

KfK 5427
Dezember 1994

Wissensbasierte Modellierung des LIGA-Fertigungsprozesses

I. H. Brauch
Institut für Angewandte Informatik

Kernforschungszentrum Karlsruhe

KERNFORSCHUNGSZENTRUM KARLSRUHE

Institut für Angewandte Informatik

KfK 5427

Wissensbasierte Modellierung des LIGA-Fertigungsprozesses*)

Ilko Holger Brauch

*) von der Fakultät für Maschinenbau der Universität Karlsruhe
genehmigte Dissertation

Kernforschungszentrum Karlsruhe GmbH, Karlsruhe

Als Manuskript gedruckt
Für diesen Bericht behalten wir uns alle Rechte vor

Kernforschungszentrum Karlsruhe GmbH
Postfach 3640, 76021 Karlsruhe

ISSN 0303-4003

Wissensbasierte Modellierung des LIGA-Verfahrens

Zusammenfassung

Seit seiner Einführung wurde das am Kernforschungszentrum Karlsruhe entwickelte LIGA-Verfahren zur Herstellung von dreidimensionalen Mikrostrukturen unterschiedlicher Geometrie und Materialien stetig weiterentwickelt und verbessert. Heute liegt daher ein äußerst komplexer Fertigungsprozeß mit einer großen Anzahl von Prozeßabläufen, Alternativen und gegenseitigen Abhängigkeiten vor.

In der vorliegenden Arbeit wurde ein Modell für die rechnergestützte Repräsentation und Verarbeitung von Prozeßwissen entwickelt. Durch die rechnergestützte Modellierung ist es möglich Prozeßwissen, welches mit herkömmlichen Methoden nicht archiviert werden kann, zu erfassen und zu verarbeiten. Darüber hinaus bildet die Repräsentation von Prozeßwissen auf dem Rechner die Grundlage für eine durchgängige informationstechnische Unterstützung des gesamten Fertigungsprozesses, da auch andere Komponenten einer CIM-Umgebung, wie Systeme zum Entwurf oder der Vermessung, auf fertigungstechnisches Wissen zugreifen müssen.

Auf der Grundlage einer Analyse des Fertigungsprozesses wurde daher in der vorliegenden Arbeit ein Modell entwickelt, welches die wesentlichen allgemeinen Elemente zur Beschreibung des Prozesses berücksichtigt. Die Beschreibungselemente wurden auf ein objektorientiertes Modell abgebildet, in welchem die wesentlichen statischen und funktionalen Aspekte erfaßt werden. Das Modell erlaubt die Definition hierarchischer, zeitlicher und logischer Beziehungen zwischen den Elementen. Dadurch sind die Voraussetzungen für eine tiefgehende Erfassung von Prozeßwissen gegeben.

Damit auch quantitative Aspekte des Prozeßwissens verarbeitet werden können, wurden im Modell die Prozeßparameter und ihre gegenseitigen Abhängigkeiten integriert. Parameterwerte lassen sich dabei durch numerische oder linguistische Ausdrücke darstellen und durch entsprechende funktionale Abhängigkeiten verknüpfen. Für eine vollständige formale Beschreibung wurden sowohl scharf als auch unscharf charakterisierte Parameterwerte in die Modellierung einbezogen. Es wurde gezeigt, daß sich unscharfe Zusammenhänge zwischen Parametern mit Hilfe des plausiblen Schließens darstellen lassen, das auf der Grundlage des logischen Schließens mit unscharfen Mengen basiert.

Die Verarbeitung scharfer und unscharfer Zusammenhänge erfolgt innerhalb eines einzigen Modells. Parameter können daher beliebig vernetzt werden, wodurch sich Abhängigkeiten zwischen den Parametern unterschiedlicher Teilprozesse beschreiben lassen. Auf der Basis dieser Beschreibung von Zusammenhängen können Simulationen durchgeführt werden, die es erlauben, verschiedene Parameterkonstellationen zu testen und die erhaltenen Ergebnisse zu begründen.

Alle beschriebenen Modelle wurden auf einer Workstation in einem lauffähigen System realisiert. Die Realisierbarkeit der beschriebenen Konzepte wurde anhand einer Wissensbasis, welche reale Prozeßsituationen aus dem Bereich der Zwischenmaskenherstellung für die Röntgenlithographie beschreibt, demonstriert.

Knowledge Based Modelling of the LIGA Fabrication Process

Abstract

The LIGA technique for the fabrication of three dimensional microstructures with variable geometry and material, developed at the Nuclear Research Center Karlsruhe, has been steadily improved during the last years. Today it is an extremely complex fabrication process involving a lot of process steps, alternatives and mutual dependencies.

In the present paper a model for the computer based representation and processing of knowledge has been developed. It is possible to process knowledge, which can't be stored with conventional methods. Further the computer based representation of process knowledge is fundamental for a complete computer based support for the whole fabrication process, because other components of a CIM-based environment like systems for the design or automatic measurement have to use fabrication knowledge.

Based on an analysis of the fabrication technique a model was developed, which uses the fundamental general elements for the description of the process. The basic elements have been mapped into an object oriented model, which considers the most important static and dynamic aspects. The model permits the definition of hierarchical, time based and logical connections between the elements. This is a very important precondition for a deeper modelling of process knowledge.

In order to be able to include quantitative aspects of the process knowledge, the process parameters and their mutual dependencies have been integrated in the model. Values of the parameters can be represented by numerical and linguistic expressions and can be connected by functional dependencies. For a more complete formal description, as well as crisp as fuzzy characterized parameter values have been considered. It has been shown, that dependencies between fuzzy parameter values, can be represented by plausible reasoning, which is based on logical reasoning with fuzzy sets.

The processing of crisp and fuzzy dependencies is integrated in a single model. Parameters can be arbitrarily connected. So dependencies between parameters of different partial processes can be described. Based on this description, simulations can be performed, testing variable parameter constellations and explaining the results.

All models described above have been implemented on a workstation. The realisability of the concepts developed could be proved by a knowledge base, which describes real process situations in the area of the intermediate mask fabrication for x-ray lithographic.

1.	Einleitung	1
1.1	Motivation und Problemstellung.....	1
1.2	Zielsetzung.....	4
1.3	Vorgehensweise	5
2.	Analyse des LIGA-Verfahrens und Systemanforderungen.....	7
2.1	Allgemeine Strukturierung des Prozeßwissens beim LIGA-Verfahren.....	7
2.1.1	Ablauf und Beschreibung der Prozeßdurchführung.....	7
2.1.2	Beziehungen zwischen Einstellparametern und Produkteigenschaften ..	11
2.2	Anforderung an die rechnergestützte Modellierung des LIGA-Verfahrens	15
2.2.1	Grundlegendes Modell und Anforderungen an die Wissensverwaltung.	15
2.2.2	Repräsentation von Oberflächenwissen und Tiefenwissen	16
2.2.3	Einbeziehung von unscharfem Wissen	16
3.	Grundlagen der Prozeßmodellierung	17
3.1	Methoden zur Wissensdarstellung auf dem Rechner.....	17
3.1.1	Semantische Netze	17
3.1.2	Das Relationale Datenmodell.....	19
3.1.3	Regelbasierte Darstellung	21
3.1.4	Objektorientiertes Paradigma.....	23
3.2	Bewertung der Methoden im Hinblick auf die Prozeßmodellierung	26
4.	Objektorientierte Analyse und Modellierung des Fertigungswissens	29
4.1	Prozeßschritte.....	29
4.2	Arbeitsmaterialien und Ausrüstungsgegenstände	31
4.3	Prozeßparameter.....	31
4.4	Parameterbeziehungen und Abhängigkeiten.....	32
5.	Beschreibung unscharfen Prozeßwissens	36
5.1	Methodik zur Repräsentation unscharfen Prozeßwissens.....	36
5.1.1	Inferenz auf der Basis des verallgemeinerten Modus Ponens.....	39
5.1.2	Eigenschaften der verwendeten Zugehörigkeitsfunktionen	40
6.	Simulation von Prozeßparameterabhängigkeiten	42
6.1	Parameterbeziehungen und Wertebereiche.....	42
6.2	Parameterabhängigkeiten und Integration	44
6.3	Simulation von Wirkungsketten zwischen Parametern	48

7.	Realisierung.....	54
7.1	Hard- und Softwarehilfsmittel zur Realisierung von LIMES	54
7.2	Wissenserwerb und Prozeßumfang	57
7.3	Visualisierung von Parameterabhängigkeiten des Prozesses.....	61
7.4	Simulation von Parameterwirkungsketten	63
7.5	Erklärungskomponente für die Ergebnisse	68
8.	Zusammenfassung und Ausblick	69
8.1	Zusammenfassung.....	69
8.2	Ausblick	70
	Anhang A.....	72
	Anhang B.....	80
	Anhang C.....	83
	Literatur.....	87

1. Einleitung

1.1 Motivation und Problemstellung

Die Mikrosystemtechnik ist ein junges Forschungsgebiet, in welchem Erfahrungen unterschiedlicher Disziplinen wie Maschinenbau, Elektrotechnik, Chemie, Informatik und Verfahrenstechnik zusammenfließen. Es werden Produkte hergestellt, die bei mikroskopischen Abmessungen Funktionen erfüllen, die bisher nur im makroskopischen Bereich oder überhaupt nicht verwirklicht werden konnten. Der Grundgedanke ist, bei gleichzeitiger Verringerung der Kosten und Einsparung von Rohstoffen, Produkte ständig wachsender Leistung hervorzubringen. So erhofft man sich, daß die Mikrosystemtechnik einen ähnlich erfolgreichen Entwicklungsweg einschlägt, wie ihn die Mikroelektronik in der Vergangenheit durchschritten hat.

Eine Schlüsselstellung in Mikrosystemen nehmen die mechanischen, optischen und chemischen Aktor- und Sensorelemente ein. Sie stellen die eigentliche technische Neuerung dar, welche die Herstellung von Mikrosystemen überhaupt erst möglich macht. Deshalb kommt den Fertigungsverfahren zur Herstellung dieser Elemente eine entscheidende Bedeutung zu.

Notwendigkeit der systematischen Aufarbeitung von Fertigungswissen

Das LIGA-Verfahren (**L**ithographie, **G**alvanik, **A**bformung) ist ein Verfahren zur Herstellung dreidimensionaler mechanischer und optischer Mikrostrukturen. Von anderen Mikrostrukturierungsverfahren unterscheidet es sich insbesondere durch die große Vielfalt der einsetzbaren Materialien [ZUMG93] und die Möglichkeit zur beliebigen lateralen Formgebung. Bedingt durch dieses Potential weist das Verfahren im heutigen Entwicklungsstand eine sehr hohe Komplexität auf. Dies wird bereits an der großen Anzahl von Prozessschritten zur Herstellung eines Produktes und der Zahl möglicher Alternativen der Prozeßführung für die Fertigung von Endprodukten unterschiedlichster Art deutlich. Im Zuge der laufenden Entwicklungsarbeiten werden überdies ständig einzelne Prozessschritte verbessert und neue Varianten entwickelt, die ältere Vorgehensweisen teilweise ergänzen oder vollständig ablösen [MENZ93].

Vor dem Hintergrund der bereits begonnenen kommerziellen Nutzung des LIGA-Verfahrens ist es in Zukunft erforderlich, die mit dem Herstellungsprozeß verbundenen Abläufe weitestgehend informationstechnisch zu unterstützen, um vorhandenes Wissen zeit- und kostengünstig einzusetzen.

Der Bedarf an detaillierten Informationen zum Fertigungsprozeß für LIGA-Mikrostrukturen bereits während der Konstruktionsphase wird in [LESS92] unterstrichen:

„Bei der weitergehenden Ausgestaltung (von LIGA-Mikrostrukturen) werden mit steigendem Detaillierungsgrad tieferegehende Informationen benötigt, um einerseits den genauen Fertigungsablauf festlegen, und andererseits das Produkt optimal an den Fertigungsprozeß anpassen zu können.“

Bisher liegt fertigungstechnisches Wissen über das LIGA-Verfahren vorwiegend in Form schriftlicher Aufzeichnungen oder als Wissen von Prozeßfachleuten vor. Ein großer Teil dieses Wissens kann durch die herkömmlich genutzten Beschreibungsformen nicht adäquat repräsentiert und auf dem neuesten Stand gehalten werden. So sind heute wichtige, den Gesamtprozeß betreffende Informationen, wie beispielsweise eine detaillierte, aktuelle Beschreibung der Abfolge einzelner Prozeßvarianten nur schwer und unvollständig verfügbar.

Aus diesem Grund und wegen des Umfangs und der Komplexität des vorliegenden Fertigungswissens ist in Zukunft eine rechnergestützte Repräsentation notwendig. Die Darstellung allgemeinen Fertigungswissens auf dem Rechner setzt ein *Modell* des Prozesses voraus [BRAU93]. Ein geeignetes Modell für die rechnergestützte Repräsentation von Prozeßwissen eines so vielseitigen und komplexen Prozesses wie dem LIGA-Verfahren fehlt bislang. Im folgenden werden die grundsätzlichen Schwierigkeiten bei der Modellbildung für einen komplexen Prozeß näher betrachtet.

Modellierung komplexen Prozeßwissens auf dem Rechner

In den Natur- bzw. Ingenieurwissenschaften werden Modelle mit unterschiedlichen Absichten aufgestellt. In den Naturwissenschaften sucht man in erster Linie Modelle realer Vorgänge, um die beobachteten Phänomene zu *erklären* und zu *verstehen*. So ist es ein Ziel der modernen Physik, ein einheitliches Modell für die Beschreibung der bekannten Wechselwirkungen zwischen Materie zu suchen, um diese dadurch grundlegend zu verstehen. In den Ingenieurwissenschaften dienen Modelle dazu, die Kenntnis der Gesetzmäßigkeiten, die über den Ablauf bestimmter Vorgänge bekannt sind zu *nutzen*, um diese gezielt zu beeinflussen. Beispielsweise ist für die Herstellung einer möglichst spannungsarmen Schicht bei einem Sputtervorgang die Kenntnis des Einflusses bestimmter Parameter wie Temperatur oder Gasdruck auf die Schichteigenschaften notwendig. Die Art des benutzten Modells kann dabei unterschiedlich sein. So kann es in empirischer Form vorliegen (welches etwa durch Messung der Zusammenhänge gewonnen wurde) oder in Form einer analytischen bzw. numerischen Lösung von Differentialgleichungen, die auf der physikalisch-mathematischen Modellierung des Sputtervorganges beruhen.

Die Vorteile der physikalisch-mathematischen Modellierung von Vorgängen bestehen in der erreichbaren Präzision der Vorhersagen und ihrer Objektivierbarkeit durch den zugrundeliegenden mathematischen Formalismus. Die physikalische Modellierung ist dabei jedoch im wesentlichen auf zwei Arten von Systemen beschränkt [KLIR89]:

- Die erste Art von Systemen beschreibt das Verhalten durch die Berücksichtigung weniger entscheidender Systemparameter. Parameter, welche das Systemverhalten nur geringfügig beeinflussen, gehen in die Modellierung nicht ein. Ein Beispiel hierfür ist der freie Fall eines Körpers auf der Erde im Vakuum, der schon durch wenige Parameter und die Newtonschen Axiome sehr genau beschrieben werden kann.

- Bei der zweiten Art von Systemen gehen extrem viele Objekte der Realität in die Beschreibung des Systems ein. Die Annahme der Gleichartigkeit dieser Objekte und der Wechselwirkungen zwischen ihnen ermöglicht jedoch die Beschreibung des Systems mit Hilfe statistischer Methoden. Diese Vorgehensweise führt z.B. in der Thermodynamik zur erfolgreichen Beschreibung von Phänomenen.

Die beiden genannten Arten von Systemen stellen jedoch Extremfälle dar. Viele Vorgänge bei der Durchführung realer technologischer Prozesse sind zu komplex für eine physikalische Modellierung, d.h. es muß eine große Zahl von Parametern bzw. unterschiedlichen Objekten und Beziehungen betrachtet werden. Daraus würden Modellgleichungen resultieren, die mit vertretbarem Aufwand überhaupt nicht mehr oder nur unter sehr einschränkenden Näherungen gelöst werden könnten. Dadurch ginge letztendlich die Aussagekraft des Modells verloren.

Der Mensch kann jedoch auch komplexe technologische Abläufe beschreiben und kontrollieren. Er bedient sich zu diesem Zweck auch unscharfer Modellvorstellungen, die zwar ein weniger tiefes und allgemeingültiges Verständnis und eine weniger genaue Beschreibung ermöglichen, dafür jedoch flexibler und leichter nachvollziehbar sind und auf einen wesentlich größeren Teil des gesamten Prozesses angewendet werden können.

Einsatz wissensbasierter Methoden

Herkömmliche Hilfsmittel für die rechnergestützte Problemlösung sind für die Beschreibung komplexer Systeme schlecht geeignet. So lassen sich mit prozeduralen Programmiersprachen Systeme zwar durch Zerlegung in funktionale Einheiten beschreiben, andere wichtige Systemaspekte, wie die Klassifizierung und Beschreibung der Systembestandteile oder ihrer gegenseitigen Abhängigkeiten lassen sich jedoch nur unzureichend ausdrücken [BOOC91]. Ein wesentlicher Teilbereich der Forschung im Gebiet der Künstlichen Intelligenz (KI) ist daher den Grundlagen *Wissensbasierter Systeme* gewidmet. Definitionen dieses Begriffes sind in der Literatur zahlreich und fallen recht unterschiedlich aus. Gemeinsam ist Wissensbasierten Systemen, daß sie auf menschliches Wissen zugreifen, welches zuvor in eine formalisierte Gestalt überführt wurde, um eine maschinelle Wissensverarbeitung zu ermöglichen. Grundsätzliche Anforderungen an die Repräsentation von Wissen in Wissensbasierten Systemen sind dabei:

- Trennung von problemspezifischem Wissen und aufgabenspezifischem Wissen.
- Flexible Änderbarkeit und Erweiterbarkeit des Wissens durch geeignete Strukturierung der Wissensbasis.
- Leichte Nachvollziehbarkeit der Ableitung geschlossenen Wissens, indem vom System verständliche Erklärungen für die abgeleiteten Ergebnisse gegeben werden.

Da diese Eigenschaften für ein in der Praxis einsetzbares System zur Darstellung von Prozeßwissen unerlässlich sind, wird bei der Modellierung des Prozesses in dieser Arbeit auf wissensbasierte Methoden zurückgegriffen.

1.2 Zielsetzung

Modellierung von Fertigungswissen

Dem Fertigungswissen kommt im Rahmen der Herstellung von Mikrostrukturprodukten eine besondere Bedeutung zu. Zum einen korrelieren die Fertigungsbedingungen eines Produktes eng mit dessen Eigenschaften während des gesamten Produktzyklus. Zum anderen können weitere für die Produktherstellung notwendige Teilaufgaben, wie die Konstruktion oder die Qualitätssicherung, ohne fertigungstechnisches Wissen nicht durchgeführt werden. Die systematische Aufarbeitung von Entwurfs- und Fertigungswissen des LIGA-Verfahrens und anderer Mikrostrukturierungsverfahren ist Gegenstand aktueller Forschungs- und Entwicklungsarbeiten [SENT92, JOHN93].

Ein Modell des Fertigungswissens muß sowohl den Ablauf und die Organisation des Prozesses erfassen, als auch die Integration der verschiedenen Beschreibungsformen von Zusammenhängen zwischen Parametern der Prozeßführung gestatten, um eine tiefere kausale Modellierung zu ermöglichen. Das auf dem Rechner umgesetzte Modell kann in zweierlei Weise genutzt werden:

- Zum einen läßt sich das System aufgrund des über den Prozeßablauf und die Prozeßdurchführung vorhandenen Wissens in Form eines *Informationssystems* nutzen. Der Anwender erhält dabei eine umfassende Beschreibung der Abläufe und gegenseitigen Abhängigkeiten zwischen Fertigungsparametern und Produkteigenschaften und kann die Auswirkungen bestimmter Prozeßeinstellungen anhand von Simulationen ermitteln.
- Zum anderen kann auf das im Rechner vorliegende Modell des Fertigungswissens auch maschinell zugegriffen werden. Damit steht dieses anderen Softwarewerkzeugen ebenfalls direkt zur Verfügung, beispielsweise zur Unterstützung bei der Konstruktion oder Qualitätssicherung. Das System stellt also einen *Baustein für eine durchgängige Rechnerunterstützung* der gesamten Herstellung von Mikrostrukturen dar.

Einbeziehung heuristischen und unscharfen Wissens in die Prozeßbeschreibung

Ein wichtiger Aspekt bei der Modellbildung für komplexe reale Prozesse ist die Berücksichtigung von heuristischem Wissen. Vielfach lassen sich Teilaspekte des Prozesses mit Hilfe von Heuristiken weitaus weniger aufwendig als durch ein physikalisch-mathematisches Modell beschreiben. Dies ist insbesondere dann pragmatisch sinnvoll, wenn andere Teilaspekte des Prozesses, die für die Eigenschaften der Endprodukte wesentlicher sind, noch nicht ausreichend untersucht wurden.

Ein weiterer wichtiger Aspekt der Modellbildung beim Menschen ist die Toleranz von Unschärfe bei der Beschreibung eines Prozesses oder Systems. Ein komplexer Prozeß besitzt im allgemeinen Parameter, deren Werte aufgrund der Abstraktionsstufe, auf welcher er gerade betrachtet wird, nicht exakt angegeben werden können. Die ungenaue Kenntnis dieser Parameterwerte und ihrer Zusammenhänge genügt jedoch in vielen Fällen für die praktische

Durchführung des Prozesses, auch wenn die zugrundeliegenden Wirkmechanismen nicht exakt verstanden sind. In der Arbeit wird gezeigt, wie solche unscharfen Parameter und Zusammenhänge mit Hilfe unscharfer Mengen und der darauf aufbauenden Logik beschrieben werden können.

Simulation von Parameterabhängigkeiten

Für die konkrete Durchführung und Beurteilung von Prozeßvarianten ist die Kenntnis der gegenseitigen Abhängigkeiten zwischen den Parametern der einzelnen Prozeßschritte, welche ein Produkt durchläuft wesentlich. Aufsetzend auf der Beschreibung der scharf oder unscharf formulierten Parameterabhängigkeiten wird das Modell um die Möglichkeit zur Simulation erweitert. D.h. aus der Vorgabe (scharf oder unscharf formulierter) Parametereinstellungen und Randbedingungen und dem im System vorliegenden Wissen können Ergebniswerte für bestimmte Zielparameter ermittelt werden.

Zusammengefaßt besteht das Ziel dieser Arbeit in der Entwicklung eines Modells für die systematische rechnergestützte Erfassung und Darstellung von Fertigungswissen des LIGA-Verfahrens. Das Modell muß die Simulation sowohl scharf als auch unscharf beschriebener Zusammenhänge zwischen Prozeßparametern ermöglichen. Für die Realisierung des Modells sollen wissensbasierte Methoden eingesetzt werden, um das im System vorhandene Wissen in der Praxis optimal handhaben zu können. Zur Überprüfung der Leistungsmerkmale des entwickelten Konzepts muß ein lauffähiges System erstellt und für die Beschreibung realer Prozeßabläufe eingesetzt werden.

1.3 Vorgehensweise

Im nachfolgenden *Kapitel 2* werden zunächst allgemeine Eigenschaften des LIGA-Verfahrens im Hinblick auf den Aufbau eines *informationstechnischen Modells* erarbeitet. Eine detailliertere Analyse beschränkt sich dann auf die Herstellung von Masken für die Röntgenlithographie, die im Vergleich zum Gesamtverfahren zwar einen relativ kleinen, jedoch repräsentativen Verfahrensausschnitt darstellt. Das Ergebnis dieser Untersuchungen führt zu einem Anforderungsprofil an das Prozeßmodell.

In *Kapitel 3* werden *Methoden für die rechnergestützte Repräsentation von Wissen* diskutiert. Bezüglich der aus der Analyse des LIGA-Verfahrens stammenden Anforderungen werden verschiedene wichtige Methoden der Wissensrepräsentation auf ihre Eignung als Grundlage für die Prozeßmodellierung untersucht.

In *Kapitel 4* wird auf der *Grundlage der objektorientierten Wissensrepräsentation* ein Basismodell erarbeitet, welches die wichtigsten Beschreibungselemente des Prozeßwissens und ihre gegenseitigen Beziehungen in enger Analogie zu der realen Sicht auf den Prozeß beschreibt.

Die detaillierte qualitative und quantitative Beschreibung von Parameterzusammenhängen eines technologischen Prozesses setzt voraus, daß nicht nur scharfe sondern auch unscharfe Parameterwerte, Zusammenhänge und Randbedingungen beschrieben werden können. Wegen der engen Verknüpfung scharfer und unscharfer Informationen in der Realität müssen beide in analoger Weise dargestellt und gemeinsam verarbeitet werden können. Deshalb wird in *Kapitel 5* die *Methode zur Repräsentation unscharfen Wissens* näher diskutiert und ihre Eignung zur Beschreibung von Parameterzusammenhängen bei der Modellierung des LIGA-Verfahrens untersucht.

In *Kapitel 6* wird die *Beschreibung von Beziehungen zwischen Prozeßparametern* untersucht, die auf unterschiedliche Art voneinander abhängen können. Anschließend wird die Problematik des Zusammenwirkens von verschiedenen Parameterbeziehungen in Parameterwirkketten behandelt. Darauf aufbauend wird ein Modell zur Simulation von Parameterwirkketten unter Berücksichtigung scharf und unscharf beschriebener Abhängigkeiten entwickelt.

In *Kapitel 7* wird die *Funktionalität des Systems LIMES (LIGA Modelling and Engineering System)* vorgestellt und gezeigt, wie dieses System zur Beschreibung realen Prozeßwissens eingesetzt werden kann. Dazu wird anhand von Beispielen des LIGA-Zwischenmaskenprozesses zunächst die Beschreibung grundlegender Prozeßelemente und ihrer gegenseitigen Beziehungen und anschließend die Simulation von vernetzten Parameterabhängigkeiten aufgrund der im System vorliegenden elementaren Zusammenhänge demonstriert.

2. Analyse des LIGA-Verfahrens und Systemanforderungen

Das LIGA-Verfahren ist ein Verfahren zur Herstellung von Mikrokomponenten für die Mikrosystemtechnik. Daneben existieren jedoch auch andere Möglichkeiten. Die wichtigsten hiervon sind die Verfahren zur mikromechanischen Strukturierung durch anisotropes Ätzen von Silizium und die Silizium-Oberflächenmechanik, mit der dreidimensionale Strukturen durch Entfernung von Opferschichten in einer Schichtfolge hergestellt werden können. Weitere Verfahren sind die am Kernforschungszentrum Karlsruhe entwickelte mechanische Mikrostrukturierung mit Hilfe von Diamantwerkzeugen [BIER88], sowie die aus der Mikroelektronik abgeleitete Dickschichttechnologie zur Herstellung von freitragenden Strukturen [STEC87].

Das LIGA-Verfahren unterscheidet sich von den genannten Verfahren insbesondere durch die große Flexibilität in bezug auf die Wahl von Material und Geometrie des Produktes. Aufgrund dieser Möglichkeiten wird beim LIGA-Verfahren eine größere Bandbreite an technologischen Grundprozessen und Varianten benötigt als bei den anderen Verfahren. Für die Modellierung des Prozeßwissens müssen daher Beschreibungselemente gefunden werden, die einerseits allgemein genug sind, um den Prozeß in seiner gesamten Breite zu erfassen, die andererseits aber auch als Basis einer detaillierten Modellierung dienen können. Dazu ist zunächst eine Analyse des Prozesses notwendig.

2.1 Allgemeine Strukturierung des Prozeßwissens beim LIGA-Verfahren

Das für die Beschreibung und Durchführung des LIGA-Verfahrens benötigte Wissen läßt sich den folgenden Bereichen zuordnen:

- Wissen über die grundsätzliche Organisation und den Ablauf des Prozesses.
- Wissen über die Durchführung des Ablaufs, also Anleitungen, Vorschriften und Einstellungen zur Prozeßdurchführung.
- Wissen über Zusammenhänge zwischen den kontrollierten Prozeßparametern und die Auswirkungen auf die Produkteigenschaften.
- Physikalisches, chemisches und technisches Grundlagenwissen zu den Prozeßabläufen.

In den folgenden Unterabschnitten wird der Prozeß unter den genannten Gesichtspunkten eingehender betrachtet.

2.1.1 Ablauf und Beschreibung der Prozeßdurchführung

In diesem Abschnitt soll untersucht werden, wie der grundsätzliche Prozeßablauf und seine Durchführung beschrieben werden kann.

Der Prozeß beruht trotz seiner Komplexität im wesentlichen auf vier Grundprozessen zum lokalen Aufbau oder Entfernen von Material [LESS92]:

- Lithographie
- Selektives naßchemisches Ätzen von Opferschichten
- Galvanoformung
- Formgebung mit Mutterformen.

Diese Grundprozesse werden dazu genutzt, Zwischenprodukte zu strukturieren und umzukopieren, damit das gewünschte Endprodukt entsteht. Abb. 2.1 gibt einen stark vereinfachten Überblick über das Verfahren.

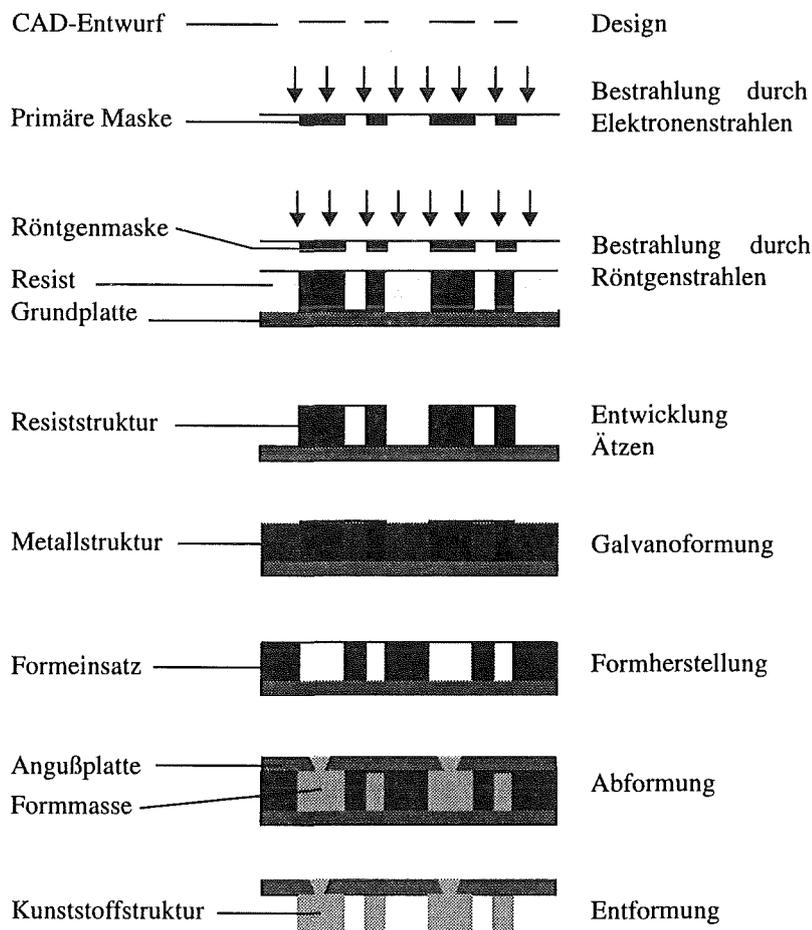


Abb. 2.1 Überblick über den Ablauf des LIGA-Verfahrens. Über die grundlegenden Schritte Lithographie und Galvanik kann direkt ein Produkt oder ein Werkzeug zur Abformung vieler Kunststoff- oder Keramikendprodukte gefertigt werden.

Abb. 2.1 stellt einen grundsätzlichen Prozeßablauf dar. Den Ausgangspunkt bildet die Beschreibung der Geometrie einer Struktur durch ein mit einem CAD-System erstelltes Datenfile. Aus dieser Beschreibung wird mit Hilfe eines Elektronenstrahlschreibers eine primäre Chrommaske hergestellt. Aus der Maske entsteht eine Arbeitsmaske für die Röntgentiefenlithographie. Bei diesem Schritt wird der Resist an den nicht von der Maske abgedeckten Stellen belichtet. Bei der Entwicklung werden die belichteten (oder je nach Resist-Entwicklersystem unbelichteten) Resistteile gelöst. Durch galvanische Auffüllung der entstandenen Hohlräume entsteht ein Formeinsatz, der als Abformwerkzeug für die Herstellung weiterer Kunststoff- oder Keramikprodukte dient.

Im folgenden soll die Problematik einer detaillierten Beschreibung des Prozesses aufgezeigt werden. Dazu wird zunächst näher auf den Prozeß zur Herstellung einer Röntgenzwischenmaske eingegangen. Dieser stellt dabei selbst nur einen Teil der Prozeßfolge dar, die notwendig ist, um von einer primären Maske (aus Chrom) zu einer Röntgenarbeitsmaske zu gelangen (vgl. Abb. 2.1).

Für die Herstellung einer Röntgenzwischenmaske gibt es zwei grundsätzlich verschiedene Vorgehensweisen. Da sich die primäre Chrommaske für die Röntgenlithographie nicht eignet und auch die optische Strukturierung eines dicken Resists mit der für Arbeitsmasken erforderlichen Genauigkeit nicht möglich ist, wird die Arbeitsmaske gewöhnlich über eine Zwischenmaske strukturiert. Die Übertragung der Strukturgeometrie auf die Zwischenmaske kann entweder durch optisches Kopieren einer Chrommaske [SCHO90], oder durch direkte Strukturierung mit Hilfe eines Elektronenstrahlschreibers erfolgen [HEIN92]. Werden sehr kleine Strukturen benötigt, kann auch der sogenannte Tri-Level-Prozeß angewandt werden, bei dem eine dünne Schicht durch reaktives Ionenätzen vorstrukturiert und dann als Maske für die Strukturierung des eigentlichen Resists (Polyimid) im Sauerstoffplasma verwendet wird [MENZ92].

Die eigentliche Herstellung von Zwischenmasken gliedert sich weiter auf in die Herstellung und Vorbereitung des Substrats, die Strukturierung des Resists durch Lithographie, die Galvanik zum Aufbau des Absorbers [MANE88] sowie die Entfernung des bei der Entwicklung übriggebliebenen Resists durch eine Plasmabehandlung.

Auch die zuletzt genannten Abläufe können wiederum verschiedene Alternativen besitzen. So können die Trägerfolien (üblicherweise aus Titan) auf den zuvor massiven Rahmen einer Maske aufgesputtert und anschließend freigeätzt werden, oder in einem Übertragungsverfahren nachträglich auf den Rahmen aufgeklebt werden. Für geringere Genauigkeitsanforderungen können die Masken auch aus einer Polyimidfolie bestehen, die über einen Rahmen aus Kupfer oder Glas eingespannt ist [MÜNCH87].

Die genannten Abschnitte der Maskenherstellung lassen sich außerdem weiter verfeinern. So läßt sich z.B. das Aufbringen des Resists in zwei aufeinanderfolgende Schritte zur Belackung mit anschließenden Temperschritten und eine abschließende visuelle Kontrolle zerlegen. Solche elementaren Schritte beschreiben die physikalischen oder chemischen Vorgänge, die zum Erreichen bestimmter Eigenschaften des Produktes oder Zwischenproduktes (Dicke und Homogenität eines Resists bei der Belackung) bei vorgegebenen Parameterwerten und Randbedingungen ablaufen müssen.

Bereits die Beschreibung dieses Teilausschnitts des Prozesses zeigt die Schwierigkeit, das Prozeßgeschehen übersichtlich zu dokumentieren. Deshalb ist eine systematische Strukturierung notwendig.

Der Gesamtprozeß stellt eine Folge von Abläufen dar. Die einzelnen Abläufe setzen sich dabei selbst wieder aus einer Folge von Abläufen oder elementaren Prozeßschritten zusammen. Dies kann so lange fortgesetzt werden, bis alle Abläufe durch elementare Prozeßschritte beschrieben sind. Jeder Ablauf oder elementare Prozeßschritt kann über Alternativen verfügen. Abb. 2.2 veranschaulicht das Modell, in welchem die wesentlichen Informationen zum Prozeß beschrieben werden können. Diese betreffen zunächst den zeitlichen Ablauf, die möglichen Varianten und den unterschiedlichen Abstraktionsgrad, unter welchem die einzelnen Prozeßschritte betrachtet werden.

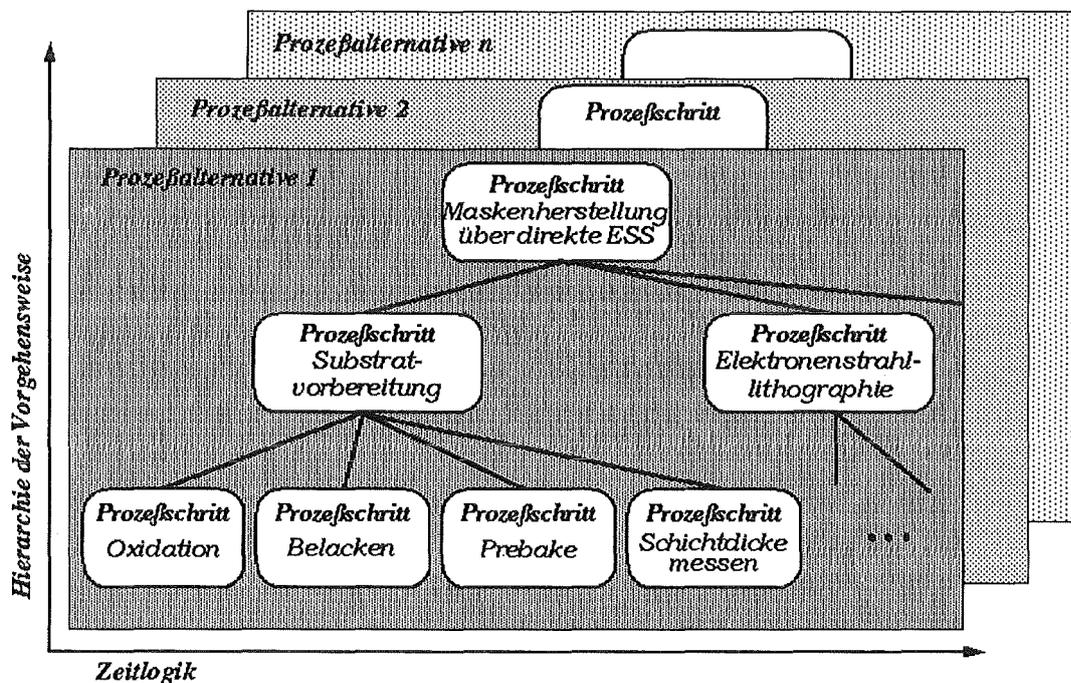


Abb. 2.2 Strukturierung des Prozesses durch Prozeßschritte.

Weitere wichtige Informationen betreffen die benötigten Arbeitsgegenstände und die bei der Fertigung eingesetzten Arbeitsmaterialien. Diese werden sinnvollerweise den entsprechenden Prozeßschritten zugeordnet. Auf diese Weise entsteht ein (Semantisches) Netz, dessen Knoten von den Beschreibungselementen Prozeßschritte, Arbeitsmaterialien und Ausrüstungsgegenstände gebildet werden, und dessen Kanten die Beziehungen zwischen diesen Beschreibungselementen ausdrücken. Dies ist der erste Schritt für eine formale Erfassung des Prozeßablaufes.

Fixierte Beschreibungen der Durchführung des LIGA-Verfahrens liegen vor allem in Form schriftlicher Dokumente vor.

Wichtig sind dabei die *Prozeßvorschriften*, welche in komprimierter Form die Prozeßdurchführung durch die Angabe von Arbeitsvorschriften, Parametereinstellungen, verwendeten Arbeitsmaterialien sowie Besonderheiten und Vorsichtsmaßnahmen beschreiben. Bisher liegen nur für einen Teil des LIGA-Prozesses genaue Prozeßvorschriften vor. Wichtige Aspekte, die bei der rechnergestützten Modellierung beachtet werden müssen, sind:

- Die genauen Parametereinstellungen und Zusammenhänge sind in vielen Fällen nur dem für die Prozeßdurchführung verantwortlichen Personenkreis bekannt.
- Zusätzliches Wissen existiert darüberhinaus in schriftlicher Form oder als Fachwissen des Personals über die Bedienung und den Umgang mit komplexen Anlagen, wie z.B. dem Elektronenstrahlschreiber oder den Anlagen zum Trockenätzen.
- Im Falle von Abweichungen und Schwierigkeiten bei der Prozeßführung muß auf vertiefende, das jeweilige Grundlagenwissen einbeziehende Informationen zurückgegriffen werden.
- Diese Informationen sind in den internen und externen Veröffentlichungen der durchgeführten Forschungs- und Entwicklungsarbeiten und als Grundlagenwissen der am Prozeß beteiligten Fachleute vorhanden.

2.1.2 Beziehungen zwischen Einstellparametern und Produkteigenschaften

Voraussetzung für ein tiefergehendes Modell des Prozesses ist die Erfassung der Beziehungen zwischen den Prozeßparametern der einzelnen Prozeßschritte.

Damit kann die Frage nach den Auswirkungen von Änderungen der Werte von Prozeßparametern beantwortet werden. Dieses Wissen wird für die praktische Prozeßdurchführung, bei der Konstruktion und Qualitätssicherung, bei der Planung und bei der Suche nach Fehlerursachen benötigt. Der vorliegende Abschnitt behandelt anhand konkreter Prozeßbeispiele die Beziehungen zwischen den Prozeßparametern.

Unter einer *Beziehung* wird hier die Beschreibung der Abhängigkeit eines Zielparameters von den ihn *direkt* beeinflussenden Parametern verstanden. Im folgenden werden also die Abhängigkeiten zwischen den kontrollierten einflußnehmenden Parametern und den von ihnen direkt beeinflussten Zielparametern, die auch Produkteigenschaften beschreiben können, näher untersucht.

Für eine Reihe von Einzelschritten und Verfahren existieren bereits grundlegende physikalisch-mathematische Modelle, teilweise werden auch bereits rechnergestützte Simulationswerkzeuge eingesetzt [HEIN92]. Bei der Herstellung von LIGA-Strukturen werden jedoch viele prinzipiell unterschiedliche Verfahren hintereinander angewandt. Ein großer Teil beruht auf Zusammenhängen, denen keine physikalisch-mathematischen, sondern heuristische Modelle oder empirische Annahmen zugrundeliegen. Ferner können Abhängigkeiten zwischen den unterschiedlichen Verfahren bestehen, welche ebenfalls nur auf heuristischer oder empirischer Grundlage bekannt sind.

Um das Wissen über die Parameterabhängigkeiten zu strukturieren, wird zunächst eine Reihe konkreter Zusammenhänge betrachtet. Daraus kann eine abstrakte Klassifizierung der Typen der Parameterwerte abgeleitet werden, durch die sich die realen Abhängigkeiten der Parameter beschreiben lassen.

Betrachtet man den elementaren Prozeßschritt der Belackung einer Maskenträgerfolie mit einem Resist, so stellt die Dicke der aufgeschleuderten Schicht den Zielparameter einer Beziehung dar. Wie aus Abb. 2.3 ersichtlich ist, kann der Zusammenhang zwischen der „Schichtdicke“ und ihren Einflußparametern „Feststoffgehalt“ und „Spincoatfrequenz“ in mathematischer Form durch eine parameterisierte Kurvenschar beschrieben werden. Hier ergibt sich der Wert des Zielparameters durch einen numerischen Wert (nämlich den Wert einer Messung).

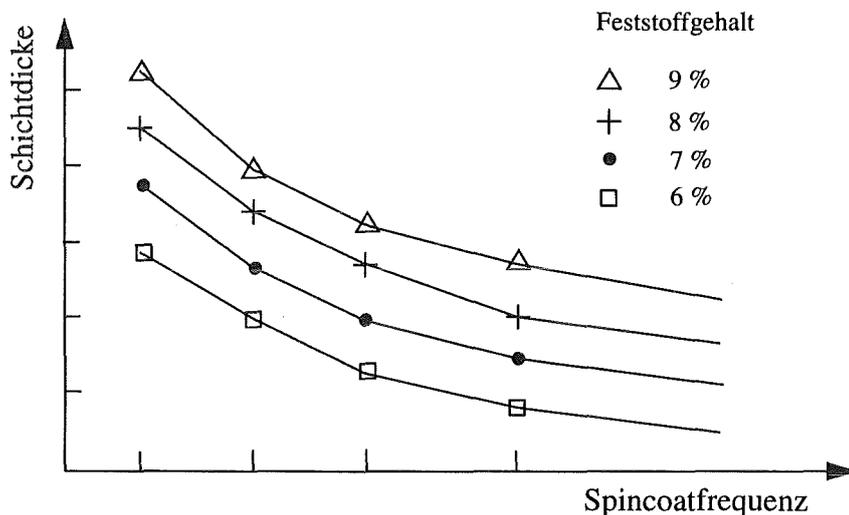


Abb. 2.3 Abhängigkeit der Beschichtungsdicke von der Frequenz beim Aufschleudern einer Resistschicht für unterschiedliche Feststoffanteile.

Die Werte der Einflußparameter in Abb. 2.4 lassen sich durch numerische oder linguistische Werte beschreiben. Der Wertebereich der numerischen Parameter kann dabei entweder kontinuierlich (Spincoatfrequenz) oder diskret (Molekulargewicht) sein, womit ausgedrückt wird, daß nur Resists mit bestimmtem Molekulargewicht vorkommen können, während die Belackungsfrequenz innerhalb vorgegebener Grenzen frei variieren kann. Die Beschreibung des Zusammenhanges zwischen den Parametern erfolgt im Fall diskreter linguistischer und numerischer Parameter in Form von Aufzählungen, bei kontinuierlichen Parametern durch kontinuierliche Funktionen.

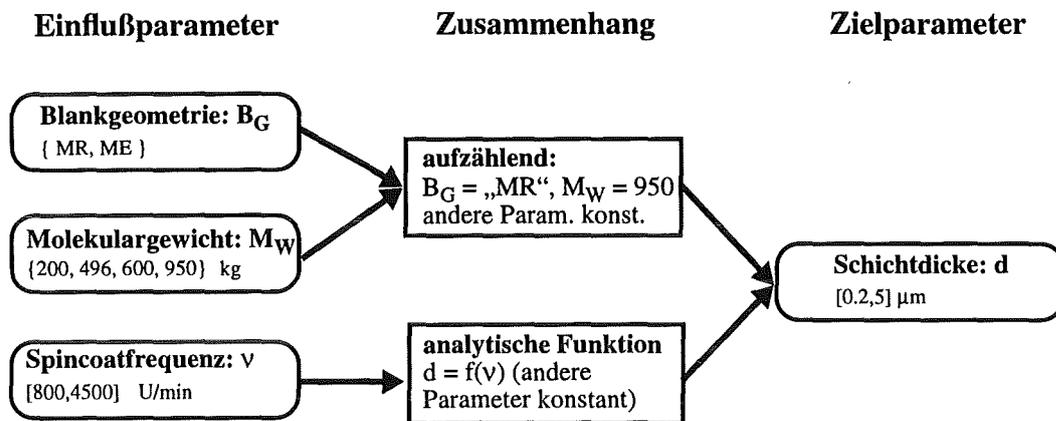


Abb. 2.4 Einfluß von scharf charakterisierten Eingangsparametern unterschiedlichen Typs auf die Schichtdicke beim Belacken einer Maske.

Ein anderes Beispiel einer Beziehung stellt die Beeinflussung der Homogenität der aufgetragenen Resistschicht dar (Abb. 2.5). Dieser Parameter kann aufgrund des vorhandenen Prozeßwissens nur durch einen unscharfen Wertebereich beschrieben werden. Dabei kann die Homogenität einer Schicht beispielsweise als „akzeptabel“ oder „schlecht“ eingestuft werden. Durch den linguistischen Ausdruck „akzeptabel“ wird dabei nicht ein einziger Wert,

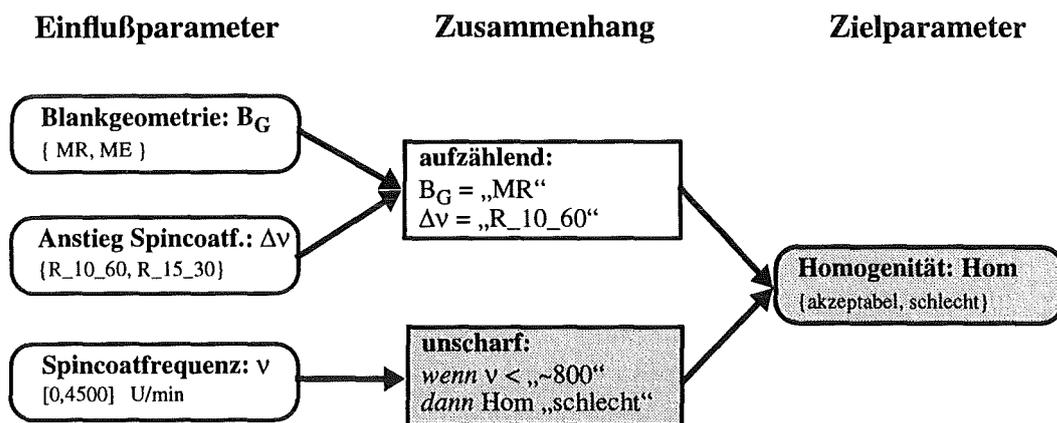


Abb. 2.5 Scharfe (weißer Kasten) und unscharfe (grauer Kasten) Zusammenhänge zwischen scharf charakterisierten Eingangsparametern und der Homogenität der Resistschicht beim Belacken einer Maske.

sondern ein ganzes Intervall numerischer Werte bezeichnet, welche ein akzeptables Homogenitätsmaß darstellen.¹ Der Zusammenhang wird dann wie bei den diskreten scharfen Parameterwerten durch Relationen beschrieben. Die Relationen drücken diesmal jedoch Beziehungen zwischen Wertemengen anstatt zwischen einzelnen Werten aus.

Im letzten Beispiel wird die Oxidation einer Maskenträgerfolie betrachtet. Dabei werden sowohl die Einflußparameter als auch der Ergebnisparameter durch unscharfe Werte charakterisiert. In Abb. 2.6 ist der Einfluß der Oxidationszeit und Temperatur auf die Färbung einer Titan-Maskenträgerfolie dargestellt, die wiederum Rückschlüsse auf die Haftung der anschließend aufgebauten Galvanikstrukturen ermöglicht.

Im Gegensatz zum Beispiel aus Abb. 2.5 werden diesmal ausschließlich Parameter betrachtet, die durch unscharfe Werte charakterisiert werden. Die zur Beschreibung des Zusammenhanges zwischen den Parametern benutzten Aufzählungen lassen sich in diesem Fall als unscharfe Regelzusammenhänge auffassen (vgl. Abschnitt 5.2).

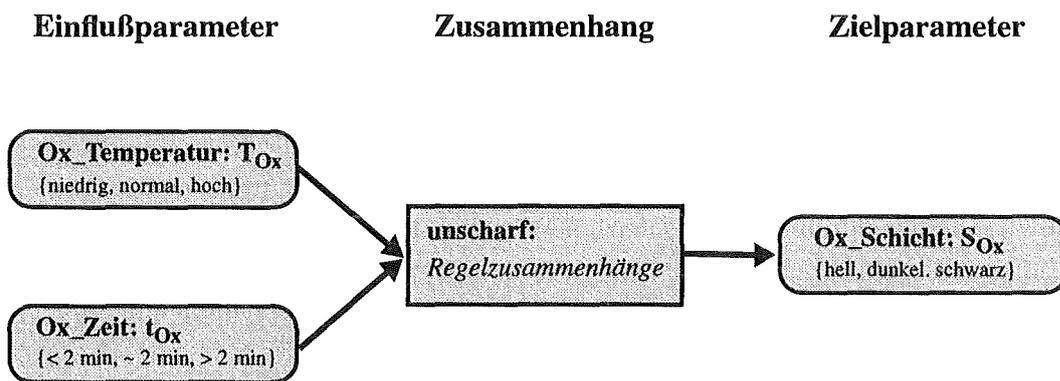


Abb. 2.6 Einfluß von unscharf charakterisierten Eingangsparametern unterschiedlichen Typs auf die Färbung der Oxidschicht bei der Oxidation einer Maskenträgerfolie.

Bei allen aufgeführten Beispielen ist zu beachten, daß außer den aufgeführten noch weitere Parameter zur Beschreibung einer Beziehung zu beachten sind. Die genaue Zahl dieser weiteren Parameter hängt von dem speziell betrachteten Zielparameter, der Genauigkeit der Modellierung und dem Wissen über die betrachteten Beziehungen ab. Selten übersteigt sie jedoch die Zahl von 6 - 8 direkten Einflußparametern auf einen Zielparameter, da eine ausführliche Berücksichtigung von noch mehr Parametern sowohl experimentell als auch in theoretischen Modellen äußerst schwierig ist und eine Beschränkung auf die wesentlichen Einflüsse im allgemeinen zu realistischeren Modellen führt.

1. Dieses kann z.B. aus der Messung der Schichtdicke an verschiedenen Stellen der Maske gewonnen werden.

2.2 Anforderung an die rechnergestützte Modellierung des LIGA-Verfahrens

Eine primäre Anforderung an die rechnergestützte Modellierung ist es, die Komplexität des Verfahrens durch eine geeignete systematische Repräsentation des Prozeßwissens handhabbar zu machen. Die Analyse des Prozesses und der Parameterabhängigkeiten im letzten Abschnitt läßt die Komplexität des LIGA-Verfahrens erkennen. Diese ist eine Folge grundlegender Prozeßeigenschaften:

- Der Ablauf des Verfahrens wird durch eine große Zahl¹ verschiedener sequentiell ablaufender Prozeßschritte und Prozeßvarianten bestimmt.
- Die Zustände des Prozesses und die resultierenden Produkteigenschaften hängen von den Wechselwirkungen einer Vielzahl von Prozeßparametern ab, wobei der gegenseitige Einfluß teilweise nur ungenau und unvollständig bekannt ist.
- Es bestehen außerdem Abhängigkeiten zwischen den Parametern und Ergebnissen unterschiedlicher Prozeßschritte über verschiedene Prozeßschritte hinweg.

In den folgenden Abschnitten werden die aus der Analyse resultierenden Systemanforderungen zusammengefaßt.

2.2.1 Grundlegendes Modell und Anforderungen an die Wissensverwaltung

Die Strukturierung des Prozeßablaufs ist eine Voraussetzung für jede weitergehende wissensbasierte Unterstützung des Prozesses. Durch die Strukturierung müssen die zeitliche Abfolge, der Abstraktionsgrad und die Prozeßvarianten erfaßt werden. Darüber hinaus müssen weitere Elemente zur Beschreibung des Prozeßablaufs wie Prozeßparameter, Materialien und Ausrüstungsgegenstände innerhalb der Ablaufstruktur beschrieben werden können. Dadurch entsteht ein Basismodell, welches als Grundlage für eine standardisierte, vollständige Beschreibung der Prozeßabfolge und -durchführung dient.

Da ständig neue Prozeßvarianten für das LIGA-Verfahren entwickelt und bestehende Varianten weiterentwickelt und verbessert werden, muß das Modell eine hohe Flexibilität im Umgang mit dem Prozeßwissen ermöglichen. Insbesondere muß das Modell die Akquisition, Fortschreibung und Erklärung der Wissensinhalte des Systems unterstützen. Diese Anforderungen betreffen, im Gegensatz zu den bisher aufgestellten, nicht das im System repräsentierte Prozeßwissen, sondern dessen Handhabung und Präsentation.

Unter diesen Gesichtspunkten ist auch die Änderbarkeit und Fortschreibung der Wissensinhalte, für die in Wissensbasierten Systemen im allgemeinen eine eigene Wissenserwerbskomponente vorgesehen ist, von Bedeutung. Der Aufwand für die Übertragung des über den Prozeß vorhandenen Wissens in das Prozeßmodell soll möglichst gering sein. Das Modell muß aus diesem Grund die Beschreibung des Wissens in einer dem Prozeßumfeld entnommenen Sprache ermöglichen. Dadurch wird ein mit Hilfe eines graphisch-interaktiv geführten Dialoges durchgeführter Wissenserwerb des Systems möglich.

1. Für ein Produkt, welches eine vollständige Prozeßkette einschließlich der Kunststoffabformung durchläuft, können weit über 100 Prozeßschritte benötigt werden.

2.2.2 Repräsentation von Oberflächenwissen und Tiefenwissen

Um von der Durchführung des Prozesses zu den Eigenschaften des Produktes zu gelangen, muß detailliertes Wissen über die Werte der Prozeßparameter und ihre gegenseitige Beeinflussung verfügbar sein. Dieses Wissen wird in Anlehnung an [JUNG92] als *Oberflächenwissen* bezeichnet. Das Oberflächenwissen beschreibt also Zusammenhänge, nicht aber die Grundlagen, die zu deren Ermittlung bzw. Herleitung notwendig sind. Die Abhängigkeiten der Prozeßparameter müssen im System so repräsentiert sein, daß es möglich ist, aufgrund der Beschreibung der Zusammenhänge zwischen den Parametern die Ergebnisse bestimmter Parametereinstellungen vorherzusagen oder abzuschätzen. Damit läßt sich schneller und zuverlässiger als bisher ermitteln, wie auf prozeßbedingte Schwankungen und Abweichungen reagiert werden muß, um gleichmäßige zufriedenstellende Prozeßergebnisse zu erreichen.

Für die Behandlung unerwarteter Probleme bei der Durchführung des Prozesses reicht die Kenntnis der gegenseitigen Abhängigkeiten der Prozeßparameter ohne zusätzliche Informationen nicht aus. In diesem Fall muß weitergehendes Wissen über die physikalischen und chemischen Grundlagen der Prozeßabläufe vorhanden sein. Dieses Wissen wird im folgenden als *Tiefenwissen* [JUNG92] bezeichnet. Die Beherrschung von Tiefenwissen erfordert den kreativen Umgang mit vorhandenem Wissen und die Reaktion auf unvorhersehbare Situationen, Leistungen, die im erforderlichen Ausmaß nur von Prozeßfachleuten erbracht werden können. Die Unterstützung des Benutzers durch Tiefenwissen muß sich daher auf die strukturierte Darstellung schriftlicher oder graphischer Information und zusätzlicher Verweise auf Quellen beschränken.¹

2.2.3 Einbeziehung von unscharfem Wissen

Die Diskussion in Abschnitt 2.1.2 zeigte, daß für die Beschreibung der Prozeßzustände und ihre gegenseitigen Abhängigkeiten die Berücksichtigung von Parameterwerten und Zusammenhängen unterschiedlichen Typs notwendig ist. In der Realität treten oftmals Größen als Einflußfaktoren oder Randbedingungen auf, deren genaue Auswirkung auf den Prozeß nicht bekannt sind. Eine Vernachlässigung dieser Größen ist jedoch in vielen Fällen nicht zu vertreten, da sonst aufgrund nichtberücksichtigter Größen unerklärliche Abweichungen der Ergebnisse zustande kommen können. Daher ist die kombinierte Darstellung und Verarbeitung scharfer und unscharfer Zusammenhänge in einem einheitlichen Modell notwendig. Darüber hinaus ermöglicht die Einbeziehung unscharfen Wissens auch die Abschätzung von Auswirkungen bisher nicht exakt untersuchter Zusammenhänge aufgrund von Erfahrungswissen und heuristischen Modellen.

1. Im Prozeßmodell von LIMES ist vorgesehen, Textstellen und graphische Informationen an die Objekte zur Beschreibung des Prozesses anzubinden. Eine zusätzliche Organisation dieser Informationen durch ein Hypertext System wäre an dieser Stelle ebenfalls von Nutzen.

3. Grundlagen der Prozeßmodellierung

Im vorhergehenden Kapitel wurden semantische Eigenschaften des Prozeßwissens beim LIGA-Verfahren untersucht. Dabei wurde zugunsten inhaltlicher Gesichtspunkte weniger auf die formale Organisation des Wissens geachtet. Für die Realisierung eines Prozeßmodells auf dem Rechner ist jedoch eine geeignete formale Methode der *Wissensrepräsentation* eine unerläßliche Voraussetzung. Im folgenden werden verschiedene Methoden auf ihre Eignung für die Modellierung des LIGA-Verfahrens untersucht. Es zeigt sich dabei, daß die objektorientierte Vorgehensweise als methodische Basis die Voraussetzungen für die allgemeine Beschreibung von Prozeßwissen in einem praxisgerechten System am besten erfüllt. Für die Modellierung bestimmter Teilaspekte erscheint jedoch eine regelbasierte Wissensrepräsentation geeigneter. Im zweiten Abschnitt des Kapitels wird basierend auf einem objektorientierten Entwurf ein Basismodell entwickelt, das als Grundlage für die Beschreibung wesentlicher qualitativer und quantitativer Aspekte des Prozeßwissens dient.

3.1 Methoden zur Wissensdarstellung auf dem Rechner

Ein wichtiger Grundgedanke wissensbasierter Systeme ist, Wissen nicht gemäß seines Inhaltes sondern durch die Strukturen, in denen es repräsentiert wird, zu kategorisieren. Dadurch lassen sich allgemeine Strategien verwenden, um Wissen unabhängig von seinem Inhalt darstellen und weiterverarbeiten zu können. Um eine Wissensbasis für das LIGA-Verfahren zu erstellen, ist eine sorgfältige Auswahl der Methode zur Wissensstrukturierung erforderlich. In welcher Weise das Wissen organisiert ist, hat entscheidende Auswirkungen auf so wesentliche Gesichtspunkte wie die Wissenserfassung, die Effizienz der Wissensdarstellung und Verarbeitung sowie die Wartbarkeit und Benutzerfreundlichkeit des Gesamtsystems. Daher werden im folgenden die wichtigsten Methoden der Wissensrepräsentation eingehender diskutiert.

3.1.1 Semantische Netze

Semantische Netze wurden ursprünglich zur Beschreibung von natürlichsprachlichen Sätzen in der Spracherkennung verwendet [SOWA87]. Das Wissen wird in Form gerichteter Graphen dargestellt, deren Knoten bestimmte Entitäten und deren Kanten Relationen zwischen den Entitäten darstellen. Die Kanten der Netze können beschriftet sein, um verschiedenartige Beziehungen zwischen den Knoten zu kennzeichnen. Die Bedeutung der Kanten kann prinzipiell frei gewählt werden. Besondere Bedeutung besitzen jedoch exakt definierte allgemeine Relationen zwischen den Knoten. Die Eigenschaften dieser Relationen lassen sich gesondert untersuchen und ihre Mechanismen unabhängig von einer speziellen Anwendung festlegen. Die beiden wichtigsten Vertreter hiervon sind die Inklusion \subseteq (Teilmengenbeziehung) und die Elementbeziehung \in .

Die *Inklusionsbeziehung* drückt eine Teilordnung der verbundenen Knoten bezüglich der Spezialisierung ihrer Beschreibung aus. Daher müssen Elemente der Untermenge nur einmal aufgezählt werden, die Obermengen können dann durch die Angabe der zusätzlichen Elemente und der Inklusionsbeziehungen vollständig beschrieben werden.

Im Gegensatz dazu drückt die *Elementbeziehung* aus, daß ein Knoten ein konkretes Element einer ebenfalls als Knoten repräsentierten Menge darstellt. Von einem Knoten, der eine Menge repräsentiert, können mehrere Elementkanten ausgehen. Der Vorteil dabei ist, daß gemeinsame Eigenschaften der Elemente der Menge zugeordnet werden und nicht für jedes Element einzeln angegeben werden müssen. Dies erhöht die Übersichtlichkeit und vermindert den Aufwand bei Änderungen von Eigenschaften der Mengen durch die Elimination von Redundanzen.

Mechanismen zur Ableitung neuer Schlußfolgerungen aus dem im Netz ausgedrückten Wissen sind in der Wissensdarstellung durch Semantische Netze nicht enthalten. Sie müssen zusätzlich beispielsweise in Form von Such- und Abfragestrategien, als prozedurale Erweiterungen oder Constraints hinzugenommen werden. Dies ist häufig bei Rahmen (frames) [RATH87] der Fall.

Für das Prozeßwissen besonders wichtig ist der Vorteil Semantischer Netze, die Komplexität in der Darstellung durch die Vermeidung von Redundanzen möglichst gering zu halten. Dies gilt insbesondere, wenn die Beschreibungselemente zu Mengen mit gemeinsamen Eigenschaften zusammengefaßt werden können. So läßt sich der Prozeßablauf in Prozeßschritte unterteilen, die durch gemeinsame Attribute und gemeinsame Beziehungen zu anderen Beschreibungselementen charakterisiert werden. Diese Gemeinsamkeiten müssen in einem Semantischen Netz nur einmal festgelegt werden. Die Beschreibung der großen Zahl von Prozeßschritten, die eine Prozeßvariante ausmachen, besteht dann lediglich in der Angabe der für den betrachteten Schritt spezifischen Eigenschaften und Beziehungen zu anderen Beschreibungselementen. Ein weiterer Vorteil Semantischer Netze ist die Repräsentation des Wissens durch Graphen, die die direkte Umsetzung in eine übersichtliche und anschauliche Darstellung der Zusammenhänge erlaubt.

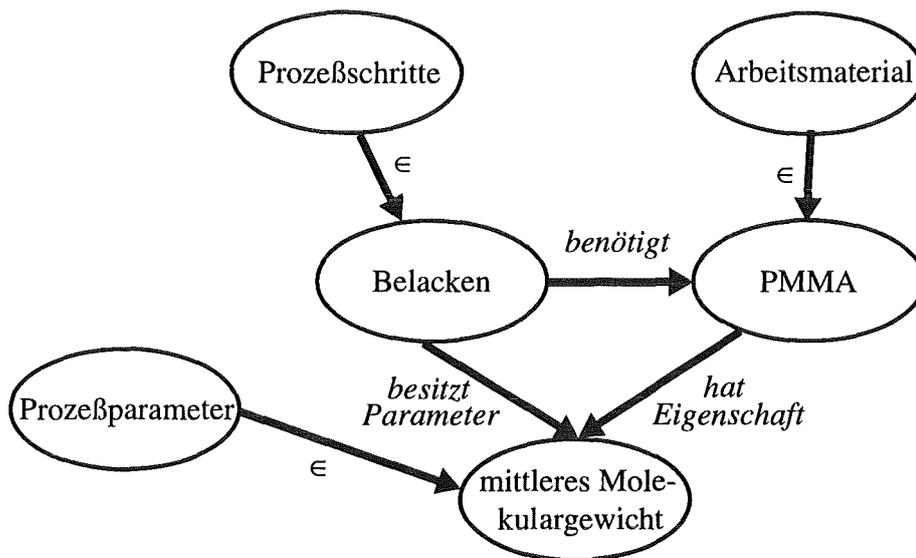


Abb. 3.1 Semantisches Netz für die Beschreibung des Zusammenhanges einiger Grundelemente der Prozeßbeschreibung.

Abb. 3.1 zeigt, wie die Einbettung eines Prozeßparameters (hier das mittlere Molekulargewicht eines Resists) in den Prozeßkontext mit Hilfe eines Semantischen Netzes beschrieben werden kann. Der Abbildung läßt sich entnehmen, daß das mittlere Molekulargewicht ein Prozeßparameter ist, welcher bei einem speziellen Prozeßschritt (der Belackung einer Maskenträgerfolie) eine Rolle spielt. Weiter ist zu ersehen, daß das mittlere Molekulargewicht gleichzeitig die Eigenschaft eines bestimmten Arbeitsmaterials (PMMA¹) ist, welches bei der Belackung benötigt wird.

Insgesamt sind Semantische Netze geeignet, die große Menge qualitativer Daten, die für die Prozeßbeschreibung notwendig ist, zu strukturieren. Quantitative numerische und vage Beziehungen durch Semantische Netze auszudrücken, ist weniger erfolgversprechend. In diesem Fall sind die Beziehungen zwischen den einzelnen Elementen nicht gleichartig und können daher nicht durch so allgemeine mathematische Strukturen wie Inklusions- und Elementbeziehungen beschrieben werden. Semantische Netze sind also zur Modellierung der Organisationsstruktur des Prozeßwissens gut geeignet, nicht jedoch für die Beschreibung quantitativer Abhängigkeiten im Prozeß.

3.1.2 Das Relationale Datenmodell

Das Relationale Datenmodell geht auf Codd [CODD70] zurück und besitzen heute große praktische Bedeutung. Es beschreibt Wissen in Form von Relationen, die in Tabellenform dargestellt werden können. Die Relationen bilden das *Datenbank-Schema*, welches die Beschreibungselemente durch eine Anzahl von Attributen charakterisiert. Die Bezeichnungen der Attribute beschriften die Spalten der Tabellen. Die einzelnen Zeilen beschreiben bestimmte Ausprägungen aller Attribute und stellen damit die Repräsentanten eines Schemas dar.

Im allgemeinen eignet sich das relationale Modell für die Darstellung großer Mengen einfach und regelmäßig strukturierter Daten. Prozeßwissen besteht im Gegensatz dazu jedoch aus komplex strukturierten Beschreibungselementen. Ein Prozeßschritt wird nicht nur durch einfache Attribute wie seine Bezeichnung, sondern auch durch zusammengesetzte Strukturen charakterisiert, z.B. einen Datentyp zur Beschreibung der Dokumentation des Prozeßschrittes. Ein solcher Typ kann selbst wieder eine Struktur aufweisen, in welcher z.B. das Datenformat oder Eigenschaften der Darstellung des Dokumentes beschrieben werden.

Das relationale Datenmodell weist grundsätzliche Schwierigkeiten bei der Repräsentation solcher komplex strukturierter Datentypen auf. Beim Aufstellen der Relationen sind nicht nur inhaltliche Zusammenhänge von Attributen zu beachten. Zur Vermeidung von Anomalien, die z.B. im Zusammenhang mit Redundanzen auftreten können, muß darüberhinaus eine Normalisierung [LOCK87] durchgeführt werden.

Diese für die Modellierung wichtige Problematik wird im folgenden anhand von Abb. 3.2 und Abb. 3.3 erläutert. Das Problem der in Abb. 3.2 gezeigten Relation liegt darin, daß das Attribut „Unterprozeßschritt“ für die Oxidation mehrfach (redundant) angegeben werden muß. Soll für die Oxidation nun ein Unterprozeßschritt eingetragen oder geändert werden,

1. Das Resistmaterial PMMA oder Polymethylmethacrylat ist auch unter dem Namen Plexiglas bekannt.

so muß dies zweimal geschehen, da der Prozeßschritt Oxidation in der Relation zweimal vorkommt. Diese Schwierigkeit läßt sich durch eine Überführung der Relation in die zweite Normalform¹ beseitigen, die in Abb. 3.3 dargestellt ist. Diesmal muß für den Eintrag eines Unterprozeßschrittes für die Oxidation nur eine Stelle der Relation „Verfeinerung“ aus Abb. 3.3 geändert werden. Das Beispiel zeigt also, daß bei der Durchführung der Normalisierung im allgemeinen semantisch zusammengehörige Attribute auf verschiedene Relationen verteilt werden.

Relation : Prozeßschritte

Name	Unterprozeßschritt	Ausrüstungsgegenstand	Dokumentation
Oxidation	keine	Hotplate	IMT_104-12
Oxidation	keine	Magnetrührer	IMT_104-12
Vorbereitung	Oxidation	s. Teilschritte	IMT_102-16
...			

Abb. 3.2 Stark vereinfachte Darstellung eines Prozeßschrittes in einer Relation in erster Normalform. Die Relation ist nicht in zweiter Normalform, da mehrere Ausrüstungsgegenstände für einen einzigen Prozeßschritt vorkommen können.

Relation : Verfeinerung

Name	Unterprozeßschritt	Dokumentation
Oxidation	keine	IMT_104-12
Vorbereitung	Oxidation	IMT_102-16
...		

Relation : Gegenstandszuordnung

Name	Ausrüstungsgegenstand
Oxidation	Hotplate
Oxidation	Magnetrührer
...	

Abb. 3.3 Bereits die Überführung der Relation Prozeßschritte aus Abb. 3.2 in die zweite Normalform erfordert eine Aufspaltung in zwei Relationen.

1. Normalformen unterschiedlicher Stufe im relationalen Datenmodell dienen zur Beseitigung unterschiedlicher Anomalien, die beim Datenbankentwurf auftreten können.

Bei der Normalisierung größerer relationaler Datenmodelle unter Verwendung spezieller Regeln für die Normalisierung werden teilweise sogar Attribute völlig unterschiedlicher Objekttypen in einer einzigen Relation vermengt [HEUE92]. Dies führt einerseits dazu, daß das Datenbank-Schema unübersichtlich und schwer wartbar wird und außerdem die Effizienz von Anfrage- und Updateoperationen¹ abnimmt, da zunächst die Semantik der dargestellten Informationen aus den verschachtelten Relationen umständlich wiederhergestellt werden muß.

Semantische Beziehungen zwischen Attributen können im relationalen Modell nur dadurch ausgedrückt werden, daß bestimmte Attribute gleichzeitig in mehreren Relationen vorkommen. So besteht eine Verbindung zwischen den Relationen „Verfeinerung“ und „Gegenstandszuordnung“, indem das Attribut „Name“ in beiden Relationen vorkommt. Dies ist die einzige Möglichkeit, Beziehungen zwischen den durch die Relationen modellierten Entitäten darzustellen.

Bei der Ausführung von Anfrage- oder Updateoperationen werden jedoch unterschiedliche Beziehungen benötigt. Beispielsweise hat das Löschen eines Prozeßschrittes verschiedene Auswirkungen auf die ihm zugeordneten unterschiedlichen Komponenten. Ein Ausrüstungsgegenstand kann auch unabhängig von einem Prozeßschritt existieren. Die Entfernung eines Prozeßschrittes zieht dagegen zwangsläufig auch die Entfernung aller Verfeinerungen dieses Prozeßschrittes mit sich. Da dem relationalen Modell in der Grundstruktur die Möglichkeit fehlt, zwischen verschiedenen Arten semantischer Beziehungen zu unterscheiden, müssen diese ohne Beziehung zum Datenmodell bei den Anfrage- oder Updateoperationen hinzugefügt werden.

Zusammenfassend läßt sich feststellen, daß das relationale Modell keine geeignete Grundlage für eine wissensbasierte Prozeßmodellierung darstellt, da die einheitliche Darstellung von Beziehungen, die eine Stärke des Modells bei der Modellierung weniger komplex strukturierter Informationen ist, bei der Anwendung auf komplexe Datenstrukturen zu einem Verlust an Übersichtlichkeit und Effizienz der Repräsentation führt.

3.1.3 Regelbasierte Darstellung

In der regelbasierten Darstellung wird Wissen in Form von Regelmengen ausgedrückt. Dies sei an einem einfachen Beispiel aus der Maskentechnik erläutert:

Regelprämisse	{	<i>Wenn</i>	Innere Spannung der Maskenmembran zu niedrig.
		<i>und</i>	Lack aufgebracht.
		<i>und</i>	Lack getempert.
Regelkonklusion	{	<i>Dann</i>	Durchbiegen der Membran.

1. Im Bereich relationaler Datenbanken versteht man unter Update die Operationen Einfügen, Löschen und Ändern, im sonstigen Sprachgebrauch wird der letzte Begriff oft als Update bezeichnet.

Eine Regel besteht demnach aus einem Bedingungsteil, welcher die *Regelprämisse* enthält, und einem *Konklusionsteil*. Die Regelprämisse kann sich aus mehreren, durch logische Operatoren verknüpften atomaren Bedingungen zusammensetzen. Der Konklusionsteil kann ebenfalls aus mehreren *und*-Verknüpften Schlußfolgerungen bestehen. Sind bestimmte Fakten gegeben, die alle Voraussetzungen des Bedingungsteils der Regel erfüllen, so treffen die Folgerungen im Konklusionsteil der Regel ebenfalls zu.

Verschiedene Regeln können voneinander abhängen, indem sich ihre Bedingungs- und Konklusionsteile überschneiden. Für die Verarbeitung solcher Regelsysteme existieren zwei verschiedene Strategien:

- Beim sogenannten *Vorwärtsschließen* wird die Inferenz nach der Vorgabe aller Fakten angestoßen. Dabei wird verglichen, ob die Fakten den Bedingungsteil einer Regel erfüllen. Darauf wird der Aktionsteil der Regeln ausgewertet, wodurch externe Aktionen, wie Eingabe-Ausgabeaktionen angestoßen oder neue Fakten für die Wissensbasis abgeleitet werden. Die bei der Regelauswertung erhaltenen Fakten können die Bedingungsteile weiterer Regeln erfüllen.
- Beim *Rückwärtsschließen* wird von einem Ziel ausgegangen und überprüft, ob dieses im Konklusionsteil einer Regel vorkommt. Sind die Prämissen einer so gefundenen Regel die Konklusion einer weiteren Regel, so kann weiter rückwärts geschlossen werden.

Bei der Rückwärtspropagation wird das Ziel in Form der Konklusion einer Regel vorgegeben. Es werden dann nur Regeln ausgewertet, die auf die Prämissen der vorgegebenen Konklusion einwirken. Daher werden nur Regeln aktiviert, die zur Ermittlung des Zieles notwendig sind.

Bei der Vorwärtspropagation werden die in den Regelprämissen vorkommenden Daten vorgegeben. Beispielsweise wird die Innere Spannung einer Membran als niedrig vorausgesetzt. Bei der Ermittlung des Zieles werden alle Regeln aktiviert, für welche die vorgegebenen Daten als Prämisse zutreffen. Dies führt dazu, daß unter Umständen sehr viele Regeln aktiviert werden, deren Auswertung für die gegebene Fragestellung irrelevant ist. Der Einsatz von Vorwärts- oder Rückwärtspropagation hängt daher von den Anforderungen an das System ab. Vorwärtspropagation eignet sich besonders, wenn mehrere oder alle Auswirkungen eines bestimmten Zustandes untersucht werden sollen. Die Rückwärtspropagation ist dann vorzuziehen, wenn man an einer einzigen Ableitung oder Begründung für einen vorgegebenen Zustand interessiert ist.

Eine wichtige Eigenschaft von Regeln ist, daß sie gleichzeitig die Grundelemente der Methode der Wissensrepräsentation und abgeschlossene Wissensseinheiten aus inhaltlicher Sicht darstellen. Bei kleineren Regelbasen führt dies zu einer hohen Übersichtlichkeit und Flexibilität der Wissensdarstellung. Bei großen Regelbasen, die viele voneinander abhängige Regeln enthalten, kann die Änderung einer einzigen Regel sich jedoch auf die gesamte Regelbasis auswirken. Eine Regel kann in diesem Fall nicht mehr als selbständige Wissensseinheit betrachtet werden.

Die Darstellung von Wissen in Regelform zielt insbesondere auf eine effiziente Verarbeitung ab. Die Möglichkeiten zur Organisation der darzustellenden Wissensinhalte durch Regeln sind dagegen gering. Insbesondere fehlen Prinzipien zur Abstraktion und Modularisierung bei der Wissensrepräsentation. Für eine strukturierte Darstellung der vielfältigen Beschreibungselemente, welche das Prozeßwissen ausmachen, sind Regeln allein also wenig geeignet.

Für den Teil des Prozeßwissens, welcher sich mit den Zusammenhängen zwischen Prozeßparametern befaßt, erscheint die regelorientierte Darstellung geeignet. Vergleicht man die Beispiele aus Abb. 2.4 bis Abb. 2.6 mit der Struktur einer Regel, so beschreibt die Regelprämisse die Bedingungen an die Werte der beeinflussenden Parameter und die Regelkonklusion den resultierenden Wert des Zielparameters. Der Unterschied bei der Darstellung von Parameterabhängigkeiten im Vergleich zum eingangs gegebenen Beispiel einer Regel ist, daß hier die Bedingungen aus Werten von Prozeßparametern bestehen und durch unterschiedliche Wertetypen beschrieben werden. Für die Behandlung der Zusammenhänge ist daher ein Regelkonzept zu entwickeln, welches die gemeinsame Verarbeitung von Parametern unterschiedlichen Wertetyps erlaubt. Ein solches Konzept gestattet es dann, die Vorteile der Erklärbarkeit und effizienten Wissensverarbeitung für die Modellierung von Parameterzusammenhängen zu nutzen.

Zusammenfassend läßt sich feststellen, daß eine auf Regeln basierende Darstellung aufgrund fehlender Organisationsprinzipien für eine allgemeine Beschreibung des Prozeßwissens wenig geeignet sind. Für die Beschreibung von Zusammenhängen zwischen Prozeßparametern stellt der Ansatz jedoch einen erfolversprechenden Ausgangspunkt dar.

3.1.4 Objektorientiertes Paradigma

Die objektorientierte Wissensdarstellung ist eine relativ neuartige Methode, die in vielen Grundlagen auf die Sprache SIMULA (SIMUlation LAnguage) [DAHL66] zurückgeht, in der die Verwendung des Klassenkonzeptes bei Programmiersprachen erstmals eingeführt wurde. Die Basis objektorientierter Methoden bildet die Erforschung der Organisationsprinzipien des menschlichen Verstandes im Umgang mit der Komplexität realer Problemfelder.

Eine wesentliche Erkenntnis ist dabei, daß durch die bisher vorherrschenden Methoden zur Wissensrepräsentation jeweils nur bestimmte Probleme oder Teile komplexer Problemfelder adäquat beschrieben werden. So eignet sich beispielsweise die prozedurale Sicht der meisten höheren Programmiersprachen besonders für Probleme, bei welchen der Schwerpunkt in der algorithmischen Lösung liegt. Dagegen ist sie für die Organisation menschlichen Wissens über ein breiteres Problemfeld wenig angemessen.

Beim Umgang mit komplexen Problemen bedient sich der menschliche Verstand einer Reihe fundamentaler Prinzipien [ATKI90, BOOC91, COAD91]. Im folgenden werden diese Prinzipien, welche dem objektorientierten Paradigma zugrundeliegen, näher betrachtet.¹

1. In der Literatur werden verschiedentlich noch weitere Prinzipien genannt. Diese betreffen jedoch eher bestimmte Aspekte der Realisierung eines Systems, wie Nebenläufigkeiten oder Persistenz von Objekten, als daß sie unmittelbare Prinzipien der Behandlung von Komplexität realer Probleme darstellen.

Dekomposition und Abstraktion

Zur Reduktion der Komplexität eines Problemfeldes ist zunächst eine *Dekomposition* (Zerlegung) in Teilprobleme notwendig. Die Art der Dekomposition kann dabei nach den Grundelementen, in welche die Zerlegung erfolgt, unterschieden werden. Bei der funktionalen Dekomposition besteht die Beschreibung des Gesamtproblems aus einer Menge von Funktionen und Subfunktionen. Dagegen wird beim objektorientierten Ansatz das Problem in eine Menge von *Objekten* zerlegt. Die Objekte stellen dabei eine *Abstraktion* der zu beschreibenden Gebilde dar, d.h. die Verkörperung der für eine konkrete Aufgabe wichtigen Beschreibungselemente des Problemfeldes.

Ein Objekt wird repräsentiert durch einen Datenteil zur Beschreibung seiner Eigenschaften (Attribute) und einen Funktionsteil (Methoden), durch dessen Funktionen die Daten des Datenteils ausschließlich manipuliert werden können. Die Dekomposition in Objekte weist gegenüber der funktional orientierten Dekomposition insbesondere den Vorzug einer größeren Nähe zur menschlichen Sicht auf ein Problemfeld auf. Vorteile der konsequenten Nutzung der Prinzipien Dekomposition und Abstraktion sind die erhöhte Wiederverwendbarkeit von Programmteilen, was zu kleineren Gesamtsystemen führt, eine verbesserte Wartbarkeit durch die konsequente Verwendung kleiner Teilsysteme (Modularisierung) sowie eine höhere Stabilität des Entwurfs gegenüber Änderungen in den Anforderungen, da die Grundelemente der Beschreibung eines Problemfeldes eine höhere Beständigkeit besitzen als die Aktionen oder Abhängigkeiten in einer rein funktionalen Beschreibung.

Zusammenfassend läßt sich feststellen, daß mit Hilfe der Prinzipien Dekomposition und Abstraktion ein Problemfeld durch Objekte repräsentiert wird. Diese stellen Abstraktionen der zur Beschreibung des Problemfeldes notwendigen Begriffe dar.

Strukturierte Organisation und Hierarchie

Das Ergebnis der Dekomposition und Abstraktion eines Systems kann zu einer sehr großen Zahl von Objekten führen. Um den Umgang damit zu ermöglichen, benutzt der Mensch unterschiedliche Organisations- und Strukturierungsprinzipien:

Durch das erste Prinzip werden Objekte in *Klassen* zusammengefaßt. Die in den Klassen enthaltenen Objekte (Instanzen) besitzen zwar gemeinsame Eigenschaften, die jedoch bei verschiedenen Objekten unterschiedliche konkrete Ausprägungen annehmen können.

Das zweite Prinzip der *Generalisierung und Spezialisierung* wird zur Organisation der Klassen eingesetzt. Dieses Prinzip entspricht der Inklusionsbeziehung bei Semantischen Netzen. Dabei werden ausgehend von vorhandenen, speziellere oder allgemeinere Klassen durch Hinzufügen oder Weglassen von Eigenschaften gebildet. Das Prinzip der *Vererbung* erlaubt dabei die Übernahme von Eigenschaften einer allgemeinen Basisklasse durch speziellere abgeleitete Klassen.

Das dritte grundlegende Organisationsprinzip für die Behandlung komplexer Probleme ist die *Beziehung zwischen einem Teil und dem Ganzen*. Im Unterschied zur Generalisierung und Spezialisierung zwischen Klassen besteht diese Beziehung zwischen konkreten Objekten und drückt aus, aus welchen Unterobjekten oder Teilen ein bestimmtes Objekt besteht.

Außer den genannten Prinzipien existiert noch ein weiteres, weniger grundlegendes Prinzip, die *Assoziation*. Damit ist die semantische Zuordnung zwischen Objekten gemeint. So wird beispielsweise die Beziehung zwischen einem Prozeßschritt und den zu seiner Durchführung notwendigen Parametern durch eine assoziative Beziehung repräsentiert. Das Prinzip der Assoziation dient weniger der fundamentalen Strukturierung eines Problemfeldes als der Repräsentation inhaltlicher Zusammengehörigkeiten von Objekten.

Kapselung

Das Prinzip der Kapselung wird zwar den Prinzipien des objektorientierten Paradigmas zugerechnet, ist aber im Vergleich zu den bisher vorgestellten Prinzipien stärker auf die Implementierung als auf den Entwurf ausgerichtet.

Die Kapselung sei an einem einfachen Beispiel verdeutlicht. Ein Prozeßschritt-Objekt enthält eine Menge von Attributen (Texte, Datumsangaben, Graphiken etc.), die zur Beschreibung eines Prozeßschrittes notwendig sind. Diese Attribute werden dem Prozeßschritt-Objekt als *private* Daten zugeordnet. Auf private Daten dürfen nur Methoden des Objektes selbst zugreifen. Für andere Objekte sind die Attribute eines Prozeßschrittes geheim und können nur über den Aufruf von Methoden des Prozeßschritt-Objektes geändert werden. Ist es beispielsweise notwendig, ein Graphik-Attribut in einem anderen Format abzuspeichern, so muß dies nur in der Implementierung des Prozeßschritt-Objektes berücksichtigt werden. Für andere Objekte bleibt diese Änderung unsichtbar. Die konsequente Einhaltung des Prinzips der Kapselung erhöht daher die Modularisierung und verbessert dadurch insbesondere die Wart- und Erweiterbarkeit des Systems, da sich die Auswirkungen von Änderungen auf einen möglichst kleinen Teil des Gesamtsystemes beschränken.

Kommunikation durch Botschaften

Um die Interaktion verschiedener Systemelemente zu beschreiben, muß ein Prinzip für die Kommunikation zwischen Objekten existieren. Das objektorientierte Paradigma sieht für die Kommunikation Botschaften (messages) vor. Methoden und Daten bilden gleichberechtigte Teile innerhalb eines Objektes. Daher werden in einem Objekt sowohl statische (Daten) als auch dynamische Aspekte (Daten und Funktionen) des modellierten Beschreibungselementes zusammengefaßt. Neben den Attributen, die ein Prozeßschritt besitzt kann ein Prozeßschritt-Objekt also auch Methoden besitzen, beispielsweise eine Methode sich selbst und alle mit sich zusammenhängenden Objekte zu löschen. Die öffentlichen Methoden eines Objektes können dann durch Botschaften von anderen Objekten aus aktiviert werden. Eine allgemeine Abarbeitungsstrategie wird durch die Zusammenarbeit unterschiedlicher Objekte erreicht. Auf diese Weise schließt der objektorientierte Ansatz das Prinzip der funktionalen Zerlegung zur Darstellung dynamischer Aspekte eines Problemfeldes mit ein.

3.2 Bewertung der Methoden im Hinblick auf die Prozeßmodellierung

Die aufgeführten Methoden der Wissensrepräsentation lassen sich in zwei Gruppen einteilen. Die erste Gruppe zielt eher auf die problemgerechte *Organisation* des Wissens ab. Zu dieser Gruppe gehören die Semantischen Netze und das relationale Datenmodell. Bei der Repräsentation durch Regeln liegt der Schwerpunkt mehr auf einer effektiven Inferenzstrategie, also der *Verarbeitung* des Wissens. Beim objektorientierten Ansatz werden schließlich umfangreiche Prinzipien zur Wissensorganisation mit Elementen einer funktionalen Beschreibung kombiniert.

Für die Beurteilung der Eignung verschiedener Repräsentationsmethoden sind Kriterien erforderlich. Diese ergeben sich aus den Anforderungen an die Prozeßmodellierung. In Tab.3.1 werden anhand fünf wesentlicher Kriterien die verschiedenen Repräsentationsmethoden beurteilt.

Unter „Organisation des Wissens“ werden die primären Konzepte zur Repräsentation des Wissens unterschieden. Während Konzepte wie die Regeldarstellung oder das Relationenmodell das Problem auf eine einzige, sehr einfache, mathematisch beschreibbare Grundstruktur abbilden, bietet das objektorientierte Paradigma einen ganzen Satz solcher Strukturen zur Organisation vorliegenden Wissens an.

In der zweiten Spalte der Tabelle werden die einzelnen Methoden bezüglich ihrer Eignung für die Modellierung von Zusammenhängen zwischen Prozeßparametern einander gegenübergestellt. Durch Regeln läßt sich der kausale Charakter von Parameterabhängigkeiten besonders gut ausdrücken. In der Prämisse einer Regel werden Bedingungen an die Einflußparameter gestellt, im Konklusionsteil wird der Zielwert aufgrund der die Prämisse erfüllenden Eingangswerte berechnet.

Um auch unscharfes Wissen in der Modellierung zu berücksichtigen, wurden die Repräsentationsmethoden auf die Möglichkeit der Einbeziehung dieser Wissensform hin untersucht. Für den regelbasierten Ansatz und auch den relationalen sind bereits Erweiterungen in diese Richtung bekannt. Mit Hilfe des objektorientierten Ansatzes lassen sich problemangepaßte Klassen zur Behandlung unscharfen Wissens entwerfen. Hierauf wird in Kapitel 5 näher eingegangen.

Die vierte Spalte behandelt die zur Umsetzung der Methoden verfügbaren Werkzeuge. Hier bieten das relationale Modell und die objektorientierten Programmiersprachen die besten Voraussetzungen. Mit Werkzeugen für diese Methoden liegen umfangreiche Erfahrungen vor, die teilweise zu weit verbreiteten Standards geführt haben. Fehlen solche Standards oder Schnittstellen zur Benutzerkommunikation, Datenhaltung etc. kann der Einsatz eines Werkzeuges einen relativ hohen Anpassungsaufwand erforderlich machen.

Methode	Organisation des Wissens	Repräsentation von Parameterabhängigkeiten	Einbeziehung unscharfen Wissens	Umsetzung Werkzeugunterstützung	Vorteile / Nachteile
Semantische Netze	Graphen mit Knoten und Kanten.	numerische Abhängigkeiten nicht darstellbar.	-	Basiskonzept für Implementierung realer Probleme zu allgemein.	gute Organisation und Strukturierung des Wissens / wenig formal, Wissensverarbeitung nicht im Konzept.
Relationale Modelle	Relationen	Abhängigkeiten diskreter Wertemengen durch Relationen. Kontinuierliche Abhängigkeiten nicht darstellbar.	Erweiterung des Ansatzes zur Ermöglichung unscharfer Anfragen.	Werkzeuge in Form ausgereifter und erprobter Produkte.	theoretisch fundiert, gute Werkzeuge / umständliche Beschreibung komplexer Objekte, Wissensverarbeitung schwach.
Regeln	Regeln, wenig strukturiert.	Möglichkeit für Repräsentation: Regelprämisse Eingangsparameter Regelkonklusion Zielparameter	Erweiterung des Ansatzes auf unscharfe Regeln möglich.	Shells zur Entwicklung regelbasierter Systeme auf dem Markt erhältlich.	Einfachheit des Konzeptes / fehlende Organisation, Unübersichtlichkeit bei großen Regelbasen
Objektorientierter Ansatz	Klassen Objekte Vererbung Kapselung Beziehungen	Beschreibung von Abhängigkeiten durch prozedurale Elemente möglich.	Aufbau von Bibliotheken zur Behandlung unscharfen Wissens möglich.	Objektorientierte Programmiersprachen Framesysteme.	Wissensorganisation, umfassendes Konzept / nur ansatzweise formale mathematische Grundlagen

Tab. 3.1 Gegenüberstellung verschiedener Methoden für die Wissensrepräsentation.

Zusammenfassend erfüllen die verschiedenen Methoden für die Prozeßmodellierung erforderliche Aspekte in unterschiedlichem Grade. Der objektorientierte Ansatz zeichnet sich vor allem durch seine hervorragenden Möglichkeiten zur *Organisation* komplexer Informationen aus. Da Werkzeuge zur Umsetzung dieser Methode in Form objektorientierter Programmiersprachen auf relativ niedrigem Abstraktionsniveau verfügbar sind, können andere Ansätze, wie der regelbasierte Ansatz oder Datenbankeigenschaften aus objektorientierten Grundmodulen aufgebaut werden. Für die Darstellung von *Parameterbeziehungen* scheinen Regeln aufgrund ihrer Einfachheit und Erklärungsfähigkeit besonders geeignet. Insbesondere, da im

Fall der Modellierung von Parameterabhängigkeiten die Abhängigkeitsstrukturen verschiedener Regeln vergleichsweise überschaubar sind. Parameterabhängigkeiten können durch relativ kleine Regelbasen wiedergegeben werden. Dadurch fallen in der Literatur beschriebene Nachteile großer regelbasierter Systeme weniger ins Gewicht [LI91].

Um das System auch in der Praxis einzusetzen, wurde aus den genannten Gründen der objektorientierte Ansatz als Grundlage für die gesamte Modellierung gewählt. Die Funktionalität für die Modellierung zusätzlich benötigter Methoden wurde in Form vorhandener Klassenbibliotheken bzw. selbst entwickelter Klassen aufgebaut. Letztere umfassen im wesentlichen Klassen zur Verarbeitung unscharfen Regelwissens (s. Kapitel 5) und zur regelbasierten Behandlung von Parameterbeziehungen (s. Kapitel 6). Durch die beschriebene Vorgehensweise können die Vorteile verschiedener Repräsentationsmethoden kombiniert werden. Dabei bleibt die Einheitlichkeit des Ansatzes, die z.B. den Übergang auf eine objektorientierte Datenbank ermöglicht, gewahrt.

4. Objektorientierte Analyse und Modellierung des Fertigungswissens

Bei einer objektorientierten Modellierung können die semantisch grundlegenden Begriffe des Anwendungsgebietes als Klassen in das Datenmodell übertragen werden. Im vorliegenden Abschnitt werden die Beschreibungselemente des LIGA-Verfahrens und ihre objektorientierte Modellierung vorgestellt. Auf diese baut die gesamte Beschreibung des Prozesses und die in Kapitel 6 eingehend behandelte Simulation der Beziehungen zwischen Prozeßparametern auf.

Ein Prozeß stellt allgemein eine Folge von Abläufen dar, die auf ein System verändernd einwirken. Ein Prozeß bewirkt daher eine Aufeinanderfolge von Zuständen eines Systems. Ein Teilabschnitt eines Prozesses, der einen gegebenen Eingangszustand des Systems in einen Zielzustand überführt, wird im folgenden als *Prozeßschritt* bezeichnet. Ein Prozeßschritt kann gemäß dieser Definition auch eine Reihe elementarer Prozeßschritte umfassen, dies wird durch das jeweils gewählte Abstraktionsniveau der Betrachtung bestimmt.

Einem Prozeßschritt wird eine Anzahl von *Parametern* zugeordnet, durch welche die Prozeßbedingungen sowie die Produkteigenschaften beschrieben werden können. Um zu erfassen wie die Parameter zusammenhängen, müssen mehrere Parameter zusammengefaßt und ihre gegenseitigen Abhängigkeiten beschrieben werden. Die hierzu benötigten Objekte wird im folgenden als *Beziehungen* bezeichnet.

Um auch Abhängigkeiten zwischen unscharfen Parametern behandeln zu können, werden Klassen zur Modellierung von *Fuzzy-Regeln*, *Fuzzy-Mengen* und *Fuzzy-Operatoren* benötigt, deren Modellierung im letzten Teil des Kapitels beschrieben wird.

4.1 Prozeßschritte

Zunächst werden alle Prozeßschritte in einer Klasse zusammengefaßt. Inhaltlich lassen sich dadurch einzelne, in sich abgeschlossene Handlungsabläufe, die zur Fertigung eines Produktes notwendig sind, beschreiben.

Prozeßschritte besitzen eine wichtige Funktion bei der strukturellen Gliederung des Prozeßablaufes. Die Gliederung erfolgt nach (vgl. Abb. 2.2):

- **Hierarchie**

Der Prozeß muß auf unterschiedlichen Ebenen der Konkretisierung beschrieben werden. Dies ist notwendig, um von einem umfassenden Überblick in eine detaillierte Beschreibung überzugehen.

- **Zeitabfolge**

Im allgemeinen muß eine bestimmte Reihenfolge für die Durchführung der Prozeßschritte eingehalten werden. Die Prozeßschritte sind also zeitlich geordnet.

- **Alternativen**

Bestimmte Fertigungsziele können oft durch eine Reihe unterschiedlicher Prozeßschritte oder durch unterschiedliche Sequenzen von Prozeßschritten erreicht werden. Prozeßschritte besitzen also im allgemeinen Alternativen, welche selbst wiederum durch die genannten Prinzipien strukturiert werden können.

Ein Prozessschritt wird durch ein Objekt der Klasse aller Prozessschritte repräsentiert. Abb. 4.1 veranschaulicht die Beschreibung der Daten und Methoden, sowie der Beziehungen zwischen Objekten der Klasse Prozessschritte durch die komprimierende Notation nach Coad und Yourdon [COAD91]¹. Im unteren Teil von Abb. 4.1 wird in dieser Notation die Klasse Prozessschritte, also die Menge aller Prozessschritt-Objekte, durch den inneren schwarzen Kasten dargestellt. Dieser enthält auch die Attribute des Datenteils der Klasse, in welchen für jeden Prozessschritt detaillierte Informationen z.B. über Ablauf, Zuständigkeiten und Aktualität erfaßt werden. Dabei können Attribute auch selbst wiederum Objekte separater Klassen sein und den Prozessschritten als Komponentenobjekte zugeordnet werden.²

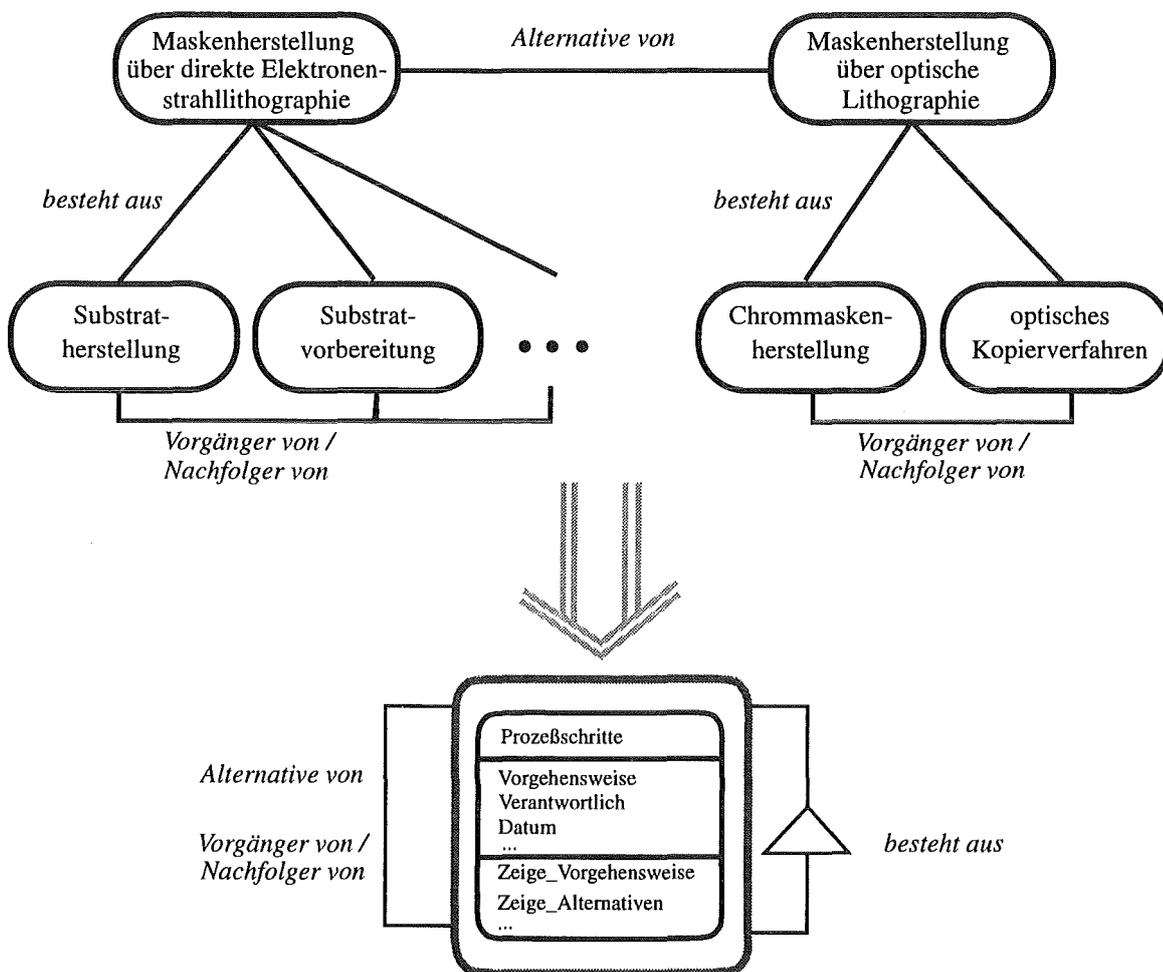


Abb. 4.1 Darstellung der objektorientierten Beschreibung des Prozeßablaufes durch Prozessschritte in der Notation von Coad und Yourdon.

1. Coad und Yourdon teilen ihre Notation in fünf Schichten ein, die Attribut- und Service- Schichten werden hier aus Gründen der Übersichtlichkeit nicht gezeigt, da lediglich die Struktur des Systems insgesamt dargestellt werden soll.

2. In den folgenden Abschnitten werden die Attribute der einzelnen Beschreibungselemente nicht explizit genannt, da dies die Darstellung in diesem Rahmen überlasten würde.

Der Vergleich mit dem oberen Teil der Abbildung veranschaulicht die abkürzende Beschreibung von Beziehungen zwischen Objekten durch die Notation von Coad und Yourdon. Der äußere graue Kasten beschreibt die Objekte der Klasse Prozeßschritte selbst. Die beschrifteten Kanten links und rechts des grauen Kastens, welcher die Prozeßschritt-Objekte repräsentiert bezeichnen unterschiedliche Beziehungen zwischen den Objekten. Durch die *durchgezogene* Kante links werden assoziative Beziehungen beschrieben. Diese kennzeichnen im vorliegenden Fall einen Prozeßschritt als Alternative, Vorgänger oder als Nachfolger eines anderen. Das *Dreiecksymbol* auf der rechten Kante identifiziert eine Beziehung als Teil-Ganzes Beziehung (*besteht aus*). Bei den Prozeßschritten wird hierdurch eine Verfeinerung beschrieben, die ausdrückt, daß einem Schritt eine Menge von Unterschritten zugeordnet wird, die diesen in ihrer Gesamtheit wiederum ausmachen.

4.2 Arbeitsmaterialien und Ausrüstungsgegenstände

Die Beschreibung von Arbeitsmaterialien und Ausrüstungsgegenständen erfolgt in separaten Klassen. Dies ist notwendig, da sowohl Arbeitsmaterialien als auch Ausrüstungsgegenstände durch spezielle, eigene Attribute und Methoden beschrieben werden.

Materialien kommen im Prozeß als Substrat-, Endproduktmaterialien und Hilfsstoffe, wie sie z.B. für Entwicklungs- und Galvanikprozesse benötigt werden, vor. Materialien mit gemeinsamen Eigenschaften, wie Metalle oder Resists stellen Spezialisierungen der allgemeinen Klasse Arbeitsmaterialien dar und werden in abgeleiteten Klassen zusammengefaßt. Eine beliebige Anzahl von Instanzen dieser Klassen werden dann einem Prozeßschritt zugeordnet, um die bei seiner Durchführung benötigten Materialien zu beschreiben.

Analog zu den Arbeitsmaterialien läßt sich jedem Prozeßschritt eine Menge von Ausrüstungsgegenständen zuordnen. Dabei kann die Beschreibung des Aufbaus komplexer Ausrüstungsgegenstände aus einzelnen Teilen analog zu den Prozeßschritten durch eine Verfeinerungsrelation ausgedrückt werden.

4.3 Prozeßparameter

Bisher wurde in erster Linie der Prozeßablauf in einem Modell erfaßt, um Basiswissen über den Prozeß auf dem Rechner darzustellen. Um jedoch auch tiefergehendes Wissen über quantitative und qualitative Zusammenhänge beschreiben zu können, müssen für jeden Prozeßschritt die Parameter erfaßt werden. Die Parameter charakterisieren die Voraussetzungen des Ablaufs eines Prozeßschrittes (Eingangsparameter) sowie die möglichen Zielzustände (Ausgangsparameter) durch Werte unterschiedlichen Typs.

Prozeßparameter stellen ein grundlegendes Element der Prozeßbeschreibung dar. Zu ihrer Spezifikation sind ebenfalls verschiedene Attribute notwendig, z.B. ihr Wert und Wertebereich oder ihre detailliertere Beschreibung. Außerdem werden auch Methoden, beispielsweise zur Ermittlung beeinflussender oder beeinflusster weiterer Parameter, benötigt. Daher werden auch die Prozeßparameter durch eine eigenständige Klasse repräsentiert.

Abb. 4.2 zeigt die Modellierung unter Berücksichtigung der bisher beschriebenen Klassen. Ein Prozeßschritt besteht danach nicht nur aus Unterprozessen, sondern auch aus Instanzen der Klassen, welche die übrigen Prozeßelemente beschreiben. Außerdem werden einem Prozeßschritt semantische Alternativen zugeordnet, was durch die durchgezogene Kante auf der linken Seite des Objektsymbols dargestellt ist. Auch Ausrüstungsgegenstände und Arbeitsmaterialien können wiederum aus gleichartig strukturierten Objekten bestehen (Dreieckssymbol auf den Kanten), während Prozeßparameter durch eine inhaltliche Beziehung, nämlich „beeinflußt“ oder „beeinflußt von“, mit anderen Parametern verknüpft sind (durchgezogene Kante).

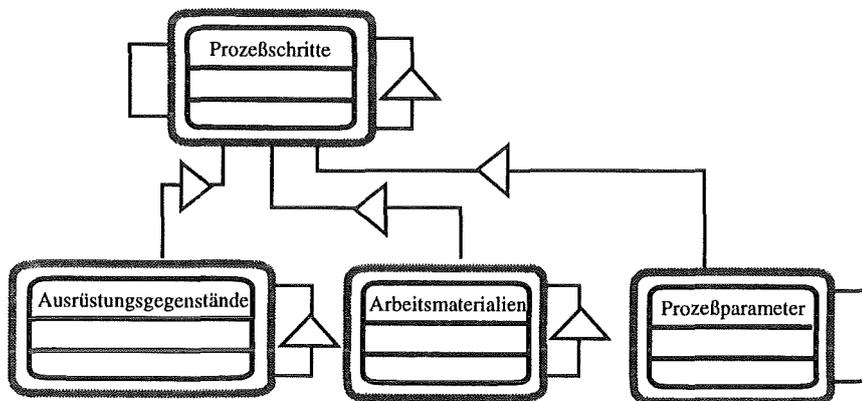


Abb. 4.2 Modellierung unter Berücksichtigung der Klassen Prozessschritte, Ausrüstungsgegenstände, Arbeitsmaterialien und Prozeßparameter.

4.4 Parameterbeziehungen und Abhängigkeiten

Für die vollständige Beschreibung der Prozeßführung ist außer der Erfassung der Prozeßparameter die Kenntnis ihrer gegenseitigen Abhängigkeiten notwendig. Diese werden ebenfalls als Objekte beschrieben und in eine Klassenhierarchie eingeordnet (vgl. Abschnitt 6.1). Stellt man die Parameter als Knoten, ihre gegenseitige Beeinflussung als Kanten eines Graphen dar, so entsteht ein Netz, welches die Abhängigkeitsstruktur der Prozeßparameter widerspiegelt.

Betrachtet man in einem solchen Netz die Abhängigkeit eines bestimmten Zielparameters von den ihn beeinflussenden Parametern, so läßt sich diese gesamte Beeinflussung in dem elementaren Konzept einer *Beziehung* zusammenfassen.

Die Beschreibung einer Beziehung besteht aus der Spezifikation der beteiligten Prozeßparameter, Bedingungen für die Eingangswerte sowie der Beschreibung der quantitativen oder qualitativen Zusammenhänge zwischen den Parametern unter Berücksichtigung der Eingangsbedingungen. In der objektorientierten Modellierung wird für die Repräsentation dieses Konzepts die Klasse „Beziehungen“ eingeführt.

Die direkte Beeinflussung eines konkreten Zielparameters wird also durch eine Instanz der Klasse Beziehungen dargestellt, die eine Grundeinheit für die Beschreibung längerer Parameterwirkketten bildet. Abb. 4.3 zeigt das Modell zur Repräsentation einer Beziehung.

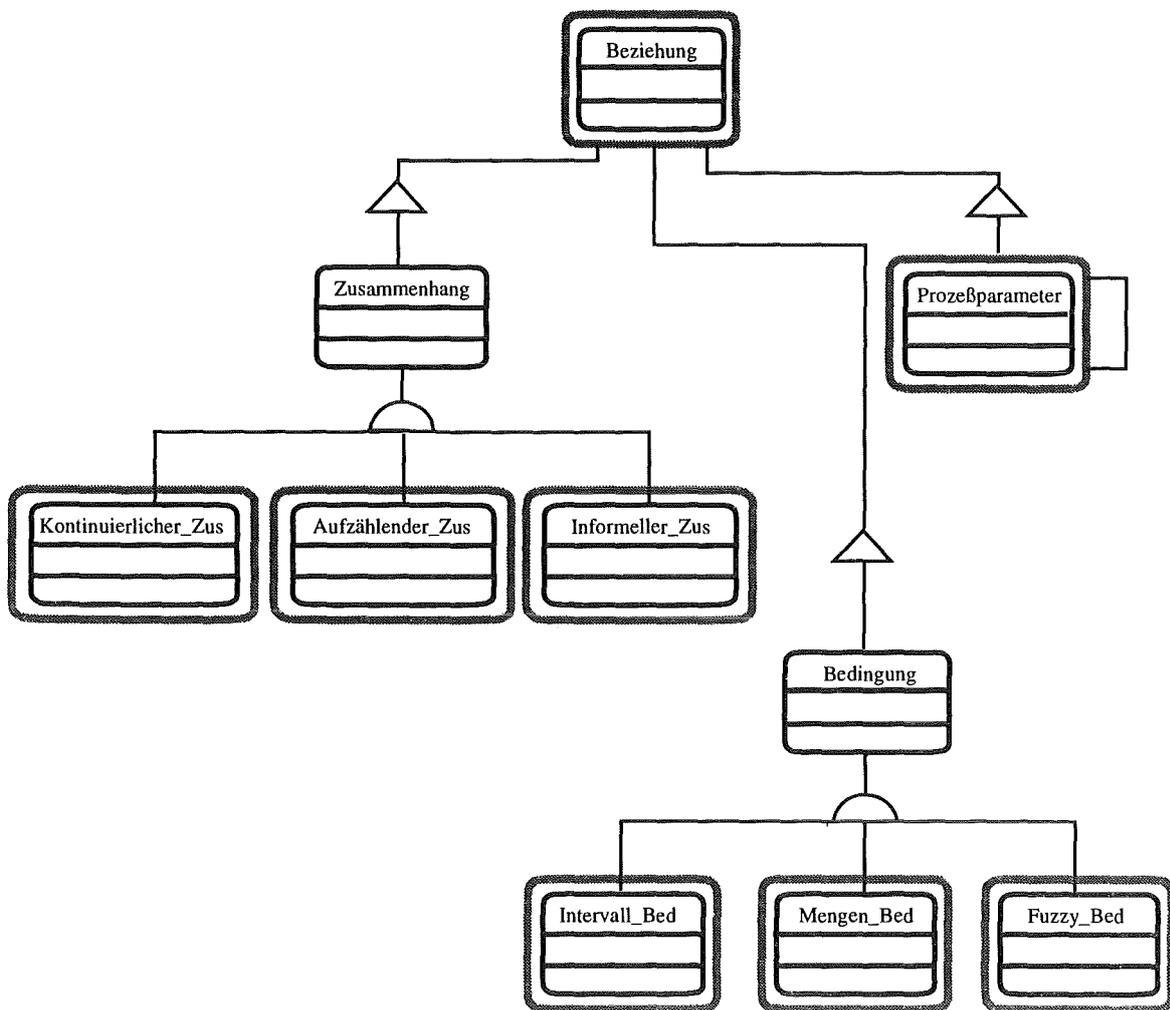


Abb. 4.3 Repräsentation einer allgemeinen Beziehung zwischen Prozeßparametern, welche durch unterschiedliche Wertetypen und Zusammenhänge charakterisiert ist.

Das Halbkreissymbol auf einer Kante zeigt eine Spezialisierungsbeziehung an. Klassen unterhalb des Halbkreises sind von darüberstehenden Basisklassen abgeleitet. Die Klassen „Zusammenhang“ und „Bedingung“ kommen in unterschiedlichen Varianten vor, welche durch die unterschiedlichen Typen der zu beschreibenden Parameterwerte und die unterschiedlichen Arten von Zusammenhängen begründet sind. Hierauf wird in Kapitel 5 näher eingegangen.

Um die Auswirkungen von Parametern über die Grenzen einer einzigen direkten Beziehung hinaus simulieren zu können, wird ein zusätzliches Konzept benötigt. Bei einer solchen Simulation muß der betrachtete Prozeßausschnitt in Form der mitberücksichtigten Beziehungen festgelegt werden. Daher besteht eine Simulation aus mindestens einem Beziehungsobjekt. Es kann erforderlich sein, Simulationen unter Einbeziehung unterschiedlicher Parameterabhängigkeiten parallel durchzuführen. Simulationen werden daher durch Klassen beschrieben. Eine Simulation unter Berücksichtigung bestimmter Parameterabhängigkeiten wird dann durch eine Instanz der Klasse Simulation repräsentiert. Der Ablauf einer Simulation und die Beschreibung der Parameterabhängigkeiten in einer Beziehung wird in Kapitel 6 ausführlich behandelt.

Die genannten Klassen stellen die Grundlage für die Modellierung des Prozeßwissens dar. Dabei wurde eine große Zahl weiterer Klassen, die zur Umsetzung des Konzepts notwendig sind, hier nicht beschrieben, da sie für die Darstellung des Grundkonzeptes eine weniger zentrale Bedeutung besitzen.

Hierzu gehören Klassen zur Repräsentation unscharfer Parameterwerte und Klassen für die Dokumenttypen zur Beschreibung des Prozesses. Der Aufbau der erstgenannten Klassen ist in Abb. 4.4 dargestellt. Das der gegebenen Klassifizierung zugrundeliegende theoretische Konzept wird im nächsten Kapitel und im Anhang ausführlicher behandelt.

Darüber hinaus wird eine Vielzahl von Klassen benötigt, die nicht unmittelbar die Semantik des Prozeßmodells betreffen. Hierzu gehören Klassen für die graphische Benutzeroberfläche, zur Wahrung der Konsistenz des eingegebenen Wissens, zur Bereitstellung grundlegender Datenstrukturen, wie Listen oder Graphen, sowie zur Ermöglichung der Persistenz der in der Wissensbasis abgespeicherten Informationen. Viele dieser Klassen konnten für die vorliegende Aufgabe direkt aus öffentlich verfügbaren Bibliotheken übernommen bzw. durch geringfügige Erweiterungen adaptiert werden.

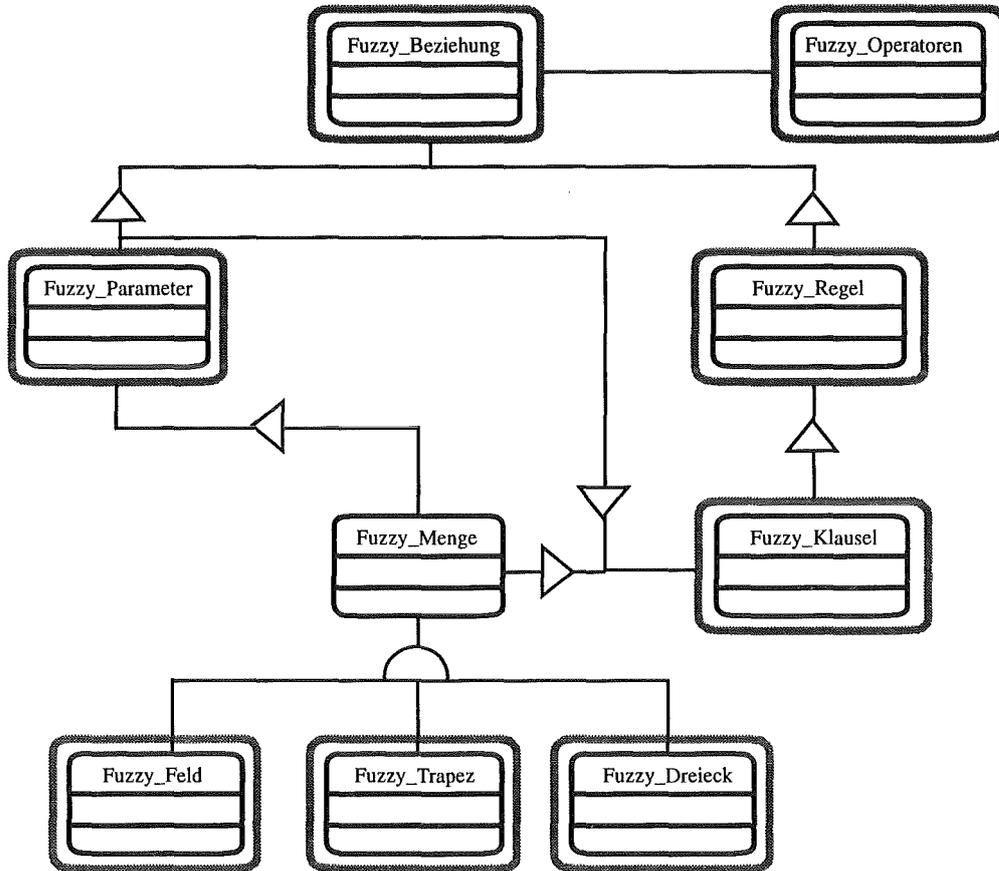


Abb. 4.4 Klassenkonzept zur Behandlung unscharfer Zusammenhänge zwischen Prozeßparametern.

5. Beschreibung unscharfen Prozeßwissens

In Kapitel 3 wurde festgestellt, daß ein wichtiger Teil des Wissens über den Fertigungsprozeß in Form scharfer Abhängigkeiten zwischen Prozeßparametern ausgedrückt wird. Wie die Beispiele in Kapitel 2 zeigen, können jedoch viele Beziehungen und Randbedingungen nicht allein durch scharfe Angaben beschrieben werden. Um auch diese Informationen im Rahmen eines einheitlichen Konzeptes zu erfassen, muß zunächst eine geeignete Methode zur Repräsentation und Verarbeitung unscharfen Wissens gefunden werden.

5.1 Methodik zur Repräsentation unscharfen Prozeßwissens

Die Methodik zur Repräsentation unscharfen Prozeßwissens soll nicht isoliert, sondern im Rahmen eines wissensbasierten Gesamtmodells eingesetzt werden. Daher sind zunächst die hiermit zusammenhängenden grundlegenden Anforderungen festzulegen.

- **Integrierbarkeit**

Die Methodik muß geeignet sein, unscharfes Wissen nicht nur isoliert darzustellen, sondern zusammen mit scharfen, d.h. durch scharfe Funktionen oder Relationen beschriebenen Zusammenhängen zu verarbeiten.

- **Erklärbarkeit**

Insbesondere muß für den Benutzer das Zustandekommen von Ergebnissen, die aus unsicherem Wissen und plausiblen Schließen resultieren, zurückverfolgbar und nachvollziehbar sein. Daher sollte die Verarbeitung des unscharfen Wissens auf der Grundlage eines mathematischen Modells erfolgen, das die Erklärbarkeit des erschlossenen Wissens ermöglicht.

- **Adäquatheit**

Die Methodik muß einerseits über die notwendigen Ausdrucksmittel verfügen, um die vorliegenden unscharfen Zusammenhänge zu beschreiben. Sie muß andererseits flexibel genug sein, um den Aufwand für die Erfassung, Darstellung und Verarbeitung dem Informationsgehalt des zu verarbeitenden Wissens anpassen zu können.

Bevor auf die Fuzzy Logik als Methode zur Repräsentation unscharfen Wissens näher eingegangen wird, soll der Begriff der Unschärfe detaillierter gefaßt und vom Begriff Unsicherheit unterschieden werden. *Unschärfe* stellt eine Form vagen Wissens dar, d.h. ein Faktum kann nicht durch eine eindeutige Auswahl aus einer Menge scharfer Alternativen beschrieben werden. *Unsicherheit* entsteht dadurch, daß ein Faktum nicht eindeutig einem Element einer Menge von scharf abgegrenzten Alternativen zugeordnet werden kann, sondern verschiedene Möglichkeiten mit unterschiedlicher Sicherheit angenommen werden.

Zur Verarbeitung *unsicheren* Wissens existieren eine Reihe von Methoden und Werkzeugen. So werden Unsicherheiten im Bereich der Künstlichen Intelligenz oft mit Hilfe von Gewißheitsfaktoren (certainty factors) modelliert [BUCH85]. Eine mathematisch fundiertere (und aufwendigere Methode) stellt die Anwendung wahrscheinlichkeitstheoretischer (probabilistischer) Ansätze, wie zum Beispiel dem Bayesschen Kalkül, dar. Dies setzt voraus, daß das interessierende unsichere Wissen in Form statistischen Datenmaterials vorliegt, wie es in der medizinischen Diagnostik oft der Fall ist.

Die Verarbeitung von *unscharfem* Wissen im oben beschriebenen Sinn ist bisher im Rahmen wissensbasierter Systeme, abgesehen von Fuzzy Control Systemen im Bereich regelungstechnischer Anwendungen, weit weniger verbreitet. In neuerer Zeit wurden hierfür jedoch durch die Weiterentwicklung der Theorie der Fuzzy Logik und die Erfahrungen mit ihrer Anwendung wichtige Grundlagen geschaffen. In den folgenden Abschnitten wird gezeigt, wie mit Hilfe der Fuzzy-Mengen Theorie und der Fuzzy Logik die oben gestellten Anforderungen an eine Methodik zur Verarbeitung unscharfen Wissens erfüllt werden können.

Bei der Wissensverarbeitung mit Fuzzy-Logik wird das Wissen zunächst in Form unscharf formulierter Aussagen bzw. Regeln niedergelegt. Diese beschreiben Zusammenhänge zwischen Prozeßparametern, welche nicht in Form numerisch-funktionaler oder linguistisch scharfer Abhängigkeiten spezifiziert werden können. Das in dieser Form niedergelegte *generelle* Wissen beinhaltet also das vergleichsweise statische allgemeine Hintergrundwissen über den Prozeß. In der Praxis besteht nun die Aufgabe für ausgewählte (unscharfe) Parameter Zielwerte zu ermitteln, welche aufgrund des generellen Wissens und bestimmter vorgegebener Parametereinstellungen möglich sind (*Inferenz*). Im folgenden soll die Inferenz, wie sie im allgemeinen im Bereich von Fuzzy Control Systemen üblich ist, anhand eines einfachen Prozeßbeispiels beschrieben werden.

Inferenz bei Fuzzy Control Systemen

Der Zusammenhang zwischen den Prozeßparametern Zeit, Temperatur und der Färbung einer Oxidationsschicht sei bekannt. Die linguistischen Variablen dieser Größen seien:

Temperatur	Wertebereich = [40, 9] (entspricht der Temperatur in °C)
Zeit	Wertebereich = [1, 4] (entspricht der Zeit in Minuten)
Färbung	Wertebereich = [0, 10] (entspricht dem Reflexionsgrad)

Die linguistischen Bezeichnungen der unscharfen Wertebereiche seien:

Temperatur	$L^1 = \{\text{niedrig, normal}\}$
Zeit	$L^2 = \{<2, \sim 2, >2\}$
Färbung	$L^3 = \{\text{hell, dunkel, schwarz}\}$

Durch die linguistischen Bezeichner allein sind die unscharfen Parameterwerte nicht klar definiert. Deshalb werden den Parameterwerten die *Zugehörigkeitsfunktionen* (membership functions) aus Abb. 5.1 zugeordnet. Diese normierten Verteilungen für die Temperatur, Zeit und Färbung beschreiben, zu welchem Grad die scharfen Werte des Grundbereiches (x-Achse) einem vorgegebenen unscharfen Wertebereich (z.B. hell) zugeordnet werden.

Die Regelbasis wird dann konkret durch Regeln der Form:

R1: wenn Zeit = ~2 und Temperatur = normal dann Färbung = hell

R2: wenn Zeit = >2 und Temperatur = normal dann Färbung = dunkel

Abb. 5.1 skizziert den Ablauf der Inferenz. Die Regeln R1 und R2 der Regelbasis werden getrennt abgearbeitet.

Zunächst wird eine *Gesamtkompatibilität* des (in Abb. 5.1 scharfen) Eingangswertes mit der Regel ermittelt. Dazu wird der Grad der Übereinstimmung der vorgegebenen Eingangswerte der Parameter Temperatur und Zeit mit der jeweiligen Regelprämisse verglichen und durch Minimumsbildung zum Gesamtkompatibilitätsgrad vereinigt (waagrechte Linie). Danach wird die resultierende Verteilung der Regel beim Wert der ermittelten Kompatibilität abgeschnitten.

Dies wird für alle Regeln der Regelbasis durchgeführt. Anschließend werden die Ausgangsverteilungen der einzelnen Regeln durch Bildung des Maximums aller Verteilungen zur endgültigen Ausgangsverteilung zusammengefügt, die in der Abbildung unten rechts gezeigt wird.¹ Diese Ausgangsverteilung wird in regelungstechnischen Anwendungen in eine scharfe Stellgröße umgewandelt (defuzzifiziert). In einem wissensbasierten System kann die Verteilung in einen linguistischen Wert verwandelt werden, um dem Benutzer ein übersichtliches und leicht interpretierbares Ergebnis zur Verfügung zu stellen.

Die Ergebnisse des eben beschriebenen Inferenzablaufes widersprechen auf verschiedene Weise der intuitiven Auffassung eines plausiblen Schließens:

- Je mehr Regeln zur Ermittlung des Ergebnisses beitragen, desto breiter und damit ungenauer wird die resultierende Verteilung.
- Liegt die Eingangsverteilung in der Schnittmenge zweier Regelprämissen, so liegt im allgemeinen die Regelkonklusion nicht in der Schnittmenge der resultierenden Verteilungen. In Abb. 5.1 liegt der Eingangswert für die Zeit zwischen ~2 min und >2 min. Die resultierende Verteilung liegt jedoch nicht zwischen hell und dunkel, sondern ist so breit wie beide Verteilungen zusammengenommen. Sie stellt daher eine sehr viel ungenauere Schlußfolgerung dar, als es aufgrund der verfügbaren Informationen plausibel wäre. Dies ist insbesondere beim Hintereinanderschalten (chaining) von Regeln ein gravierender Nachteil.
- Die resultierende Verteilung ist in der Regel nicht normalisiert. Die Verteilung sollte diese Eigenschaft jedoch besitzen, da von vorneherein in den Regeln vorausgesetzt wird, daß zumindest einer der möglichen Werte mit Sicherheit angenommen wird.

1. Es wird hier lediglich der Spezialfall der sogenannten min-max Inferenz beschrieben. Anstelle des Minimumoperators zur Ermittlung der Gesamtkompatibilität und der Maximumbildung zur Bestimmung der resultierenden Verteilung können auch andere Operatoren verwendet werden.

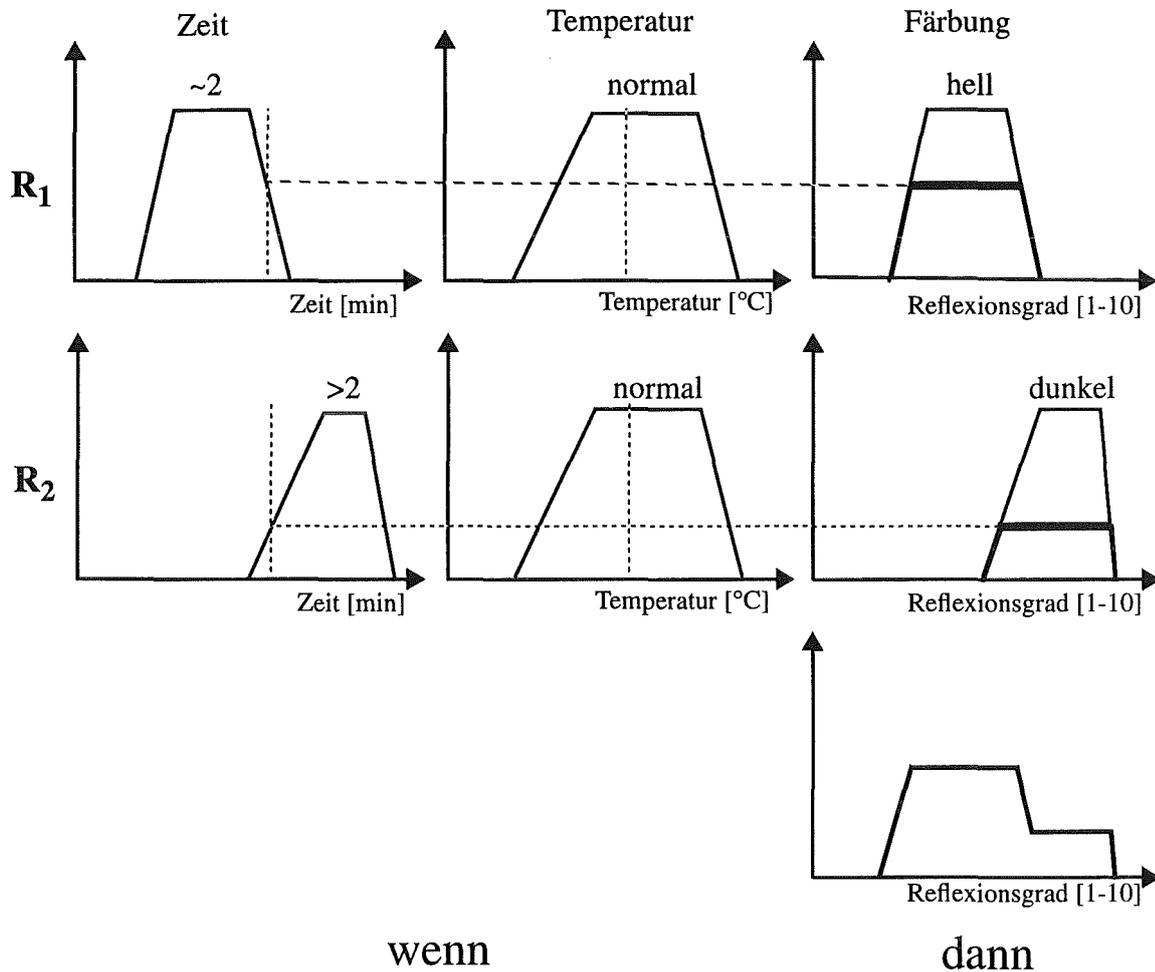


Abb. 5.1 Veranschaulichung der max-min Inferenz mit zwei Fuzzy Regeln R1 und R2, welche die unscharfen Zusammenhänge zwischen den linguistischen Variablen Zeit, Temperatur und Färbung ausdrücken.

5.1.1 Inferenz auf der Basis des verallgemeinerten Modus Ponens

Aus den zuvor dargelegten Gründen eignet sich die üblicherweise in Fuzzy Control Systemen angewandte Form der Inferenz weniger für den Einsatz in wissensbasierten Systemen. Für diesen Zweck erscheint ein anderer Ansatz geeigneter, der auf einer Verallgemeinerung des aus der zweiwertigen Logik bekannten Schlußprinzips des Modus Ponens basiert. Dieses Prinzip erlaubt es aus einer Regel, z.B. R1 aus dem letzten Abschnitt und einer Voraussetzung (z.B. Zeit = ~2 min, Temperatur = normal) die Konklusion der Regel (Färbung = hell) abzuleiten.

In der Fuzzy Logik wird die Regel durch einen Fuzzy-Operator modelliert, die Werte der Eingangs- und Ausgangsparameter werden durch entsprechende Zugehörigkeitsfunktionen wie in Abb. 5.1 ausgedrückt.

Die Inferenz auf der Grundlage des verallgemeinerten Modus Ponens unterscheidet sich in wichtigen Punkten von der im vorhergehenden Abschnitt beschriebenen Inferenz (s. Anhang A), die üblicherweise bei Anwendungen im Fuzzy Control Bereich eingesetzt wird. Für die Anwendung wichtige Eigenschaften dieser Methode sind:

- Liegt der Wert des Eingangsparameters zwischen den Prämissen zweier Regeln für diesen Parameter, so liegt auch die mit Hilfe der beiden Regeln gewonnene resultierende Verteilung zwischen den Ergebniswerten der einzelnen Regeln. Für das Beispiel aus Abb. 5.1 bedeutet dies, daß für Zeiten, welche zwischen ~ 2 min und < 2 min liegen, die resultierende Verteilung zwischen hell und dunkel liegen muß. Dies entspricht der intuitiven Auffassung weitaus besser als dem Resultat bei der Verwendung der Inferenz aus dem Fuzzy Control Bereich.
- Stimmt die Eingangsverteilung mit der Regelprämisse überein, so ist die resultierende Verteilung höchstens so spezifisch wie die Verteilung der Regelkonklusion. Im Beispiel bedeutet dies, daß aus der Regel „wenn Zeit = ~ 2 min dann Färbung = hell“ nicht gefolgert werden kann, daß die Haftung durch eine spezifischere als die in der Regelkonklusion gegebene Verteilung für den linguistischen Wert hell beschrieben werden kann, auch wenn die Eingangswerte spezieller sind als die Regelprämisse ~ 2 min.
- Stimmen Werte der Eingangsverteilung nicht mit der Regelprämisse überein, so führt dies zu einer Aufweitung der Schlußfolgerung. Entspricht die Eingangsverteilung also einer breiteren Verteilung als ~ 2 min, so wird die resultierende Verteilung ebenfalls breiter als die Verteilung für den Wert dunkel sein.
- Regeln können hintereinandergeschaltet werden. Insbesondere kann dann aus der Gültigkeit von $A \rightarrow B$ und $B^* \rightarrow C$ die Regel $A \rightarrow C$ gefolgert werden, wenn die Verteilung von B^* die Verteilung von B umfaßt.

Die aufgeführten Punkte zeigen, daß die Inferenz auf der Basis des Modus Ponens zu intuitiv besser nachvollziehbaren Ergebnissen führt und sich auch für das Hintereinanderschalten von Regeln besser eignet, als die Inferenz aus dem Fuzzy Control Bereich.

5.1.2 Eigenschaften der verwendeten Zugehörigkeitsfunktionen

In den seltensten Fällen ist eine detaillierte Beschreibung der Parameterwerte durch die Benennung einer analytischen Zugehörigkeitsfunktionen oder Umgekehrt die Interpretation von Details einer solchen Verteilung möglich.

Die verwendeten Zugehörigkeitsfunktionen werden durch λ -trapezförmige Funktionen angenähert. Die λ -trapezförmigen Zugehörigkeitsfunktionen werden dabei durch fünf Werte $(a_1, a_2, a_3, a_4, \lambda)$ repräsentiert (s. Abb. 5.2).

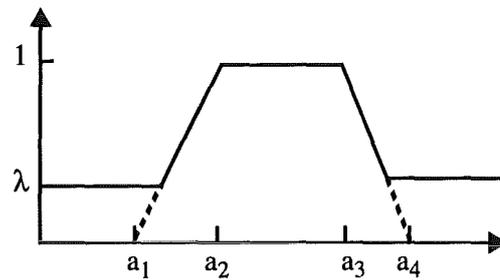


Abb. 5.2 λ -trapezförmige Zugehörigkeitsfunktionen

Für die vorgesehene Anwendung ist die Beschränkung auf einfache Zugehörigkeitsfunktionen gegenüber allgemeinen mathematischen Funktionen besonders aus zwei Gründen sinnvoll. Zum einen können die einfachen λ -trapezförmigen Funktionen durch eine geringe Datenmenge beschrieben und daher effizient gespeichert und verarbeitet werden. Zum anderen kommt es der Natur unscharfen Wissens näher, dieses nicht durch die Angabe komplexer mathematischer Funktionen, sondern durch die Erfassung der wesentlichen Informationen darzustellen. Das heißt das Wissen wird durch Bereiche, in denen eine Tatsache sicher ist, Bereiche in welchen sie unmöglich ist und dazwischenliegende Übergangszone modelliert.

6. Simulation von Prozeßparameterabhängigkeiten

Die Kenntnis der Abhängigkeiten von Prozeßparametern untereinander sowie zwischen Prozeß- und Produktparametern ist für die Durchführung und Weiterentwicklung des LIGA-Verfahrens von grundlegender Bedeutung. In diesem Kapitel wird untersucht, wie sich die Auswirkungen von Änderungen beliebiger Prozeßparameter auf direkt und indirekt abhängige Prozeß- und Produktparameter simulieren lassen. Dabei wird von der Beschreibung der unterschiedlichen Abhängigkeiten zwischen Prozeßparametern ausgegangen, welche durch elementare Beziehungen beschrieben werden, auf denen die spätere Simulation aufbaut.

6.1 Parameterbeziehungen und Wertebereiche

Die Komplexität bei der Beschreibung von Beziehungen zwischen Prozeßparametern beruht im wesentlichen auf folgenden Ursachen:

- Das LIGA-Verfahren weist eine sehr große und damit schwer überschaubare Zahl von Prozeßparametern und Beziehungen zwischen diesen auf.
- Sowohl die Werte der Prozeßparameter, als auch die Abhängigkeiten zwischen ihnen werden in unterschiedlicher Form beschrieben.
- Die zur Beschreibung eines Prozesses notwendigen Parameter und Beziehungen stehen nicht fest, sondern können sich gemäß des jeweiligen Wissensstandes ständig ändern.

Beschreibung des Konzeptes einer Parameterbeziehung

Um die Komplexität der Beschreibung von Parameterbeziehungen zu reduzieren, lassen sich gemäß der objektorientierten Vorgehensweise zunächst die Prinzipien der Abstraktion und Dekomposition anwenden. Daher werden die kleinsten Einheiten betrachtet, die einen allgemeinen Prototyp für die Beschreibung beliebiger Parameterbeziehungen repräsentieren. Diese Einheiten stellen die in Abschnitt 4.4 eingeführten Beziehungen dar.

Wirken n Einflußparameter auf einen Zielparameter, so kann diese Abhängigkeit durch eine n -parametrische Funktion beschrieben werden. Der Zusammenhang zwischen bestimmten Einflußgrößen und der Zielgröße kann dann durch Schnitte der Funktionsdarstellung veranschaulicht werden. Dazu werden bestimmte Parameterwerte fixiert, während die übrigen frei variieren können. Es zeigt sich jedoch, daß die gewohnte Auffassung einer Funktion als mathematische Vorschrift zur Berechnung eines numerischen Wertes aus einem anderen zu eng gefaßt ist, um der Beschreibung der Abhängigkeiten gerecht zu werden, wie sie in der Prozeßpraxis verlangt wird. Um die Parameterbeziehungen rechnergestützt beschreiben zu können, müssen die unterschiedlichen Formen, in denen der Zusammenhang zwischen Parametern vorliegen kann, klassifiziert und in das Prozeßmodell einbezogen werden.

Sind die Zielparameterwerte in Abhängigkeit der Werte eines Einflußparameters explizit bekannt, können also durch die Auswertung einer Funktion beschrieben werden, so wird der Einfluß auf den Zielparameter als direkt bezeichnet. Die tatsächliche Beeinflussung des Zielparameters kann danach auch bei einem direkten Zusammenhang in der Realität über einen Zwischenparameter erfolgen, welcher bei der gewählten Sicht des Problems nicht berücksichtigt werden muß.

Wertebereiche von Prozeßparametern

Wie in Kapitel 2 bereits gezeigt wurde, werden Prozeßparameter durch Wertebereiche unterschiedlichen Typs beschrieben. Abb. 6.1 stellt die Klassifizierung der Typen von Parameterwerten dar. Zunächst wird zwischen scharfen und unscharfen Wertetypen unterschieden. Scharfe Werte lassen sich durch die Angabe eines eindeutigen Wertes des vorgegebenen Wertebereiches beschreiben. Unscharfe Werte werden dagegen nach Kapitel 5 durch die Angabe einer Teilmenge möglicher Werte ihres Wertebereiches charakterisiert. Sowohl scharfe als auch unscharfe Werte können numerischer oder linguistischer Natur sein.

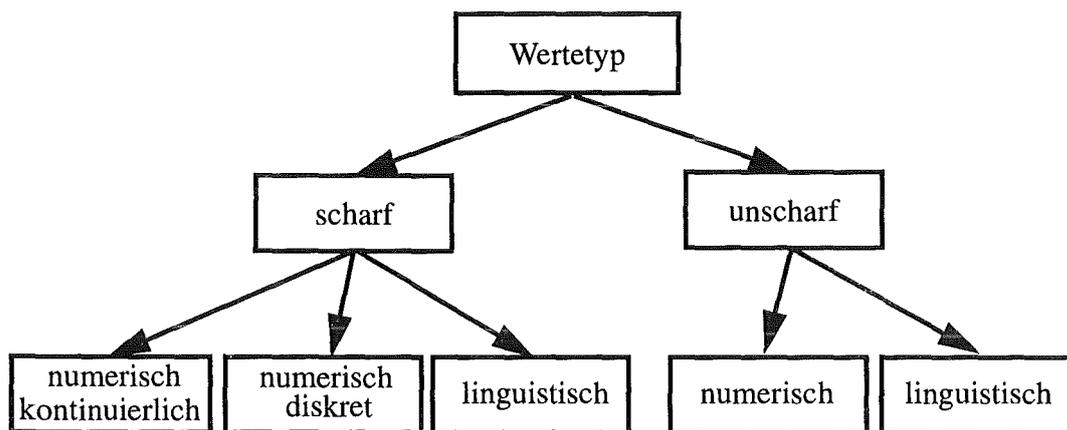


Abb. 6.1 Klassifizierung der Typen von Prozeßparametern.

Bei scharfen Wertetypen wird weiter zwischen kontinuierlichen und diskreten Wertebereichen unterschieden, um ausdrücken zu können, daß in vielen Fällen nur bestimmte diskrete Parameterwerte angenommen werden können. Diese Unterscheidung ist notwendig, weil für die Beschreibung der Zusammenhänge zwischen Parametern des diskreten Typs im Gegensatz zum kontinuierlichen Typ bereits eine Aufzählung von Wertepaaren anstelle einer Berechnungsvorschrift genügen kann. Diskrete numerische und linguistische Parameter unterscheiden sich durch das ihnen zugrundeliegende Skalenniveau [STEV46]. Numerischen Werten liegt eine Ordinalskala (Ratioskala) zugrunde, diese können miteinander verglichen und nach bestimmten Merkmalen geordnet werden. Linguistischen Werten liegt eine Nominalskala zugrunde, sie unterscheiden sich daher nur durch ihre Benennung.

Unscharfe Wertetypen können analog in linguistisch und numerisch unscharfe Wertetypen eingeteilt werden. Linguistisch unscharfe Wertetypen werden durch linguistische Variablen repräsentiert, die den linguistischen Bezeichner durch eine Abbildung auf ein numerisches Intervall beschreiben. Numerisch unscharfe Größen lassen sich mit Hilfe unscharfer Zahlen repräsentieren, für welche die Rechenoperationen mit natürlichen Zahlen verallgemeinert werden können [KAUF85]. Zur Repräsentation der in der Prozeßbeschreibung vorkommenden Zusammenhänge werden diese Rechenoperationen jedoch nicht benötigt, da sie zu intuitiv nicht nachvollziehbaren Ergebnissen führen und außerdem einen hohen Aufwand an Rechenzeit erfordern.

Die Wertetypen werden folgendermaßen beschrieben:

- $I_i, i \in \mathbb{N}$: Menge numerischer Intervalle
- D : diskrete numerische Menge
- L : Menge diskreter, linguistischer Ausdrücke
- F : Menge unscharfer linguistischer Variablen

6.2 Parameterabhängigkeiten und Integration

Ebenso wie die Parameterwerte, lassen sich auch die unterschiedlichen Arten von Zusammenhängen klassifizieren. Dazu wird in Abb. 6.2 analog zu Abb. 6.1 zwischen scharfen und unscharfen Zusammenhängen unterschieden. Zusammenhänge zwischen scharf charakterisierten Wertetypen werden entweder aufzählend oder in Form einer Funktionsvorschrift beschrieben.

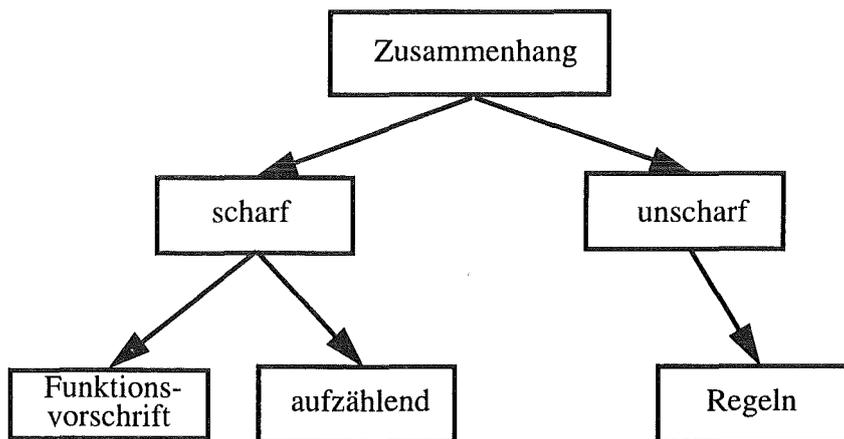


Abb. 6.2 Beschreibung von Zusammenhängen.

Unschärfe Zusammenhänge werden in Form unscharfer linguistischer Regeln dargestellt. Linguistische Regeln unterscheiden sich von scharfen linguistischen Zusammenhängen dadurch, daß die in der Regel vorkommenden linguistischen Ausdrücke durch eine Zugehörigkeitsfunktion genauer spezifiziert werden.

Setzt man die vier oben beschriebenen Wertetypen voraus, so bestehen für die Beziehungen zwischen einem Eingangs- und einem Ausgangsparameter 16 mögliche Kombinationen. Die Klassifizierung dieser Kombinationen wird im folgenden anhand von Tab. 6.1 diskutiert¹.

Beziehung zwischen scharfen numerischen Größen:

Das erste Diagonalelement stellt den analytischen Zusammenhang zwischen kontinuierlich variierenden Parametern dar. Das zweite beschreibt Zusammenhänge zwischen diskreten Prozeßgrößen.

Ist die Wertemenge des Eingangsparameters diskret und die des Ausgangsparameters kontinuierlich (Element D-I), so wird jedem Eingangswert ein Intervall von Ausgangswerten zugeordnet. Es liegt dann keine Abbildung im Sinne der oben gegebenen Definition vor, da einem Eingangsparameter mehr als ein Zielwert zugeordnet wird. Diese Art der Beziehung spiegelt eine Verletzung der Kausalität wider, da es unmöglich ist, den Zielzustand *exakt* aus der Kenntnis des Anfangszustandes abzuleiten. Bei experimentell bestimmten Zusammenhängen zwischen numerischen Parametern werden durch diese Art von Beziehungen die Meßfehler beschrieben.

Ist umgekehrt nur der Ausgangsparameter diskret (Element I-D), so stellt die Beziehung eine unstetige Funktion dar. Der Prozeß kann nur die durch den diskreten Wertebereich des Ausgangsparameters definierten Zustände annehmen.

Linguistische Ausdrücke:

Um die Möglichkeiten zur Beschreibung von Parameterzusammenhängen zu erweitern, werden scharfe linguistische Ausdrücke als Parameterwerte mitberücksichtigt. Für die Beziehungen zu kontinuierlichen Parametern gelten hier die für die scharfen numerischen Werte getroffenen Feststellungen. Die Beziehungen sind in der dritten Zeile und Spalte der Tabelle dargestellt.

Für Parameter mit unscharfem Wertebereich werden die Beziehungen als Regeln für die unscharfen Größen formuliert (s. Kapitel 5). Für die Behandlung von Beziehungen zwischen scharfen und unscharfen Größen müssen zunächst die scharfen Werte in unscharfe umgewandelt werden (Fuzzyfizierung). Anschließend kann die Auswertung der Beziehungen erfolgen und danach, falls notwendig, eine Rückwandlung der Ergebnisse in scharfe Werte durchgeführt werden (Defuzzyfizierung). Bei den Tabellenelementen der zweiten und dritten Spalte (erste Zeile) ist die Zugehörigkeitsfunktion der Zielparameter ein Singleton, also das Ergebnis ein scharfer Wert.

1. Es wird vorausgesetzt, daß die Wertemenge des Zielparameters genau aus den bei der Variation der Eingangswerte erreichten Werten besteht, die Abbildung zwischen den Wertemengen der Parameter also surjektiv ist.

	$x \rightarrow$ $z \downarrow$	I _i	D	L	F
Wertebereich Zielparameter 	I _i	funktionaler Zusammenhang $f(x)=z$	Unsicherheit, Spezialfall von D-F	Unsicherheit, Eingang diskret, Ausgang kontinuierlich	linguistische Regel, wobei $\mu(z)=const.$
	D	Sprungfunktion numerischer Wertebereich	$f(x_i)=z_i$ fkt. Zusammenhang zwischen diskreten Werten	Relation (x_i, z_i) scharfer linguistischer Wert numerischer Wert	Spezialfall: $\mu(z)=$ Singleton
	L	Sprungfunktion symbolischer Wertebereich	Relation (x_i, z_i) numerischer Wert scharfer linguistischer Wert	Relation (x_i, z_i) zwischen scharfen linguistischen Werten	Spezialfall: $\mu(z)=$ Singleton
	F	Fuzzyfizierung $\mu_i(x)=const.$	Spezialfall: $\mu(x)=$ Singleton	Spezialfall: $\mu(x)=$ Singleton	„unscharfe“ linguistische Regel

Wertebereich Eingangsparameter 

Tab. 6.1 Bedeutung der Zusammenhänge zwischen unterschiedlichen Wertetypen für die Prozeßbeschreibung.

Bei den Elementen der zweiten und dritten Zeile (letzte Spalte) wird die Eingangsgröße als scharfer Wert durch einen Singleton¹ repräsentiert. Sind die Wertebereiche von Ein- und Ausgangsparameter unscharf (erste Zeile, letzte Spalte), so wird die Beziehung durch eine Fuzzy-Regel mit frei wählbaren Zugehörigkeitsfunktionen beschrieben. Damit können auch

1. Ein Fuzzy-Singleton ist eine Fuzzy-Menge mit einer Zugehörigkeitsfunktion, die nur für einen scharfen Wert verschieden von null ist.

Parameter und Zusammenhänge mit in die Modellierung aufgenommen werden, die nicht exakt bekannt sind, deren Kenntnis aber trotzdem für die Durchführung des Prozesses von Bedeutung ist.

Integration unterschiedlicher Beschreibungsformen von Zusammenhängen

Die verschiedenen Einflußgrößen einer Beziehung besitzen im allgemeinen unterschiedliche Wertetypen. Um die Zusammenhänge in einer Beziehung als einheitliches Konzept zu beschreiben, werden sie alle wie Regeln behandelt. So besteht jede Beziehung aus Bedingungen, welche an die Werte der Einflußparameter gestellt werden und einer Berechnung, die dann aktiviert wird, wenn der Bedingungsteil der Beziehung erfüllt ist. Bedingungen können durch Tabellen dargestellt werden. Tab. 6.2 zeigt Bedingungen am Beispiel der Prozeßparameter Geometrie, mittleres Molekulargewicht und Spincoatfrequenz. Die Geometriebedingung wird als scharfer linguistischer Wert, welcher zwischen runden und rechteckigen Substraten unterscheidet, spezifiziert. Die Bedingung an das mittlere Molekulargewicht kann ein scharfer numerischer Wert oder eine Aufzählung von Werten sein. Die Bedingung an die Spincoatfrequenz wird in Form eines scharfen numerischen Intervalls gestellt. Sind bestimmte aktuelle Werte für die Eingangsparameter vorgegeben, wird zeilenweise überprüft, ob die Bedingungen erfüllt werden. Treffen die Bedingungen für alle Einflußparameter zu, so wird anhand der zur Beziehung gehörigen funktionalen Zusammenhänge, hier der Abhängigkeit zwischen Spincoatfrequenz und Schichtdicke aus Abb. 2.4, der aktuelle Wert der Schichtdicke ermittelt. Dieser wird in Abhängigkeit der Werte der Eingangsparameter durch unterschiedliche Funktionen, die von den kontinuierlichen Parametern (im Beispiel der Schichtdicke v) abhängen, berechnet.

Bedingungen			Konklusion
Eingangsparameter			Ausgangsparameter
Geometrie	Molekulargewicht kg	Spincoatfrequenz v U/min	Schichtdicke μm
„MR“	950	[800,4500]	$d = f(v)$
„MR“	600	[800,4500]	$d = g(v)$
„MR“	496	[600,4000]	$d = h(v)$
„ME“

Tab. 6.2 Darstellung scharfer Bedingungen an Eingangsparameter in tabellarischer Form

Unschärfe Beziehungen zwischen Prozeßparametern können ebenfalls in tabellarischer Form beschrieben werden. Tab. 6.3 stellt die Beschreibung des Zusammenhanges aus Abb. 2.6 dar, in der die beteiligten Parameterwerte unscharf sind. Die Bedingungen an die unscharfen Parameter Oxidationstemperatur und Oxidationszeit lassen sich gemäß Tab. 6.3 analog zu den Bedingungen an scharfe Parameter darstellen.

Bedingungen		Konklusion
Eingangsparameter	Eingangsparameter	Ausgangsparameter
Oxidations- temperatur	Oxidations- zeit	Oxidationsschicht- Färbung
„normal“	„~2“	„dunkel“
„niedrig“	„~2“	„hell“
„hoch“	„<2“	„schwarz“
„niedrig“

Tab. 6.3 Darstellung unscharfer Bedingungen.

Die Bedingungen in der Tabelle gleichen den Bedingungen an den scharfen linguistischen Parameter Blank-Geometrie aus dem vorausgegangenem Beispiel. Der Unterschied ist jedoch, daß die Parameterwerte diesmal durch unscharfe Wertetypen beschrieben werden. Zu diesen gehört neben der linguistischen Bezeichnung eine unscharfe Menge, welche die Zugehörigkeit bestimmter scharfer Parameterwerte, also beispielsweise einer bestimmten Oxidationstemperatur, zur Menge der als „normal“ aufzufassenden Oxidationstemperaturen festlegt. Zur Bestimmung des unscharfen Zielwertes Oxidationsschicht-Färbung wird die in Kapitel 5 beschriebene Inferenzmethode benutzt. Diese ist im Prinzip äquivalent zur einfachen Suche einer zutreffenden Bedingungszeile im Fall scharfer Prozeßparameter. Die in den Zeilen der Tabelle vorgegebenen Bedingungen können diesmal jedoch zu einem beliebigen Grad erfüllt sein, welcher durch einen Wert aus dem Intervall $[0,1]$ beschrieben wird. Daher kann der Ergebniswert von mehreren Zeilen der Tabelle bzw. Regeln abhängen.

6.3 Simulation von Wirkungsketten zwischen Parametern

Wie die Diskussion von Parameterabhängigkeiten im letzten Abschnitt zeigte, stehen die Beziehungen, welche den einzelnen Prozeßschritten zugeordnet werden, im Prozeß nicht isoliert. Im allgemeinen besitzen sie Auswirkungen auf weitere Parameter, die ihrerseits wiederum andere Parameter beeinflussen. Betrachtet man daher den zeitlichen Prozeßab-

lauf, so existiert ein Parametergraph, der die Wirkungsketten von Parametern beschreibt. Als einfaches Beispiel ist in Abb. 6.3 die Beeinflussung der Dicke der Resistschicht bei der Maschenherstellung durch die Parameter eines Belackungs- und anschließenden Temperschlittes gezeigt¹.

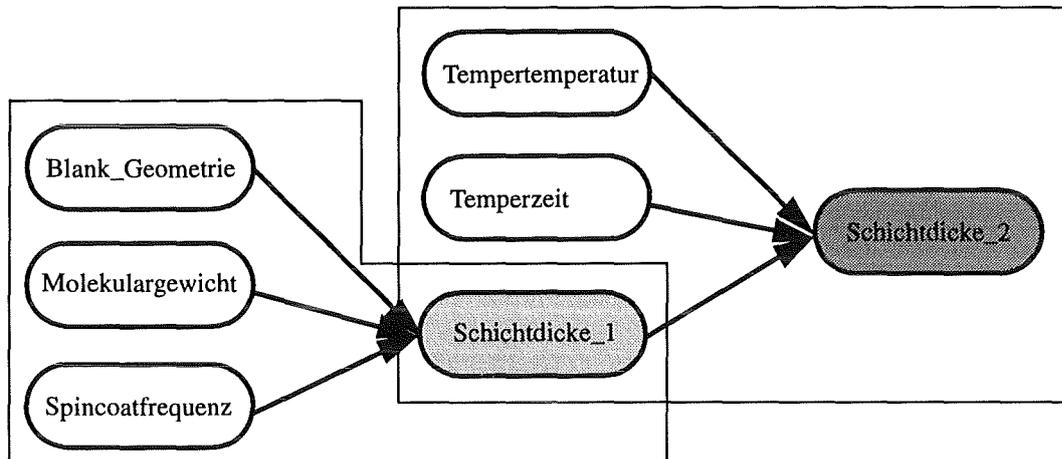


Abb. 6.3 Beispiel einer mehrstufigen Parameterabhängigkeit.

Für die Prozeßdurchführung ist es wichtig, die Wirkungen einzelner Prozeßparameter zu kennen. Aus dem Parametergraphen können alle Parameter bestimmt werden, auf die sich ein bestimmter Parameter auswirken kann. Um zusätzlich auch quantitative Informationen darüber zu erhalten, wie sich die Änderung bestimmter Parameterwerte auf andere Parameter auswirkt, können auch sequentiell mehrere Beziehungen zwischen den Parametern ausgewertet werden.

Beziehungen stellen damit die elementaren Bausteine für die Simulation quantitativer und qualitativer Abhängigkeiten zwischen Parametern dar. Bei einer Simulation wird von den Standardwerten, die beim gewöhnlichen Prozeßbetrieb verwendet werden, ausgegangen. Diese können durch den Benutzer verändert werden. Nach der Wertänderung wird der Einfluß des neuen Parameterwertes auf einen Zielparameter unter Berücksichtigung der bekannten Parameterzusammenhänge ermittelt.

Probleme bei der Simulation von Parameterwirkketten bestehen in der Weiterverarbeitung der Ergebnisse einer Beziehung, der Steuerung des Ablaufes der Auswertung verschiedener an einer Simulation beteiligten Beziehungen und der Wahrung der Konsistenz bei der Änderung von Parameterwerten. Diese Probleme werden in den folgenden Abschnitten ausführlicher behandelt.

1. Aus Übersichtlichkeitsgründen sind in der Abbildung nur einige der bekannten Einflußparameter auf die Schichtdicke dargestellt.

Erfüllung der Bedingungen durch Parameter unterschiedlichen Typs

Bei der Beschreibung der Zusammenhänge zwischen den Prozeßparametern in einer Beziehung wird überprüft, ob die an die Eingangparameter gestellten Bedingungen mit dem Typ der Parameter verträglich sind. So kann z.B. für einen Parameter mit linguistischem Wertetyp nicht überprüft werden, ob er innerhalb eines bestimmten numerischen Bereiches liegt. Tab. 6.4 zeigt, welche Bedingungen an Parameterwerte unterschiedlichen Typs gestellt werden können.

Wertetyp	Bedingungstypen
scharf linguistisch	Mengenbedingung
scharf numerisch diskret	Intervallbedingung, Mengenbedingung, unscharfe Bedingung
scharf numerisch kontinuierlich	Intervallbedingung, Mengenbedingung, unscharfe Bedingung
unscharf linguistisch	unscharfe Bedingung

Tab. 6.4 Mögliche Bedingungen für verschiedene Wertetypen.

An Parameter mit scharfem linguistischem Wertetyp können nur Mengenbedingungen, also durch Aufzählung beschreibbare Bedingungen, gestellt werden. Für numerische Werte kann außerdem durch eine Intervallbedingung überprüft werden, ob sie in beliebigen offenen oder abgeschlossenen Intervallen liegen. Sowohl an unscharfe als auch an scharfe numerische Wertetypen kann eine unscharfe Bedingung gestellt werden, da ein scharfer Wert ein Spezialfall eines unscharfen Wertes ist. Dadurch können bei der Simulation auch unscharfe Bedingungen berücksichtigt werden. Die Verarbeitung unscharfer Abhängigkeiten ist somit also in das Simulationsmodell scharfer Zusammenhänge integriert.

Strategie der Abarbeitung von Beziehungen

Für die Strategie der Abarbeitung von Beziehungen sind aus dem Bereich regelbasierter Systeme bekannte Methoden des Vorwärtsschließens, Rückwärtsschließens oder hybride Lösungen denkbar. Zur Auswahl einer geeigneten Strategie muß die Problemstellung bei der Simulation von Parameterabhängigkeiten betrachtet werden. Ziel einer Simulation ist es, die Auswirkungen von Änderungen der Werte einer Menge von Einflußparametern auf bestimmte Zielparameter zu verfolgen. Gewöhnlich ist bereits vor der Simulation bekannt, welche Zielparameter simuliert werden sollen. Daher kann der Zielparameterwert entsprechend einer Rückwärtspropagation als unbekannt vorausgesetzt werden.

Anhand Abb. 6.4 wird die Abarbeitungsstrategie erläutert. Die Rahmen fassen die Parameter einer Beziehung zusammen. Die Zahlen in den Ecken dienen zur Unterscheidung der Beziehungen. Durch spezielle Einstellungen oder Standardwerte sind die Werte der Einflußparameter E_1 bis E_5 vorgegeben. Zunächst soll der Wert des Zielparameters Z_3 ermittelt werden. Daher wird versucht den Wert von Z_2 zu berechnen. Da der Wert des Parameters Z_2 der Beziehung 2 nicht bekannt ist, muß zunächst diese ausgewertet werden. Hier sind die Werte aller Einflußparameter bekannt, es kann also nun der Wert von Z_2 berechnet werden und mit diesem auch der Zielwert von Beziehung 1.

Im Gegensatz zu einer Vorwärtssimulation wird also der Parameter Z_1 nicht berechnet. In größeren Netzen würden bei der Vorwärtspropagation Beziehungen, die zur Lösung der gestellten Aufgabe nicht beitragen, ebenfalls ausgewertet werden. Es lassen sich also bei der Rückwärtssimulation erheblich kürzere Simulationszeiten erreichen.

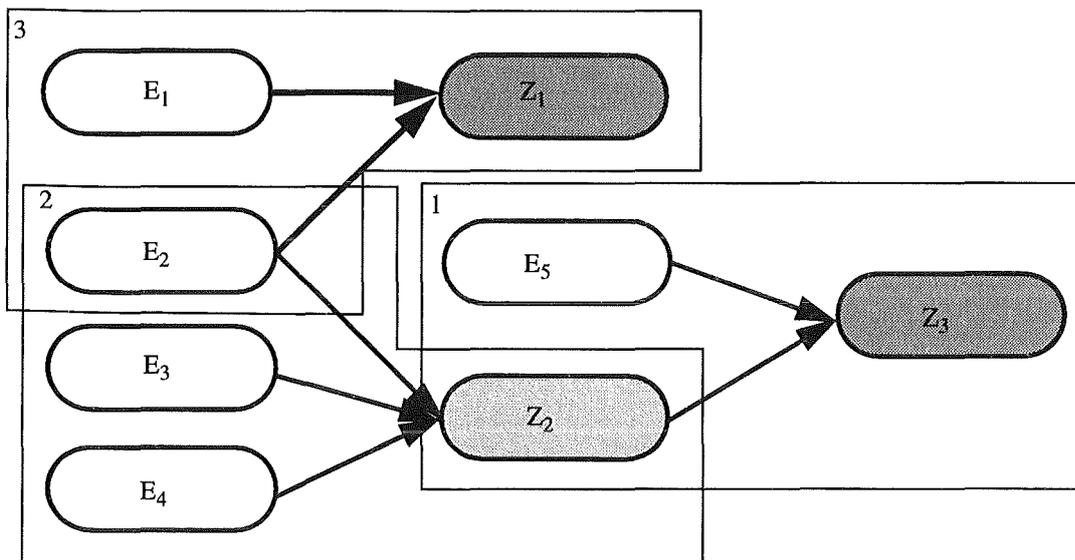


Abb. 6.4 Beispiel einer mehrstufigen Parameterabhängigkeit.

Die Methode der Rückwärtspropagation erlaubt die Berechnung des Wertes *eines* vorgegebenen Zielparameters. Oft ist es jedoch notwendig, die Auswirkung der eingestellten Parameterwerte auf mehrere Zielparameter gleichzeitig zu kennen. Dies wäre durch die Verwendung der Vorwärtspropagation möglich. Wie bereits festgestellt wurde, ist die Vorwärtspropagation jedoch ungeeignet, da hier immer sämtliche möglichen Zielwerte berechnet werden.

Um nur die notwendigen Berechnungen durchzuführen und dennoch die Auswirkungen von Einstellungsänderungen auf unterschiedliche Zielparameter ermitteln zu können, ist es möglich, sequentiell verschiedene Zielparameter für die Simulation anzugeben. Die Ergebniswerte dieser Parameter werden dann basierend auf den eingestellten Werten der Einflußparameter durch Rückwärtspropagation ermittelt. Der Parameter Z_1 kann also nachträglich durch Rückwärtspropagation berechnet werden. Dadurch lassen sich die Nebenwirkungen von Wertänderungen auf verschiedene Zielparameter untersuchen.

Konsistenzbetrachtungen

Bisher wurde vorausgesetzt, daß die Abhängigkeiten der Prozeßparameter so beschrieben sind, daß die Einflußparameter einer Beziehung als voneinander unabhängig betrachtet werden können. In der Realität kommen jedoch auch Abhängigkeiten vor, bei denen Parameter in verschiedenen Beziehungen als Einflußgrößen auftauchen. Als Beispiel sei das mittlere Molekulargewicht eines Resists in der Maskentechnik betrachtet. Dieses wirkt sich sowohl direkt auf die im Resist abgelagerte Flächendosis aus, als auch über die Schichtdicke, die von der Molekulargewichtsverteilung des Resists beeinflusst wird.

Abb. 6.5 zeigt den geschilderten Sachverhalt schematisch. Aus der Abbildung ist zu erkennen, daß P_G und P_1 nicht unabhängig voneinander variiert werden können, da P_G P_1 beeinflusst. Im gezeigten Beispiel muß immer P_G zuerst variiert werden, damit läßt sich P_1 bestimmen. Aus den mit der Prozeßbeschreibung konsistenten Werten von P_G und P_1 kann dann P_Z ermittelt werden.

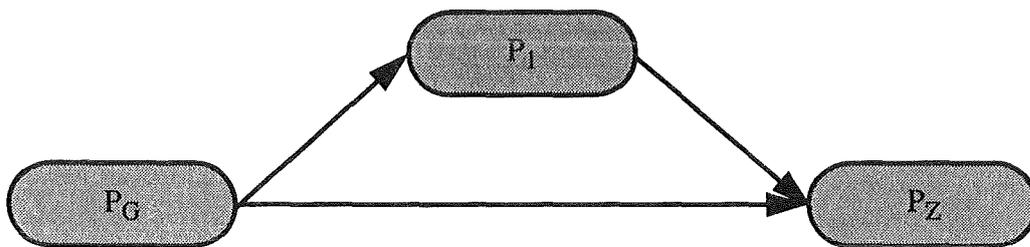


Abb. 6.5 Gleichzeitige direkte und indirekte Beeinflussung eines Zielparameters.

Allgemeiner läßt sich das Problem folgendermaßen formulieren:

Beeinflußt ein Einflußparameter P_1 einen Zielparameter über verschiedene Wege des Parametergraphen, so können die Parameter, die auf diesen Wegen liegen, nicht voneinander unabhängig variiert werden.

Wie Inkonsistenzen, die sich hieraus ergeben, vermieden werden können, wird anhand von Abb. 6.6 erläutert. Durch einen gemeinsamen Parameter P_G beeinflusste Eingangsparameter dürfen nur über den Zusammenhang mit P_G variiert werden. Soll daher ein Parameterwert P_1 in einer Beziehung verändert werden, so muß zunächst geprüft werden, ob ein Parameter P_G existiert, der P_1 und gleichzeitig andere Einflußparameter der Beziehung über verschiedene Wege des (gerichteten) Parametergraphen beeinflusst. Diese Wege werden hier als zu P_1 assoziierte Wege bezeichnet. Bevor die Auswirkungen eines Parameterwertes simuliert werden können, müssen also aus Konsistenzgründen die diesem Parameter assoziierten Wege in der Simulation berücksichtigt werden. Der Parameter kann dann nur über den Wert des Parameters P_G beeinflusst werden, von welchem verschiedene assoziierte Wege ausgehen.

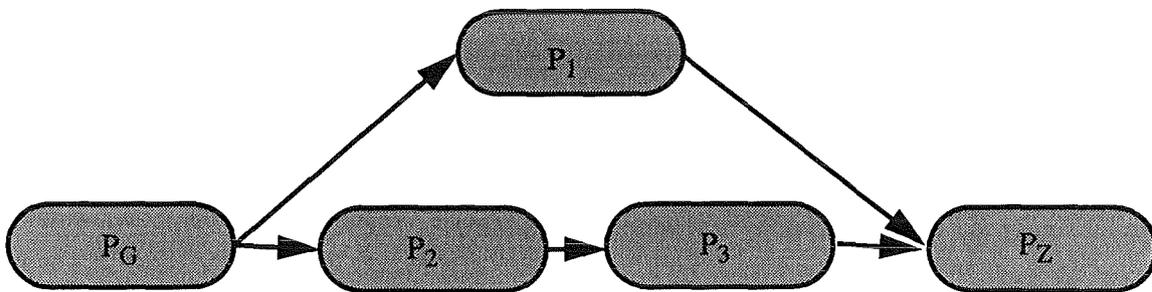


Abb. 6.6 Gleichzeitige direkte und indirekte Beeinflussung eines Zielparameters.

Sollen die Auswirkungen von Parametereinstellungen ohne die Berücksichtigung der gegenseitigen Beeinflussung zwischen den Beziehungen untersucht werden, so können Beziehungen auch losgelöst vom Prozeßkontext dargestellt und simuliert werden (s. Abschnitt 7.3).

7. Realisierung

In diesem Kapitel wird der Prototyp des Systems beschrieben, welches das in den vorausgehenden Kapiteln beschriebene Konzept für die rechnergestützte Verarbeitung von Prozeßwissen des LIGA-Verfahrens realisiert. Das hier beschriebene System behandelt in der vorliegenden Version die Möglichkeiten zur Darstellung und Verarbeitung von Prozeßwissen. Im folgenden Abschnitt werden die Hard- und Softwarevoraussetzungen und die verwendeten Softwarehilfsmittel genannt.

7.1 Hard- und Softwarehilfsmittel zur Realisierung von LIMES

Als Hardware-Implementierungsplattform für das System wurden Workstations der Firma Sun Microsystems eingesetzt. Die Rechner werden unter dem Betriebssystem UNIX und dem X Window System betrieben.

Als Sprache für die Implementierung wurde C++ gewählt, da diese Sprache zum einen die Anforderungen an eine objektorientierte Programmiersprache erfüllt, und andererseits eine hohe Kompatibilität zu C aufweist. Dadurch erschließen sich die Vorteile von C, die vor allem in seiner großen Verbreitung, Effizienz und der guten Verträglichkeit mit den verwendeten Standards UNIX und X Window System liegen.

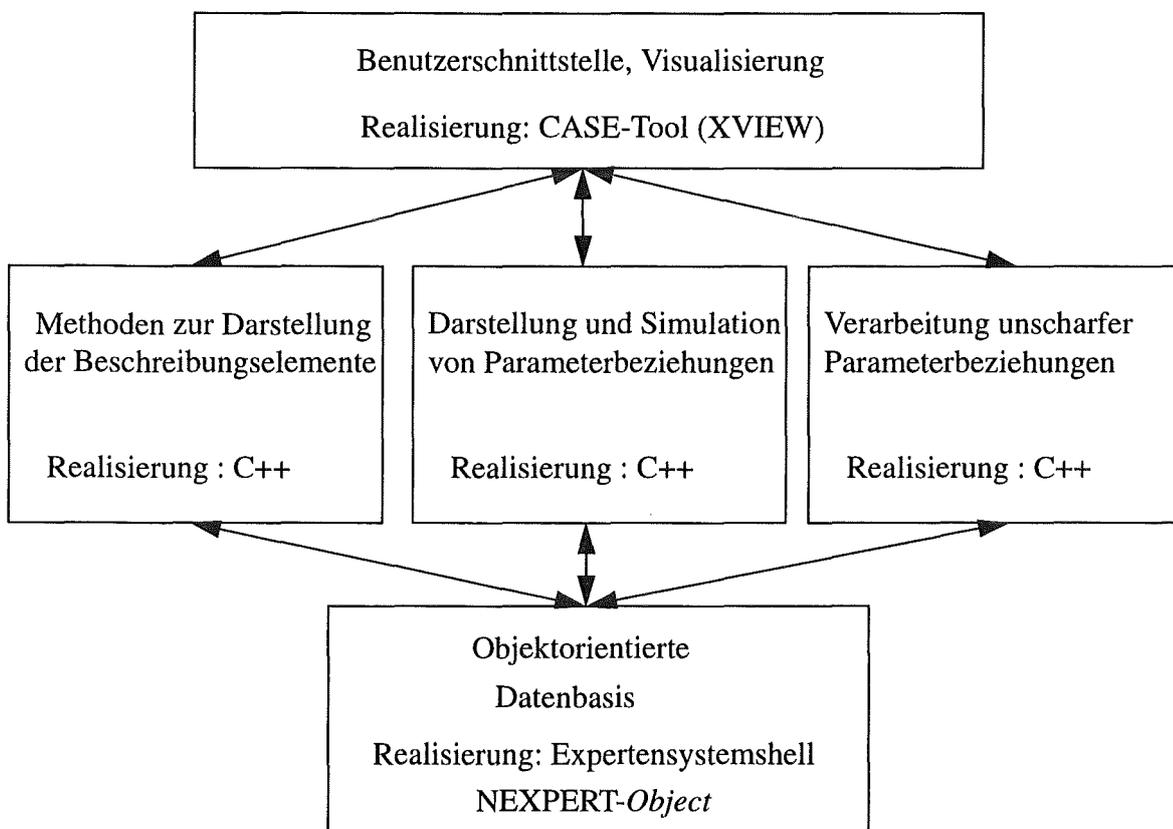


Abb. 7.1 Veranschaulichung der Gesamtstruktur von LIMES.

Abb. 7.1 zeigt einen Überblick über die Gesamtstruktur der für LIMES implementierten Software. Soweit möglich wurden für die Implementierung bereits vorhandene Softwarewerkzeuge benutzt. Diese Vorgehensweise reduziert nicht nur den Implementierungsaufwand, sondern verbessert auch die Qualität des Systems, das auf diese Weise aus häufig benutzten und daher gut ausgetesteten Modulen besteht. Bei der Auswahl der Werkzeuge wurde außerdem Wert auf die freie Verfügbarkeit und das Vorhandensein des Quellcodes gelegt, um einerseits die Kosten für das System und andererseits die Abhängigkeit von bestimmten Softwareanbietern zu minimieren.

Eine Ausnahme bildet der Einsatz der Expertensystemshell *NEXPERT-Object* [NEXP91]. Dieses Werkzeug der Firma Nexus ist nicht frei verfügbar. Ein benutzerfreundlicher Zugang ermöglicht die Definition von Klassen- und Objektstrukturen sowie der Attribute und Methoden der Klassen und Objekte. Darüberhinaus besitzt *NEXPERT-Object* im Unterschied zu einer Programmiersprache wie C++ die Fähigkeit, Klassen- und Objektstrukturen abzuspeichern, ermöglicht also die Persistenz des in der Wissensbasis vorhandenen statischen Wissens.

Für die Erstellung der graphischen Benutzerschnittstelle des Systems wurde das CASE-Tool *guide* von Sun Microsoft eingesetzt. Dieses Werkzeug erlaubt die direkte graphische Spezifikation der Systemoberfläche und der Aktionen, die aufgrund der Interaktion des Benutzers mit der Oberfläche ausgeführt werden sollen. Aus dieser Spezifikation wird durch einen Übersetzungsvorgang direkt C++ Quellcode erzeugt, in den sich der übrige Programmtext integrieren läßt. Ein besonderer Vorzug der eingesetzten Version des Werkzeuges gegenüber früheren ist, daß die Oberfläche erweitert werden kann, ohne daß das bisherige System von den Änderungen betroffen wird.

Zur Realisierung grundlegender Datenstrukturen wie z.B. Listen oder Strings wurden die C++ Klassen-Bibliothek LEDA Version 3.0 eingesetzt, die am Max Planck Institut Saarbrücken entwickelt wurde.

Auf die C++ Realisierung der drei Blöcke in Abb. 7.1 zwischen der Datenbasis und der Benutzerschnittstelle wurde bereits in Kapitel 4.3. bzw. 4.4 näher eingegangen.

Der linke Block faßt die Objekte und Methoden zur Navigation durch den Prozeßablauf und die Darstellung der Prozeßbeschreibungselemente zusammen. Wegen der größeren Effizienz von C++ werden hierzu zunächst die in der NEXPERT- Datenbasis vorhandenen Objekte als C++ Objekte erzeugt. Um die Objekte weiter zu realisieren werden außerdem unabhängige Tools z.B. zur Visualisierung von Texten oder Graphiken eingebunden.

Der mittlere Block beschreibt das Modul zur Darstellung und Simulation quantitativer Zusammenhänge. Die in Kapitel 4.4 genannten Klassen zur Umsetzung des Konzeptes wurden in C++ implementiert. Um Funktionsausdrücke bei der Simulation auszuwerten, wurde ein Parser benötigt, mit welchem die Syntax einer einfachen Sprache zur Berechnung mathematischer Funktionen, wie sie z.B. in einer höheren Programmiersprache zur Verfügung stehen und zur Überprüfung der Korrektheit der Beschreibung von Beziehungen umgesetzt werden kann. Hierzu wurde BISON, eine objektorientierte Erweiterung des standardmäßigen UNIX-Parsergenerators YACC eingesetzt. Damit ist es möglich eigene Klassen zur Über-

prüfung und Auswertung von Ausdrücken nach einer zuvor spezifizierten Syntax zu implementieren. Für die Darstellung von Funktionen und Graphiken wurden Routinen der frei verfügbaren Bibliothek PLPLOT verwendet. Die Realisierung der Behandlung scharfer Zusammenhänge wurde im Rahmen einer Zusammenarbeit mit der TU-Dresden durchgeführt.

Der rechte Block steht für die in Kapitel 4.4 zusammengefaßten Klassen zur Behandlung unscharfer Zusammenhänge und der Fuzzy-Inferenz. Der Ablauf der Auswertung unscharfer Zusammenhänge ist in Anhang C näher ausgeführt.

Abb. 7.2 zeigt das zu Beginn erscheinende Kontrollfenster zur Aktivierung grundlegender Systemfunktionen.

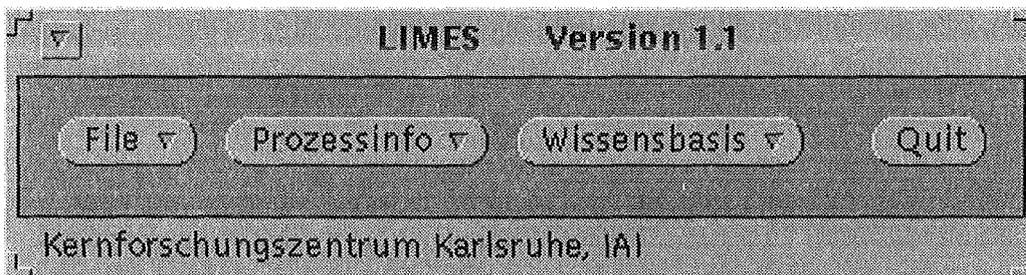


Abb. 7.2 LIMES-Kontrollfenster.

Mit Hilfe des *File*-Menüs können Teilwissensbasen über den Prozeß geladen oder gelöscht werden.

Das *Prozessinfo*-Menü dient zur Aktivierung der Darstellung des Prozeßablaufes.

Das Menü *Wissensbasis* ermöglicht Eingabe und Änderungen an der Wissensbasis.

In der vorliegenden Realisierung wird der Umfang der beschreibbaren Prozesse lediglich durch die zur Verfügung stehende Information und den vorhandenen Speicherbereich begrenzt. Es können also jederzeit beliebig viele neue Prozeßschritte und andere Beschreibungselemente in das Modell aufgenommen oder bezüglich ihrer Beschreibung erweitert werden. Die Anzahl der erfaßten Prozeßschritte umfaßt derzeit ca. 70 Prozeßschritte zur Herstellung von Röntgenmasken. Der Detaillierungsgrad der Beschreibung der einzelnen Schritte unterscheidet sich dabei in Abhängigkeit der zur Verfügung stehenden Informationen sowie der Komplexität der beschriebenen Schritte. Ein Teil der Prozeßschritte (Herstellung einer Zwischenmaske über die direkte Elektronenstrahlolithographie) wurde dazu ausgewählt, die Simulation von Parameterabhängigkeiten zu testen. Einem Prozeßschritt werden normalerweise mehrere Parameter (abhängig von der Komplexität des Schrittes ca. 4-10) zugeordnet. Obwohl nicht allen Prozeßschritten neue Parameter zugeordnet werden (Inspektionsschritte sind auch Prozeßschritte, besitzen jedoch keine eigenen Parameter), übersteigt also die Zahl der Prozeßparameter die der Prozeßschritte deutlich.

7.2 Wissenserwerb und Prozeßumfang

Die generelle Strategie des vorliegenden Ansatzes ist es, die Struktur der Wissensbasis so zu entwerfen, daß der Wissenserwerb des Systems direkt von den Prozeßfachleuten selbst durchgeführt werden kann. Dies setzt natürlich eine interaktive, benutzerführende graphische Oberfläche voraus, die es gestattet, vorhandenes Wissen direkt am Arbeitsplatz zu ergänzen und zu korrigieren. Diese Vorgehensweise besitzt den Vorteil, daß der Dialog mit einem Wissensingenieur entfällt. Weiterhin erübrigt sich eine Übersetzung des Prozeßwissens in das Format der Wissensbasis, welche eine zu hohe zusätzliche Belastung verursachen und eine zusätzliche Fehlerquelle darstellen würde. Da der Schwerpunkt der Arbeit in der Modellierung des Prozeßwissen liegt, wurde die Komponente zur Wissensakquisition nur insoweit realisiert, wie es für einen flexiblen Aufbau der prototypischen Wissensbasis notwendig war.

Bei der Wissenserfassung muß die Art der Repräsentation des vorliegenden Wissens berücksichtigt werden. Für die Akquisition objekt- bzw. framebasierter Wissensinhalte kann die graphische Benutzeroberfläche von NEXPERT-*Object* genutzt werden. Diese wird über das Menü *Wissensbasis* aktiviert. Durch die graphischen Browser und Editoren können das Klassen- und Objektschema angezeigt und verändert und die Beschreibung der Attribute einzelner Objekte vorgenommen werden (Abb. 7.3).

Die Eingaben werden von NEXPERT-*Object* syntaktisch und auf einfache Konsistenzbedingungen (Namensgleichheit etc.) überprüft, bevor sie als Fakten in die Wissensbasis übernommen werden. Die graphischen Möglichkeiten erlauben das rasche Auffinden gesuchter Objekte und Attribute, um Änderungen durchzuführen. Das Datenmodell von NEXPERT-*Object* ermöglicht Attribute nicht nur durch Daten, sondern auch durch intern zur Verfügung gestellte oder externe in der Sprache C geschriebene Methoden zu beschreiben. Über die Methoden können die Attributwerte der Objekte verändert und Methoden anderer Objekte aktiviert werden. Durch die Attribute lassen sich also umfassende Operationen, wie sie z.B. für die Anzeige und Änderung bestimmter Dokumente oder die Darstellung von Bildmaterial notwendig sind, beschreiben.

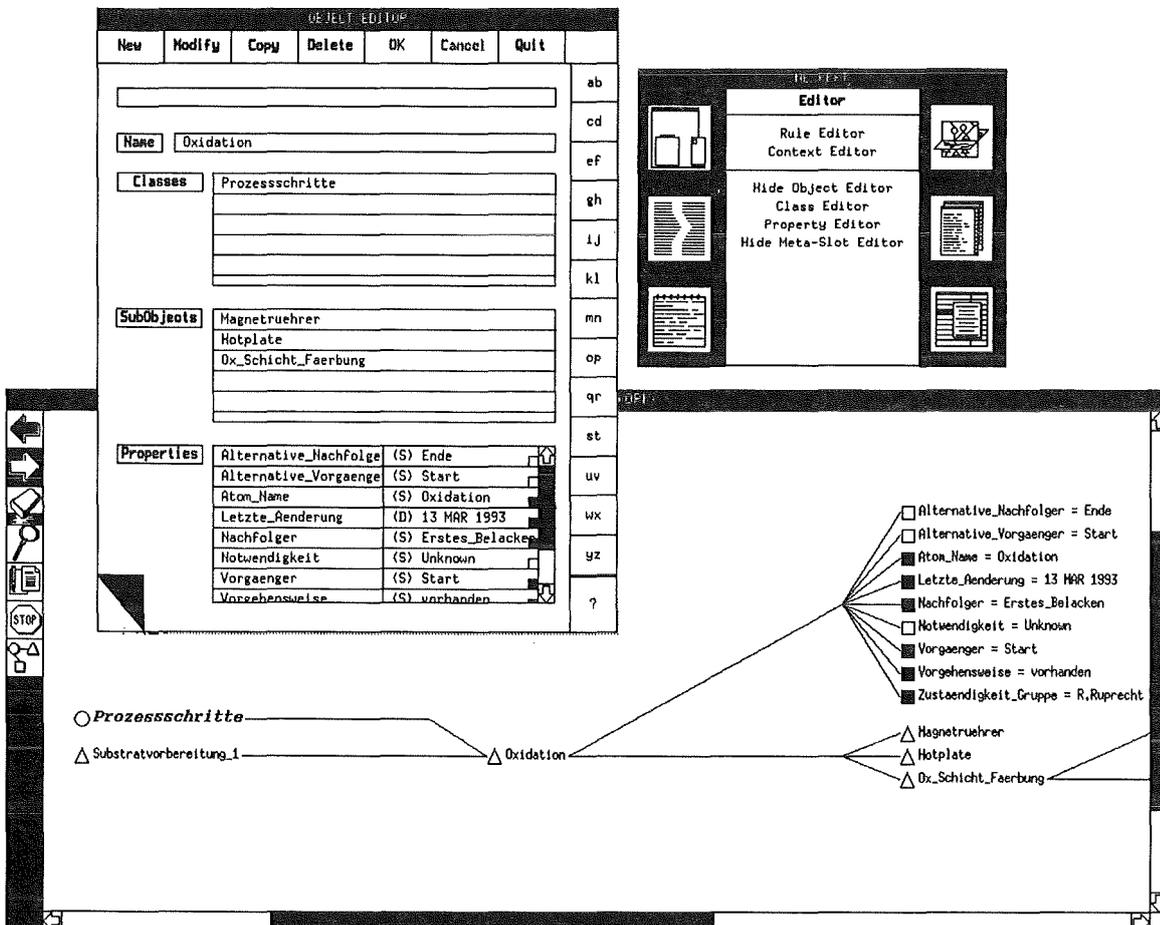


Abb. 7.3 Benutzeroberfläche von NEXPERT-Object mit Editor für Prozessschritte und graphischem Browser zur Anzeige von Objektbeziehungen und Attributen.

Die Erfassung scharfer Zusammenhänge und Beziehungen, sowie der Regeln zur Beschreibung unscharfer Zusammenhänge und Randbedingungen erfolgt bisher in speziellen Dateien, in welchen diese Informationen in einer eigenen Syntax niedergelegt werden.

Die graphische Benutzerschnittstelle von NEXPERT-Object bietet die Möglichkeit einer effektiven Wissenserfassung während der Entwicklungs- und Testphase eines Systems. Eine Benutzerführung durch die Oberfläche, die für die Abfrage von Wissen von Prozeßfachleuten unbedingt benötigt wird, ist jedoch nicht vorhanden.¹ NEXPERT-Object fehlen hierzu wichtige Mechanismen, um den Ablauf eines Benutzerdialoges effektiv zu steuern oder die Eingabe komplexer Datenstrukturen in übersichtlicher, z.B. tabellarischer Form zu unterstützen.

Für eine praktisch einsetzbare Version der Wissenserwerbskomponente des Systems, die die

1. NEXPERT Object stellt daher lediglich die Möglichkeit bereit, alle ausführbaren Funktionen über Bibliotheksfunktionen (in C) zu aktivieren, die in Benutzerprogramme eingebunden werden können.

direkte Wissensangabe durch die Prozeßfachleute unterstützt, wird es also erforderlich sein, das System um eine den Benutzer führende Dialogkomponente zu erweitern.

Durch Auswahl des *Prozeßinfo* Menüs wird die Darstellung des Prozeßablaufs aktiviert (Abb. 7.4). Diese erlaubt die Anzeige durchzuführender Prozeßschritte, ihrer Alternativen sowie ihrer zeitlichen und hierarchischen Ordnung und stellt damit den zentralen Ausgangspunkt für die Präsentation detaillierterer Informationen über den Prozeß dar.

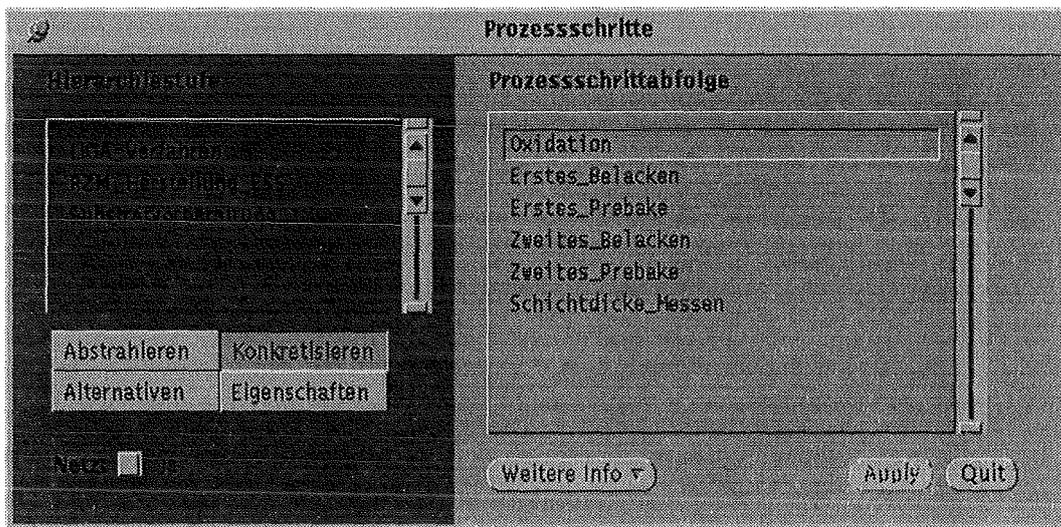


Abb. 7.4 Darstellung des Prozeßablaufes.

Auf der rechten Seite der Prozeßablauf-Darstellung wird die zeitlich geordnete Prozeßabfolge zur Realisierung eines Verfahrensabschnittes angegeben. Durch die Funktionen *Abstrahieren* und *Konkretisieren* kann zwischen verschiedenen Abstraktionsebenen für die Sicht auf den Prozeß gewählt werden. Auf der linken Seite wird durch Angabe der Hierarchiefolge gezeigt, welcher Prozeßabschnitt gerade dargestellt wird.

Durch die Wahl des *Alternativen*-Schalters werden mögliche Alternativen für einen Prozeßschritt oder Alternativvarianten einer Prozeßschrittfolge dargestellt und ausgewählt. Die Auswahl des Schalters *Eigenschaften* führt zur Anzeige der Bezeichnung der Attribute, durch die die Prozeßschritte näher beschrieben werden (Abb. 7.5).

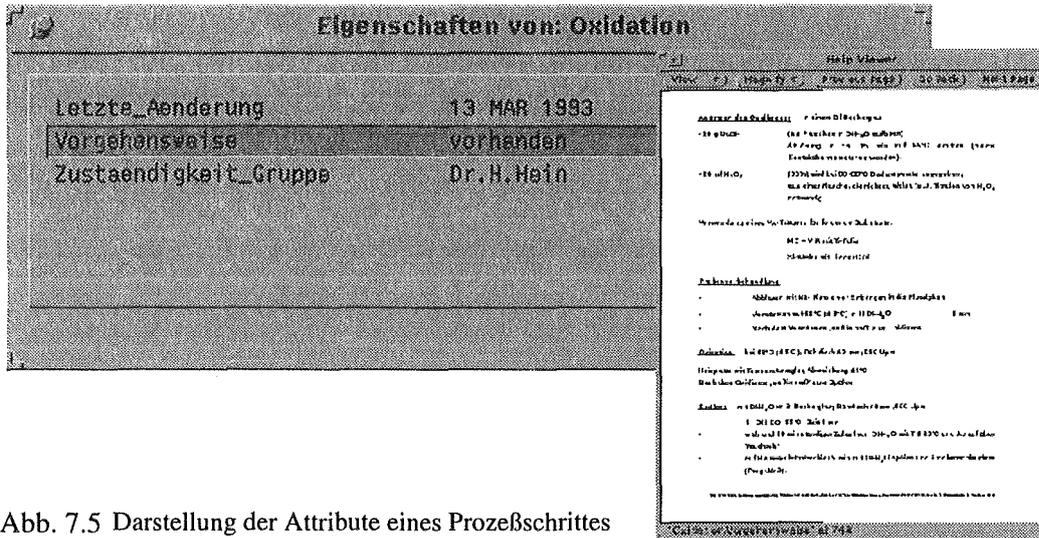


Abb. 7.5 Darstellung der Attribute eines Prozessschrittes

Die Attribute können völlig unterschiedliche Datentypen besitzen. Sie werden daher im allgemeinen durch Objekte beschrieben, welche die Darstellung von numerischen Werten, Dokumenten, Bildern etc. durch die ihnen assoziierten Methoden selbst übernehmen.

Durch das Menü *Weitere Info* aus Abb. 7.6 lassen sich die einem Prozessschritt zugeordneten Arbeitsmaterialien, Ausrüstungsgegenstände und Prozeßparameter ermitteln und anzeigen. Diese Beschreibungselemente des Prozesses werden ebenfalls durch Objekte repräsentiert, deren spezifische Eigenschaften analog zu den Prozessschritten aus ihren Attributen entnommen werden können.

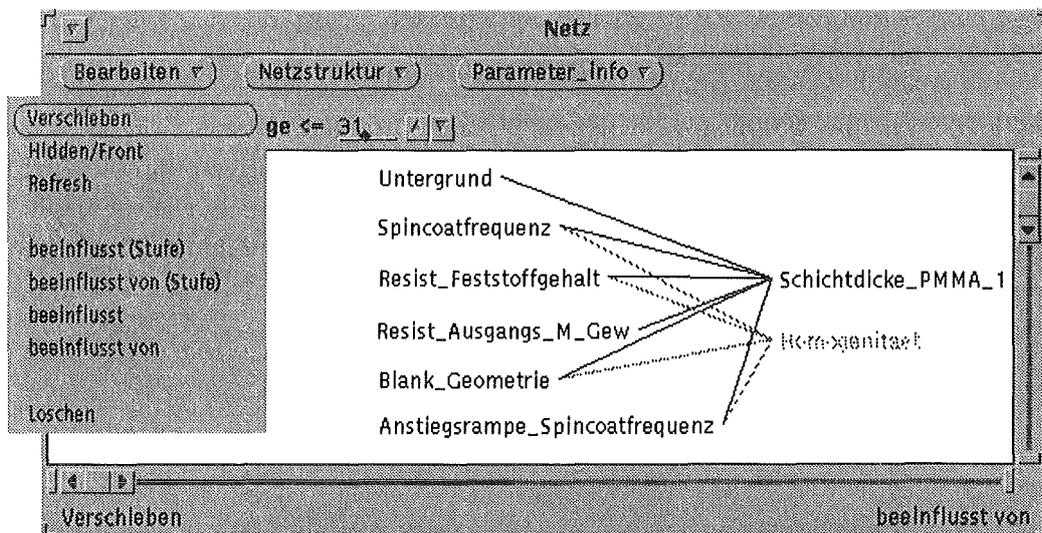


Abb. 7.6 Graphische Darstellung des Parameterabhängigkeitsnetzes.

7.3 Visualisierung von Parameterabhängigkeiten des Prozesses

Um das Ergebnis eines Prozessschrittes quantitativ zu beschreiben, ist die Kenntnis der ihn charakterisierenden Parameter, ihrer Werte und Beziehungen erforderlich. Wird in der Prozeßablauf-Darstellung der Wahlschalter *Netz* gewählt (Abb. 7.6), so wird eine graphische Netzdarstellung aktiviert. Diese faßt die Informationen über die an einem Prozessschritt beteiligten Parameter und ihre gegenseitigen qualitativen Abhängigkeiten zusammen.

Durch das *Bearbeiten*-Menü kann die Darstellung des Netzes zur Erhöhung der Übersichtlichkeit beeinflusst werden. Dies geschieht durch *Verschieben* von Parametern auf der Zeichenfläche, die Hervorhebung der Beeinflussung oder des Einflusses eines bestimmten Parameters (*beeinflusst, beeinflusst von*) oder das Entfernen von Parametern von der Zeichenfläche (*Löschen*).

Durch das Parameternetz wird bisher keine quantitative Information über die Parameterabhängigkeiten ausgedrückt. Wählt man aus dem Menü *Prozessinfo* den Punkt *Darstellung* aus, so erfolgt die quantitative Darstellung einer Beziehung (Abb. 7.7).

Die Darstellung einer Beziehung umfaßt den Zielparameter der Beziehung und die auf ihn direkt einwirkenden Einflußparameter. Durch Selektieren eines Wertes kann der entsprechende Wertebereich eines Parameters angezeigt, und der Parameterwert geändert werden. Bei der Ermittlung des Zielparameterwertes werden die Einstellparameter als fixiert betrachtet. Einer der fixierten Einflußparameter kann als variabler Parameter markiert werden. Die Darstellung des Zielparameters erfolgt dann in Abhängigkeit dieses variablen Parameters.

Die Art der Darstellung der Abhängigkeit hängt vom Wertetyp der beteiligten Parameter und der Art des Zusammenhanges ab. Aufzählende Zusammenhänge werden in Form einer Tabelle ausgegeben (vgl. Abb. 7.7 unten), kontinuierliche Abhängigkeiten entweder ebenfalls tabellarisch für bestimmte Werte des Wertebereichsintervalles oder graphisch als Funktionsplot (Abb. 7.8).

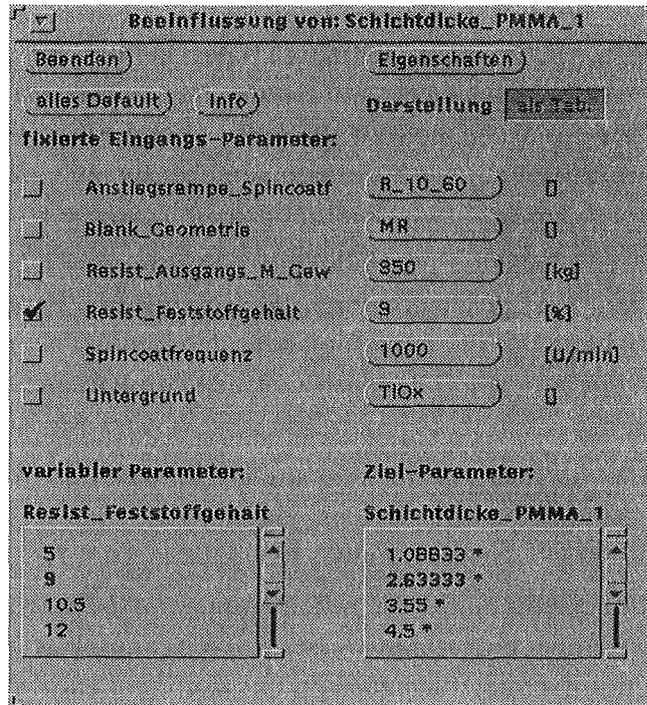


Abb. 7.7 Tabellarische Darstellung eines aufzählenden Zusammenhanges.

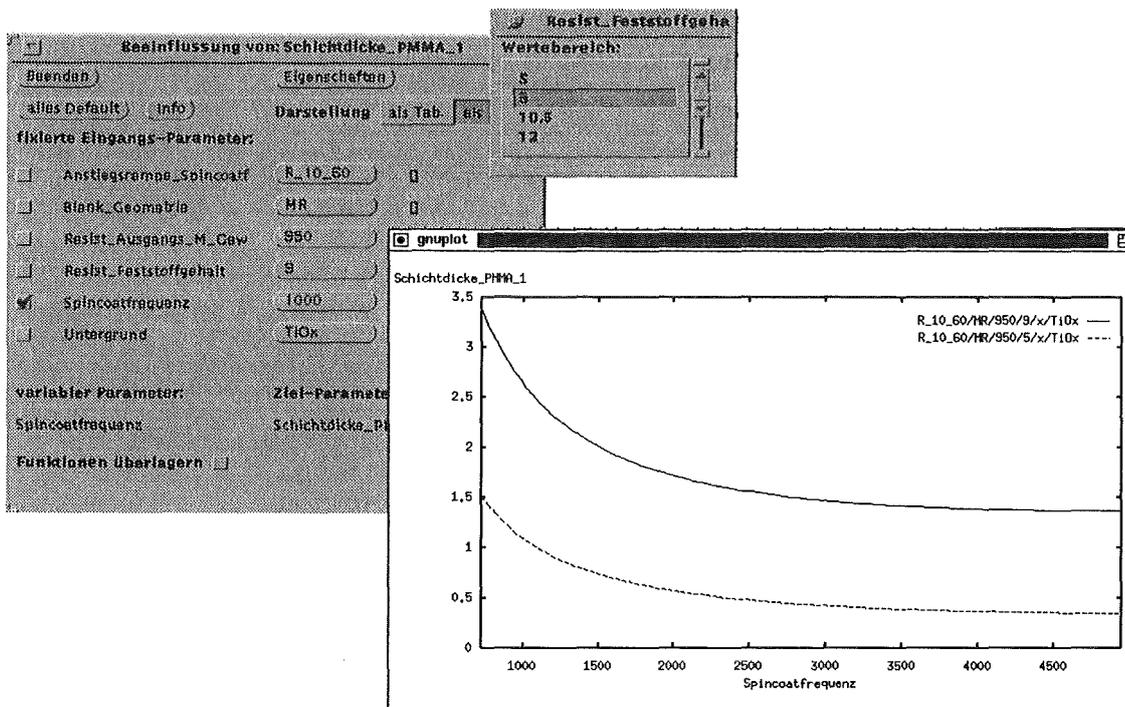


Abb. 7.8 Darstellung kontinuierlicher Zusammenhänge als parametrisierte Funktionenschar.

Unscharfe Parameter unterscheiden sich von scharfen Parametern in Bezug auf die Darstellung ihrer Werte (Abb. 7.9). Ein unscharfer Wert drückt aus, daß verschiedene Werte eines scharfen Wertebereiches angenommen werden können. Die Plausibilität für das Zutreffen bestimmter Werte aus dem Wertebereich wird mit Hilfe einer trapezförmigen Zugehörigkeitsfunktion visualisiert, die jedem linguistischen Bezeichner des unscharfen Wertebereiches zugeordnet wird. Dadurch lassen sich unscharf formulierte Aussagen erfassen und in die Behandlung scharfer quantitativer Größen einbeziehen.

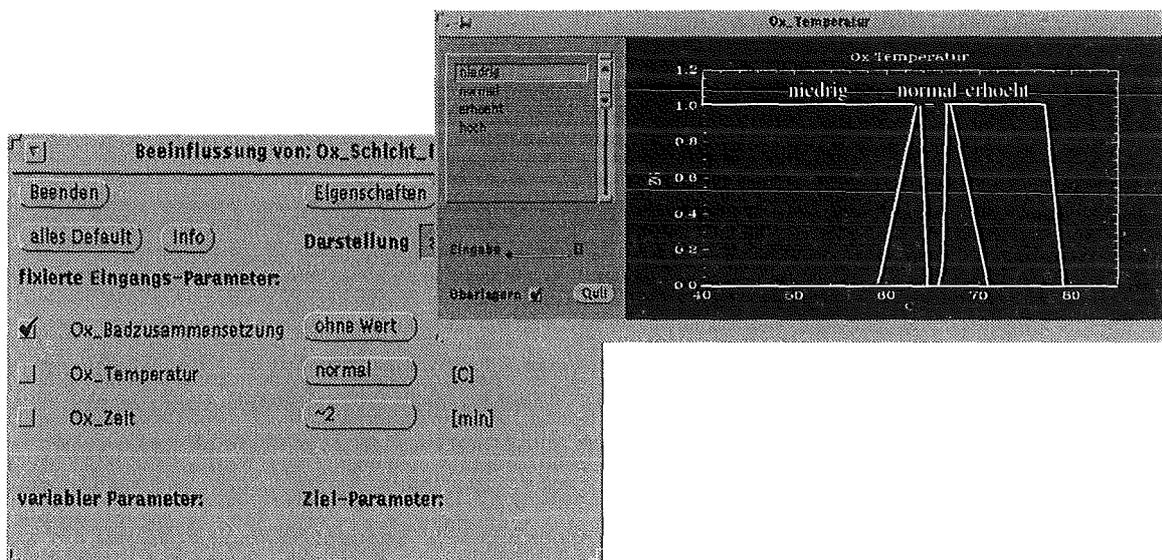


Abb. 7.9 Darstellung unscharfer linguistischer Werte durch Zugehörigkeitsfunktionen.

7.4 Simulation von Parameterwirkungsketten

Die Darstellung der Parameterabhängigkeiten kann auf die simultane Berücksichtigung mehrerer Beziehungen unterschiedlicher Prozessschritte ausgedehnt werden. Dies geschieht bei der Simulation von Parameterwechselwirkungen. Die Möglichkeiten der Simulation sollen am Beispiel des Resistauftrags für eine Röntgenzwischenmaske diskutiert werden.

Zur Ausdehnung des Parameternetzes auf Parameter unterschiedlicher Prozessschritte können die Unterpunkte *beeinflusst* und *beeinflusst von* des *Netzstruktur*-Menüs benutzt werden. Abb. 7.10 zeigt ein Parameternetz, welches die Abhängigkeiten beim mehrfachen Belacken

und Tempern eines Resists für eine Röntgenzwischenmaske beschreibt.

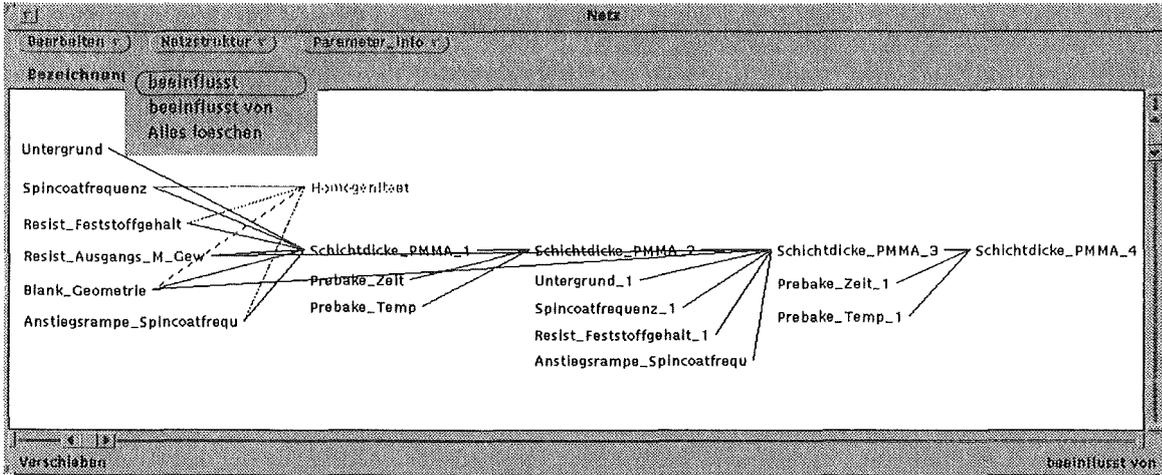


Abb. 7.10 Parameterabhängigkeiten über mehrere Prozessschritte.

Der Menüpunkt *Simulation* aus dem Menü *Parameter_info* und die Wahl eines Zielparameters durch Anklicken im Parameternetz erlaubt es, ein quantitatives Abhängigkeitsnetz analog dem Abhängigkeitsnetz von Abb. 7.10 aufzubauen. Dies ist in Abb. 7.11 dargestellt.

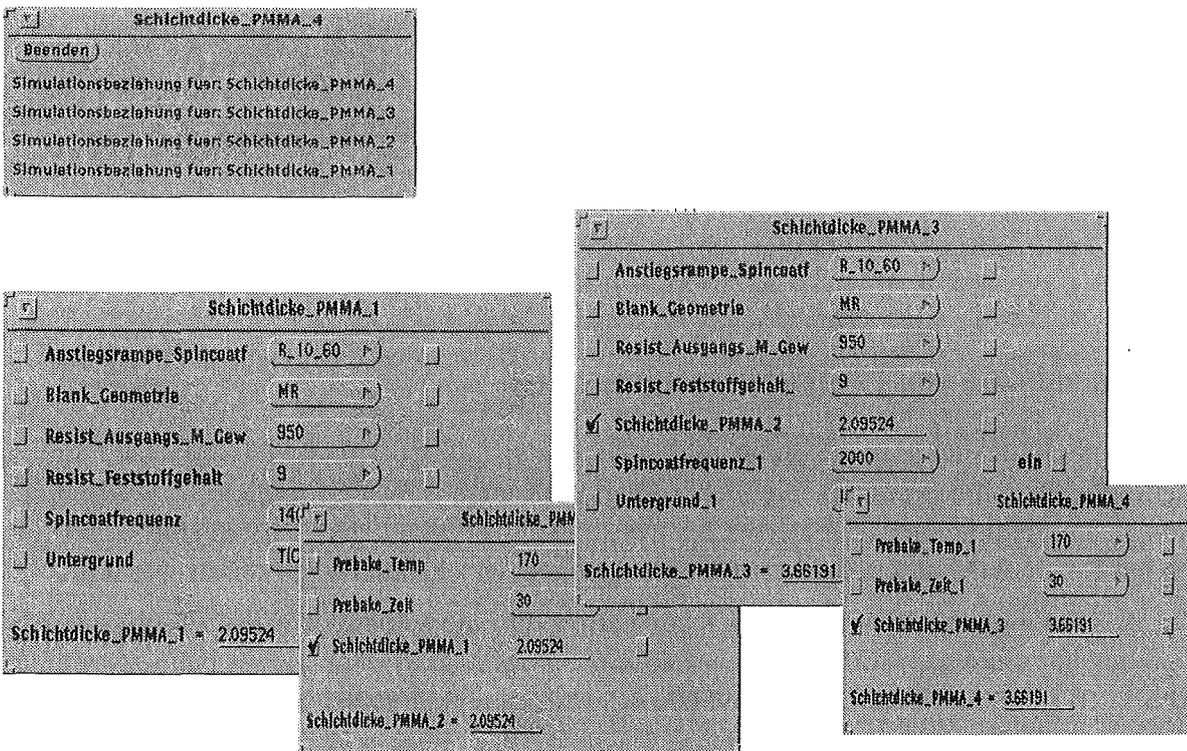


Abb. 7.11 Simulationsnetz bestehend aus fünf Parameterbeziehungen.

Ziel einer Simulation ist die Bestimmung des Zielparameterwertes in Abhängigkeit der ihn beeinflussenden und in die Simulation einbezogenen Parameter.

Bei der Simulation stehen die Beziehungen der Parameter nicht isoliert, sondern sind durch gemeinsame Parameter verbunden, die in einer Beziehung als Ziel- und in einer anderen als Einflußparameter auftauchen. Für Einstellparameter, also diejenigen Parameter, die in keiner Beziehung einer Simulation als Zielparameter vorkommen, werden Standardwerte, die für die Prozeßführung verwendet werden, angenommen. Die Werte der Einstellparameter können wahlweise über ein Eingabefeld oder ein Auswahlménü geändert werden. Darauf werden die Zielparameter der berücksichtigten Beziehungen berechnet und in den Ergebnisfeldern angezeigt.

Dabei lassen sich in der Simulation auch Seiteneffekte berücksichtigen. Beispielsweise wirkt sich die Spincoatfrequenz aus der Beziehung *Schichtdicke_PMMA_1* aus Abb. 7.11 gemäß Abb. 7.10 nicht nur auf die Schichtdicke nach dem anschließenden Temperprozeß aus, sondern auch auf die Homogenität der Resistschicht. Um diesen Einfluß zu berücksichtigen, kann von der Beziehung *Schichtdicke_PMMA_1* aus die Beziehung *Homogenität* in die Simulation eingebunden werden. Dadurch lassen sich die Randbedingungen, die aus der gleichzeitigen Erfüllung mehrerer Zielwerte resultieren, in der Simulation ebenfalls mitefassen.

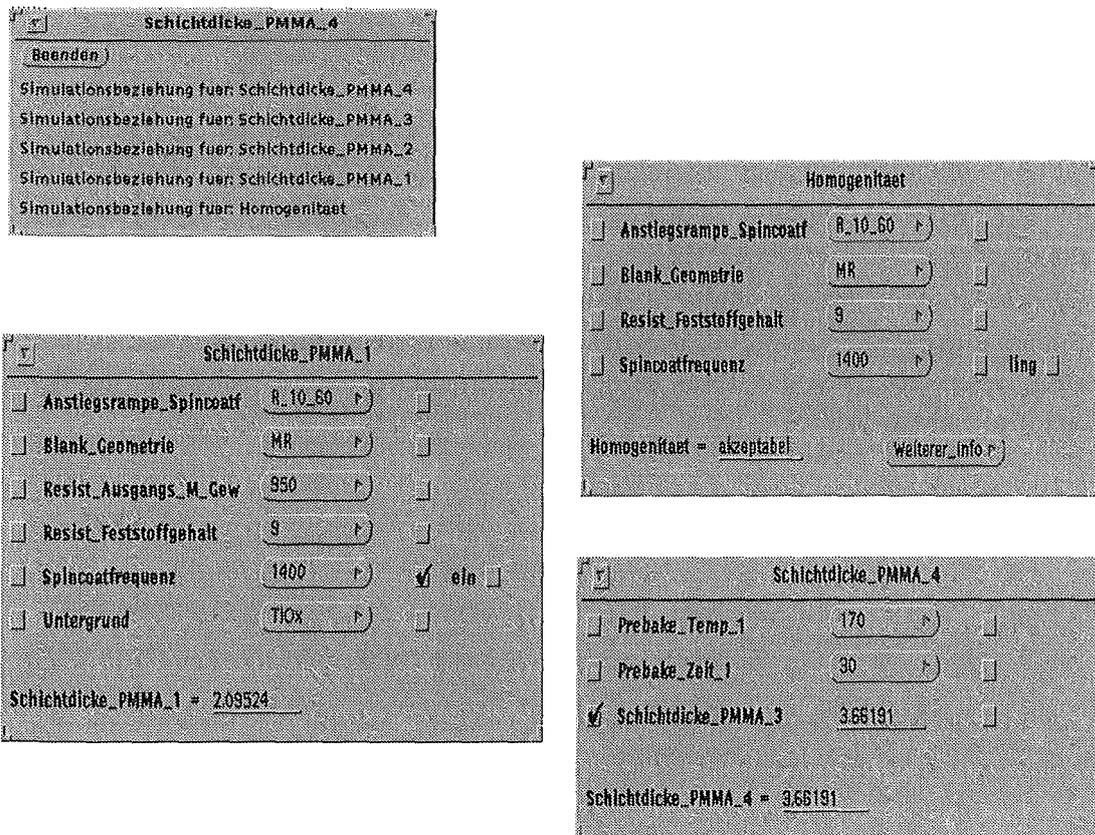


Abb. 7.12 Gleichzeitige Berücksichtigung verschiedener Auswirkung eines Einflußparameters.

Die Auswirkung eines Parameterwertes im Modell kann durch Zusammenhänge unterschiedlichen Typs berücksichtigt werden. So kann die Auswirkung der Spincoatfrequenz in Abb. 7.12 über einen scharfen funktionalen Zusammenhang beschrieben werden, während die Beschreibung der Wirkung auf die Homogenität nur in ungenauer Weise vorliegt. Das Ergebnis des Einflusses der Spincoatfrequenz wird hier also in Form des Wertes einer linguistischen Variable angezeigt.

Die Behandlung linguistischer Variablen wird im folgenden am Beispiel der Oxidation einer Maskenträgerfolie aus Abschnitt 2.1.3 diskutiert. In Abb. 7.13 ist die Charakterisierung der Helligkeit der erreichten Oxidationsschicht unter Berücksichtigung der Oxidationszeit und Oxidationstemperatur dargestellt.

Für die Oxidationszeit wird ein Standardwert von 2 min angenommen, der Temperaturwert unterscheidet sich in Abbildung Abb. 7.13 mitte und unten.

In der Mitte wird der Temperaturwert selbst als unscharfe Menge zur Beschreibung eines „normalen“ Temperaturwertes vorgegeben (vgl. Abb. 7.9). Das Ergebnis ist gemäß Tab. 6.3 eine unscharfe Menge, die den Wert „dunkel“ der Oxidationsschicht charakterisiert, da in der Regelbasis zur Beschreibung des Zusammenhanges eine Regel existiert, für welche exakt die eingestellten Vorbedingungen zutreffen. Die Wertemenge der Zugehörigkeitsfunktion beschreibt dabei die gemessene oder geschätzte Intensität des unter gleichen Bedingungen reflektierten Lichtes. Diese Bedingungen können dabei als Informationstext einer Beziehung zugeordnet werden.

Im unteren Teil der Abbildung wird ein scharfer Wert als Eingabe für die Temperatur vorausgesetzt. Der Wert der Eingabe liegt diesmal mit 67°C zwischen den als „normal“ bzw. „erhöht“ bezeichneten Temperaturbereichen (vgl. Abb. 7.9). Daher liegt auch das Ergebnis der Inferenz zwischen den mit „dunkel“ bzw. „schwarz“ bezeichneten Bereichen (vgl. Abb. 7.13 oben).

Dies unterscheidet die hier gewählte Inferenzmethode von der üblicherweise im Fuzzy Control Bereich eingesetzten Inferenz. Bei dieser würde im letzten Fall als Ergebnis eine breite Verteilung resultieren, die die Bereiche der Mengen „dunkel“ und „schwarz“ der Schichtfärbung umfassen würde.

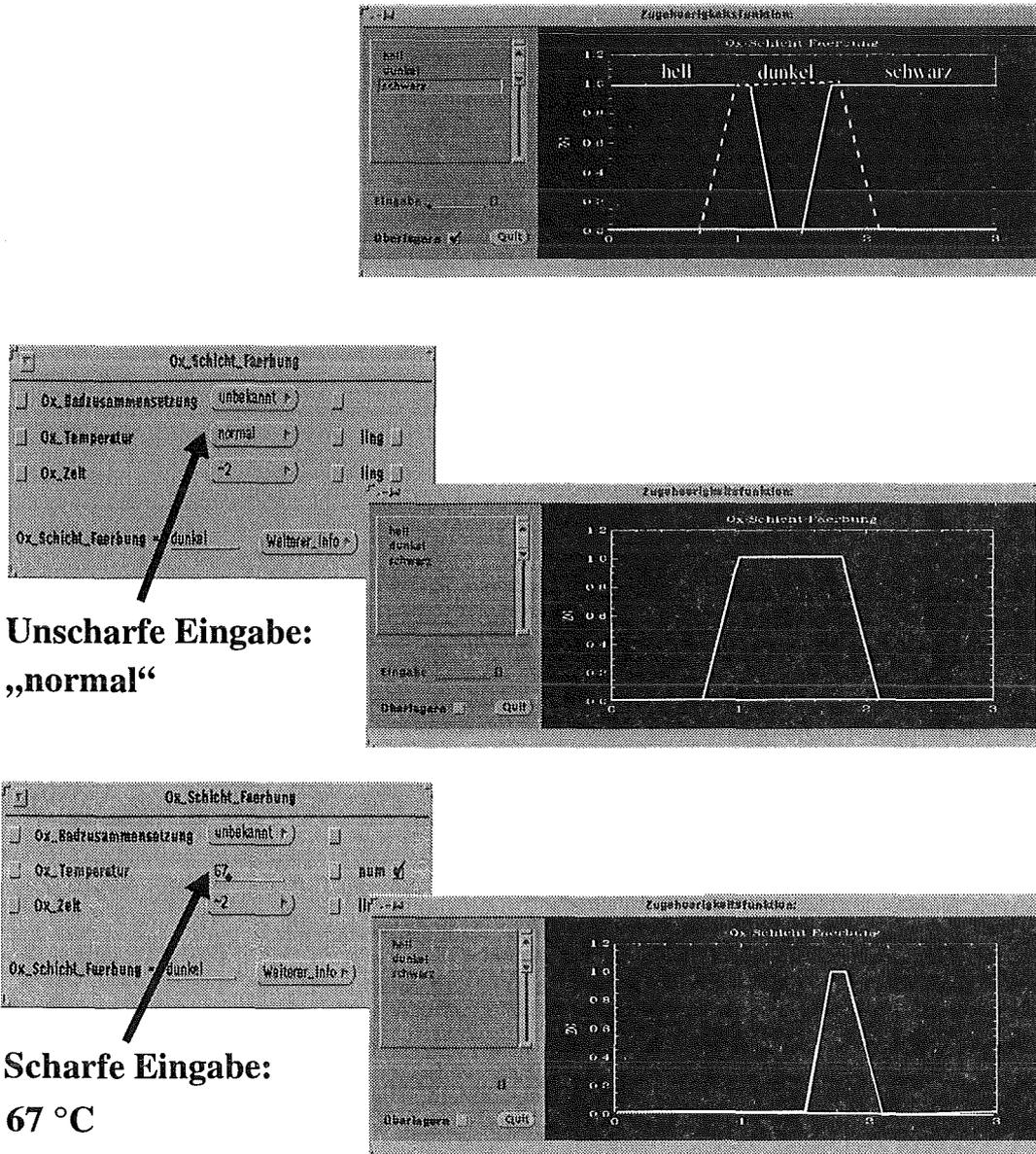


Abb. 7.13 Darstellung linguistischer Variablen als Ergebnisse bei unscharfen (mitte) und scharfen (unten) Eingangswerten.

7.5 Erklärungskomponente für die Ergebnisse

Im Rahmen der Simulation unscharfer Zusammenhänge werden bei der Ermittlung der Zugehörigkeitsfunktion im allgemeinen mehrere Regeln gleichzeitig aktiviert. Durch den Menüpunkt *Begründung_des_Ergebnisses* des Menüs *Weitere_Info* kann die Darstellung der zum Ergebnis beitragenden Regeln aktiviert werden (Abb. 7.14). Unter der Liste dieser Darstellung werden die erfüllten Regeln angezeigt, die Liste selbst enthält die Anforderungen an die Eingangsparameter der Regel (*wenn* Teil). Im *dann* Teil wird der Wert des Zielparameters unter der Voraussetzung, daß alle Eingangsbedingungen erfüllt sind, angezeigt. Auf diese Weise kann auch das Zustandekommen von Ergebnissen, die aufgrund vagen Wissens abgeleitet wurden erklärt und begründet werden.

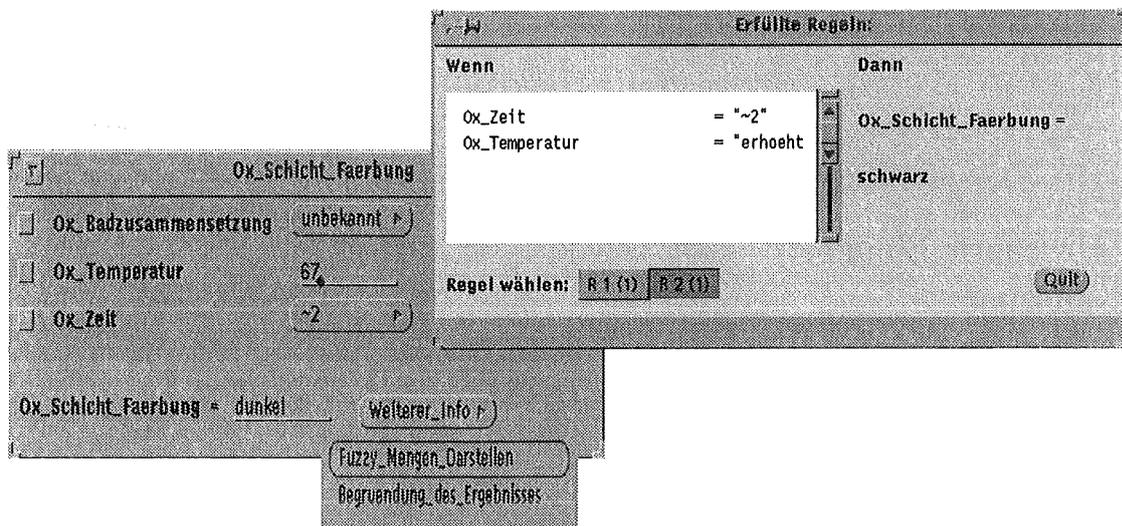


Abb. 7.14 Begründung des Zustandekommens der Ergebnisse bei der Inferenz mit unscharfen Werten.

Im Fall der Simulation scharfer Zusammenhänge ist die Begründung der Ergebnisse ebenfalls möglich. Hier muß aufgrund der modularen Beschreibung der Parameterabhängigkeiten lediglich das Zustandekommen der Zielparameter der an einer Simulation beteiligten Beziehungen nachvollzogen werden. Zu diesem Zweck lassen sich die Darstellungen der einzelnen Beziehungen nutzen, in denen die Beschreibung der Abhängigkeiten in Form von Tabellen, graphischen Plots oder in schriftlicher Form (*Info-Schalter*) untersucht werden kann.

8. Zusammenfassung und Ausblick

8.1 Zusammenfassung

Das LIGA-Verfahren nimmt aufgrund seiner vielfältigen Möglichkeiten eine Sonderstellung unter den Mikrostrukturierungsverfahren ein. Hinter dem Verfahren verbirgt sich eine ganze Palette von Prozessen und Prozeßvarianten zur Realisierung unterschiedlichster Geometrie-Material- und Genauigkeitsanforderungen an die zu fertigenden Mikrokomponenten. Gerade wegen der Vielgestaltigkeit und Komplexität des Verfahrens kommt der Repräsentation von Fertigungswissen eine besondere Bedeutung zu.

Folgende Gründe lassen insbesondere eine systematische rechnergestützte Repräsentation des Prozeßwissens unverzichtbar erscheinen:

- Der große, sich ständig ändernde Umfang von Prozeßschritten und Prozeßvarianten zur Herstellung von Mikrostrukturen in LIGA-Technik.
- Die Notwendigkeit, Wissen als wichtigen Produktionsfaktor zentral und möglichst unabhängig von menschlichen Fehlerquellen zu dokumentieren und verfügbar zu machen.
- Die Aufgabe, in Zukunft Informationen für andere rechnergestützte Werkzeuge zur Herstellung von Mikrokomponenten und -systemen bereitzustellen.

Ziel der Arbeit war es zunächst das über den Prozeß vorliegende Wissen zu strukturieren so daß es im Rahmen eines Informationssystems auf dem Rechner verfügbar gemacht werden kann.

In der Arbeit wurde daher auf der Grundlage objektorientierter Prinzipien ein Basismodell entwickelt, in welches komplexes Wissen über die wesentlichen Prozeßelemente wie Prozeßschritte, Prozeßparameter, Arbeitsmaterialien und Ausrüstungsgegenstände abgelegt und verarbeitet werden kann. Dabei wurde sowohl die Struktur der einzelnen Elemente in Form der zu ihrer Beschreibung notwendigen Attribute, als auch ihre gegenseitigen Abhängigkeiten mitefaßt.

Da bei der Durchführung und Weiterentwicklung des Prozesses die Kenntnis der Auswirkungen bestimmter Parametereinstellungen auf Produkteigenschaften im Vordergrund steht, waren diese Abhängigkeiten im Modell ebenfalls zu erfassen.

Bei der tiefgehenden Modellierung des Prozesses standen daher Beziehungen zwischen Prozeßparametern, welche sowohl die Eingangsgrößen als auch die Ergebnisse eines Prozeßschrittes charakterisieren, im Vordergrund. Zur Erfassung dieser Zusammenhänge auf dem Rechner wurde ein Modell für die Simulation von Parameterbeziehungen entwickelt und zusammen mit dem Basismodell zu einem wissensbasierten Prozeßmodell integriert. Dadurch wurde die Möglichkeit geschaffen, Abhängigkeiten zwischen Parametern, die für die Produkteigenschaften wesentlich sind vollständig zu erfassen und aus, innerhalb vorgegebener Grenzen, frei wählbaren Parametereinstellungen bestimmte Zielparameterwerte zu ermitteln.

Ein komplexer Prozeß, wie das LIGA Verfahren ist nicht durchzuführen, ohne daß auch heuristisches und unscharfes Wissen in die Prozeßdurchführung miteinfließt. Es war daher notwendig auch diese Aspekte bei der Beschreibung der Parameterabhängigkeiten und der Simulation zu berücksichtigen.

Eine wesentliche Rolle spielte die Einbeziehung von unscharf formulierten Zusammenhängen in die Modellierung. Die Anwendung von Fuzzy Logik erlaubt es, unscharfe Zusammenhänge und Randbedingungen für die Parameterbeziehungen in Form allgemeinverständlicher linguistischer Regeln zu formulieren. Da die Beschreibung unscharfer Werte in der Fuzzy Mengentheorie letztlich mit Hilfe von Elementen scharfer Wertemengen erfolgt, lassen sich unscharf formulierte Zusammenhänge auch auf scharfe Parameterwerte anwenden. Damit läßt sich die Beschreibung unscharfer Beziehungen völlig in das Modell integrieren.

Die Parameter von Prozeßschritten besitzen im allgemeinen Auswirkungen auf die Ergebnisse nachfolgender Prozeßschritte. Im Fall der langen, komplexen und ständigen Änderungen unterworfenen Prozeßketten des LIGA-Verfahrens sind diese Auswirkungen oft schwer zu überblicken und vorherzusehen. Das entwickelte Modell gestattet es daher, die Auswirkungen bestimmter Parametereinstellungen auf vorgegebene Zielparameter unter Berücksichtigung der im System beschriebenen Abhängigkeiten zu simulieren.

Um die entwickelten Konzepte zu testen, wurden diese in einer prototypischen Implementierung des Systems LIMES realisiert. Anhand einer Beschreibung des Prozesses zur Herstellung von Röntgenzwischenmasken wurde gezeigt, daß die im Konzept vorgegebenen Möglichkeiten geeignet sind, das vorhandene Wissen über den Prozeßablauf zu repräsentieren und zu dokumentieren. Weiter konnten Zusammenhänge zwischen Prozeßparametern vollständig beschrieben und realistische Vorhersagen von Ergebnissen aufgrund von Änderung der Parametereinstellungen getroffen werden.

Als wesentlicher Aspekt der Realisierung des Systems wurde berücksichtigt, daß das System offen gegenüber Änderungen und Erweiterungen der Wissensbasis ist. Dies betrifft sowohl die Beschreibung des Basismodells, als auch die Beziehungen zwischen den Prozeßparametern. Damit wurde nachgewiesen, daß das vorgeschlagene Konzept eine tragfähige Grundlage für eine rechnergestützte Dokumentation und Simulation des LIGA-Verfahrens darstellt.

8.2 Ausblick

In der vorliegenden Arbeit wurde schwerpunktmäßig ein wissensbasiertes Modell zur Repräsentation und Simulation von Prozeßwissen entwickelt. Die Leistungsfähigkeit des Systems konnte anhand einer prototypischen Realisierung des Systems LIMES demonstriert werden. Der durch den Einsatz objektorientierter Methodik erhaltene Systementwurf (Abschnitt 3.2) kann im Prinzip für eine praktisch einsetzbare Version von LIMES verwendet werden. Für eine real einsetzbare Version sind dabei folgende Gesichtspunkte besonders zu berücksichtigen:

- Objekte, die zur Beschreibung des Prozesses angelegt werden, müssen persistent sein. Die Persistenz der Objekte muß die Aspekte der Datensicherheit und -integrität, einen effektiven Zugriff und eine Versionshaltung ermöglichen.
- Es werden Mechanismen benötigt, um den gleichzeitigen Zugriff von verschiedenen Benutzern und von verschiedenen Rechnern aus auf die Objekte zu ermöglichen.
- Für die Eingabe und Änderung von Prozeßwissen muß, wie dies bereits für die Wissensdarstellung der Fall ist, eine graphisch interaktive und dialogorientierte Benutzerschnittstelle zur Verfügung stehen, um das Prozeßwissen direkt von den Systemanwendern zu erfassen.

Die ersten beiden Punkte sprechen dafür, das System auf der Basis eines objektorientierten Datenbank Management Systems (OODBMS) zu reimplementieren. Da die bisher verwendete Sprache C++ Grundlage vieler OODBMS's ist, können die bisher realisierten Klassen mit geringfügigen Änderungen übernommen werden. Die in den ersten beiden Punkten gestellten Forderungen werden dann weitgehend durch das Datenbanksystem erfüllt. Auf die Notwendigkeit den dritten Punkt weiter zu verwirklichen, wurde bereits in Kapitel 7 hingewiesen.

Bei der Konzeption und Umsetzung des Modells wurde stets darauf geachtet, daß Erweiterungen am Modell und des eingegebenen Wissens so weit wie möglich ohne Änderungen des bisherigen Modells oder Wissens durchgeführt werden können. Diese Eigenschaft soll auch bei der Implementierung unter einem OODBMS erhalten bleiben. Deshalb bleibt für den dritten Punkt zu klären, inwieweit die Mechanismen der Integrität und der Versionshaltung bei objektorientierten Datenbanken dazu eingesetzt werden können, die Konsistenz bei Änderungen und Löschen von Objekten zu wahren. Diese sind dann an die entsprechenden Aktionen der Benutzeroberfläche der Systemkomponente zum Wissenserwerb zu koppeln.

Für die Zukunft scheint das Ziel einer Integration des vorliegenden Systems mit anderen Systemen zur Unterstützung der Entwicklung und Realisierung von Mikrokomponenten und -systemen erstrebenswert. In einer solchen Umgebung können wichtige Informationen, z.B. aus Konstruktion, Fertigung und Qualitätssicherung auf Rechnern zur Verfügung gestellt werden, damit die Mikrosystemtechnik ebenso wie die Mikroelektronik vom Einsatz rechnergestützter Werkzeuge profitieren kann.

Anhang A

Grundlagen der Fuzzy-Mengen Theorie

Der vorliegende Abschnitt soll keine geschlossene Einführung in die Fuzzy-Mengen Theorie geben. Es werden lediglich Begriffe bereitgestellt, die für die Diskussion der Anwendung und Eigenschaften verschiedener Inferenzmechanismen benötigt werden (für eine umfassendere Einführung siehe z.B. [KRUS93], [ZIMM93]).

A.1 Fuzzy-Mengen und Logik

Zunächst wird der zentrale Begriff einer Fuzzy-Menge definiert. Dazu wird von der charakteristischen Funktion $\Pi(x)$ ausgegangen, die in der Mathematik zur Beschreibung einer gewöhnlichen (scharfen) Menge X verwendet wird, und die folgendermaßen definiert ist:

$$\Pi : X \rightarrow \{0, 1\}$$
$$\Pi(x) = \begin{cases} 1, & x \in X \\ 0, & \text{sonst.} \end{cases}$$

Eine Fuzzy-Menge ist eine Verallgemeinerung der charakteristischen Funktion, die es erlaubt die Zugehörigkeit zur Menge durch beliebige Werte aus dem Intervall $[0, 1]$ auszudrücken.

Definition 2

Eine Fuzzy-Menge μ von X ist eine Funktion der Grundmenge X in das abgeschlossene Intervall $[0, 1]$.

$$\mu : X \rightarrow [0, 1].$$

Mit $F(X)$ sei die Menge aller Fuzzymengen auf X bezeichnet.

Eine Fuzzy-Menge, die mindestens ein sicher zur Menge gehöriges Element besitzt ($\exists x \in X$ mit $\mu(x) = 1$), kann auch als Possibilitätsverteilung $\pi(x)$ über X interpretiert werden. Die semantische Bedeutung einer Possibilitätsverteilung kann durch die Gegenüberstellung mit einer Wahrscheinlichkeitsverteilung veranschaulicht werden.

Gewöhnlich wird durch die Abbildung

$$P : P(X) \rightarrow [0, 1]$$
$$P(A) = \sum_{x \in A} p(x)$$

einer Menge $A \subseteq P(X)$ mittels der Wahrscheinlichkeitsverteilung $p(x)$ ein Wahrscheinlichkeitsmaß $P(A)$ zugeordnet. $P(A)$ besagt, mit welcher Wahrscheinlichkeit ein in der Menge A liegendes Ereignis für eine gemäß $p(x)$ verteilten Zufallsgröße eintreffen wird. Durch die folgenden Beziehungen werden mit Hilfe der Possibilitätsverteilung $\pi(x)$ dem Wahrscheinlichkeitsmaß entsprechende verallgemeinerte Maße definiert.

Definition 3

$\pi(x)$ sei eine Possibilitätsverteilung über X , $P(X)$ die Potenzmenge von X , $A \in P(X)$,

die Abbildung

$$Poss: P(X) \rightarrow [0, 1] \quad \text{mit}$$

$$Poss(A) = \sup \{ \pi(\omega) \mid \omega \in A \}$$

heißt Possibilitätsmaß von π .

Die Abbildung

$$Nec: P(X) \rightarrow [0, 1] \quad \text{mit}$$

$$Nec(A) = \inf \{ 1 - \pi(\omega) \mid \omega \notin A \}$$

heißt Notwendigkeitsmaß von π .

$Poss(A)$ bzw. $Nec(A)$ drücken aus mit welcher Möglichkeit, bzw. Notwendigkeit ein gemäß $\pi(x)$ verteiltes $x \in X$ in A liegt¹. Possibilitätsmaße können auch für Relationen $R \subseteq A^n$ ($A \in P(X^n)$) definiert werden. $\Pi(A^1, A^2, \dots, A^n)$ wird dann als Möglichkeit interpretiert, mit welcher ein gemäß $\Pi(A^1, A^2, \dots, A^n)$ verteiltes $x \in X^n$ im n-Tupel (A^1, A^2, \dots, A^n) liegt.

Die bisher eingeführten Konzepte ermöglichen es, Aspekte der Unschärfe mit formalen Mitteln zu beschreiben. Konkret drückt sich *Unschärfe* z.B. dadurch aus, daß aufgrund unvollständig vorliegender Information nicht entschieden werden kann, welcher (scharf charakterisierbaren) Qualitätsklasse ein zu fertigendes Produkt zugeordnet werden kann. Unschärfe wird im Rahmen der Fuzzy-Mengen Theorie durch Possibilitätsmaße repräsentiert.

Ein Beispiel für *Unschärfe* ist die vage Beschreibung eines Temperaturwertes als „hoch“. Hier kann eine Fuzzy-Menge verwendet werden, um jedem Temperaturwert des betrachteten Temperaturbereiches einen Zugehörigkeitsgrad zur Menge der als „hoch“ bezeichneten Temperaturen zuzuordnen.

Um eine Aussage über die Exaktheit der Beschreibung eines Objektes durch eine Fuzzy-Menge zu machen, benötigt man den Begriff der Spezifität.

1. Die genannten Fuzzy Maße sind keine Wahrscheinlichkeitsmaße, da sie die Kolmogoroffschen Axiome der Wahrscheinlichkeitstheorie lediglich in einer abgeschwächten Form erfüllen, Wahrscheinlichkeitsmaße sind jedoch Spezialfälle von Possibilitätsmaßen.

Definition 4

Eine Fuzzy-Menge $\mu(x) \ x \in X$ heißt spezifischer als $\nu(x)$, d.h. $\mu(x) \ll \nu(x)$ falls gilt:

$$\mu(x) \neq \nu(x)$$

und

$$\mu(x) \leq \nu(x), \forall x \in X.$$

Analog zu scharfen Mengen können auch für unscharfe Mengen Operationen wie Komplementbildung, Vereinigung und Durchschnitt definiert werden. Die Operatoren zur Realisierung der Mengenoperationen sind Abbildungen von $[0,1] \times [0,1]$, also dem Bildbereich der Zugehörigkeitsfunktionen der verknüpften Mengen, auf das Intervall $[0,1]$ zur Charakterisierung des Zugehörigkeitsgrades zur Ergebnismenge.

Die Operatoren der Fuzzy-Mengentheorie werden über axiomatische Forderungen definiert. Dabei können die Mengenoperationen gemäß der Axiome durch verschiedene Funktionen realisiert werden. In der Literatur wird eine große Zahl von Operatoren, welche die Axiome erfüllen, genannt. Erstmals wurden von Zadeh die folgenden Funktionen vorgeschlagen:

$\min(a,b)$ *Durchschnitt (t-Coorm)*

$\max(a,b)$ *Vereinigung (t-Norm)*

$1-a$ *Komplement*

Für das Verständnis des Schließens mit Fuzzy-Mengen ist die Beziehung zur Logik von Bedeutung. Es ist bekannt, daß die gewöhnliche Mengentheorie äquivalent zur klassischen (zweiwertigen) Aussagenlogik ist, wenn die in Tab. 5.1 gezeigten Operatoren benutzt werden [KLIR89].

Mengentheorie	Aussagenlogik	Beispiel
\cup	\vee	max
\cap	\wedge	min
-	\neg	$1-\mu$
\subseteq	\rightarrow	I_G (S. 78)
X	1	
\emptyset	0	

Tab. 5.1 Zuordnung zwischen logischen Operatoren und Mengenoperationen.

Die Äquivalenz zur Logik kann auf die Fuzzy-Mengentheorie ausgedehnt werden, wenn für die Mengenoperatoren spezielle Funktionen eingesetzt werden. Damit wird die Fuzzy-Mengentheorie jedoch nicht zur zweiwertigen Logik, sondern zu einer Logik mit unendlich vielen, im Intervall $[0,1]$ liegenden Wahrheitswerten äquivalent¹. Andere Ersetzungen der logischen Operatoren, die gemäß der Axiomatik möglich sind, ergeben bei unscharfen Logiken unterschiedliche inhaltliche Bedeutungen. Durch geeignete Wahl der Operatoren ist es daher möglich menschliches plausibles Schließen in gewissem Grad nachzuvollziehen.

A.2 Grundlegende Vorgehensschritte beim Schließen mit Fuzzy Logik

Im vorliegenden Abschnitt wird der prinzipielle Ablauf des Schließens auf der Grundlage der Fuzzy-Logik erläutert. Der Ablauf der Inferenz geschieht in folgenden Schritten:

Im *ersten Schritt* müssen die für die Beschreibung wesentlichen Größen (z.B. Prozeßparameter $p_1, \dots, p_i, \dots, p_n$) und ihre Definitionsbereiche (X^1, X^2, \dots, X^n) benannt werden. Für die i -te Größe muß der Grundbereich $\text{Dom}(X^i)$ z.B. als Intervall reeller Zahlen festgelegt werden. Über dem Grundbereich werden anschließend unscharfe Werte definiert. Dazu wird für jede Größe p_i eine Menge linguistischer Bezeichner L^i eingeführt, z.B. $L^i = \{\text{gering, mittel, hoch}\}$. Für jedes Element l_j^i der Menge L^i wird dann eine Possibilitätsverteilung $\xi_j^i(x)$ zur detaillierten Beschreibung des unscharfen Wertebereiches angegeben.

Im *zweiten Schritt* wird die Abhängigkeit zwischen den durch die linguistischen Bezeichner benannten unscharfen Werten näher beschrieben. Die Abhängigkeit läßt sich mit Hilfe unscharfer Regeln der Form

$$\text{wenn } \xi' = \xi_j \quad \text{dann } \eta' = \eta_k$$

erfassen. Dabei wird mit Hilfe der Possibilitätsverteilungen ξ_j und η_k das allgemeine, aus Erfahrung bekannte Wissen über den Zusammenhang zwischen dem j -ten und k -ten Element der unscharfen Wertebereiche der Eingangs- und der Ausgangsgröße ausgedrückt. ξ' beschreibt eine, beispielsweise durch Beobachtung erhaltene Eingangsgröße, η' beschreibt die resultierende Verteilung für die Zielgröße.

Der Bedingungsteil der Regel, die *Prämisse*, kann auch aus mehreren Bedingungen bestehen, die verschiedene Eingangsgrößen betreffen. Die rechte Seite der Regel wird auch als *Konklusion* bezeichnet.

Der *dritte Schritt* ist die Inferenz. Hier wird aufgrund der vorgegebenen Eingangsfakten ξ' die am wenigsten spezifische² Possibilitätsverteilung η' bestimmt, die mit dem durch die Regeln ausgedrückten allgemeinen Wissen über die Zusammenhänge vereinbar ist. Auf diesen Schritt wird im folgenden Abschnitt ausführlich eingegangen.

1. Genauer wird die Fuzzy- Mengentheorie durch bestimmte Ersetzungen der Operatoren zu speziellen Lukasiewicz-Logiken isomorph.

2. Das Prinzip der minimalen Spezifität ist ein Grundprinzip der possibilistischen Theorie. Es ist vergleichbar mit dem Prinzip der maximalen Entropie in der Wahrscheinlichkeitstheorie und besagt, daß stets der unspezifischste, mit dem bekannten possibilistischen Wissen vereinbare Zustand, angenommen wird.

Im *vierten Schritt* muß das Ergebnis, welches in Form einer Possibilitätsverteilung vorliegt, noch weiter interpretiert werden, entweder durch die Umwandlung in einen scharfen Wert oder durch eine verbale Interpretation des Ergebnisses. Ersteres findet unter der Bezeichnung *Defuzzifizierung* vor allem bei regelungstechnischen Anwendungen statt, in denen ein scharfer Ausgangswert als Stellgröße benötigt wird. Bei wissensbasierten Systemen wird das Ergebnis im allgemeinen durch eine linguistische Approximation in einen verbalen Ausdruck umgewandelt, um einem menschlichen Benutzer in verständlicher Weise mitgeteilt zu werden. Dazu wird im vorliegenden System die Bezeichnung derjenigen Possibilitätsverteilung ermittelt, die zur resultierenden Verteilung der Inferenz die geringste Distanz aufweist¹.

A.3 Plausibles Schließen mit unscharfen Mengen

Interpretation unscharfer Regeln

Eine unscharfe Regel wird mit Hilfe einer bedingten Possibilitätsverteilung $\pi_{x|y}$ ausgedrückt. Diese beschreibt die Möglichkeit mit der ein scharfer Wert durch $\eta(y)$ beschrieben wird, unter der Voraussetzung, daß für den Eingangswert die Prämisse $\xi(x)$ zutrifft. Unscharfe Regeln lassen sich aufgrund der Vieldeutigkeit unscharfer Logik unterschiedlich interpretieren. Sie lassen sich gemäß ihrer Bedeutung in drei Klassen einteilen [DUBO92b].

Im Bereich der Regelungstechnik wird eine Regel häufig als unscharfe Relation auf dem karthesischen Produkt XY aufgefaßt. Der Grad, zu welchem die resultierende Verteilung zutrifft, wird in diesem Fall durch das Maß der Übereinstimmung der Eingangswerte mit den Regelprämissen beschrieben.

Die in der zweiten Klasse zusammengefaßten Regeln besitzen die gemeinsame Interpretation, daß bei einer besseren Übereinstimmung des Eingangsfaktes mit der Regelvoraussetzung ein spezifischeres Ergebnis ermittelt wird. Die Bedeutung einer solchen Regel kann daher durch „je mehr $x = A$ desto mehr $y = B$ “ beschrieben werden. Diese Regeln werden auch als *graduelle Regeln* bezeichnet [DUBO92a].

Durch Regeln, die der dritten Klasse angehören, lassen sich Unsicherheiten ausdrücken. Eine solche „certainty qualifying“ Regel kann folgendermaßen interpretiert werden: „je mehr $x = A$ desto größer die Möglichkeit bzw. Sicherheit bzw. Wahrscheinlichkeit ..., daß $y = B$ ist“.

Abb. A.1 zeigt unterschiedliche Interpretationen einer Regel mit scharfem Eingangswert x_i ($i=1,2$):

$$\text{wenn } x_i = \xi \text{ dann } \eta' = \eta.$$

Dabei wird die unterschiedliche Semantik von Regeln des zweiten und dritten Typs verdeutlicht. Die Verteilungen $\xi(x)$ und $\eta(y)$ charakterisieren die zum Ausdruck des Regelwissens notwendigen unscharfen Mengen. Die obere und untere Hälfte der Abbildung zeigen die Ergebnisverteilungen (fettgezeichnet) für unterschiedliche Eingangsfakten x_0 und x_1 . In der

1. Ein mögliches Maß für die Distanz zweier Possibilitätsverteilungen stellt der Schwerpunktsabstand der beiden Verteilungen dar.

oberen Hälfte ist die Übereinstimmung des Eingangsfaktes mit der Regelprämisse, ausgedrückt durch α , größer als in der unteren. Beim Vergleich fällt auf, daß eine geringere Übereinstimmung bei graduellen Regeln zur Abnahme der Spezifität des Ergebnisses führt. $\eta(y)$ in Abb. A.1.e) ist überall größer als oder gleich $\eta(y)$ in Abb. A.1.b) und daher weniger spezifisch (vgl. Def. 3). Bei „certainty qualifying“ Regeln erzeugt eine geringere Übereinstimmung dagegen eine Unsicherheit von $1 - \alpha$. Dies kann an den Verteilungen (c, f) gesehen werden, da sie eine Notwendigkeit von $1 - \alpha$ wiedergeben, den Ergebniswert außerhalb des Trägers von $\eta(y)$ vorzufinden (vgl. Def.2).

Die Beschreibung von Zusammenhängen zwischen Prozeßparametern muß es in erster Linie ermöglichen, aufgrund der angegebenen Regeln die Werte von Zielgrößen zu ermitteln. Die Gültigkeit der Regel und des Ergebnisses wird dabei als sicher vorausgesetzt. Dies kann in Analogie zu scharfen funktionalen Zusammenhängen gesehen werden, bei denen die zur Ermittlung eines Parameterwertes verwendeten Funktionen als sicher gültig vorausgesetzt werden.

„Certainty qualifying“ Regeln sind daher für die Modellierung von Zusammenhängen zwischen Prozeßparametern ungeeignet. Ihr Anwendungsgebiet liegt eher im Bereich von Entscheidungs- oder Diagnosesystemen, bei welchen unterschiedliche Alternativen bewertet und beurteilt werden müssen [BUIS86].

Die semantische Interpretation gradueller Regeln erfüllt dagegen besser die Anforderungen an eine Beschreibung von Parameterzusammenhängen. Gemäß ihrer Interpretation hängt die Unschärfe der Ergebnisse von der Schärfe der beobachteten Fakten und der Übereinstimmung mit der Regelprämisse ab. Zusätzlich kann das Ergebnis auch eine Unsicherheit als Maß dafür aufweisen, zu welchem Grad die beobachteten Fakten nicht mit den Voraussetzungen der Regeln übereinstimmen.

Regeln in Fuzzy Control Systemen, bei welchen $\pi_{x|y}$ zumeist durch einen Durchschnittsoperator anstelle einer Implikation beschrieben wird, unterscheiden sich nicht nur in ihrer semantischen Bedeutung, sondern gewöhnlich auch im Ablauf der Inferenz von den beiden übrigen Klassen.

Die insbesondere in der Regelungstechnik eingesetzte Inferenzmethode, bei welcher $\pi_{x|y}$ durch $\min(x, y)$ beschrieben wird, liefert Ergebnisse, die der intuitiven Auffassung eines plausiblen Schließens widersprechen (s. Kapitel 5). Für „certainty qualifying“ Regeln entspricht die semantische Interpretation der Regeln nicht der für die Beschreibung von Parameterzusammenhängen geforderten Bedeutung. Die semantische Interpretation gradueller Regeln erscheint für die Erweiterung der Beschreibung scharf charakterisierter Zusammenhänge zwischen Parametern auf den Fall unscharfer Informationen geeignet. Daher wird im folgenden näher auf die Inferenz mit diesem Regeltyp eingegangen.

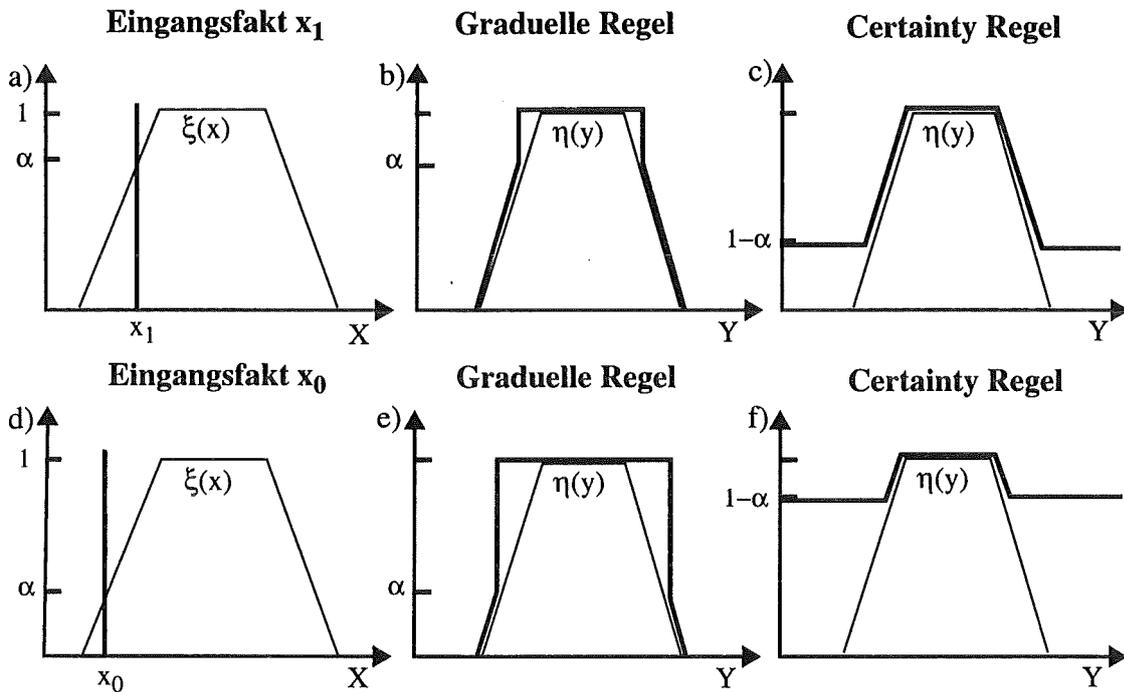


Abb. A.1 Ergebnisse von „certainty qualifying“ und graduellen Regeln bei unterschiedlichem scharfem Eingangsfakt x_1 (oben) und x_0 (unten).

Inferenz auf der Basis des verallgemeinerten Modus Ponens

Die Auswertung gradueller Regeln basiert auf einer Verallgemeinerung des aus der zweiwertigen Logik bekannten *Modus Ponens*. Dieses Schlußprinzip läßt sich folgendermaßen darstellen:

aus	A	
und	$A \rightarrow B$	
folgt	B	

Aus der Implikation $A \rightarrow B$ und der Gültigkeit der Prämisse A kann also auf die Wahrheit der Konklusion B geschlossen werden. Eine Verallgemeinerung dieses Prinzips für die unscharfe Logik wurde erstmals von Zadeh eingeführt [ZADE79].

In der Schreibweise der Logik kann der Modus Ponens auch durch

$$B = A \wedge (A \rightarrow B)$$

dargestellt werden.

Ersetzt man gemäß Tab. 5.1 den Operator \wedge durch den Durchschnittsoperator \min und die Implikation \rightarrow durch den Operator $I_G(\mu, \nu)$:¹

$$I_G(\mu, \nu) = \begin{cases} 1, & \text{für } \mu(a) \leq \nu(b) \\ \nu(b), & \text{sonst,} \end{cases}$$

so erhält man die Verallgemeinerung des Modus Ponens für einen scharfen Eingangswert. Durch Supremumbildung wird von allen Werten einer unscharfen Verteilung der zur Schlußfolgerung am meisten beitragende Wert auf den Grundbereich der Zielvariablen projiziert. Es gilt:

$$(1) \quad \eta'(a) = \sup_u \min(\xi'(u), I(\xi_j(u), \eta_i(a)))$$

Die resultierende Verteilung wird durch $\eta'(a)$ beschrieben, a stammt dabei aus dem Grundbereich der unscharfen Zielgröße. Der generelle Zusammenhang zwischen der Eingangsverteilung und der resultierenden Verteilung ist durch die Implikation $I(\xi, \eta) := I_G(\xi, \eta)$ festgelegt. $\xi'(u)$ bezeichnet die Verteilung zur Charakterisierung des Eingangswertes.

Die Beziehung (1) gilt für den Fall von Regeln mit einer einzigen Prämisse. Die Beziehung läßt sich auf den Fall von Regeln mit mehreren Prämissen verallgemeinern [DUBO91]:

$$(2a) \quad \text{wenn } \xi^{i^1} = \xi_j^1 \text{ und } \xi^{i^2} = \xi_k^2 \dots \text{ und } \xi^{i^m} = \xi_l^m \text{ dann } \eta' = \eta_i.$$

$$(2b) \quad \eta'(a) = \sup_{u, v, \dots} \min(\xi^{i^1}(u), \xi^{i^2}(v), \dots, I(\min(\xi_j^1(u), \xi_k^2(v), \dots), \eta_i(a)))$$

Die Implikation I wird nun auf das Minimum der Eingangsverteilungen angewendet, da bei mehreren gleichzeitig zu erfüllenden Prämissen der Wahrheitsgrad der Schlußfolgerung durch die am wenigsten zutreffende Bedingung begrenzt wird.

Um den Fall mehrerer paralleler, sich auf dieselben Ein- und Ausgangsgrößen beziehenden Regeln der Form (2a) zu beschreiben, müssen die Ergebnisse der Einzelregeln geeignet kombiniert werden. Hierzu wird in (2b) die Implikation I durch $I_{\text{ges}} = \min(I_i)$ ersetzt werden, wobei der Index i über alle in der Wissensbasis vorliegenden parallelen Regeln läuft.

1. Diese Festlegung erscheint plausibel, wenn man die zweiwertige Logik betrachtet. Hier ist die Implikation $a \rightarrow b$ ebenfalls nur dann 1, wenn $a = 0$ oder $b = 1$, d.h. $a \leq b$ gilt.

Anhang B

Zur Simulation von Parameterabhängigkeiten

Im folgenden wird am Beispiel der Belackung einer Zwischenmaske gezeigt, wie die Auswirkung verschiedener Einflußparameter auf einen Zielparameter mit Hilfe des Simulationsteils der LIMES-Software untersucht werden kann¹.

Tab. B.1 zeigt wesentliche Parameter, die die Schichtdicke bei der Belackung beeinflussen. Die Werte dieser Parameter sind von unterschiedlichem Typ. So wird die Geometrie durch einen linguistische Parameter beschrieben, welcher z.B. runde (MR) oder rechteckige Träger für die Maskenblanks unterscheidet. Der Untergrund für die Belackung wird ebenfalls durch eine linguistische Bezeichnung charakterisiert (im vorliegenden Fall besteht er aus oxidiertem Titan). Wichtige Eigenschaften des Resists sind der Feststoffanteil und die Molekulargewichtsverteilung. Durch die Frequenz mit welcher die Resistschicht aufgeschleudert wird (Spincoatfrequenz) kann überdies die Schichtdicke in weitem Rahmen beeinflußt werden.

Geometrie	Untergrund	Feststoffgehalt [%]	Molekulargewicht [kg]	Spincoatfrequenz [U/min]	Schichtdicke [μm]
MR	TiOx	9	950	1000	2,33

Tab. B.1 Einflußparameter auf die Schichtdicke bei einer Belackung.

Um die komplexen Auswirkungen von Änderungen der Werte von Parametern nachvollziehen zu können, wird ein beliebiger Parameter als variabel gewählt und der Zielparameter in Abhängigkeit von dessen möglichen Werten dargestellt (dies kann durch eine Tabelle oder auch in Form einer graphischen Darstellung geschehen). Tab B.2 zeigt Ergebnisse, bei welchen der Feststoffgehalt des Resists als variabler Parameter gewählt wurde.

Geometrie	Untergrund	Feststoffgehalt [%]	Molekulargewicht [kg]	Spincoatfrequenz [U/min]	Schichtdicke [μm]
MR	TiOx	5	950	1000	0,79
MR	TiOx	9	950	1000	2,33
MR	TiOx	10,5	950	1000	3,25
MR	TiOx	12	950	1000	4,2

1. Die hier genannten Zahlen und Zusammenhänge dienen lediglich zur Verdeutlichung des Prinzips. Die Werte der tatsächlichen Prozeßführung sind nicht zur Veröffentlichung bestimmt.

Tab. B.2 Schichtdicke bei Variation des Resist Feststoffanteils.

Eine Erhöhung des Feststoffgehalts und damit der Viskosität zieht demnach eine höhere Schichtdicke nach sich. Soll nun trotz eines erhöhten Feststoffgehalts (z.B. 10,5%) die Schichtdicke 2,3 μm erreicht werden, so ist dies durch eine Variation der Spincoatfrequenz möglich.

Hierzu kann die Schichtdicke für den veränderten Feststoffgehalt in Abhängigkeit der Spincoatfrequenz ermittelt werden. Aus Tab. B.3 läßt sich entnehmen, daß für eine Frequenz von 1500 U/min die gewünschte Schichtdicke von 2,3 μm erreicht wird.

Geometrie	Untergrund	Feststoffgehalt [%]	Molekulargewicht [kg]	Spincoatfrequenz [U/min]	Schichtdicke [μm]
MR	TiOx	10,5	950	600	5,17
MR	TiOx	10,5	950	1000	3,25
MR	TiOx	10,5	950	1400	2,50
MR	TiOx	10,5	950	1500	2,30
MR	TiOx	10,5	950	1800	2,07
MR	TiOx	10,5	950	2200	1,85
MR	TiOx	10,5	950	2600	1,72
MR	TiOx	10,5	950	3000	1,60
MR	TiOx	10,5	950	3400	1,61
MR	TiOx	10,5	950	3800	1,60
MR	TiOx	10,5	950	4200	1,59
MR	TiOx	10,5	950	4600	1,59

Tab. B.3 Variation der Schichtdicke mit der Spincoatfrequenz.

Der Darstellung der Vernetzung der Parameter kann entnommen werden, daß sich die Spincoatfrequenz auch auf die Homogenität der aufgeschleuderten Resistschicht auswirkt. Ein genauer Zusammenhang zwischen der Homogenität der Resistschicht und der Spincoatfrequenz ist nicht bekannt. Das hierüber vorliegende Wissen kann jedoch mittels unscharfer Zusammenhänge formuliert werden.

Wegen der Abhängigkeit zwischen Spincoatfrequenz und Homogenität kann die Schichtdicke durch Änderungen der Spincoatfrequenz nicht beliebig variiert werden. So läßt sich bei einer Frequenz von 600 U/min gemäß Tab. B.3 zwar eine Schichtdicke von 5,17 μm erreichen. Dieser Frequenz ist jedoch über die entsprechende Zugehörigkeitsfunktion (vgl. Kap. 5.1) die linguistische Bezeichnung „sehr niedrig“ zugeordnet. Nach Tab. B.4 wird dann die Homogenität als „schlecht“ eingestuft¹. Eine Prozeßdurchführung mit dieser Parameterkonstellation kann daher nicht ohne weiteres erfolgen.

Spincoatfrequenz	Geometrie	Feststoffgehalt [%]	Homogenität
"sehr niedrig"	MR	10,5	„schlecht“
"niedrig"	MR	10,5	„bedingt akzeptabel“
"normal"	MR	10,5	„gut“
"hoch"	MR	10,5	„bedingt akzeptabel“

Tab. B.4 Unscharfer Zusammenhang zwischen Spincoatfrequenz und Homogenität der Resistschicht.

1. Quantitativ läßt sich die Güte der Homogenität z.B. durch eine Dickenmessung an verschiedenen Positionen des Substrat und dem Vergleich der Abweichungen dieser Messungen erfassen.

Anhang C

Zur Software Struktur

LIMES basiert grundsätzlich auf einem graphisch interaktiven Dialog mit dem Benutzer, durch welchen die erforderlichen Systemfunktionen aktiviert werden. Zur Navigation durch den Prozeßablauf, das Parameternetz und für die Visualisierung von Beschreibungselementen (Prozeßschritte, Prozeßparameter, Arbeitsmaterialien, Ausrüstungsgegenstände) werden direkt Methoden der entsprechenden Objekte aufgerufen. Die Ablaufstruktur ist daher aus der Beschreibung der Benutzerschnittstelle und Oberfläche in Kapitel 7 ersichtlich.

Die Simulation von scharfen und unscharfen Parameterabhängigkeiten ist vom Ablauf her komplexer. Eine vollständige Dokumentation der hierzu implementierten Software, muß daher in einer objektorientierten Notation (z.B. Coad and Yourdon) durchgeführt werden. Allerdings ist eine solche Darstellung im vorliegenden Rahmen zu umfangreich. Für das grundsätzliche Verständnis wird im folgenden das Prinzip der komplexeren Abläufe der Software anhand der Blockdiagramme in Abb. C.1 und Abb. C.2 erläutert.

Prinzip der Simulation:

Start:

Zunächst wird die Beziehung vorgegeben werden, deren Zielparameter ermittelt werden soll. Diese Beziehung muß zu Beginn initialisiert sein. Dazu muß die *Bedingung*, d.h. die Voraussetzung unter welcher die Beziehung ausgewertet werden kann und der *Zusammenhang*, welcher dann ausgewertet wird, wenn die Beziehung zutrifft von einer Datei gelesen werden.

Eingangswerte Ermitteln:

In diesem Schritt werden die Eingangswerte der zur Beziehung gehörigen Parameter vom Benutzer, einer anderen Beziehung oder einem speziellen Softwarewerkzeug ermittelt.

Prüfe ob alle Eingangswerte vorliegen:

Es wird geprüft ob für jeden der Eingangsparameter der Beziehung ein Wert vom Benutzer vorgegeben wurde. Falls nicht wird zur Ermittlung dieses Parameterwertes die zugehörige Beziehung zunächst ausgewertet.

Prüfen ob Vorbedingungen erfüllt:

Für jeden Parameter der Beziehung wird geprüft, ob der Eingangswert im zulässigen Bereich liegt und ob eine der zur Beziehung gehörenden Bedingungen zutrifft.

Zusammenhang auswerten:

Der Zusammenhang wird mit Hilfe der Eingangswerte ausgewertet, d.h. der Wert des Zielparameters der Beziehung wird ermittelt. Sowohl die Werte der Eingangsparameter als auch der Wert des Zielparameters und die Zusammenhänge können durch unterschiedliche Wertetypen bzw. Klassen beschrieben werden (vgl. Kapitel 6.2).

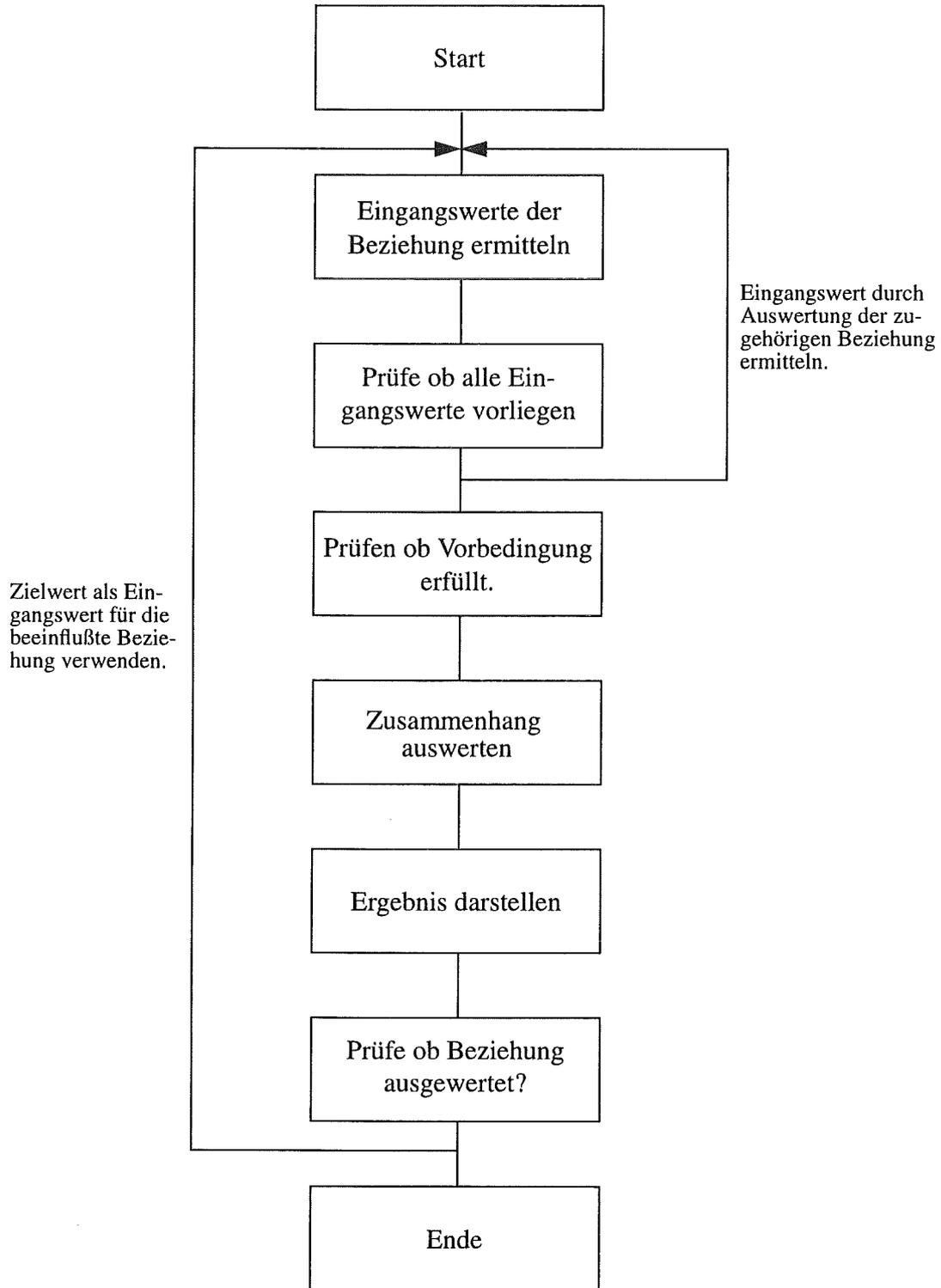


Abb. C.1 Blockstruktur für die Simulation scharfer Zusammenhänge.

Ergebnis Darstellen:

Der mit Hilfe des Zusammenhanges ermittelte Wert des Zielparameters der Beziehung wird zusammen mit den Eingangswerten dargestellt.

Prüfe ob Beziehung ausgewertet:

Es wird geprüft, ob die zu Beginn vorgegebene Beziehung bereits ausgewertet wurde.

Ende:

Zurück in die Warteschleife für neue Benutzeranforderungen.

Prinzip der Auswertung unscharfer Zusammenhänge:**Start:**

Zu Beginn werden die zu der Beziehung gehörigen Regeln, welche den unscharfen Zusammenhang zwischen den verknüpften Parametern beschreiben, eingelesen.

Werte der Eingangsparameter ermitteln:

Die Werte der Eingangsparameter können vom Benutzer, einer anderen Beziehung oder einem speziellen Softwarewerkzeug ermittelt, und als Eingang für die Simulation verwendet werden.

Zugehörigkeitsfunktion des Zielparameters ermitteln:

Dieser Schritt umfaßt die in Anhang A beschriebene Fuzzy-Inferenz auf der Grundlage des verallgemeinerten Modus Ponens. Dazu werden die scharfen oder unscharfen Eingangswerte für die vorgegebenen Regeln (die das generell vorhandene Wissen über die unscharfen Zusammenhänge ausdrücken) mit Hilfe von Gleichung 2b im Anhang ausgewertet. Die ermittelte Zugehörigkeitsfunktion beschreibt den unscharfen Zielparameterwert, welcher mit dem in den unscharfen Regeln abgelegten generellen Wissen vereinbar ist.

Ergebnis an Benutzer weitergeben:

Es wird unterschieden, ob das Ergebnis dem Benutzer weitergegeben, oder intern weiterverarbeitet werden soll. In letzterem Fall wird die Zugehörigkeitsfunktion des Zielparameters direkt weiterverwendet.

Zugehörigkeitsfunktion in linguistische Werte umwandeln:

Für den Benutzer ist es oft ausreichend und übersichtlicher, außer der Zugehörigkeitsfunktion auch einen linguistischen Wert für den Zielparameter zu erhalten. Dazu wird der linguistische Bezeichner gewählt, dessen Zugehörigkeitsfunktion eine möglichst geringe Distanz zur ermittelten Zugehörigkeitsfunktion des Zielparameters aufweist. Die Distanz von Zugehörigkeitsfunktionen kann auf verschiedene Weise bestimmt werden. Eine einfache Wahl ist der Abstand des Schwerpunktes beider Funktionen.

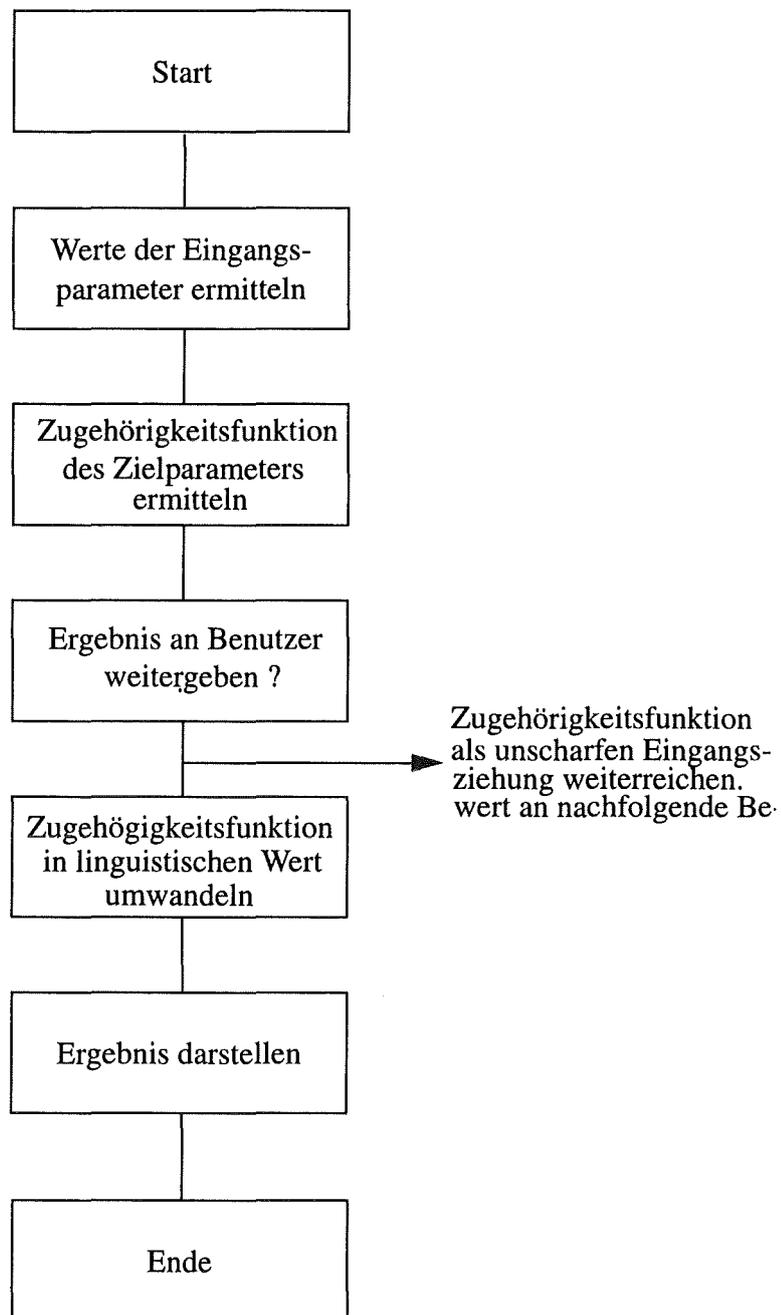


Abb. C.2 Blockstruktur der Auswertung unscharfer Zusammenhänge

Ergebnis darstellen:

Die Zugehörigkeitsfunktion des Zielparameters wird dem Benutzer zusammen mit der ermittelten linguistischen Bezeichnung graphisch präsentiert.

Ende:

Zurück in die Warteschleife für neue Benutzeranforderungen.

Literatur

- ATKI90 M. Atkinson, D. Witt et al.: The Object-Oriented Database System Manifesto; North Holland, 1990
- BIER88 W. Bier, K. Schubert: Herstellung von Mikrostrukturen mit großem Aspektverhältnis durch Präzisionszerspanung mit Formdiamanten; KfK-Bericht 4363, 1988
- BOCK84 S.F. Bocklisch: Prozeßanalyse mit unscharfen Verfahren; VEB Verlag Technik Berlin, 1984
- BOOC91 G. Booch: Object Oriented Design with Applications; Benjamin/Cummings Pub. Comp. California, 1991
- BRAU93 I. Brauch, H. Eggert, K.P. Scherer, P. Stiller: Unterstützung des LIGA- Herstellungsprozesses mit wissensbasierten Methoden; ZWF CIM; Carl Hanser Verlag München pp.38, 1991
- BUCH85 B.G. Buchanan, E.H. Shortliffe: Rule- Based Expert Systems; Adison Wesley Pub. Company pp.248, 1985
- BUIS86 J.C. Buissonne, H. Farreny: Dealing with Imprecision and Uncertainty in the Expert System DIABETO-III; pp.705-721, 1986
- COAD91 P. Coad, E. Yourdon: Object Oriented Analysis; Yourdon Press Englewood Cliffs, 1991
- CODD70 E.F. Codd: A Relational Model of Data for Large Shared Data Banks; Communications of the ACM, 13 (6): pp.377-387, 1970
- DAHL66 O.J. Dahl, K. Nygaard: SIMULA an ALGOL based Simulation Language; CACM 9, p.671-678, 1966
- DUBO85 D. Dubois, H. Prade: The Generalised Modus Ponens under sup-min Composition; Approximate reasoning in Expert Systems, Elsevier Science Publishers B.V. pp.217-232, 1985
- DUBO88 D. Dubois, H. Prade: Practical Computing in Fuzzy Logik; Fuzzy Computing, Elsevier Science Publishers B.V. pp.11-34, 1988
- DUBO90 D. Dubois, H. Prade: Fuzzy Rules in Knowledge-Based Systems; Kluwer Academic Pub. Dordrecht in the Netherlands, 1990
- DUBO91 D. Dubois, H. Prade: Fuzzy Sets in Approximate Reasoning, Part 1: Inference with possibility distributions; Fuzzy Sets and Systems 40 pp.143-202, 1991
- DUBO92a D. Dubois, H. Prade: Gradual Inference Rules in Approximate Reasoning; Information Sciences 61 pp.103-122, 1992
- DUBO92b D. Dubois, H. Prade: Possibility theory as a Basis for Preference Propagation in Automated Reasoning; IEEE pp.821-832, 1992

- ESHR79 F.Eshragh, H.Mamdani: A General Approach to Linguistic Approximation ; Int.J.Man-Machine Studies 11 501-519, 1979
- HEIN92 H.Hein, P.Bley, J.Göttert, U.Klein: Elektronenstrahlolithographie zur Herstellung von Röntgenmasken für das LIGA-Verfahren; Carl Hanser Verlag, München, 1992
- HEUE92 A.Heuer: Objektorientierte Datenbanken; Konzepte, Modelle, Systeme; Addison-Wesley Publishing Company, 1992
- JUNG92 L.Junghanns, S.Weinmeister: Modellierung des Wissens für die Simulation und Analyse von Fertigungsverfahren der Mikrosystemtechnik; Abteilung-sinterner Bericht in Zusammenarbeit mit der TU Dresden, 1992
- KRUS93 R.Kruse, J.Gebhardt, F.Klawonn: Fuzzy-Systeme; B.G.Teubner Stuttgart, 1993
- KAUF85 A.Kaufmann, M.M.Gupta: Introduction to Fuzzy Arithmetic: Theory and Applications; Van Nostrand Reinhold, New York, 1985
- KEMK91 C.Kemke: Darstellung von ungenauem Wissen in taxonomischen Wissensbasen; KI Fachbeiträge 2 pp.12-19, 1991
- KLIR89 G.J.Klir, T.A. Folger: Fuzzy Sets Uncertainty and Information; Prentice Hall International, Inc.
- LESS92 C.Leßmöllmann: Fertigungsgerechte Gestaltung von Mikrostrukturen für die LIGA-Technik; Dissertation Karlsruhe, 1992
- LOCK87 P.C.Lockemann, J.W.Schmidt: Datenbank-Handbuch; Springer-Verlag Berlin, Heidelberg, New York pp.527, 1987
- LI91 X.LI: What's so bad about Rule-Based Programming?; IEEE Software 10, 1991
- MANE88 A.Maner, W.Ehrfeld, R.Schwarz: Galvaniformung von Absorbermasken aus Gold auf Masken für die Röntgenlithographie, 1988
- MART89 R.Martin-Clouaire: Semantics and Computation of the Generalized Modus Ponens: The Long Paper; International Journal of Approximate Reasoning, Elsevier Science Publishers B.V. pp.195-217, 1989
- MENZ92 W.Menz, P.Bley: Mikrosystemtechnik für Ingenieure; VCH Verlagsgesellschaft Weinheim, 1992
- MENZ93 W.Menz: Die LIGA-Technik und ihr Potential für die industrielle Anwendung ; 1.Statuskolloquium des Projektes Mikrosystemtechnik, KfK- Bericht 5238, 1993
- MIZU82 M.Mizumoto, H.-J. Zimmerman: Comparison of Fuzzy Reasoning Methods; North Holland pub. Company, 1982
- MÜNCH87 D.Münchmeyer, W.Ehrfeld, A.Maner, W.Schelb: Aufbau von Fertigungseinrichtungen für die Röntgentiefenlithographie; KfK-Nachr. Jahrg. 19 Vol.4 pp.180-191, 1987

- NEXP91 NEXPERT-*Object* Functional Description, Neuron Data Inc., 1991
- RATH87 C.Rathke: Framebasierte Wissensrepräsentation mit Hilfe Objektorientierter Programmierung; German Chapter of the ACM pp.64-85, 1987
- RETT84 J. Retti et al.: Artificial Inteligence; Teubner Verlag Stuttgart, 1984
- SCHO90 W. Schomburg, P.Bley, H.Hein, J.Mohr: Masken für die Röntgentiefenlithographie; VDI Berichte Nr.870, 1990
- SCHU93 J.Schulz, J.Mohr: Optische Lithographie als Strukturierungsmethode der Mikrotechnik und spezielle Anwendungen für den LIGA- Prozeß; 1.Statuskolloquium des Projektes Mikrosystemtechnik am Kernforschungszentrum Karlsruhe, 1993
- SENT92 S.D.Senturia et.al.: A Computer- Aided Design System for Microelectromechanical Systems (MEMCAD); Journal of Microelectromechanical Systems, vol 1, no. 1 March, 1992
- SOWA87 J.Sowa: Semantic Networks: Encyclopedia of Artificial Intelligence; John Wiley & Sons, Inc. Vol.2, 1987
- STEC87 G.Stecher: Proceedings 6th European Microelectronics Conf. pp.421 Bournemouth, England, 1987
- STEV46 S. Stevens: On the Thoery of Scales of Measurement; Science 103 p.677-680, 1946
- STRO92 B. Stroustrup: Die C++ Programmiersprache; Addison-Wesley (Deutschland), 1992
- TILLI 93 Tilli: Fuzzy Logik Grundlagen und Anwendungen; Franzis' Verlag 3. Auflage, 1993
- ZADE65 L.A.Zadeh: Fuzzy Sets; Information and Control 8, p.338-353, 1965
- ZADE79 L.A.Zadeh: A theory of approximate reasoning; Machine Intelligence, Vol.9 (E.Hayes et al., eds), Elsevier, New York, pp 149-194, 1979
- ZIMM93 H.J.Zimmermann: Fuzzy Technologien Prinzipien Werkzeuge Potentiale; VDI-Verlag GmbH Düsseldorf, 1993
- ZUMG93 K.-H.Zum Gahr: Materialforschung für Mikrosysteme; 1.Statuskolloquium des Projektes Mikrosystemtechnik, KfK- Bericht 5238, 1993

Danksagung

Die vorliegende Arbeit wurde in den Jahren 1991 bis 1994 am Institut für Angewandte Informatik des Forschungszentrums Karlsruhe erstellt.

Herzlich danken möchte ich Herrn Prof. Dr. W. Menz für die Übernahme des Referats. Ebenfalls danke ich herzlich Herrn Prof. Dr. H. Trauboth für die Übernahme des Koreferats.

Weiter möchte ich Herrn Dr. H. Eggert für die engagierte Unterstützung der Arbeit am Institut danken. Ebenfalls Dank an die Mitarbeiter der Abteilung Mikrosysteminformatik am IAI, besonders an meine Kollegen Herrn Dr. K.P. Scherer und Herrn P. Stiller.

Für die gute Zusammenarbeit danke ich den Mitarbeitern des IMT, die durch ihre fachlichen Beiträge wesentliche Impulse für die Arbeit gegeben haben. Insbesondere danke ich Herrn Dr. H. Hein für seine freundliche Unterstützung.