Planung automatisierter Montageanlagen
Dissertation am Lehrstuhl für Fertigungsautomatisierung und Produktionssystematik der Friedrich – Alexander Universität Erlangen – Nürnberg
München, Wien: Carl Hanser Verlag, 1989
- /26/ Warnecke, H.J.:
Der Produktionsbetrieb
Berlin, Heidelberg: Springer-Verlag, 1984
- /27/ Herzog, H.-D.:
Serienfertigung und Werkstattfertigung
In: VDI-Z 132 (1990), Nr. 6, S.46–52
- /28/ Busch, U.
Entwicklung eines PPS-Systems
Berlin: Erich Schmidt Verlag GmbH, 1987
- /29/ Weule, H. und Schmitt, E.:
Ein hierarchischer Lösungsansatz zur Auftragsreihenfolgeplanung bei mehrstufiger Werkstattfertigung
In: wt 79 (1989), Nr. S. 333–336
- /30/ Köhler, C.:
Der elektronische Leitstand – Befehlsempfänger der PPS oder Partner der Werkstatt?
In: VDI-Z 132 (1990), Nr. 3, S. 14–19
- /31/ Schrödel, O. und Reichel, H.:
Marktanalyse von Werkstattplanungs- und Steuerungssystemen
Interner Bericht am Lehrstuhl für Fertigungsautomatisierung und Produktionssystematik, 1990

- /32/ Friedrichs, P. und Gromotka, W.:
Fertigungsleittechnik
In: VDI-Z 132 (1990), Nr. 3, S. 145–152
- /33/ Ploenzke Informatik (Hrsg.)
Fertigungsleitstand-Report
Competence Center CIM-Anwendungen und Standardsoftware, 1990
- /34/ Grob, R.:
Flexibilität in der Fertigung
Dissertation an der Rheinisch-Westfälischen Technischen Hochschule
Berlin Heidelberg: Springer Verlag, 1986
- /35/ Fischer, H.:
Verteilte Planungssysteme zur Flexibilitätssteigerung der rechnerin-
tegrierten Teilefertigung
Dissertation am Lehrstuhl für Fertigungsautomatisierung und Produk-
tionssystematik der Friedrich – Alexander Universität Erlangen – Nürnberg
München, Wien: Carl Hanser Verlag, 1990
- /36/ Severin, F.:
Planung der Flexibilität von roboterintegrierten Bearbeitungs- und Mon-
tagezellen
Dissertation an der Technischen Universität Berlin
München, Wien: Carl Hanser Verlag, 1986
- /37/ Görz, G.:
Wissensrepräsentation und die Verarbeitung natürlicher Sprache
In: it (1989), Nr. 2, S. 155–166
- /38/ Eichhorn, W.:
Eine aktive hierarchische Wissensstruktur für die Musteranalys
Dissertation am Institut für mathematische Maschinen und Datenverar-
beitung der Friedrich – Alexander Universität Erlangen – Nürnberg
Arbeitsbericht, Band 24, Nr.3, Erlangen 1991

- /39/ Richter, M.:
Prinzipien der künstlichen Intelligenz
Stuttgart: B. G. Teubner, 1989
- /40/ Gupta, M.M. und Yamakawa, T.:
Fuzzy Computing
Theory, Hardware, and Applications
Amsterdam: Elsevier Science Publishers B.V., 1988
- /41/ Bezdek, J.C.:
Analysis of Fuzzy Information
Boca Raton: CRC Press, Inc., 1987
- /42/ Bläsius, K. H. und Bürckert, H.-J.:
Deduktionssysteme
München: R. Oldenbourg Verlag GmbH, 1987
- /43/ Hemberger, A.:
Innovationspotentiale in der rechnerintegrierten Produktion durch wissensbasierte Systeme
Dissertation am Lehrstuhl für Fertigungsautomatisierung und Produktionssystematik der Friedrich – Alexander Universität Erlangen – Nürnberg
München, Wien: Carl Hanser Verlag, 1988
- /44/ Hertzberg, J.:
Planen
Zürich: Bibliographisches Institut & F.A. Brockhaus AG, 1989
- /45/ Kippe, J.:
Komponentenorientierte Repräsentation technischer Systeme
In: Technische Expertensysteme: Wissensrepräsentation und Schlußfolgerungsverfahren
München, Wien: R. Oldenbourg Verlag, 1988
- /46/ Codd, E.F.:
In: Relational database systems (Foreword)
Springer-Verlag, 1983

- /47/ Wedekind, H.:
 Datenbanksysteme I
 Mannheim, Wien, Zürich: Bibliographisches Institut AG, 1991
- /48/ Trautloft, R. und Lindner U.:
 Datenbanken: Entwurf und Anwendung
 Berlin: Verlag Technik, 1991
- /49/ Minsky, M.:
 A Framework for Representing Knowledge
 In: The Psychology of Computer Vision
 New York: McGrawHill, pp. 211–277, 1975
- /50/ Schnelder, H.-J.(Hrsg.):
 Lexikon der Informatik und Datenverarbeitung
 Wien: Oldenbourg Verlag, 1983
- /51/ Reisl, W.:
 Systementwurf mit Netzen
 Berlin, Heidelberg, New York, Tokyo: Springer Verlag, 1985
- /52/ Janzen, F. et. al.:
 Petrinetze in der Produktionstechnik
 In: Zwf 84 (1989), Nr. 3, S. 141–145
- /53/ Itter, F.:
 Einsatz von Petrinetzen zur Beschreibung von Fertigungssystemen
 In: Zwf 84 (1989), Nr. 4, S. 206–210
- /54/ Möhrle, M.:
 Petrinetze in der Produktionstechnik – Integration von Planung, Simulation und Steuerung von Produktionsanlagen
 Bochum: Dissertation am Institut für Automatisierungstechnik, 1989

- /55/ Eschenbacher, P.:
Entwurf und Implementierung einer formalen Sprache zur Beschreibung
dynamischer Modelle
Dissertation am Institut für mathematische Maschinen und Datenver-
arbeitung der Friedrich – Alexander Universität Erlangen–Nürnberg, 1988
Arbeitsbericht, Band 23, Nr.1, Erlangen 1990
- /56/ Gevarter, W. B.:
Intelligente Maschinen
Weinheim, New York: VCH Verlagsgesellschaft, 1987
- /57/ Niemann, H.:
Pattern Analysis
Berlin: Springer Verlag, 1981
- /58/ Manhart, K.:
Ein Semantisches Netz in Lisp
In: mc (1988), November, S.66–73
- /59/ Stoyan, H.:
Programmiermethoden der Künstlichen Intelligenz
Berlin, Heidelberg: Springer Verlag, 1988
- /60/ Schlaer, S. and Mellor, S.:
Object–Oriented Systems Analysis
Englewood, New Jersey: Prentice Hall Building, 1988
- /61/ Wedekind, H.:
Objektorientierung und Vererbung
– Zu den Grundlagen zweier wichtiger Softwarebegriffe –
In: it (1990), Nr. 2, S. 79–86
- /62/ Schmidt, G. und Ströhlein T.:
Relationen und Graphen
Berlin, Heidelberg, New York, Tokyo: Springer Verlag, 1989

- /63/ Winston, P.H.:
Künstliche Intelligenz
Bonn: Addison-Wesley Verlag (Deutschland) GmbH, 1987
- /64/ Meißner, J.-D.:
Heuristische Programmierung
Wiesbaden: Akademische Verlagsgesellschaft, 1978
- /65/ Pearl, J.:
Heuristics
Reading, Massachusetts: Addison-Wesley Publishing Company, 1984
- /66/ Meyer, B:
Object-Oriented Software Construction
Wood Lane End, Hemel Hempstead: Prentice Hall International Ltd, 1988
- /67/ Wedekind, H.:
Objektorientierte Schemaentwicklung
– Ein kategorialer Ansatz für Datenbanken und Programmierung
Mannheim, Wien, Zürich: Bibliographisches Institut AG, 1992
- /68/ Booch, G.:
Object-Oriented Development
In: IEEE Transactions on Software Engineering, Vol. 12, No. 2, Febr. 1986,
pp. 211–221
- /69/ Jalote, P.:
Functional Refinement and Nested Objects for Object-Oriented Design
In: IEEE Transactions on Software Engineering, Vol. 15, No. 3, March
1989, pp. 264–270
- /70/ Keramidis, S.:
Eine Methode zur Spezifikation und korrekten Implementierung von
asynchronen Systemen
Habilitation am Institut für mathematische Maschinen und Datenverarbeitung
der Friedrich – Alexander Universität Erlangen – Nürnberg
Arbeitsbericht, Band 15, Nr.4, Erlangen 1982

- /71/ Schuhmann, J. und Gerisch, M.:
Softwareentwurf – Prinzipien, Methoden, Arbeitsschritte, Rechnerunterstützung
Köln: Verlagsgesellschaft Rudolf Müller, 1986
- /72/ Scheller, J.:
Modellierung und Einsatz von Softwaresystemen für rechnergeführte Montagezellen
Dissertation am Lehrstuhl für Fertigungsautomatisierung und Produktionssystematik der Friedrich–Alexander Universität Erlangen–Nürnberg
München, Wien: Carl Hanser Verlag, 1991
- /73/ Verein Deutscher Ingenieure (Hrsg.):
Lexikon der Produktionsplanung und –steuerung
Düsseldorf: VDI–Verlag GmbH, 1983
- /74/ Strack, M.:
Organisatorische Gestaltung einer zentralen Werkstattsteuerung
Dissertation an der Rheinisch–Westfälischen Technischen Hochschule Berlin, Heidelberg: Springer Verlag, 1987
- /75/ Ochs, M.:
Entwurf eines Programmsystems zur wissensbasierten Planung und Konfigurierung
Dissertation am Institut für Werkzeugmaschinen und Betriebstechnik der Universität Karlsruhe, 1988
- /76/ Milberg, J.:
Automatisierung der Produktion – Eine ganzheitliche Aufgabe
Bericht des Instituts für Werkzeugmaschinen und Betriebswissenschaften der Technischen Universität München
- /77/ Dilling, U.:
Automatisierung der Werkstückversorgung an Werkzeugmaschinen
In: wt 81 (1991), S. 36–40

- /78/ Lorenzen, P. und Schwemmer O.:
Konstruktive Logik, Ethik und Wissenschaftstheorie
Mannheim, Wien, Zürich: Bibliographisches Institut AG, 1975
- /79/ Zeis, G.:
Methoden zur Steigerung der Wiederverwendbarkeit von Softwarekomponenten in der Fertigungsautomatisierung
Dissertation am Institut für mathematische Maschinen und Datenverarbeitung der Friedrich – Alexander Universität Erlangen – Nürnberg
Arbeitsberichte, Band 24, Nr.3, Erlangen 1991
- /80/ Hofmann, F.:
Betriebssysteme: Grundkonzepte und Modellvorstellungen
Stuttgart: B.G. Teubner, 1984
- /81/ Mackert, L.:
Modellierung, Spezifikation und korrekte Realisierung von asynchronen Systemen
Dissertation am Institut für mathematische Maschinen und Datenverarbeitung der Friedrich – Alexander Universität Erlangen – Nürnberg
Arbeitsberichte, Band 20, Nr.2, Erlangen 1987
- /82/ Hallmann, M.:
Prototyping komplexer Softwaresysteme
Stuttgart: Teubner, 1990
- /83/ Gulbins, J.:
Unix
Berlin, Heidelberg, New York, Tokyo: Springer Verlag, 1985
- /84/ Kernighan, B.W. und Pike R.:
The UNIX™ Programming Environment
Englewood Cliffs, New Jersey: Prentice Hall Inc., 1984
- /85/ Kutscha, S. und Henning, K.:
Was wird aus Unix? Stand und Trends der Standardisierung
In: atp 33 (1991), Nr.1, S.31–36

- /86/ Lippman, S.B.:
A C++ Primer
Reading, Massachusetts: Addison-Wesley Publishing Company, 1989
- /87/ Stroustrup, B.:
Die C++ Programmiersprache
Bonn: Addison-Wesley Publishing Company, 1989
- /88/ Krüger, G.:
Ein Leitstand macht noch keine Werkstattsteuerung
In: VDI-Z 133 (1991), Nr. 3, S. 43–50
- /89/ Bronstein, I.:
Taschenbuch der Mathematik
Leipzig, B.G. Teubner Verlagsgesellschaft, 1979
- /90/ Bronstein, I.:
Taschenbuch der Mathematik (Ergänzende Kapitel)
Leipzig: B.G. Teubner Verlagsgesellschaft, 1981
- /91/ Müller-Merbach, H.:
Operations Research
München: Verlag Franz Fahlen, 1973
- /92/ Zörnlein, G.:
Flexible Fertigungs-Systeme
Belegung, Steuerung, Datenorganisation
Dissertation am Institut für mathematische Maschinen und Datenverarbeitung der Friedrich – Alexander Universität Erlangen – Nürnberg
München, Wien: Carl Hanser Verlag, 1988
- /93/ Pritschow G., Spur G. und Weck M. (Hrsg.):
Leit- und Steuerungstechnik in flexiblen Produktionsanlagen
München, Wien: Carl Hanser Verlag, 1991

- /94/ Wiendahl, H.-P.:
Belastungsorientierte Fertigungssteuerung
München, Wien: Carl Hanser Verlag, 1987
- /95/ Beier, H.H. und Schwall, E.:
Fertigungsleittechnik
München, Wien: Carl Hanser Verlag, 1991
- /96/ Kiener, A.:
Objektorientierte Programmierung in flexiblen Fertigungszellen
Diplomarbeit am Lehrstuhl für Fertigungsautomatisierung und Produktionssystematik, 1990
- /97/ Hammer, H.:
Verfügbarkeitsanalyse von flexiblen Fertigungssystemen
In: Fertigungstechnisches Kolloquium, S. 59–67
Berlin Heidelberg: Springer Verlag, 1991
- /98/ Weck, M. und Lange, N.:
Universelle Ablaufsteuerung für die flexible Fertigung
In: VDI-Z 133 (1991), Nr. 3, S. 50–56
- /99/ Pomberger, G.:
Softwaretechnik und Modula-2
München, Wien: Hanser Verlag, 1984
- /100/ Rohde, J.:
Visualisierung wissensbasierter Systemkomponenten
Diplomarbeit am Lehrstuhl für Fertigungsautomatisierung und Produktionssystematik, 1991
- /101/ Weck, M. und Lopez, M.:
Konfigurierbare Bedienoberfläche einer offenen FFS-Steuerungsarchitektur
In: ZWF 86, (1991), Nr. 1, S.33–36

LEBENS LAUF

- Name:** Olaf Schrödel
geboren am 10. Juli 1962 in Bamberg
- Eltern:** Georg Schrödel
Maria Schrödel, geborene Küpfer
- Familienstand:** ledig
- Schulbildung:** 1968 – 1972 Volksschule Altendorf
1972 – 1981 Clavius Gymnasium Bamberg (math. – nat.)
- Studium:** Von 1981 bis 1987 Studium der Informatik mit Nebenfach Physik
an der Friedrich Alexander Universität Erlangen – Nürnberg
Diplomprüfung am 25. Februar 1987
- Berufstätigkeit:** In der Zeit von 1981 bis 1985 mehrmalige Werkstudententätigkeit
bei der Kraftwerk Union AG Erlangen
- Seit dem 16. März 1987 wissenschaftlicher Mitarbeiter und seit dem
1. Mai 1990 Oberingenieur für Steuerungs- und Sensortechnik am
Lehrstuhl für Fertigungsautomatisierung und Produktions-
systematik der Universität Erlangen – Nürnberg
Leiter: Prof. Dr. – Ing. K. Feldmann

Reihe Fertigungstechnik Erlangen

Band 1

Andreas Hemberger

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

208 Seiten, 107 Bilder. 1988. Kartonierte.

Band 2

Detlef Classe

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

194 Seiten, 70 Bilder. 1988. Kartonierte.

Band 3

Friedrich-Wilhelm Nolting

Projektiertung von Montagesystemen

201 Seiten, 107 Bilder, 1 Tabelle. 1989.

Kartonierte.

Band 4

Karsten Schlüter

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

177 Seiten, 97 Bilder. 1989. Kartonierte.

Band 5

Shir-Kuan Lin

Aufbau von Modellen zur Lageregelung von Industrierobotern

168 Seiten, 46 Bilder. 1989. Kartonierte.

Band 6

Rudolf Nuss

**Untersuchungen zur Bearbeitungsqualität im Fertigungssystem
Laserstrahlschneiden**

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

Band 7

Wolfgang Scholz

**Modell zur datenbankgestützten Planung automatisierter
Montageanlagen**

194 Seiten, 89 Bilder. 1989. Kartonierte.

Band 8

Hans-Jürgen Wißmeier

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

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

Band 9

Rainer Eisele

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

183 Seiten, 86 Bilder. 1990. Kartonierte.

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

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

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

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

Band 14
Stephan Biermann
Untersuchungen zur Anlagen- und Prozeßdiagnostik für das Schneiden mit CO₂ - Hochleistungslasern
VIII + 170 Seiten, 93 Bilder, 4 Tabellen. 1991. Kartoniert.

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

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

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

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

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

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

Band 21

Egon Sommer

Multiprozessorsteuerung für kooperierende Industrieroboter in Montagezellen

188 Seiten, 102 Bilder. 1991. Kartoniert.

Band 22

Georg Geyer

Entwicklung problemspezifischer Verfahrensketten in der Montage

192 Seiten, 112 Bilder. 1991. Kartoniert.

Band 23

Rainer Flohr

Beitrag zur optimalen Verbindungstechnik in der Oberflächenmontage (SMT)

186 Seiten, 79 Bilder. 1991. Kartoniert.

Band 24

Alfons Rief

Untersuchungen zur Verfahrensfolge Laserstrahlschneiden und -schweißen in der Rohkarosseriefertigung

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

Band 25

Christof Thim

Rechnerunterstützte Optimierung von Materialflußstrukturen in der Elektronikmontage durch Simulation

188 Seiten, 74 Bilder. 1992. Kartoniert.

Band 26

Roland Müller

CO₂-Laserstrahlschneiden von kurzglasverstärkten Verbundwerkstoffen

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

Band 27

Günther Schäfer

Integrierte Informationsverarbeitung bei der Montageplanung

195 Seiten, 76 Bilder. 1992. Kartoniert.

Band 28

Martin Hoffmann

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

149 Seiten, 89 Bilder. 1992. Kartoniert.

Band 29

Peter Hoffmann

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

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

Band 30

Olaf Schrödel

Flexible Werkstattsteuerung mit objektorientierten Softwarestrukturen

180 Seiten, 84 Bilder. 1992. Kartoniert.