

Design Flow Management mit Web Services in der Mikrotechnik

Vom Fachbereich Elektrotechnik und Informatik der
Universität Siegen

zur Erlangung des akademischen Grades

Doktor der Naturwissenschaften
(Dr. rer. nat.)

genehmigte Dissertation

von

Diplom Informatiker Christian Schumer

1. Gutachter: Prof. Dr. Rainer Brück
 2. Gutachter: Prof. Dr. Hans Wojtkowiak
- Vorsitzender: Prof. Dr. Udo Kelter

Tag der mündlichen Prüfung: 11.10.2004

Endgültige Fassung

urn:nbn:de:hbz:467-862

Abstract

MEMS industry in Europe is characterized by small and medium sized enterprises (SMEs) specialized on products to solve problems in specific domains like medicine, automotive sensor technology, etc. In this field of business the technology driven design approach known from micro electronics is not appropriate. Instead each design problem aims at its own, specific technology to be used for the solution. The variety of technologies at hand, like Si-surface, Si-bulk, LIGA, laser, precision engineering requires a huge set of different design tools to be available. No single SME can afford to hold licenses for all these tools. This calls for a new and flexible way of designing, implementing and distributing design software.

The key to the development of successful tools for micro electronic design has been the use of models of the design process that define precisely the steps to be taken, their constraints, their sequence and their parameters in order to turn a product specification into a technical realization. The famous Y-chart [WT85] is a good example for the extent and impact of design models. Driven by the success of the Y-chart model a dedicated design model for MEMS design is presented in this thesis. As a basis, a so-called circle-model is used, which describes a new design methodology for MEMS. This includes a class of design services to solve specific MEMS design problems and an innovative approach for developing design flows.

Based on the requirements of SMEs a novel service oriented design scenario (SODS) is presented. It contains a workflow for implementing customer specific design flows and defines the required software elements. In this context the Internet as one of today's most important communication media provides support on the basis of web services. The technology of web services is used to offer a new generation of flexible, semi-autonomous software systems via the Internet, so-called design assistants. A design assistant guides the human expert through the design flow and makes sure that the fabrication specification is complete and correct. Assistants are built dynamically depending on the requirements

of the respective MEMS design process at hand. To achieve this, a design flow integrated development environment (IDE) helps to build up assistants fast and easily.

Circle-model, design services and SODS are part of the new discipline design flow management for MEMS. The discipline shows SMEs how to reduce costs for design tool installation, operation and integration and produces a faster design flow execution with a better result.

Danksagung

Meinem Doktorvater, Herrn Prof. Dr. Rainer Brück, möchte ich für die vielfältige Unterstützung und die Förderung dieser Arbeit danken.

Herrn Prof. Dr. Wojtkowiak danke ich für das meiner Arbeit entgegengebrachte Interesse und die Übernahme des Korreferats.

Den Mitarbeitern des Instituts für Mikrosystemtechnik danke ich für die gute Zusammenarbeit und die stets vorhandene Hilfsbereitschaft.

Außerordentlich dankbar für geduldiges Korrekturlesen bin ich meinen Schulfreunden Thomas Viefhaus und Andreas Weng.

Ganz besonders bedanke ich mich bei meinen Eltern Maida und Josef Schumer, die durch ihre fortlaufende Unterstützung und Anteilnahme die Anfertigung dieser Arbeit überhaupt erst ermöglicht haben. Sie haben mir die Energie und das Selbstvertrauen gegeben, diese Arbeit zu Ende zu führen.

Last but not least danke ich meiner Frau Christiane, die es verstanden hat und versteht, die Launen eines nur allzuoft an die Arbeit denkenden Ehemanns zu ertragen und meiner lieben Tochter June verbunden mit der Entschuldigung dafür, dass sie ihren Papa oft nur sporadisch zu Gesicht bekam.

Inhaltsverzeichnis

1	Einleitung und Motivation	1
2	Grundlagen der Mikrotechnik	7
2.1	Begriffsbestimmung	7
2.2	Silizium-Mikromechanik	10
3	Entwurfsmodell	13
3.1	Begriffsbestimmung	13
3.2	Ein Modell für den Entwurf	15
3.3	Entwurfsaktivitäten	21
3.4	Konvergente und Divergente Entwürfe	23
3.5	Das Kreismodell	26
3.6	Das Y-Modell der Mikroelektronik	29
3.7	Erstellen einer Spezifikation	32
3.7.1	Informationsgewinnung	33
3.7.2	Verhaltensspezifikation	34
3.7.3	Strukturspezifikation	37
3.7.4	Physikalische Spezifikation	40
3.8	Modellbildung in der Mikrotechnik	44

4	Design Flow in der Mikrotechnik	51
4.1	Kreismodell für den physikalischen Entwurf	51
4.2	UML-Aktivitätsdiagramm	54
4.3	Entwurfsmethodik	61
4.4	Einsatzgebiet des Kreismodells	65
4.5	Elemente beim Entwurf von Mikrosystemen	69
5	Entwurfsszenarien	73
5.1	Akteure	73
5.2	Mikroelektronik-Großindustrie	75
5.3	Mikroelektronik-Design House	76
5.4	Special Design Service	77
5.5	Bewertung der Entwurfsszenarien	78
5.6	Ein neues Entwurfsszenario	80
6	Design Services	85
6.1	Anforderungen	85
6.2	Eine flexible Infrastruktur	86
6.3	Manuelle Design Services	87
6.4	Semi-automatische Design Services	87
6.4.1	Software-Architekturen	88
6.4.2	Software-Techniken	92
6.4.3	Anwendungsbeispiele	98
6.5	Voll-automatische Design Services	101
6.5.1	Web Services	101
6.5.2	Anwendungsbeispiele	105
6.6	Ein Medium für den zentralen Zugriff	107
6.7	Design Flow Entwicklung und Durchführung	109
6.7.1	Design Flow Entwicklungsumgebung	109

6.7.2	Design Flow Assistent	116
6.8	Service-orientiertes Entwurfsszenario	119
6.8.1	Realisierungskonzept	119
6.9	Design Flow Management	124
6.9.1	Daten- und Applikationsintegration	124
7	Zusammenfassung und Ausblick	131
	Abbildungsverzeichnis	134
	Literaturverzeichnis	138

Kapitel 1

Einleitung und Motivation

Das Ergebnis der beispiellosen technologischen Entwicklung der Elektronik zur Mikroelektronik in den letzten drei Jahrzehnten spiegelt sich in der Realisierung eines mikroelektronischen Systems als integrierter Schaltkreis wider. Ein integrierter Schaltkreis besitzt gegenüber diskreten elektronischen Schaltungen einige signifikante Vorteile, die zu seinem massiven Einsatz im industriellen und privaten Bereich geführt haben. Hier sind die geringen Herstellkosten, der minimale Platzbedarf, der niedrige Energieverbrauch, die sehr hohe Operationsgeschwindigkeit und die hohe Langzeitstabilität zu nennen.

Die bemerkenswerten Eigenschaften von mikroelektronischen Schaltungen führen zu der Frage, ob es nicht auch möglich ist, in anderen Technologiebereichen wie zum Beispiel der Sensorik, der Aktorik, der Feinwerktechnik oder im Maschinenbau den Schritt von der Makro- zur Mikrostrukturtechnik zu vollziehen. Dieser Schritt würde es erlauben, Strukturen wie beispielsweise Beschleunigungs-, Druck-, Strahlungssensoren, Schalter, Ventile und Motoren im Mikrometerbereich zu erstellen.

Die Herstellung von Mikrostrukturen als einzelne autonome Bauteile in dem Sinne, dass man konventionelle Bauteile durch adäquate Mikrostrukturen ersetzt, ist technisch und wissenschaftlich gesehen mit Sicherheit sehr reizvoll. Sie würde aber das Potenzial der Mikrostrukturtechnik nicht voll ausschöpfen und keine revolutionäre Entwicklung ähnlich, wie in der Mikroelektronik, einleiten. „Ziehen wir als Parallele wieder die Mikroelektronik heran, so ist es wohl einsichtig, dass die Mikroelektronik nicht so erfolgreich wäre, hätte man sich nur darauf konzentriert, möglichst viele einzelne Transistoren möglichst klein und billig herzustellen. Erst die Verknüpfung vieler Bauelemente zu Baugruppen und diese wieder zu Systemen hat die Fähigkeit der Mikroelektronik ins Vielfache gesteigert und ihren beispiellosen Erfolg begründet“ [Men97].

Der gleiche Weg ist in der Mikrostrukturtechnik zu gehen. Das Ziel sollte es sein durch Verknüpfung einer Menge von Mikrostrukturen einen synergetischen Effekt hervorzurufen und, damit verbunden, eine entsprechend hohe Leistungsfähigkeit des entstehenden Mikrosystems zu erreichen. Der Einsatz solcher Mikrosysteme eröffnet eine Vielzahl von neuen Anwendungsgebieten beispielsweise in der Mess-, der Kommunikations-, der Automobil- und der Medizintechnik.

Mikrosysteme könnten die Medizintechnik revolutionieren, wie aktuelle Forschungsergebnisse aus dem Labor der RWTH Aachen beweisen [SSR03]. Eine Minisonde wurde entwickelt, die Daten aus der Blutbahn eines Menschen per Funk senden kann. Sie besteht aus einem Mikrochip sowie einer winzigen Antenne und ist eingehüllt in Silikon (siehe Abbildung 1.1).

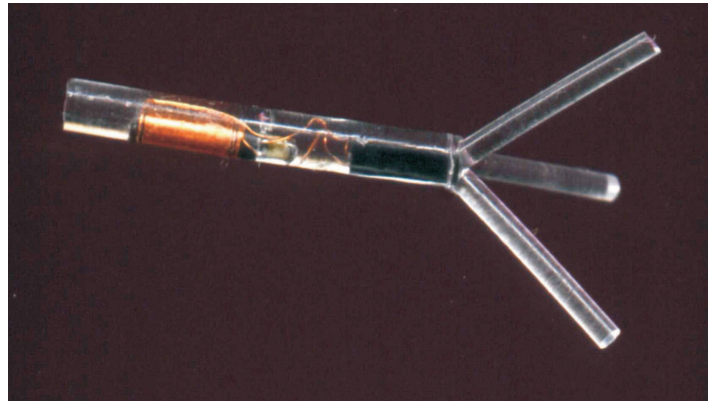


Abbildung 1.1: Minisonde für die drahtlose Überwachung von Blutwerten. Die Silikonhülle enthält Antennenspule (links) und Mikrochip. Der Durchmesser beträgt ohne Haltebeinchen 2.3 Millimeter [SSR03]

Die nur zwei Millimeter dicke Sonde wird durch einen minimalen Eingriff gezielt in einem Blutgefäß positioniert. Von dort überträgt sie drahtlos Blutdruck- und Temperaturwerte an eine am Körper getragene Lesestation. In der Minisonde ist keine Batterie eingebaut. Die Energieversorgung erfolgt mittels einer winzigen Antenne, die ihre Energie aus den elektromagnetischen Wellen des Lesegerätes schöpft. Zehnmahl pro Sekunde werden die aktuellen Temperatur- und Blutdruckwerte an die Lesestation übermittelt. Der Arzt transferiert später die gespeicherten Daten aus dem Lesegerät auf seinen Rechner zur Auswertung.

Die Sonde wird durch einen Katheterschlauch bis zu einer hautnahen Arterienverzweigung im Gesäß geführt und dann aus dem Schlauch heraus geschoben. In diesem Moment klappen drei elastische Haltebeinchen nach außen. Sie stellen

sicher, dass die Sonde an der Verzweigungsstelle hängen bleibt. Die Gefäßwand wird nur durch die Haltebeinchen berührt. Es entsteht keine Verletzung und der Blutstrom wird nicht behindert. Die Speicherung von Kalibrierungsdaten und die Signalaufbereitung für die drahtlose Datenübertragung erfolgt durch eine mikroelektronische Komponente. Die Messungen übernimmt eine nicht-elektronische Komponente in Form eines ebenfalls auf Siliziumbasis gefertigten mikromechanischen Drucksensors (siehe Abbildung 1.2).

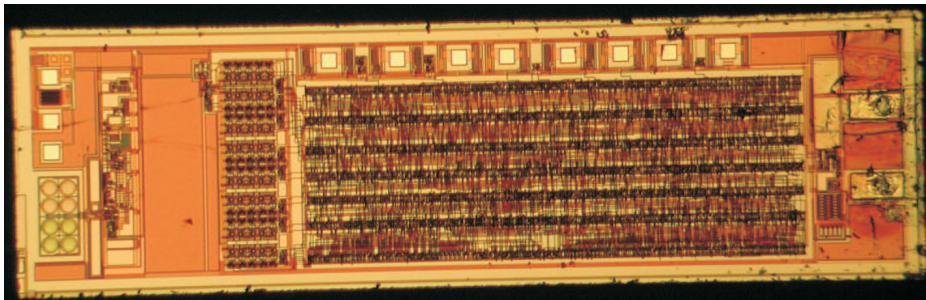


Abbildung 1.2: Mikroelektronische Komponente und Drucksensor auf einem Siliziumsubstrat. Größe 5,7 mal 1,5 Millimeter [SSR03]

Neben solchen sehr speziellen Mikrosystemen mit relativ geringen Stückzahlen gibt es Massen Anwendungen zum Beispiel im Bereich der Automobiltechnik. Um diese Märkte zu befriedigen, bedarf es eines hohen Automatisierungsgrades von Entwurf und Fertigung, um schnell Produkte mit hoher Qualität zu attraktiven Preisen anbieten zu können.

In den letzten Jahren sind Forschungsgelder primär in die Förderung fertigungsnaher Themen geflossen. Die Verfahren der Mikrostrukturtechnik wurden deutlich verbessert und besitzen besonders im Bereich der Silizium-Mikromechanik heute einen hohen Reifegrad. Zusätzlich sind zahlreiche Investitionen getätigt worden um neue Fabriken, wie die von Infineon und AMD in Dresden, zu bauen. Im Gegensatz dazu sind die Fördermittel im Bereich Entwurf ab Mitte der neunziger Jahre quasi auf Null zurückgegangen, was zu einem Stillstand bei der Entwicklung von innovativen Werkzeugen zur Entwurfsautomatisierung geführt hat. So ist zu erklären, dass der Produktivitätszuwachs im Entwurf von Jahr zu Jahr immer weiter hinter dem in der Fertigung zurückbleibt.

Ein Beispiel aus der Mikroelektronik soll verdeutlichen, welche Probleme hierdurch in naher Zukunft entstehen könnten: Die Fertigungskapazität im mikroelektronischen Bereich hat sich in den letzten Jahrzehnten pro Jahr um 60 Prozent erhöht. Die Produktivität beim Chip-Entwurf hat dagegen nur um ca. 20 Prozent pro Jahr zugenommen. Im Jahr 2010 geht die Semiconductor In-

dustry Association von einer ASIC-Schaltungskomplexität von etwa fünf Milliarden Transistoren aus. Zur Darstellung werden rund 100 Milliarden Layout Rechtecke und rund 2000 Gigabyte binärer Speicherplatz benötigt. Unter der Annahme, dass diese Datenmenge effizient zu verwalten wäre und bei einer angenommenen Designproduktivitätssteigerung von 20 Prozent pro Jahr sind für das Design eines solchen Chips drei Milliarden Euro als Budget zu verplanen. Dies gilt nur, wenn der Chipentwurf unter der marktgerechten Annahme in einem Jahr durchzuführen ist und die dafür benötigten 25.000 Entwickler zur Verfügung stehen und zu koordinieren wären [SM03].

In einer ähnlich schwierigen Situation befindet sich die Mikrostrukturtechnik. Sie ist geprägt durch anwendungsspezifische Entwürfe, die individuelle, speziell auf das Verhalten des zukünftigen Mikrosystems abgestimmte Fertigungsprozesse als Resultat produzieren. Eine Entwurfsautomatisierung ist im Augenblick nur sehr rudimentär durch IT gestützte Entwurfswerkzeuge realisiert. Die meisten Werkzeuge sind einst für die Entwurfsunterstützung in der Mikroelektronik konzipiert worden und wurden später so weit wie möglich an die Bedürfnisse des Mikrotechnik-Entwurfs angepasst. Aktuelle Mikrotechnik-Werkzeuge unterstützen entweder einzelne wenige Entwurfsprozesse oder decken nur einen Teilaspekt des Entwurfs ab. Ist der Wunsch da, eine optimale Unterstützung für den Entwurfsprozess zu erreichen muss eine sehr kosten- und zeitintensive Integration von verschiedenen Werkzeugen erfolgen. Schätzungen deuten daraufhin, dass die doppelte Summe wie für Lizenzen noch einmal für Integration und den Support der Werkzeuge aufzubringen ist [SM03]. Die von vielen **kleinen und mittelständischen Unternehmen (KMU)** sowie Forschungsinstituten geprägte Mikrotechnik-Industrie kann sich in vielen Fällen aus Kostengründen weder die Lizenzen für die Werkzeuge noch die Integrationsaufwände leisten. Erschwerend kommt hinzu, dass auch bei Nutzung der vorhandenen Werkzeuge und unbeschränktem Integrationsbudget, der Grad der Entwurfsautomatisierung nicht den Anforderungen entspricht, die von Anwenderseite bestehen.

Wie sollen die KMU in der Lage sein den derzeitig punktuell vorhandenen Forschungsvorsprung in Deutschland Gewinn bringend zu nutzen, wenn Sie keine adäquate Entwurfsunterstützung erhalten? Das Ziel der vorliegenden Arbeit ist es, einen Weg aus diesem Dilemma aufzuzeigen und ein konkretes Konzept zu erarbeiten, mit dem die immer größer werdende Lücke zwischen Entwurfs- und Fertigungsproduktivitätszuwachs zu schließen ist.

Im folgenden Kapitel erfolgt zunächst eine Bestimmung und Abgrenzung von Begriffen der Mikrotechnik. Daran schließt sich ein Überblick über Technologien zur Herstellung von mikromechanischen Bauteilen an.

Kapitel 3 beschreibt ein neues Entwurfsmodell für die Mikrotechnik. Dabei wird der Prozess des Entwurfs analysiert und in drei Phasen aufgeteilt: Verhaltensentwurf, Strukturentwurf und physikalischer Entwurf.

Kapitel 4 fokussiert sich auf den physikalischen Entwurf, der einen Bauplan für ein Mikrosystem spezifiziert. Eine Methodik wird vorgestellt, um systematisch einen anwendungsspezifischen fertigungsnahen Entwurfsprozess zu ermitteln.

Kapitel 5 beleuchtet Akteure, die beim Mikroentwurf involviert sind und analysiert bestehende Entwurfsszenarien. Anschließend wird ein neues service-orientiertes Entwurfsszenario zur verbesserten automatisierten Entwurfsdurchführung präsentiert.

Kapitel 6 stellt das Realisierungskonzept für das neue Entwurfsszenario vor, mit denen KMU zukünftig weitgehend automatisierte Entwürfe zu deutlich niedrigeren Kosten durchführen können.

Die Arbeit endet mit einer Zusammenfassung und einem Ausblick in Kapitel 7.

Kapitel 2

Grundlagen der Mikrotechnik

Zu Beginn werden die drei zentralen Begriffe beim Entwurf und der Fertigung von miniaturisierten technischen Systemen eingeführt. Danach werden anhand von Beispielen Einsatzmöglichkeiten für diese Systeme präsentiert. Das Kapitel endet mit einer Beschreibung der dafür am Markt verfügbaren Fertigungsverfahren.

2.1 Begriffsbestimmung

In der Forschungsgemeinde existieren für die Begriffe Mikrosystemtechnik, Mikrotechnik und Mikrosystem keine allgemein akzeptierten Definitionen. Es besteht daher die Notwendigkeit, zumindest für das bessere Verständnis der Arbeit, an dieser Stelle Abhilfe zu schaffen und eine Begriffsbestimmung vorzunehmen:

Definition: 1 *Unter dem Begriff Mikrosystemtechnik (MST) versteht man den Entwurf, die Fertigung und die Applikation von miniaturisierten technischen Systemen, deren Elemente und Komponenten typische Strukturgrößen im Mikrometer- und Nanometerbereich aufweisen. Die Mikrosystemtechnik behandelt dabei insbesondere die aufeinander abgestimmte Integration von unterschiedlichen funktionalen Elementen aus verschiedenen mikrotechnischen Bereichen [Brü96].* //^A

Die vorliegende Arbeit beschäftigt sich primär mit Entwurfsmodellen und Werkzeugen, die den Prozess der Herstellung von miniaturisierten technischen Systeme-

¹Das Zeichen // kennzeichnet das Ende einer Definition.

men unterstützen. In diesem Zusammenhang ist der Begriff Mikrotechnik von entscheidender Bedeutung.

Definition: 2 *Unter dem Begriff **Mikrotechnik** oder **Mikrostrukturtechnik** werden zusammenfassend alle Entwurfs- und Fertigungsaufgaben bezeichnet, die zur Herstellung miniaturisierter Strukturen aus verschiedenen Domänen verwendet werden. Die Mikrotechnik umfasst einerseits die nach Wirkprinzipien unterschiedenen Techniken wie z.B. die Mikroelektronik, die Mikromechanik und die Mikrooptik, andererseits die nach Technologien unterschiedenen Fertigungsverfahren wie z.B. die Siliziumtechnik, die LIGA-Technik oder die Feinwerktechnik [Hah99].* //

Die Mikrotechnik repräsentiert einen wichtigen Teilbereich der Mikrosystemtechnik. Sie konzentriert sich auf die Erstellung miniaturisierter Strukturen und verwendet dafür Fertigungsverfahren aus verschiedenen Technologiebereichen. Der Schwerpunkt bei der Mikrosystemtechnik liegt dagegen bei den Aspekten der Integration von miniaturisierten Strukturen zu komplexen Systemen, so genannten Mikrosystemen.

Die nachfolgende Definition beschreibt ein Mikrosystem aus technischer und funktionaler Sicht:

Definition: 3 *Ein **Mikrosystem** besteht aus Bauteilen oder Strukturen, die mit Hilfe mehrerer unterschiedlicher Mikrotechniken hergestellt werden. Funktional bestehen Mikrosysteme aus Schnittstellen zur realen Welt in denen über Sensoren bzw. Aktoren physikalische, chemische oder biologische Größen aufgenommen bzw. an die Umgebung abgegeben werden. Eine elektronische Signal- bzw. Informationsverarbeitung steuert die Ein- und Ausgabe des Systems [Hah99]. Ein Mikrosystem, das nur aus mikroelektronischen Bauteilen und mikromechanischen Strukturen besteht heißt **mikroelektromechanisches System (MEMS)**².* //

Ein Mikrosystem kann als ein der Natur nachempfunderer menschlicher Körper angesehen werden. Es verknüpft Sensoren, Signalverarbeitung und Aktoren so zu einem Gesamtsystem, dass sie fühlen, entscheiden und handeln können (siehe

²Im Gegensatz zu der hier formulierten Definition ist im allgemeinen Sprachgebrauch der Begriff MEMS oft auch als Synonym für beliebige Mikrosysteme in Verwendung.

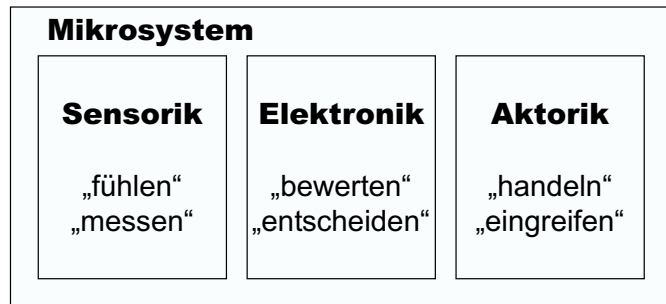


Abbildung 2.1: Mikrosystem

Abbildung 2.1). Hierbei entsprechen Sensoren den menschlichen Sinnesorganen, die Signalverarbeitung dem Gehirn und Aktoren den Gliedmaßen [Bot04].

Ein guter Überblick über bereits am Markt vorhandene oder sich noch in der Entwicklung befindliche **Mikrostrukturen** ist aus [Tsc99] zu entnehmen:

- einfache Strukturen, wie hoch präzise Spitzen, Gräben oder Löcher, z.B. für Düsen von Tintenstrahldruckern, Kanäle in chemischen Analysegeräten oder Rastertunnelmikroskop-Spitzen
- Mikrosensoren, wie Drucksensoren, Beschleunigungssensoren, Neigungssensoren, Temperatursensoren, Magnetfeldsensoren, z.B. für Navigation, Strom- und Längenmessungen, Sensoren für ausgewählte chemische Größen oder Biosensoren
- Mikroaktoren, wie steuerbare Endoskope, Mikrospiegel, Mikromotoren mit den zugehörigen Getrieben, Mikropumpen, Mikrodosierung von Flüssigkeiten, z.B. für die Medikamentendosierung oder Brennstoffeinspritzung

Der Überblick über die Bauteile von Mikrosystemen verdeutlicht, welche Innovationskraft die hieraus entstehenden Mikrosysteme besitzen können.

Ein Beispiel für ein Mikrosystem ist in Abbildung 2.2 durch ein Airbag-Auslösesystem skizziert. Es enthält Beschleunigungssensor, Schwellwertschalter und Auslösungsmechanismus, die durch Aufbau- und Verbindungstechniken zusammengefügt werden.

Ein anderes Beispiel für ein Mikrosystem ist eine Mikropumpe zur Medikamentendosierung. Kern eines solchen Medikamentendosiersystems ist eine Silizium-Mikropumpe mit einer Fläche von wenigen cm^2 und einer Dicke im Millimeterbereich. In das Silizium sind die Mikroventile mit Durchflusskanälen und einer

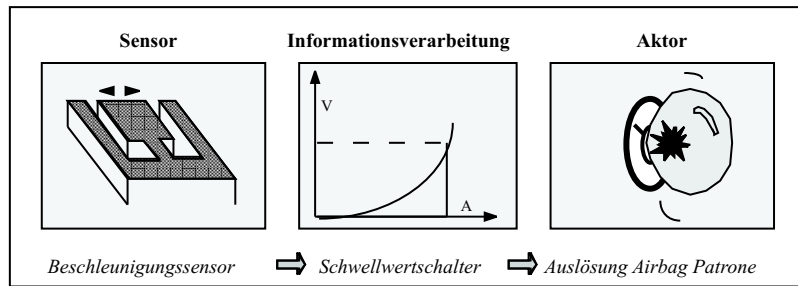
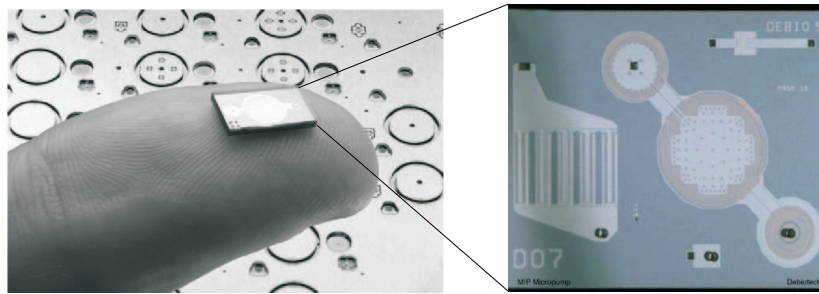


Abbildung 2.2: Airbag-Auslösesystem [Hah99]

Pumpkammer angebracht (siehe rechte Abbildung 2.3). Die Mikropumpe ist in der Lage Flüssigkeiten im Bereich von 10 bis 50 ml pro Tag zu dosieren. So ist eine kontinuierliche Medikamentenverabreichung über einen längeren Zeitraum mit der Mikropumpe möglich. Ein nahe liegendes Einsatzgebiet ist die Insulinverabreichung an Glukosepatienten [MHG⁺01].

Abbildung 2.3: Mikropumpe mit einer Größe von 6 mm x 10 mm [MHG⁺01]

Weitere Beispiele für intelligente Mikrosysteme werden in [Mad01] vorgestellt.

2.2 Silizium-Mikromechanik

Die in der industriellen Praxis wichtigsten Vertreter von Fertigungsverfahren für Mikrosysteme sind die Silizium-Mikromechanik, die LIGA-Technik, die Laserbasierten Mikrostrukturierungsverfahren und die für Mikrostrukturen angepassten Verfahren der Feinwerktechnik. Die drei letztgenannten Verfahren werden unter dem Begriff **HARMST (High-Aspect-Ratio-Micro-System-Technologies)** subsummiert. Sie kommen immer da zum Einsatz, wo hohe Aspektverhältnisse auftreten oder die Verwendung von Materialien erforderlich ist, die sich nicht auf Silizium-Substraten erzeugen lassen. Details zu den HARMST-Technologien bleiben aufgrund des begrenzten Umfangs der Arbeit unerwähnt.

Einen guten Überblick mit zahlreichen Anwendungsbeispielen geben Brück, Rizvi und Schmidt in [BRS01].

Das älteste und am meisten verbreitete Fertigungsverfahren ist die **Silizium-Mikromechanik** [Pet82]. Sie besitzt in vielen Punkten eine sehr hohe Affinität zur Mikroelektronik. Nicht nur das der Silizium-Einkristall als Basis-Werkstoff für die Mikrostrukturen zum Einsatz kommt, auch das Fertigungsverfahren ist nahezu identisch übernommen worden. Eine Weiterentwicklung einzelner mikroelektronischer Fertigungstechniken war erforderlich, da in der Mikromechanik im Gegensatz zur Mikroelektronik dreidimensionale Bauteile herzustellen sind.

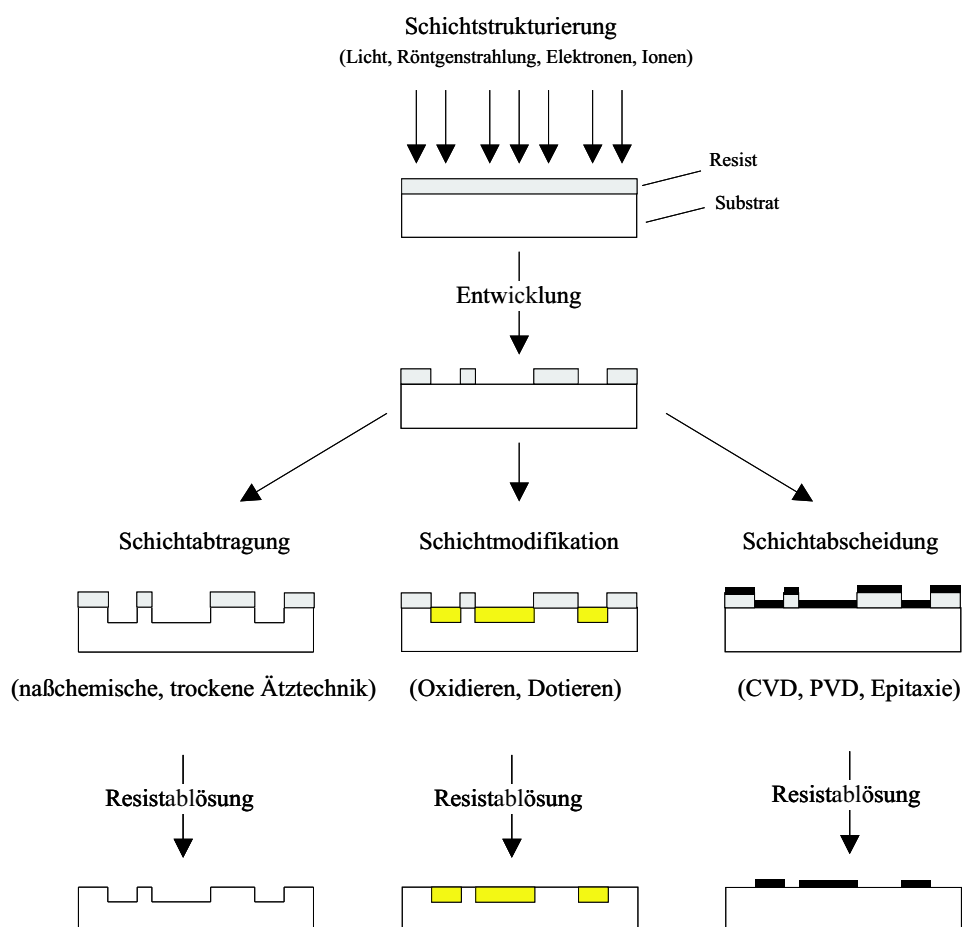


Abbildung 2.4: Prinzip der Erzeugung von Mikrostrukturen durch die Silizium-Mikromechanik [Büt94]

Die Silizium-Mikromechanik unterscheidet zwischen der **Silizium-Oberflächen-Mikromechanik (OMM)** und der **Substrat-Mikrotechnik**. Die Substrat-Mikrotechnik arbeitet durch das anisotrope, kristallografische Nassätzen

und mit Unterstützung künstlich erzeugter Ätzstoppschichten dreidimensionale Körper aus dem Silizium-Einkristall heraus [Men97]. Bei der OMM werden additiv Schichtfolgen auf dem Silizium-Wafer aufgebracht, strukturiert und selektiv geätzt.

Die verwendeten Fertigungstechniken bei der OMM stammen überwiegend aus der Mikroelektronik und lassen sich grob in vier Gruppen einteilen:

1. **Schichtstrukturierung** zur Übertragung von Strukturen auf die Substratoberfläche
2. **Schichtabtragung** zum Strukturieren und Abtragen von Schichten
3. **Schichtabscheidung** zum Aufbau von zusätzlichen Materialschichten auf dem Substrat
4. **Schichtmodifikation** zum Ändern der Struktur und/oder der chemischen Zusammensetzung des Substrats

Für die Herstellung von Mikrosystemen kommen die oben beschriebenen Fertigungstechniken iterativ zur Anwendung. Abbildung 2.4 zeigt das Prinzip der Erzeugung von Mikrostrukturen durch die Silizium-Mikromechanik grafisch.

Details zu den Fertigungstechniken sind der umfangreichen Literatur, wie zum Beispiel [Mad01], [Ehr01] und [Men97] zu entnehmen.

Ein großer Vorteil der Silizium-Mikromechanik liegt in der relativ einfachen monolithischen Integration von mikromechanischen und mikroelektronischen Bauteilen auf dem Substrat. Die Nutzung der vorhandenen Fertigungskapazitäten der Mikroelektronik ist ein weiterer Pluspunkt. Diese Gründe haben dazu geführt, dass der Silizium-Mikromechanik im weiteren Verlauf der Arbeit besondere Beachtung zukommt.

Kapitel 3

Entwurfsmodell für miniaturisierte technische Systeme

Das Kapitel beginnt mit einer Definition von grundlegenden Begriffen, die beim Entwurf von miniaturisierten technischen Systemen Verwendung finden. Danach wird ein Modell für den Entwurf vorgestellt und schrittweise verfeinert. Das Modell beschreibt, welche Aufgaben im Rahmen eines Entwurfs durchzuführen sind und legt eine Entwurfsmethodik fest, die es erlaubt, Werkzeuge für Entwurfsaufgaben abzuleiten. Aspekte wie Konvergenz und Divergenz von Entwurfsprozessen werden betrachtet. Anschließend erfolgt die Integration von Anforderungen aus der Industrie in das Modell. Am Ende steht ein Entwurfsmodell zur Verfügung, welches später für den Entwurf von Mikrosystemen und Mikrostrukturen zum Einsatz kommt.

3.1 Begriffsbestimmung

Bei der Entwicklung von mikrotechnischen Gegenständen steht am Anfang eine Idee, die es gilt beispielsweise in Form eines Mikroprozessors oder eines Mikrosensors in die Realität umzusetzen. Die Phase zwischen der Idee und der Fertigstellung einer Bauanleitung für einen Gegenstand heißt Entwurfsprozess oder Entwurf.

Im Folgenden wird eine informelle Definition des Terminus Entwurf formuliert, die eine Charakterisierung dessen liefert, was unter einem Entwurfsprozess für mikrotechnische Gegenstände zu verstehen ist.

Definition: 4 *Unter einem **Entwurfsprozess** oder **Entwurf** versteht man die schrittweise Erstellung eines Satzes von Dokumenten, die, ausgehend von einer Spezifikation des zu entwerfenden Gegenstandes, eine geschlossene, korrekte und vollständige Beschreibung darstellen, die ausreicht, diesen Gegenstand in einer der Spezifikation genügenden Ausführung zu fertigen [Brü93].* //

Folgende Definition legt die Begriffe Entwurfsgegenstand und Fertigungsgegenstand fest.

Definition: 5 *Ein Gegenstand für den ein Entwurfsprozess durchzuführen ist heißt **Entwurfsgegenstand**. Ein Gegenstand, für den ein Entwurfsprozess erfolgreich durchgeführt wurde, heißt **Fertigungsgegenstand**.* //

Was unter dem Begriff der Spezifikation zu verstehen ist, liefert die nächste Definition.

Definition: 6 *Unter einer **Entwurfsspezifikation** oder **Spezifikation** versteht man eine formale Beschreibung des Entwurfsgegenstandes (**Entwurfsbeschreibung**). Des Weiteren beinhaltet eine Spezifikation die für den Entwurfsgegenstand relevanten Restriktionen (**Entwurfsrestriktionen**).* //

Definition 4 fordert, dass am Ende des Entwurfsprozesses ein Satz von Dokumenten vorhanden ist, mit dessen Hilfe der Fertigungsprozess für den Fertigungsgegenstand anzustoßen ist. Daraus leitet sich folgende Definition für eine Fertigungsspezifikation ab.

Definition: 7 *Eine **Fertigungsspezifikation** ist ein Satz von Dokumenten, der sowohl alle Abmessungen des Fertigungsgegenstandes exakt festlegt als auch alle Informationen enthält um den Fertigungsgegenstand zu fertigen (**Fertigungsbeschreibung**). Des Weiteren beinhaltet die Fertigungsspezifikation alle vom Fertigungsgegenstand erfüllten Restriktionen (**Fertigungsrestriktionen**).* //

Es wird nicht gefordert, dass die Spezifikation des Entwurfsgegenstandes vollständig¹ ist. Gewisse Entscheidungen bezüglich Entwurfsbeschreibung und Entwurfsrestriktionen sind gegebenenfalls erst im Laufe des Entwurfsprozesses zu

¹Eine Spezifikation für einen Entwurfsgegenstand ist genau dann vollständig, wenn sie alle Informationen für die komplette Durchführung des Fertigungsprozesses enthält. Eine vollständige Entwurfsspezifikation wird daher auch als Fertigungsspezifikation bezeichnet.

fällen. Während des Entwurfs ist es deshalb möglich, neue Entwurfsrestriktionen zu formulieren oder bestehende zu verschärfen oder zu entschärfen. Wichtig dabei ist, dass die Restriktionen insgesamt nur schärfer werden können. Die Restriktionen, welche durch die Spezifikation des Entwurfsgegenstandes vorgegeben sind, legen ein Minimum fest, welches während des Entwurfs nicht zu unterschreiten ist. Um zu untersuchen, wie sich die Modifikation einer Restriktion während des Entwurfs auf andere Restriktionen auswirkt, ist es notwendig, die Beziehungen zwischen Restriktionen darzustellen.

Eine vollständige Spezifikation ist erst durch eine Fertigungsspezifikation gegeben. Sie beschreibt den Fertigungsgegenstand und seine Fertigungsrestriktionen eindeutig und ermöglicht es, den Fertigungsprozess zu aktivieren. Durch die Freiheiten des Entwurfsprozesses, gewisse Entscheidungen zu treffen, kann es für eine Spezifikation unterschiedliche Fertigungsspezifikationen geben.

Abbildung 3.1 ist ein Beispiel für eine unvollständige Spezifikation.

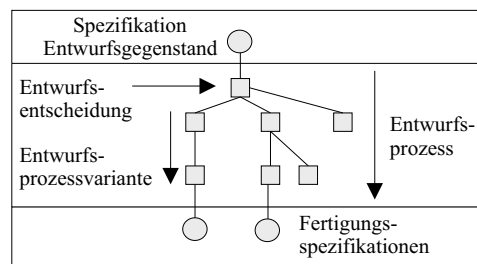


Abbildung 3.1: Entwurfsentscheidung

Daher ist im Entwurfsprozess eine **Entwurfsentscheidung** zu fällen. Es entstehen zwei **Entwurfsprozess-Varianten**, die jeweils nach zwei Entwurfsschritten zu einer Fertigungsspezifikation führen. Die Rechtecke in der Abbildung repräsentieren die einzelnen Entwurfsschritte. Zu einem späteren Zeitpunkt wird auf die hier beschriebene Problematik Entwurfsentscheidungen zu treffen noch einmal eingegangen.

Nachdem diese grundlegenden Termini eingeführt wurden, wird im nächsten Abschnitt ein Modell für den Entwurfsprozess vorgestellt.

3.2 Ein Modell für den Entwurf

Eine einfache Möglichkeit den Entwurfsprozess zu modellieren besteht darin, einen Black-Box-Ansatz zu wählen. Er erhält die Spezifikation des Entwurfsge-

genstandes als Eingabe und liefert nach dem Ablauf einer gewissen Zeitspanne das Ergebnis in Form einer Fertigungsspezifikation (siehe Abbildung 3.2).



Abbildung 3.2: Entwurfsprozess

Bevor die Spezifikation als Eingabe für den Entwurfsprozess dienen kann, wird sie auf Basis der Produktidee erstellt. Eine **Produktidee** besteht aus einer Ansammlung von gewünschten **Leistungsmerkmalen** des Fertigungsgegenstandes. Leistungsmerkmale werden beispielsweise durch ein Unternehmen festgelegt, welches plant, den Fertigungsgegenstand zu produzieren. In Gesprächen zwischen der Produktplanung und Entwicklung entstehen erste Ideen und Vorstellungen über den Fertigungsgegenstand. Darauf basierend werden die Leistungsmerkmale abgeleitet. Eine andere Möglichkeit, Leistungsmerkmale zu definieren, besteht darin, auf bereits produzierte Fertigungsgegenstände zurückzugreifen und deren Merkmale zu übernehmen oder zu verschärfen.

Leistungsmerkmale repräsentieren eine erste informelle Charakterisierung des Fertigungsgegenstandes und dokumentieren im Gegensatz zu der Spezifikation, die den Entwurfsgegenstand als Ganzes beschreibt, einzelne wichtige Produkteigenschaften. Sie bilden den Input für die sich anschließende **Spezifikationsphase**, in der eine erste Version der Spezifikation des Entwurfsgegenstandes entsteht (siehe Abbildung 3.3).



Abbildung 3.3: Spezifikationsphase

Abbildung 3.2 und 3.3 verdeutlichen die Problembereiche beim Entwurf von hochkomplexen Gegenständen. Die Spezifikationsphase und der Entwurfsprozess können ein Ergebnis nur dann liefern, wenn die Produktidee und damit verbunden die Leistungsmerkmale sowie die Spezifikation des Entwurfsgegenstandes hinreichend präzise sind. Des Weiteren ist die Fertigungsspezifikation so exakt zu formulieren, dass sich damit der anschließende Fertigungsprozess problemlos steuern lässt. Um diese Anforderungen zu erfüllen, bedarf es eines **Formalisierungsprozesses**, der aus einer informellen Charakterisierung des Fertigungsgegenstandes (hier: die Leistungsmerkmale) formale Randbedingungen ableitet, die so genannten **Restriktionen**, welche bis zur Erstellung der Fertigungsspezifikation durchgängig einzuhalten sind (siehe Abbildung 3.4).

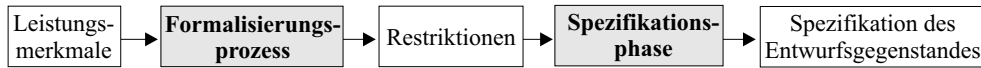


Abbildung 3.4: Formalisierungsprozess

Um diesen Formalisierungsprozess durchzuführen werden formale Sprachen eingesetzt. Mit ihrer Hilfe werden die Ergebnisse des Formalisierungsprozesses, der Spezifikationsphase und des Entwurfsprozesses beschrieben.

Definition: 8 Sei A eine endliche, nichtleere Menge, dann ist A ein **Alphabet**. Die Elemente von A heißen **Buchstaben**. //

Definition: 9 Sei $n \in \mathbb{N}$. Eine endliche Folge $w = (a_1, \dots, a_n)$ von Buchstaben aus A heißt **Wort über A** . Die Menge aller Worte über A heißt A^* . //

Definition: 10 Sei A Alphabet, jede Menge $L \subseteq A^*$ heißt **formale Sprache über A** . //

Definition: 11 Angenommen, es gibt eine formale Sprache L^0 über A^0 , mit der Restriktionen bezüglich Leistungsmerkmalen eines Fertigungsgegenstandes zu beschreiben sind. Dann heißt eine in der Sprache L^0 formulierte Beschreibung der Restriktion i bezüglich eines Leistungsmerkmals LM_i des Fertigungsgegenstandes \mathbf{R}_i^0 oder \mathbf{R}_i . Eine Menge von Restriktionen wird als $\mathbf{R}_{in}^0 := \{\mathbf{R}_1, \dots, \mathbf{R}_n\}$ bezeichnet. //

Der Formalisierungsprozess in Abbildung 3.5 erhält die informellen Leistungsmerkmale als Eingabe und erstellt als Ausgabe die Menge von Restriktionen \mathbf{R}_{in}^0 mit Hilfe der formalen Sprache L^0 .

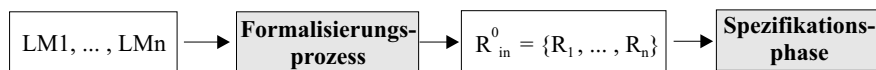


Abbildung 3.5: Restriktionen

Hier stellt sich die Frage, ob überhaupt ein Algorithmus existiert, welcher die Produktidee mit den informellen Leistungsmerkmalen in Restriktionen \mathbf{R}_{in}^0 überführt. Zumindest ist das Ergebnis also \mathbf{R}_{in}^0 nicht zu verifizieren, da die Eingabe genügend Spielraum für Interpretationen zulässt. Daraus folgt, dass Fehler beim Formalisierungsprozess nicht automatisch zu erkennen sind. Dies kann

bei größeren Projekten dazu führen, dass unter falschen Randbedingungen eine Entwicklung eingeleitet wird und das Ergebnis am Ende nicht dem entspricht, was der Kunde (Produktidee und Leistungsmerkmale) sich vorgestellt hatte. Daher ist es unerlässlich, während der gesamten Entwicklung permanent das aktuelle Ergebnis mit dem gewünschten Resultat abzugleichen und bei Abweichungen sofort durch Modifikationen am aktuellen Ergebnis zu reagieren. An dieser Stelle wird der interne Ablauf des Formalisierungsprozesses nicht weiter betrachtet.

Definition: 12 *Angenommen, es gibt die formalen Sprachen L^1 über A^1 und L^2 über A^2 , mit denen Spezifikationen für einen Entwurfs- und Fertigungsgegenstand zu beschreiben sind. Dann heißt eine in der Sprache L^1 formulierte Spezifikation des Entwurfsgegenstandes \mathbf{L}_{in}^1 und in der Sprache L^2 formulierte Fertigungsspezifikation \mathbf{L}_{out}^2 .* //

Der Entwurfsprozess fungiert als Abbildung, welche die Eingabe, gegeben durch L_{in}^1 , in die Ausgabe, gegeben durch L_{out}^2 , überführt. Die Spezifikation enthält zum einen die Beschreibung des Entwurfsgegenstandes und zum anderen Restriktionen. Die Entwurfsbeschreibung wird formal durch $L_{in,d}^1$ festgelegt, und die Entwurfsrestriktionen durch $L_{in,r}^1$. Analog dazu werden $L_{out,d}^2$ für Fertigungsbeschreibung und $L_{out,r}^2$ für die Fertigungsrestriktionen verwendet (siehe Abbildung 3.6).

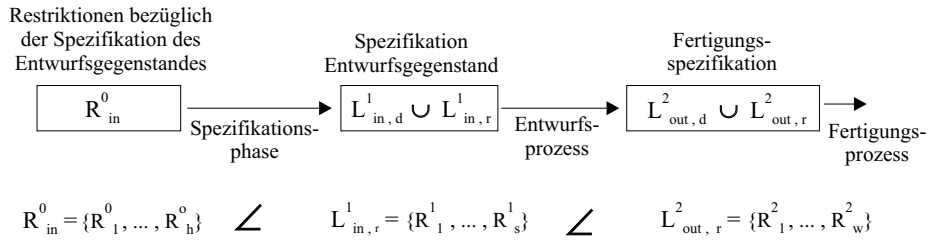


Abbildung 3.6: Einsatz von formalen Sprachen

Dabei ist zu beachten, dass $L_{in,d}^1$ und $L_{out,d}^2$ die Restriktionen implizit enthalten und sowohl $L_{in,r}^1$, wie auch $L_{out,r}^2$, diese nochmals explizit beschreiben. Sind beispielsweise Restriktionen für einen Gegenstand durch $L_{out,r}^2$ explizit gegeben, so werden die Restriktionen implizit durch den Gegenstand, der durch $L_{out,d}^2$ beschrieben ist, eingehalten (wenn $L_{out,d}^2$ korrekt ist). Erfolgt auf Basis der Fertigungsspezifikation eine Produktion des Fertigungsgegenstandes, so kann anhand der Fertigungsrestriktionen relativ einfach überprüft werden, inwieweit

das Produktionsergebnis (z.B. ein Mikrosystem) dem vorgegebenen Bauplan (Fertigungsbeschreibung) entspricht.

Zum Verständnis der vorliegenden Arbeit ist es entscheidend, den Zusammenhang zwischen den Restriktionen R_{in}^0 , den Restriktionen bei der Spezifikation des Entwurfsgegenstandes $L_{in,r}^1$ und den Fertigungsrestriktionen $L_{out,r}^2$ zu verstehen. In Abbildung 3.6 ist durch das Symbol \angle eine Beziehung zwischen den Restriktionsmengen definiert worden. Die Menge, die rechts vom Symbol \angle steht enthält insgesamt betrachtet schärfere oder zumindest äquivalente Restriktionen bezüglich der Menge auf der linken Seite des Symbols \angle . Was die zuvor getroffene Aussage wiederholt, dass die Restriktionen insgesamt nur schärfer werden können.

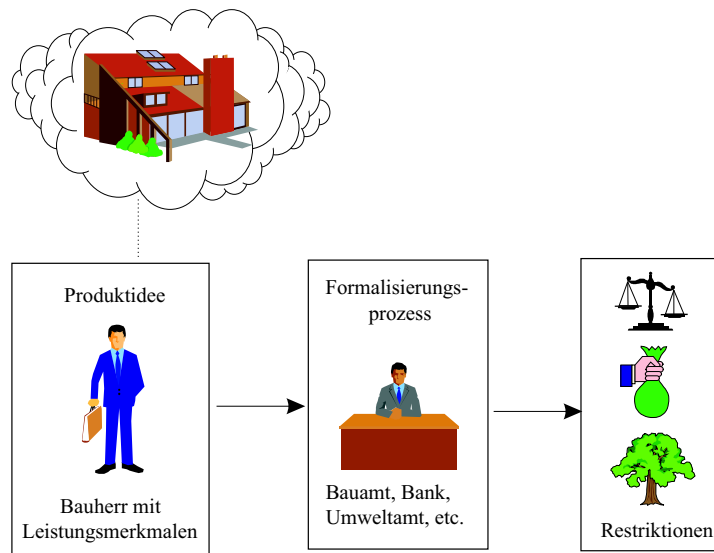


Abbildung 3.7: Beispiel: Produktidee Hausbau

Das nachfolgende Beispiel soll den Zusammenhang zwischen den Termini verdeutlichen. Als Produktidee wird der Bau eines Hauses angenommen. Diese existiert momentan nur im Kopf des Bauherrn und bildet die Basis, um eine Liste von informellen Leistungsmerkmalen aufzustellen:

- LM1: das Haus soll freistehend sein,
- LM2: das Haus soll zwei Etagen besitzen,
- LM3: das Haus soll einen große baumfreie Rasenfläche besitzen,
- LM4: das Haus soll finanziert werden.

Nach Rücksprache des Bauherrn mit der Bank sowie dem Bau- und Umweltamt ergibt sich eine erste Menge von baurechtlichen, umweltrechtlichen und finanziellen Restriktionen L_{in}^0 (siehe Abbildung 3.7). Beispielsweise sind das baurechtliche Bestimmungen, die einzuhalten (das Haus darf nur eine Etage besitzen) sind oder Umweltauflagen (bei der Einrichtung des Gartens ist der alte Baumbestand zu erhalten), die zu erfüllen sind. Die Bank legt auf Basis der Einkünfte des Bauherrn fest, wie hoch der Kredit für den Hausbau ausfällt und bestimmt auf diese Weise die einzuhaltenden finanziellen Restriktionen.

Der Architekt erstellt unter Berücksichtigung der Restriktionen eine erste noch unvollständige Bauzeichnung (Entwurfsbeschreibung) mit Angaben über die Raumaufteilung, die Bauhöhen und das äußere Erscheinungsbild des Hauses. In der Bauzeichnung sind zusätzlich zeitliche und statische Restriktionen $L_{in,r}^1$ enthalten, welche sich auf die Bauabschnitte (wie lange darf ein Bauabschnitt maximal dauern) und die Konstruktion (wie stark müssen die Wände minimal sein) des Hauses beziehen (siehe Abbildung 3.8).

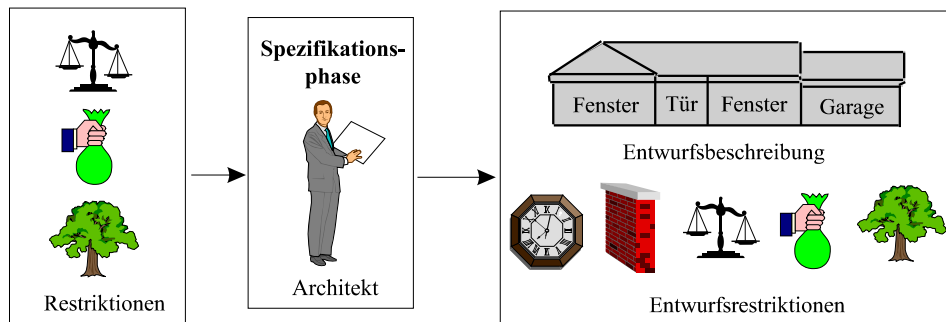


Abbildung 3.8: Beispiel: Spezifikationsphase

Die Bauzeichnung legt der Architekt dem Bauherrn und dem Bauleiter zur Prüfung vor (Start Entwurfsprozess). Der Bauherr bemängelt zum Beispiel, dass die finanziellen Restriktionen nicht eingehalten wurden. Die Konsequenz ist der Wegfall der geplanten Garage. Der Bauleiter legt die Reihenfolge der einzelnen Bauabschnitte und die dabei durchzuführenden Tätigkeiten fest. Insbesondere ist zu besprechen, welche Baumaterialien zu verwenden sind (hieraus ergeben sich Restriktionen bezüglich Materialien) und welche Handwerker zu beauftragen sind (hieraus ergeben sich Restriktionen bezüglich der Qualität von handwerklichen Tätigkeiten).

Hierdurch kann das Haus ein anderes äußeres Erscheinungsbild bekommen als der Bauherr es sich vorgestellt hat. An dieser Stelle ist zu entscheiden, inwieweit ein Kompromiss bezüglich Machbarkeit und Vorstellung des Bauherrn zu finden ist. Nachdem die Bauzeichnung vervollständigt, erweitert und geprüft wurde,

werden alle Informationen im so genannten Bauplan (Fertigungsbeschreibung, Fertigungsrestriktionen) dokumentiert. Jetzt ist es möglich mit dem Hausbau zu beginnen. Ist das Haus fertiggestellt, erfolgt die Bauabnahme auf Grundlage der Fertigungsrestriktionen $L_{out,r}^2$ (siehe Abbildung 3.9).

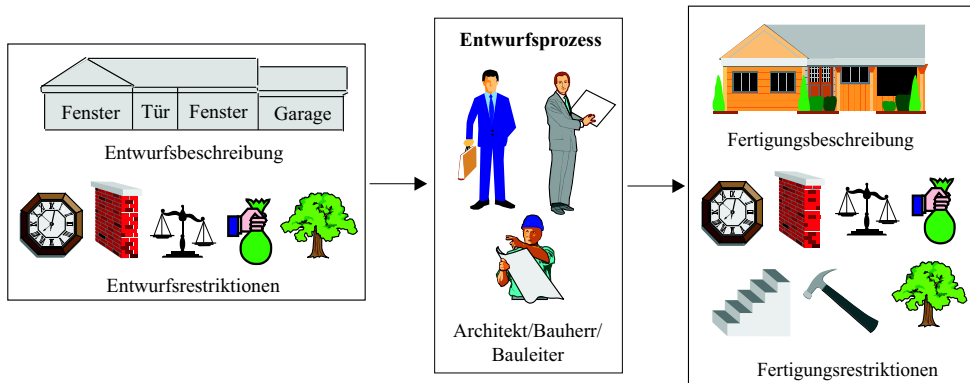


Abbildung 3.9: Beispiel: Entwurfsprozess

3.3 Entwurfsaktivitäten

Nachdem die Sprachen für die Eingaben und Ausgaben des Entwurfsprozesses spezifiziert sind, kann damit begonnen werden die Abbildung näher zu untersuchen. Welche Entwurfsaktivitäten sind durchzuführen, um die gewünschte Abbildung zu erhalten?

Die Aktivitäten beim Entwurf können nach Rammig in zwei große Bereiche eingeteilt werden [Ram89]:

- **Generierende Aktivitäten**
- **Überprüfende Aktivitäten**

Die generierenden Aktivitäten erhalten als Input $L_{in,d}^1$ und versuchen schrittweise das Entwurfsergebnis $L_{out,d}^2$ zu erzeugen. Das vorläufige Entwurfsergebnis wird durch L_{check} festgehalten und an die überprüfenden Aktivitäten übergeben. Diese werden durch $L_{in,r}^1$ gesteuert und arbeiten auf der Ausgabe L_{check} der generierenden Aktivitäten. Ist durch die überprüfenden Aktivitäten eine Abweichung zwischen L_{check} und den Restriktionen $L_{in,r}^1$ erkannt worden, erfolgt die Ausgabe von Korrekturanweisungen. Diese werden mittels L_{korrr} an die generierenden Aktivitäten übermittelt und führen zu einer Modifizierung des aktuellen Entwurfsergebnisses.

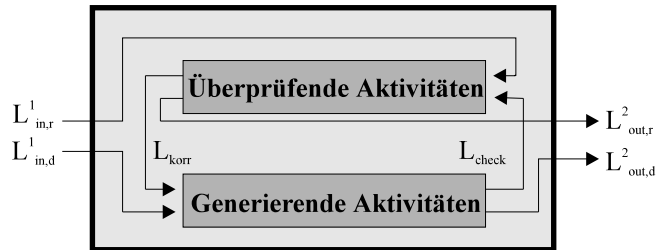


Abbildung 3.10: Makroskopisches Modell des Entwurfsprozesses nach [Ram89]

Ein makroskopisches Modell für den Entwurfsprozess ist aufgestellt worden, in dem $L^1_{in,d}$ und $L^1_{in,r}$ als Regelmechanismen dienen. Sie steuern die Rückkopplungsschleife, die sich aus generierenden und überprüfenden Aktivitäten sowie aus L_{check} und L_{korr} zusammensetzt (siehe Abbildung 3.10).

Mit Hilfe der generierenden und überprüfenden Aktivitäten ist es gelungen, die Entwurfsaufgaben transparenter darzustellen. Eine weitere Verfeinerung des Modells ist durch eine Aufteilung der Aktivitäten in bestimmte Subaktivitäten möglich. Für den Bereich der überprüfenden Aktivitäten werden drei Subaktivitäten eingeführt [Ram89]:

Verifikation: Prüft, ob das vorläufige Entwurfsergebnis (L_{check}) die Entwurfsrestriktionen einhält.

Entscheidung: Sind bei der Verifikation Abweichungen bemerkt worden, ist eine Entscheidung zu treffen, die festlegt, wie die Differenzen zu eliminieren sind. Darüber hinaus besteht die Möglichkeit einzelne Entwurfsrestriktionen zu verschärfen oder zu entschärfen.

Modifikationen festlegen: L_{korr} ist einzusetzen, um die Entscheidung so zu formulieren, dass eine Folge von Modifikationsanweisungen entsteht, die anschließend von den generierenden Aktivitäten zu berücksichtigen ist

Abbildung 3.11 stellt den Sachverhalt grafisch dar.

Während des Entwurfsprozesses werden die generierenden Aktivitäten und die Subaktivitäten Verifikation, Entscheidung und Modifikation mehrmals durchlaufen. Dies gilt unter der Annahme, dass die generierenden Aktivitäten nicht optimiert sind und daher mehrere Durchläufe benötigen. Geht man von der theoretischen Existenz von optimierten generierenden Aktivitäten aus, so ist nur ein Durchlauf nötig und die überprüfenden Aktivitäten wären überflüssig. In der Praxis sind mehrere Durchläufe notwendig und somit auch der Einsatz von überprüfenden Aktivitäten.

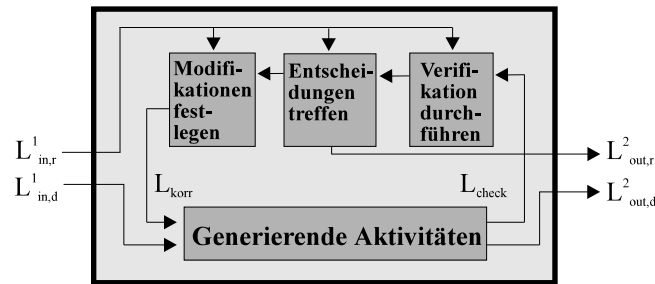


Abbildung 3.11: Verfeinertes Modell des Entwurfsprozesses nach [Ram89]

Das klassische Modell von Rammig bildet die Grundlage für ein neues Entwurfsmodell in der Mikrotechnik. Es wurde ausgewählt, da es einen sprach-basierten Ansatz besitzt und so hervorragend geeignet ist, um spätere rechnergestützte Entwurfswerkzeuge zu unterstützen, die nur auf konkret formalisierten Begebenheiten operieren können.

3.4 Konvergente und Divergente Entwürfe

In diesem Abschnitt wird die Frage untersucht, ob ein Entwurfsprozess für eine gegebene Spezifikation immer ein Ergebnis, also eine Fertigungsspezifikation in Form von $L^2_{out,d}$ und $L^2_{out,r}$, liefert.

In Abbildung 3.1 auf Seite 15 fällt auf, dass bestimmte Entwurfsprozess-Varianten zu einer Fertigungsspezifikation führen und andere nicht. Bei einer Spezifikation mit zu großen Anforderungen an den Entwurfsgegenstand kann der Fall eintreten, dass keine Entwurfsprozess-Variante zu einem Ergebnis führt. Es existiert keine Fertigungsspezifikation und somit auch kein Fertigungsprozess für den Gegenstand.

Zwischen der Abbildung 3.1 und dem Modell des Entwurfsprozesses besteht eine direkte Beziehung. Mit Hilfe des Modells sind die einzelnen Entwurfsprozess-Varianten zu erstellen. L_{check} ist durch die einzelnen Rechtecke repräsentiert. Ein Modell-Durchlauf ist durch die Verbindung zwischen zwei Rechtecken dargestellt.

Abbildung 3.12 und 3.13 zeigen zwei Entwurfsprozess-Varianten in Abhängigkeit von L_{korr} . Sie unterscheiden sich im Entwurfsergebnis. Abbildung 3.12 liefert im Gegensatz zur Abbildung 3.13 nach endlich vielen Durchläufen eine Fertigungsspezifikation.

Um Entwurfsprozesse, die für eine gegebene Spezifikation zu einem Ergebnis

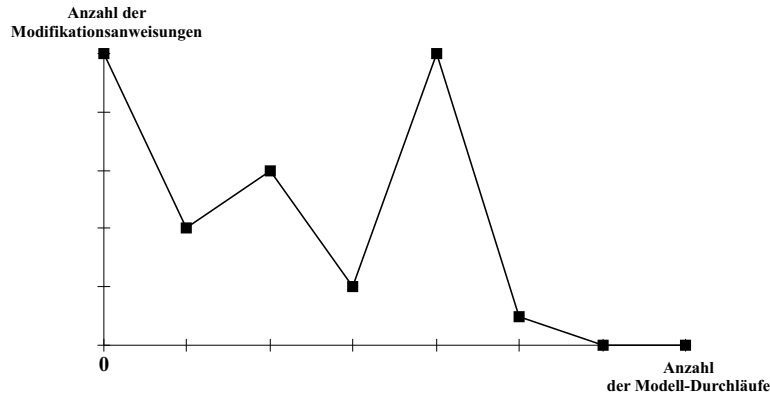


Abbildung 3.12: Entwurfsprozess-Variante mit Fertigungsspezifikation

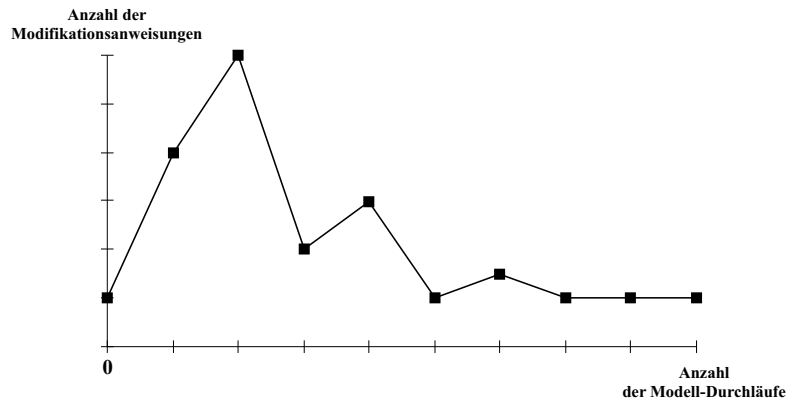


Abbildung 3.13: Entwurfsprozess-Variante ohne Fertigungsspezifikation

führen und solche die zu keinem Ergebnis kommen voneinander zu unterscheiden werden folgende Definitionen eingeführt:

Definition: 13 Sei die Spezifikation L_{in}^1 für einen Entwurfsgegenstand gegeben. Ein Entwurfsprozess wird als **konvergent** für die Spezifikation L_{in}^1 bezeichnet, wenn eine Entwurfsprozess-Variante existiert, für die nach einer endlichen Anzahl von Durchläufen L_{korr} keine Modifikationsanweisungen mehr enthält. //

Nach dieser Definition wäre der Entwurfsprozess in Abbildung 3.1 konvergent, da eine Entwurfsprozess-Variante existiert, die nach endlich vielen Entwurfsschritten (Durchläufen) eine Fertigungsspezifikation liefert. Die Ausgabe einer Fertigungsspezifikation beinhaltet implizit, dass L_{korr} keine Modifikationsanweisungen mehr enthält.

Definition: 14 Sei die Spezifikation L_{in}^1 für einen Entwurfsgegen-

stand gegeben. Ein Entwurfsprozess wird als **divergent** für die Spezifikation L_{in}^1 bezeichnet, wenn für jede Entwurfsprozess-Variante mit einer endlichen Anzahl von Durchläufen L_{korrr} noch Modifikationsanweisungen enthält. //

Abbildung 3.14 und 3.15 sind Beispiele für einen konvergenten und divergenten Entwurfsprozess.

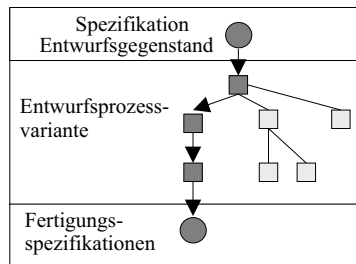


Abbildung 3.14: Konvergenter Entwurfsprozess

Die Konvergenz von Entwurfsprozessen ist nicht immer sofort zu erkennen. Hier wäre es notwendig **Konvergenzkriterien** zu definieren, mit deren Hilfe man Aussagen darüber treffen könnte, ob oder mit welcher Wahrscheinlichkeit ein Entwurfsprozess konvergiert oder nicht.

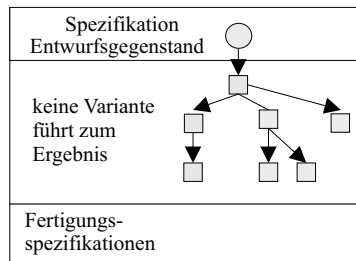


Abbildung 3.15: Divergenter Entwurfsprozess

Eine Entwurfsvariante besteht aus einer Folge von Entwurfsschritten, wobei ein Entwurfsschritt durch einen Modell-Durchlauf festgelegt ist und als Ergebnis L_{check} liefert. Der Entwurfsschritt erhält Informationen in Form von $L_{in,d}^1$, $L_{in,r}^1$ und L_{korrr} und generiert hieraus ein aktuelles L_{check} . Tritt der Fall ein, dass die Spezifikation für den Entwurfsgegenstand unvollständig ist und damit $L_{in,d}^1$ und $L_{in,r}^1$ nicht alle Informationen enthalten, ist im Entwurfsprozess eine Entscheidung zu fällen, um den Informationsmangel zu beseitigen und die Ausgabe eines neuen L_{check} zu ermöglichen. Daraus folgt, dass der Entwurfsexperte eine

zentrale Rolle beim Entwurf spielt. Er wird benötigt, um wichtige Entwurfsentscheidungen zu treffen. Ein voll-automatisierter Entwurfsprozess scheint daher zumindest zur Zeit nicht möglich.

Hat der Experte sich für eine bestimmte Entwurfsvariante entschieden, entsteht der Bedarf zu beurteilen, ob die Entwurfsvariante zum Ziel, also zur Fertigungsspezifikation führt. Hier sollte eine Funktion f vorhanden sein um den aktuellen Stand des Entwurfs zu bewerten. Diese **Bewertungsfunktion** arbeitet auf Basis von L_{korrr} und dient als Hilfsmittel bei der Entscheidung, ob die Entwurfsvariante weiter zu untersuchen ist oder nicht.

Die Menge von Prozessvarianten eines Entwurfs, die eine Fertigungsspezifikation liefern, bilden einen Lösungsraum. Ein Element im Lösungsraum ist demzufolge durch eine Fertigungsspezifikation vertreten. Die Fertigungsspezifikationen unterscheiden sich in Bezug auf den Erfüllungsgrad vorgegebener Entwurfsrestriktionen.

Beispielsweise kann eine Entwurfsrestriktion in Form eines Intervalls vorgegeben sein (z.B. $[\min, \max]$). Aus der Annahme, dass eine Fertigungsspezifikation qualitativ um so hochwertiger ist je näher die untere Intervallgrenze berührt wird, folgt dass ein optimales Entwurfsergebnis des Lösungsraumes der unteren Intervallgrenze am nächsten ist. Es gibt also Möglichkeiten die Qualität der Elemente des Lösungsraumes in Abhängigkeit vom Erfüllungsgrad der Entwurfsrestriktionen zu bewerten. Zum Beispiel könnte ein anderes Bewertungskriterium für die Elemente des Lösungsraumes sein, welches Entwurfsergebnis die kostengünstigste Fertigung erlaubt.

3.5 Das Kreismodell

Das Entwurfsmodell von Rammig spiegelt die Abläufe und Aktivitäten beim Entwurfsprozess auf einem hohen Abstraktionsniveau wider. Um dessen Eignung beim konkreten Entwurf zu erhöhen, sollen Anforderungen aus der Mikrotechnik-Industrie in das Modell einfließen.

Eine entscheidende Anforderung in diesem Zusammenhang ist die Reduzierung von Entwicklungszeiten und -kosten. Um dies zu erreichen sind beim Entwurf Probleme, die bei der Herstellung entstehen könnten, rechtzeitig zu erkennen und zu lösen. Wird das Problem erst bei oder nach der Produktion erkannt, so ist viel Zeit und Geld nutzlos investiert worden (siehe Abbildung 3.16).

Beim Entwurf ist der Herstellungsprozess so zu planen, dass in Abhängigkeit von den Parametern Zeit und Geld der Fertigungsgegenstand optimiert zu fertigen

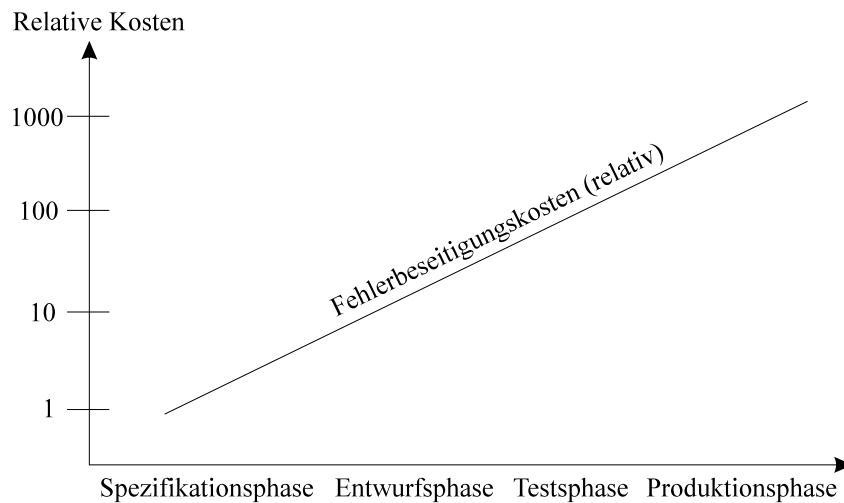


Abbildung 3.16: Relative Fehlerbeseitigungskosten bei der Entwicklung

ist. Hier geht es um die Abwägung, welche Materialien und Techniken bei der Fertigung einzusetzen sind (z.B. Selektion von Materialien unter Beachtung von Materialkosten).

Die Schlussfolgerung aus den oben beschriebenen Forderungen ist, dass der Verifikation eine entscheidende Rolle beim Entwurf zukommt. Sie repräsentiert die Aktivität beim Entwurfsprozess, welche in großem Maße Entwicklungszeiten und -kosten beeinflussen kann; positiv in dem Sinne, wenn sie vorhanden ist und eine hohe Intelligenz besitzt und negativ, wenn sie fehlt oder unterentwickelt ist.

Die Konsequenz für das Entwurfsmodell ist, dass der Verifikation stärkere Bedeutung zukommen muss. Abbildung 3.17 bringt dies zum Ausdruck.

Die Aktivität *Verifikation* bildet auf Grund ihrer Bedeutung eine separate Einheit im Entwurfsmodell, die entweder als Output das Entwurfsergebnis, also $L_{out,d}^2$ und $L_{out,r}^2$, d.h. die Fertigungsspezifikation, liefert oder die gefundenen Defekte an die zu einer Einheit zusammengefassten Aktivitäten *Entscheidungen treffen / Modifikationen festlegen* meldet. Sind Entscheidungen getroffen und die dazugehörigen Modifikationsanweisungen formuliert worden, ist durch die Stufe *Generierende Aktivitäten* die aktuelle Entwurfsbeschreibung zu überarbeiten und anschließend wieder zu prüfen.

Durch die Subaktivität *Entscheidung* ist ein Mechanismus vorhanden, der Aspekte wie Zuverlässigkeit des Entwurfsergebnisses, fehlerfreie Verfügbarkeit aller spezifizierten Funktionalitäten und eine lange Lebensdauer des Fertigungsgegenstandes steuern kann, indem entschieden wird, Restriktionen zu verschärfen

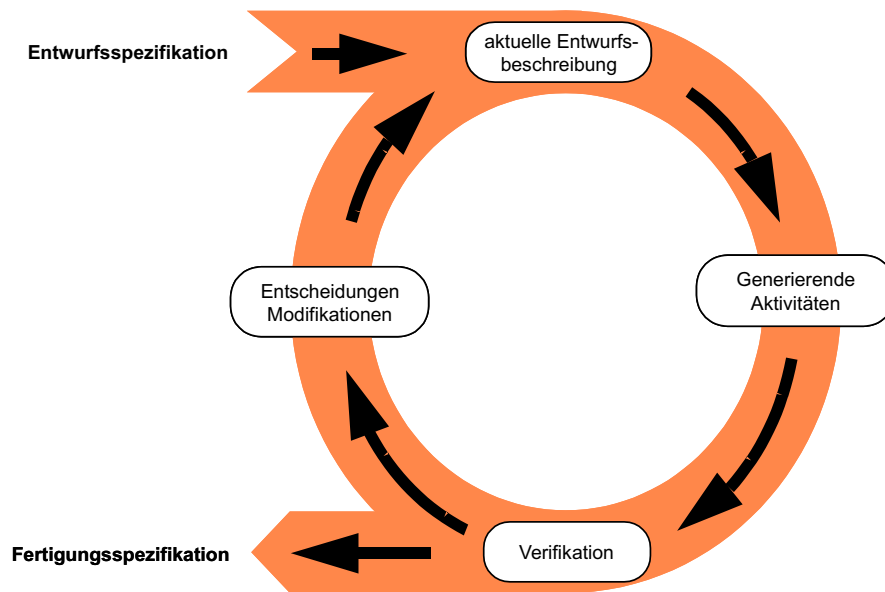


Abbildung 3.17: Kreismodell

oder neue einzuführen, oder indem festgelegt wird, welche Techniken und Materialien am besten einzusetzen sind.

Die Einheit *aktuelle Entwurfsbeschreibung* enthält die Daten, welche beim Modelldurchlauf zu bearbeiten sind. Hiermit wird explizit auf die mögliche Unvollständigkeit der Spezifikation des Entwurfsgegenstandes hingewiesen. Start- und Stop-Kriterien für den Entwurf sind durch die Eingabe der Spezifikation des Entwurfsgegenstandes und die Ausgabe der vollständigen Fertigungsspezifikation festgelegt.

Ein weitere Änderung am Modell ist durch die Darstellung als Zyklus (Kreis) erfolgt. Dadurch soll zum Ausdruck kommen, dass die Iteration (mehrere Durchläufe pro Prozess) ein bedeutsames Element beim Entwurf ist. Das Entwurfsmodell in Abbildung 3.17 wird im weiteren Verlauf der Arbeit als **Kreismodell** bezeichnet.

Die Idee des Kreismodells wurde von Brück und Schumer in *An Object-oriented Approach to Physical Microcomponent Design* vorgestellt [BS95]. Das Kreismodell visualisiert, dass kein sequentieller Prozess vorliegt, sondern ein Wechselspiel von generierenden und prüfenden Aktivitäten. So wird versucht, die Fertigungsspezifikation schrittweise und nachvollziehbar zu vervollständigen. Abbildung 3.18 verdeutlicht wie, ausgehend von einer Initialen unvollständigen Version der Entwurfsspezifikation, mit Hilfe des Kreismodells eine Fertigungsspezifikation zu erstellen ist. Es werden verschiedene Zwischenversionen

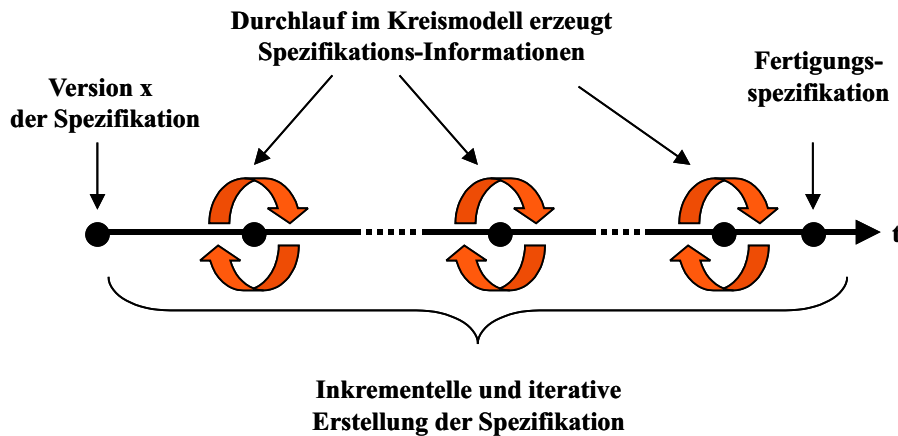


Abbildung 3.18: Spezifikationserstellung

der Spezifikation erstellt (schwarze Punkte auf der Zeitachse). Ein Durchlauf im Kreismodell liefert die dazu notwendigen Spezifikations-Informationen. Der Vorgang ist beendet, wenn ausreichend Informationen in der Spezifikation enthalten sind um den Fertigungsprozess autonom durchzuführen.

Das vorgestellte Modell erlaubt eine sehr grobe Analyse eines Entwurfsprozesses. Ziel war es zu motivieren, welche grundsätzlichen Aufgaben beim Entwurf zu bewältigen sind und dafür eine Reihenfolge zu definieren. Das Modell ist zu diesem Zeitpunkt noch unabhängig bezüglich des Entwurfsgegenstandes. Eine Ausprägung des Entwurfsmodells speziell für Mikrostrukturen und MEMS erfolgt im weiteren Verlauf der Arbeit (siehe Abschnitt 4.1 auf Seite 51).

3.6 Das Y-Modell der Mikroelektronik

Der Aufbau eines Entwurfsmodells für Mikrostrukturen und insbesondere für Mikrosysteme ist für die Mikrotechnik-Industrie von großer Bedeutung. Durch ein Entwurfsmodell wird eine Entwurfsmethodik festgelegt, die es ermöglicht Entwurfsaufgaben zu identifizieren. Entwurfsaufgaben teilen den Entwurfsprozess in überschaubare und beherrschbare Teile auf. Ihre Komplexität ist im Vergleich zum Gesamtprozess reduziert und es besteht die Möglichkeit, einzelne Entwurfswerkzeuge für die Mikrotechnik abzuleiten. Solche Werkzeuge sollen den Experten entlasten, indem sie gewisse Tätigkeiten automatisieren. Der Experte bleibt aber weiterhin mit seinem Expertenwissen die entscheidende Person beim Entwurfsprozess. Wie im vorherigen Abschnitt erläutert, wird er beispielsweise benötigt, um die Entwurfsentscheidungen zu treffen, die sich

aus der Unvollständigkeit der Spezifikation ergeben. Ziel eines Entwurfsmodells ist auch, den Arbeitsprozess des menschlichen Experten zu strukturieren und nachvollziehbar zu gestalten.

Zur Zeit existiert kein allgemein akzeptiertes Entwurfsmodell für Mikrosysteme wie etwa das **Y-Modell** [WT85] in der Mikroelektronik. Der Entwurfsgegenstand wird beim Y-Modell auf unterschiedlichen, hierarchisch strukturierten Ebenen repräsentiert: **Systemebene**, **Algorithmische Ebene**, **Register-Transfer-Ebene**, **Gatterebene**, **Schaltkreisebene**. Jede Ebene ist eine Abstraktionsstufe, die nur die relevanten Eigenschaften des Entwurfsgegenstandes der jeweiligen Ebene darstellt.

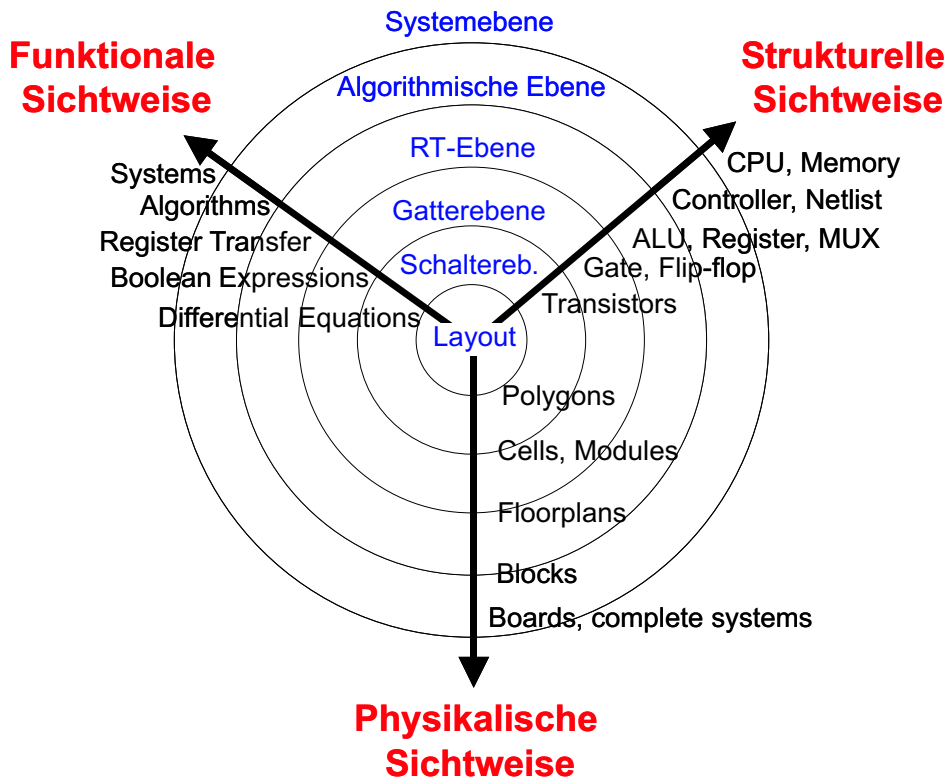


Abbildung 3.19: Y-Modell

Um den Entwurfsprozess weiter zu strukturieren wird der Entwurfsgegenstand unter drei verschiedenen Sichtweisen betrachtet: funktionale Sichtweise, strukturelle Sichtweise und physikalische Sichtweise. Die **funktionale Sichtweise** beschreibt das Verhalten (z.B.: algorithmisches Verhalten, Zeitverhalten, elektrisches Verhalten) einer Schaltung. Innerhalb der **strukturellen Sichtweise** wird eine Schaltung durch ihre Bauteile und die sie verbindenden Netze dargestellt. Die **Bauteile** einer Schaltung sind zum Beispiel Transistoren, logische

Gatter oder Prozessoren. Jedes Bauteil besitzt Anschlüsse, über die es mit anderen Bauteilen verbunden werden kann. Die Verbindungen zwischen den Bauteilen werden **Netze** genannt. Die **physikalische Sichtweise** betrachtet die konkrete Implementation des Entwurfsgegenstandes. Die Bauteile und Netze werden mit Hilfe realer physikalischer Objekte realisiert. Hierbei werden die genauen Abmessungen und Anordnungen aller Bauteile und Netze festgelegt. Abbildung 3.19 stellt die Abstraktionsebenen und Sichtweisen im Y-Modell grafisch dar.

Die Ebenen werden in der Grafik durch konzentrisch angeordnete Kreise repräsentiert, deren Abstraktionsgrad von außen nach innen abnimmt. Gleichzeitig erhöht sich damit der Detaillierungsgrad der Entwurfsbeschreibung. Den Mittelpunkt des Y-Modells bildet das **Entwurfsziel** mit der vollständigen Spezifikation (Fertigungsspezifikation). Sie beinhaltet das **Layout**, welches aus einer Menge von geometrischen Strukturen besteht, die als Vorlage für die Herstellung der integrierten Schaltung dienen.

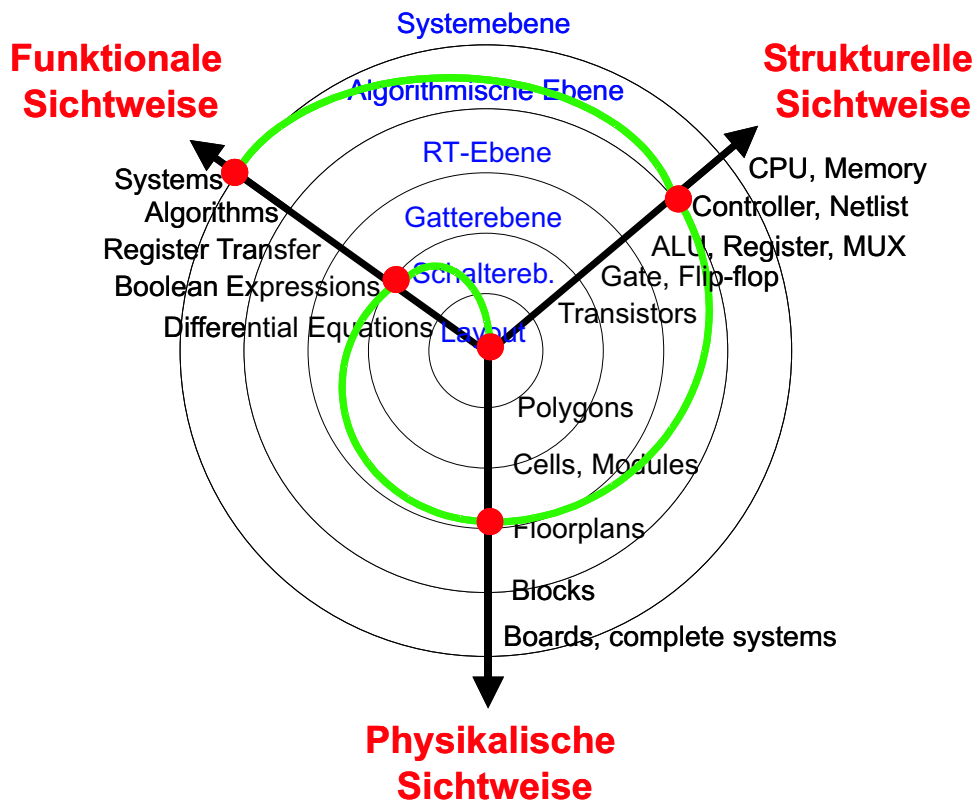


Abbildung 3.20: Beispiel für eine Entwurfsmethodik

Jede Sichtweise ist im Y-Modell durch eine aus dem Mittelpunkt entspringende Achse grafisch dargestellt. Die Schnittstellen der Achsen mit den Ebenen repräsentieren eine bestimmte Darstellungsform der Entwurfsbeschreibung in Abhängigkeit von der Abstraktionsebene und Sichtweise. Zum Beispiel werden als Darstellungsform für die Entwurfsbeschreibung in dem Schnittpunkt der funktionalen Sichtweise mit der Gatterebene Boolesche Gleichungen oder endliche Automaten verwendet.

Das Y-Modell erlaubt es, verschiedene konkrete Entwurfsmethoden abzuleiten. Ein Schnittpunkt zwischen einem der konzentrischen Kreise und einer der Achsen identifiziert einen **Entwurfzustand** im Y-Modell. Als **Entwurfsschritt** wird die Abbildung zwischen zwei Entwurfzuständen bezeichnet. Eine Folge von Entwurfzuständen, die beim Entwurfsziel endet, heißt **Entwurfsmethode**.

Im Y-Modell repräsentiert der Entwurfzustand „funktionale Sicht - Systemebene“ meist den Beginn des Entwurfs und verläuft von da aus spiralförmig auf den Mittelpunkt zu. Abbildung 3.20 gibt ein Beispiel für eine mögliche Entwurfsmethode eines mikroelektronischen Systems. Ein Entwurfsschritt in Richtung zum Mittelpunkt ist eine generierende Aktivität, und ein Entwurfsschritt in Richtung Systemebene ist eine überprüfende Aktivität. Die generierende Aktivität spezifiziert die Entwurfsbeschreibung genauer, indem sie neue Informationen einfügt, und die überprüfende Aktivität ermöglicht, die Korrektheit einer abgeschlossenen generierenden Aktivität festzustellen.

Die Disziplin der **Entwurfsautomatisierung** oder **Design Automation - DA** beschäftigt sich mit der Entwicklung von Algorithmen zur möglichst weitgehenden Automatisierung von generierenden und überprüfenden Aktivitäten. Solche Algorithmen werden dann in rechnerbasierten **Entwurfswerkzeugen** eingesetzt, um den Prozess des Entwurfs zu beschleunigen, qualitativ zu verbessern und kostengünstiger durchzuführen.

Unter Berücksichtigung der Erkenntnisse, die aus dem Y-Modell gewonnen wurden, wird das Kreismodell schrittweise auf die Bedürfnisse der Mikrotechnik angepasst. Die Termini Entwurfsmethode, Entwurfsschritt und Entwurfzustand werden analog zum Y-Modell für das Kreismodell eingeführt.

3.7 Erstellen einer Spezifikation

Der Abschnitt beschreibt, was unter der Spezifikation eines Entwurfsgegenstandes zu verstehen ist. Um eine hohe Praxisrelevanz zu gewährleisten, wird zu Be-

Die Konsequenz aus den oben identifizierten Informationen ist, dass die Restriktionen und die Spezifikation Informationen über das funktionale Verhalten, die Struktur und den Aufbau sowie den Fertigungsprozess enthalten. In Anbetracht der Erkenntnisse, die am Beispiel Chipkühlung gewonnen wurden, ist eine Verfeinerung der Definition der Termini Restriktion und Spezifikation möglich.

An dieser Stelle ist zu beachten, dass im Abschnitt 3.5 auf Seite 26 ein Entwurfsmodell beschrieben wurde, welches für eine Klasse von Gegenständen Gültigkeit besitzt. In diesem Kapitel wird das Entwurfsmodell auf einen konkreten Gegenstand (hier: Mikrostruktur) angewandt. Bedingt dadurch stehen jetzt konkrete entwurfsrelevante Informationen zur Verfügung (siehe Beispiel Chipkühlung), die eine präzisere Beschreibung des Kreismodells und der damit verbundenen Termini erlauben. Hierdurch sind Anpassungen im Modell und bei den Termini notwendig. Es wird zwischen drei Restriktionsarten unterschieden:

- $\mathbf{R}_{\text{in}}^{\text{V}}$ Restriktionen bezüglich Leistungsmerkmalen, die sich auf das funktionale Verhalten des Fertigungsgegenstandes beziehen
- $\mathbf{R}_{\text{in}}^{\text{S}}$ Restriktionen bezüglich Leistungsmerkmalen, die sich auf die Struktur und den Aufbau des Fertigungsgegenstandes beziehen
- $\mathbf{R}_{\text{in}}^{\text{P}}$ Restriktionen bezüglich Leistungsmerkmalen, die sich auf den Herstellungsprozess des Fertigungsgegenstandes beziehen

Die Entwurfsbeschreibung wird durch die Verhaltensbeschreibung, Strukturbeschreibung und physikalische Beschreibung ersetzt, die Entwurfsrestriktionen analog dazu durch die Verhaltensrestriktionen, Strukturrestriktionen und physikalischen Restriktionen. Eine Definition der Termini erfolgt im nächsten Abschnitt.

3.7.2 Verhaltensspezifikation

Die Termini Verhaltensbeschreibung, Verhaltensrestriktionen und Verhaltensspezifikation werden wie folgt definiert:

Definition: 15 *Eine Beschreibung, die festlegt, wie ein Entwurfsgegenstand auf äußere Einflüsse reagiert und welche interne Zustandsfolge er über die Zeit hinweg durchläuft, heißt **Verhaltensbeschreibung** $\mathbf{L}_{\text{in,d}}^{\text{V}}$. Restriktionen bezüglich des Verhaltens eines*

Entwurfsgegenstandes heißen **Verhaltensrestriktionen** $L_{in,r}^V$. Eine Spezifikation, die eine Verhaltensbeschreibung und Verhaltensrestriktionen für einen Entwurfsgegenstand enthält, heißt **Verhaltensspezifikation** L_{in}^V . //

Die Verhalten-Spezifikation L_{in}^V wird in der Spezifikationsphase 1 unter Beachtung der Restriktionen R_{in}^V erstellt. Danach sind die Restriktionen $L_{in,r}^V$ aus $L_{in,d}^V$ zu extrahieren. Abbildung 3.22 liefert einen Überblick über diesen Vorgang.

$$R_{in}^V \rightarrow \boxed{\text{Spezifikationsphase 1}} \rightarrow L_{in,d}^V \cup L_{in,r}^V$$

Abbildung 3.22: Spezifikationsphase 1

Die Verhaltensbeschreibung $L_{in,d}^V$ legt fest, wie sich der Entwurfsgegenstand über die Zeit hinweg verhält. Dies geschieht durch ein Portfolio von charakterisierenden Variablen und deren Werteverläufen über die Zeitachse. Mit Hilfe dieser Variablen werden sowohl die Werteverläufe an den Eingabe-/Ausgabeschnittstellen des Entwurfsgegenstandes als auch die internen Zustandsfolgen beschrieben [Ram91].

Die Grafik in Abbildung 3.23 stellt die hier beschriebenen Zusammenhänge beispielhaft dar. Die Variablen V1, V2, V3 beschreiben die Eingabeschnittstelle und die Variablen V8, V9, V10 die Ausgangsschnittstelle. Die internen Zustände werden durch die Variablen V4, V5, V6 und V7 dokumentiert.

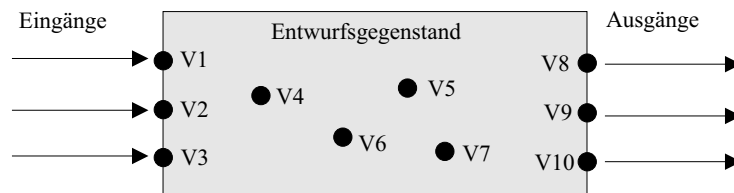


Abbildung 3.23: Charakterisierende Variablen

Jeder Variable ist eine physikalische Bedeutung bezüglich des Entwurfsgegenstandes zugeordnet. Dies erfordert vom Experten tiefgreifende Kenntnisse über die physikalischen Zusammenhänge innerhalb des Entwurfsgegenstandes. Mit Hilfe von mathematischen Gleichungen werden die Werteverläufe der charakterisierenden Variablen über die Zeitachse modelliert.

Ein neuer Funktionswert wird immer dann berechnet, wenn ein Signalwechsel an einem beliebigen Eingang des Entwurfsgegenstandes stattfindet. Die Argumente einer zu berechnenden Funktion setzen sich aus dem aktuellen internen

Zustand des Entwurfsgegenstandes und den aktuellen Werten an den Eingängen zusammen. Als Ergebnis liefert die Funktion den neuen internen Zustand des Entwurfsgegenstandes und die Werte an den Ausgängen.

Abbildung 3.24 repräsentiert die Wärmeaufnahmefähigkeit der Chipkühlung in Abhängigkeit von der Wärmeabgabe des Mikrochips.

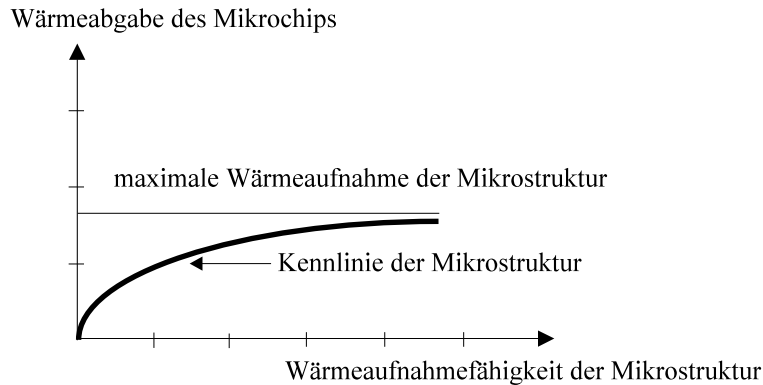


Abbildung 3.24: Wärmeaufnahmefähigkeit der Chipkühlung

Eine solche grafische Darstellung heißt **Kennlinie**. Sie repräsentiert eine Kurve in einem n -dimensionalen Koordinatensystem, die eine grafische Darstellung des Zusammenhangs zwischen technisch wichtigen Größen einer Mikrostruktur widerspiegelt. Hierfür wird eine besonders interessante Größe (unabhängige Variable) der Mikrostruktur als Funktion einer oder mehrerer anderer dargestellt (abhängige Variablen). Ein Wert für die unabhängige Variable ist durch äußere Bedingungen festgelegt. Er entspricht mindestens einem Wert der abhängigen Variablen und damit einem Punkt der Kennlinie, dem so genannten **Arbeitspunkt**. Kennlinien ermöglichen das Verhalten einer Mikrostruktur unter wechselnden äußeren Bedingungen grafisch darzustellen. Sie helfen dem Entwickler, eine erste Vorstellung des Verhaltens einer Mikrostruktur zu entwickeln.

In Abbildung 3.24 ist zu erkennen, dass die Wärmeaufnahmefähigkeit mit steigender Wärmeabgabe des Mikrochips langsam sinkt und schließlich einen Grenzwert erreicht, von dem an keine weitere Wärmeaufnahme möglich ist. Eine solche Kennlinie ergänzt um physikalische Einheiten und Skalierung kann als Restriktion R_{in}^V dienen. Die Arbeitspunkte sind bei der Erstellung der Verhaltensbeschreibung $L_{in,d}^V$ einzuhalten.

Für das Beispiel Chipkühlung enthält die Gruppe *interne Zustände* = {Wärmeabgabe Prozessor, Wärmeabgabe anderer Bauteile, Materialeigenschaften, etc.}, obwohl die Kennlinie in Abbildung 3.24 nur die Wärmeabgabe des Prozessors berücksichtigt und damit die Situation idealisiert darstellt.

Eine Idealisierung ist sowohl an dieser Stelle als auch allgemein bei Untersuchungen von Vorgängen in der Natur notwendig. Diese sind so vielgestaltig und unübersichtlich, dass der Mensch nicht fähig ist, sie exakt zu erfassen. Wird die Chipkühlung unter diesem Aspekt betrachtet, so wird deutlich, dass nicht nur die Wärmeabgabe des Mikrochips eine Rolle spielt, sondern auch indirekt die Wärmeabgabe aller Bauteile, die im Rechnersystem eingebaut sind. Des Weiteren ist zu untersuchen, inwieweit Luftfeuchtigkeit oder andere äußere Umstände die Materialeigenschaften der Mikrostruktur beeinflussen und damit auch ihre Funktionalität. Dem Menschen fällt es schwer, diesen Gesamtvorgang zu verstehen. Durch Idealisierung und Schematisierung werden daher einige Ursachen extrahiert, die ausreichend erscheinen, um das funktionale Verhalten des Entwurfsgegenstandes zu beschreiben. Inwieweit eine Idealisierung möglich ist ohne die Randbedingungen R_{in}^V zu verletzen, ist später durch die Aktivität *Verifikation* zu prüfen.

Neben der Idealisierung durch Eliminierung der Variablen besteht die Möglichkeit, durch Analogien eine Verhaltensbeschreibung zu finden. Es wird nach Analogien sowohl in dem betreffenden Wissensgebiet als auch in anderen Gebieten gesucht, um mit deren Hilfe Aussagen über das funktionale Verhalten der Mikrostruktur zu treffen. Hierbei wird eine Vereinfachung des Entwurfsgegenstandes durch idealisierende Maßnahmen vorgenommen, um die Übertragbarkeit des Problems auf eine ähnliche (bereits gelöste) Problemstellung zu erleichtern. Auch hier ist durch die Aktivität *Verifikation* zu prüfen, ob Restriktionen eingehalten wurden.

Fällt für die Variablen-Gruppe *interne Zustände* die Prämisse weg, dass jeder Variablen eine physikalische Bedeutung bezüglich des Entwurfsgegenstandes zugeordnet ist, wird versucht ausschließlich das Eingangs- und Ausgangsverhalten des Entwurfsgegenstandes zu modellieren. Die Betrachtung der internen physikalischen Zusammenhänge fällt weg und Kenntnisse in diesem Umfeld sind vom Experten nicht mehr vorzuweisen. Es ergibt sich eine Approximationsaufgabe, in der die Gruppe der internen Variablen genau so zu bestimmen ist, dass ein vorgegebenes E/A-Verhalten erfüllt wird.

3.7.3 Strukturspezifikation

Die Termini Strukturbeschreibung, Strukturrestriktionen und Strukturspezifikation werden wie folgt definiert:

Definition: 16 *Eine Beschreibung, welche eine grafische Darstellung vom Aufbau des Entwurfsgegenstandes, sowie die Topologie sei-*

Wird in dieser Spezifikationsphase kein CAD-Werkzeug eingesetzt, kann eine erste graphische Darstellung des Entwurfsgegenstandes in Form einer Skizze erfolgen. Abbildung 3.26 enthält eine Skizze für die Mikrochipkühlung. Die Skizze legt die Form sowie die Lage und Anordnung der Mikrostruktur-Bauteile (hier Kanäle) des Entwurfsgegenstandes fest.

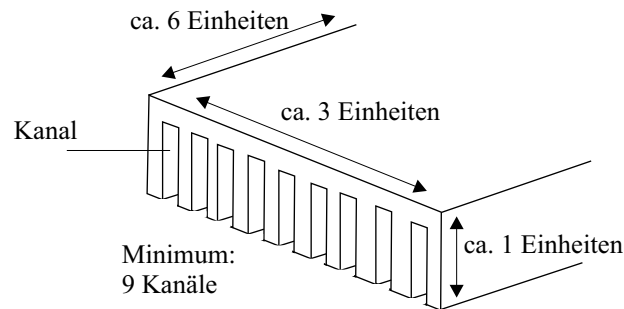


Abbildung 3.26: Form der Chipkühlung

Eine erste Version der Strukturspezifikation kann manchmal, nur gewisse Strukturrestriktionen $L_{in,r}^S$ enthalten (z.B. minimale Höhe der Chipkühlung, um die Kühlleistung zu erfüllen). Entscheidend ist, dass in diesem Fall $R_{in}^S \cup L_{in,r}^V \angle L_{in,r}^S$ gilt. Diese Restriktionen $L_{in,r}^S$ werden später im Strukturentwurf als Vorgaben für die Erstellung der Strukturbeschreibung verwendet. Hier wird als erstes versucht, auf Grundlage der Zeichnung ein **grafisches Modell** des Entwurfsgegenstandes zu erstellen. Dies kann Interaktiv unter Verwendung von kommerziellen CAD-Werkzeugen geschehen. Im nächsten Schritt erfolgt die Parametrisierung des grafischen Modells mit Hilfe der Prinzipskizze. Dabei werden die Parameter als variable Größen in das grafische Modell integriert. Ist dieser Vorgang abgeschlossen, ist ein **parametrisiertes grafisches Modell** des Entwurfsgegenstandes entstanden. Der Benutzer kann sich jetzt eine Vielzahl von grafischen Varianten des Entwurfsgegenstandes schnell und damit ohne großen Entwicklungsaufwand auf dem Bildschirm betrachten oder gegebenenfalls auf Papier ausdrucken.

Um den Wert eines Parameters exakt zu bestimmen sind **Simulationstechniken** einzusetzen [Mad01]. Sie erlauben es, die Auswirkungen einer Parameteränderung auf das Verhalten (z.B. Kennlinie) der Mikrostruktur darzustellen.

Möglicherweise existieren nicht nur verschiedene Varianten (d.h. Permutationen der Parameterwerte), sondern auch unterschiedliche Formen für den Entwurfsgegenstand, welche alle die Verhaltens- und Strukturrestriktionen erfüllen. Hier ist in Abhängigkeit von Produktionsverfahren und Aspekten wie Kosten und Zeitaufwand eine Entscheidung vom Experten bezüglich der Form zu fällen.

3.7.4 Physikalische Spezifikation

Die Termini physikalische Beschreibung, Restriktionen und Spezifikation werden wie folgt definiert:

Definition: 17 Eine Beschreibung, welche exakte Abmessungen des Entwurfsgegenstandes beinhaltet und festlegt, wie eine Fertigung des Entwurfsgegenstandes durchzuführen ist, heißt eine **physikalische Beschreibung** $L_{in,d}^P$. Restriktionen bezüglich der Abmessungen oder der Fertigung eines Entwurfsgegenstandes heißen **physikalische Restriktionen** $L_{in,r}^P$. Eine Spezifikation, die eine physikalische Beschreibung und physikalische Restriktionen für einen Entwurfsgegenstand enthält, heißt **physikalische Spezifikation** L_{in}^P . //

Die physikalische Beschreibung enthält Aussagen über die Art und Weise der Fertigung des Entwurfsgegenstandes. In diesem Zusammenhang besteht die Aufgabe darin, schrittweise einen Bauplan für den Entwurfsgegenstand zu finden. Dieser Bauplan dokumentiert detailliert die Vorgehensweise bei der konkreten Produktion und legt damit implizit sämtliche realisierungsrelevanten Eigenschaften des Fertigungsgegenstandes fest.

Bei der Generierung von $L_{in,d}^P$ sind sowohl die $L_{in,r}^S$ einzuhalten als auch die Kundenanforderungen in Form von R_{in}^P .

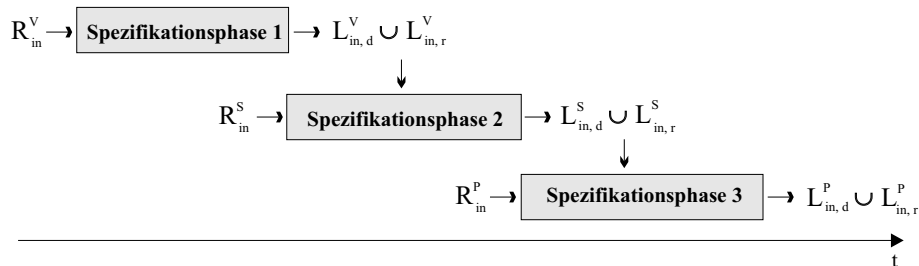


Abbildung 3.27: Spezifikationsphase 3

In der Spezifikationsphase enthält L_{in}^P eine erste noch unvollständige Version des Bauplans. Die Ursache dafür ist, dass in der Spezifikationsphase häufig Verhaltens- und Strukturspezifikationen nicht vollständig sind, sie aber eine notwendige Bedingung darstellen, um eine vollständige physikalische Spezifikation zu erstellen. Abbildung 3.27 gibt einen Überblick über die Spezifikationsphasen und beschreibt die zeitliche Reihenfolge in der die Phasen zu durchlaufen sind. Ein Sonderfall tritt auf, wenn mangelnde Informationen dazu führen, dass

$L_{in,d}^P = \emptyset$ ist. In diesem Fall ist während des Entwurfsprozesses $L_{in,d}^P$ so zu erstellen, dass mindestens R_{in}^P und $L_{in,r}^S$ dort implizit enthalten sind.

Die erste Information in der physikalischen Spezifikation bezieht sich meist auf das **Fertigungsverfahren** für den Entwurfsgegenstand. Hierzu ist beispielsweise auf den Entscheidungsbaum in Abbildung 3.28 zurückzugreifen.

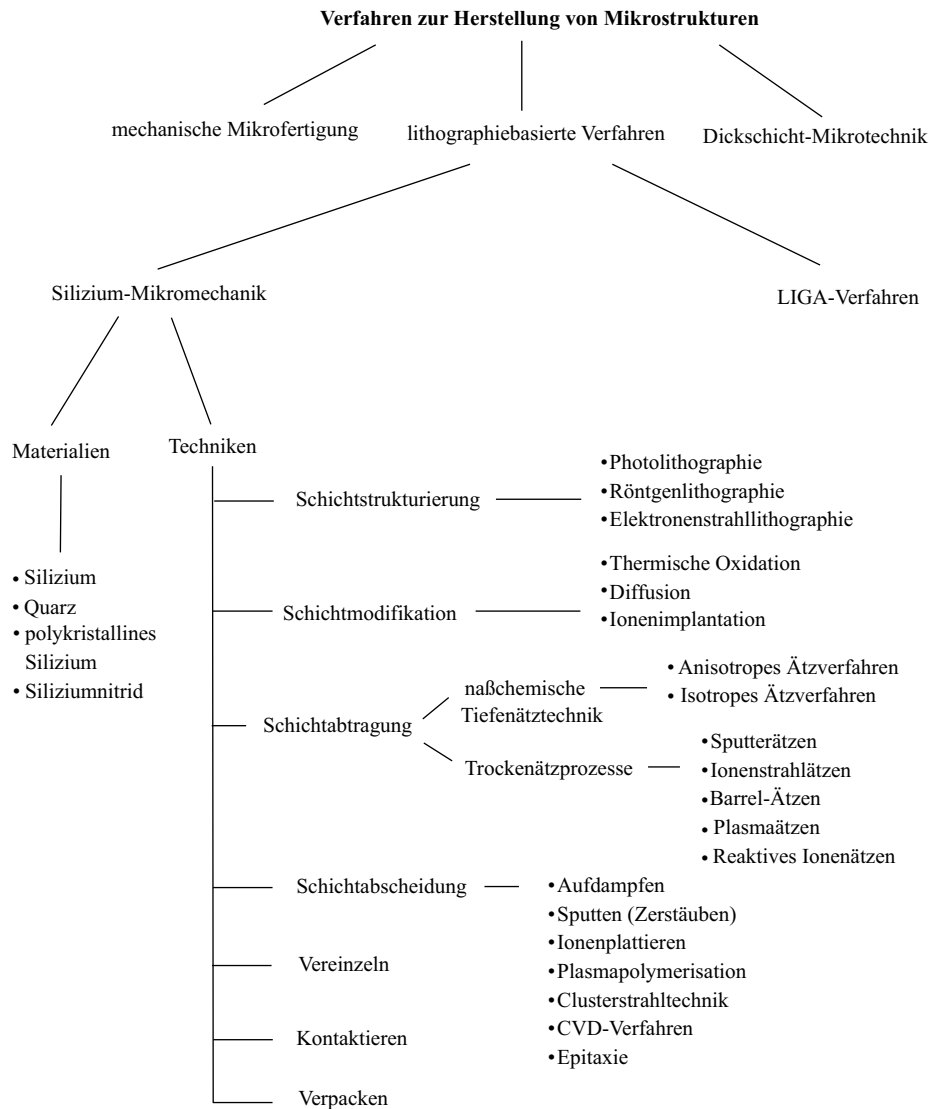


Abbildung 3.28: Entscheidungsbaum

Er gibt einen groben Überblick über die Fertigungsverfahren in der Mikrotechnik. Fällt die Entscheidung für die Silizium-Mikromechanik, ist damit festgelegt, welche Menge von **Materialien** und **Fertigungstechniken** bei der Produktion zur Verfügung stehen.

Da es technisch nicht möglich ist den Fertigungsgegenstand in einem Schritt herzustellen, wird als nächstes eine Menge von **Fertigungsprozessschritten** oder **Fertigungsschritten** definiert². Sie repräsentieren Teilaufgaben, die bei der Herstellung in einer bestimmten Reihenfolge auf dem Substrat durchzuführen sind. Sie teilen den Gesamtprozess der Fertigung in n Schritte auf. Zu jedem Fertigungsschritt gehört eine Fertigungstechnik (z.B. Lithographie, Ätzen oder Dotieren) und eine Menge von Materialien (z.B. Silizium, Quarz). Jeder Fertigungsschritt produziert dabei einen bestimmten Teil der Mikrostruktur. Alle Fertigungsschritte hintereinander angewandt erstellen die komplette Mikrostruktur. Damit wird deutlich warum die physikalische Spezifikation implizit die exakten Abmessungen des Fertigungsgegenstandes definiert.

Die Festlegung der Fertigungsschritte ist eine schwierige Aufgabe. Es stellt sich die Frage, wie viele Fertigungsschritte notwendig sind und wie diese jeweils zu konfigurieren sind um damit die dreidimensionale Mikrostruktur herzustellen. Dieses Problem wird nicht mehr in der Spezifikationsphase gelöst, sondern beim physikalischen Entwurf, der im Abschnitt 4.1 auf Seite 51 beschrieben ist.

Abbildung 3.29 repräsentiert eine Mikrostruktur, die auch als miniaturisierter Kühlkörper Verwendung finden könnte. Sie wurde mit Hilfe einer Ätztechnik produziert. Um eine solche Mikrostruktur zu produzieren, ist eine vollständige physikalische Spezifikation notwendig.

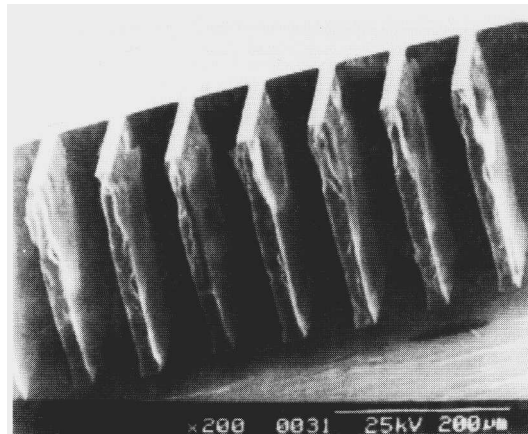


Abbildung 3.29: Beispiel für eine Mikrostruktur [Men97]

²Schichtstrukturierung, Schichtabtragung, Schichtmodifikation und Schichtabscheidung sind Beispiele für Fertigungsschritte in der Silizium-Mikromechanik Industrie. Aufbau- und Verbindungstechniken sowie die Reinigung des Substrats sind ebenfalls Fertigungsschritte und werden für den Herstellungsprozess eingesetzt [Mad01].

Die Definition des Begriffs Spezifikation wird unter Berücksichtigung der oben formulierten Definitionen neu festgelegt.

Definition: 18 *Unter einer Spezifikation L_{in}^{VSP} versteht man eine formale Beschreibung des Entwurfsgegenstandes (Verhaltens- und Strukturbeschreibung sowie physikalische Beschreibung). Des Weiteren beinhaltet die Spezifikation für den Entwurfsgegenstand relevante Restriktionen (Verhaltens- und Strukturrestriktionen sowie physikalische Restriktionen). //*

Daraus folgt, dass in der Spezifikation für einen Entwurfsgegenstand eine Verhaltens- und Strukturspezifikation sowie eine physikalische Spezifikation enthalten ist. Abbildung 3.30 verdeutlicht den Aufbau der Spezifikation L_{in}^{VSP} .

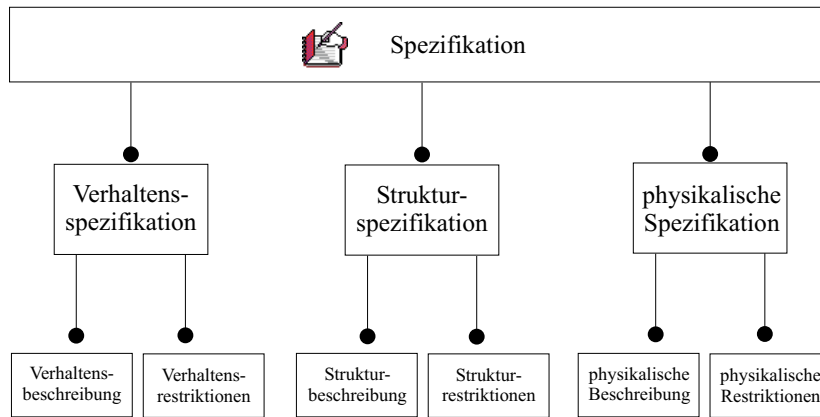


Abbildung 3.30: Aufbau der Spezifikation

Die Linien mit dem ausgefüllten Kreis an dem Ende repräsentieren eine Eigentums-Beziehung. Damit soll zum Ausdruck gebracht werden, dass beispielsweise die Strukturbeschreibung in der Strukturspezifikation enthalten ist und diese wiederum in der Spezifikation L_{in}^{VSP} . Zwischen den Beschreibungen und Restriktionen bestehen daher folgende Beziehungen:

$$\begin{aligned}
 L_{in}^1 &= L_{in}^{VSP} & := & L_{in}^V \cup L_{in}^S \cup L_{in}^P \\
 L_{in}^V & & := & L_{in,d}^V \cup L_{in,r}^V \\
 L_{in}^S & & := & L_{in,d}^S \cup L_{in,r}^S \\
 L_{in}^P & & := & L_{in,d}^P \cup L_{in,r}^P
 \end{aligned}$$

Nachdem die Eingabemenge für den Entwurfsprozess in einem höheren Detaillierungsgrad vorliegt, bedarf es demzufolge auch einer Verfeinerung des Entwurfsprozesses. Der nächste Abschnitt beschäftigt sich mit dieser Thematik.

3.8 Modellbildung in der Mikrotechnik

Es liegt nahe, den Entwurfsprozess in Teilprozesse aufzugliedern. Jeder Teilprozess erhält einen bestimmten Teil der Spezifikation als Eingabe. Dadurch wird die Gesamtkomplexität des Entwurfs reduziert und eine Spezialisierung auf bestimmte Teilaufgaben ist möglich. Der Gesamtprozess wird in drei semi-autonome Prozesse aufgeteilt.

Definition: 19 *Unter einem **semi-autonomen Prozess** versteht man einen Prozess, der nur teilweise ohne Eingriff eines menschlichen Experten abläuft. Bestimmte Prozessschritte sind nur mit Unterstützung eines menschlichen Experten durchzuführen.* //

Die semi-autonomen Prozesse³ kommunizieren über Schnittstellen miteinander und stehen hierarchisch miteinander in Beziehung. Die Prozesse heißen:

- **Verhaltensentwurf**
- **Strukturentwurf**
- **physikalischer Entwurf**⁴

Abbildung 3.31 legt eine Reihenfolge für die Teilprozesse des Entwurfs fest und verdeutlicht den Unterschied zwischen Entwurfs- und Entwicklungsprozess.

Der **Entwicklungsprozess** beinhaltet zusätzlich zum Entwurfsprozess zum einen die Erstellung der Spezifikation und des Weiteren den Fertigungsprozess. Durch die Produktidee wird der Entwicklungsprozess ins Leben gerufen. Er endet nach dem Abschluss des Fertigungsprozesses und liefert schließlich als Resultat die gewünschte Mikrostruktur.

Jeder der Teilprozesse des Entwurfs erhält zu Beginn Startinformationen und erzeugt am Ende seiner Laufzeit Ergebnisdaten. Diese dienen wiederum als Eingabe für den nächsten Prozess. Jeder Prozess deckt dabei einen Teil des Bedarfs an Informationskonzentration und -graduierung beim Entwurf einer Mikrostruktur ab. Das bedeutet, dass jeder Prozess einen bestimmten Teil an

³Der im Abschnitt 3.2 auf Seite 15 beschriebene Entwurfsprozess ist ein Beispiel für einen semi-autonomen Prozess, da in diesem Fall der Experte eine Entwurfsentscheidung treffen muss.

⁴Hahn schlägt in seiner Dissertation [Hah99] hier die Verwendung des Begriffs physischer Entwurf vor. Aufgrund der in Fachkreisen üblichen Konvention wird jedoch der gängigeren Bezeichnung physikalischer Entwurf hier der Vorzug gegeben.

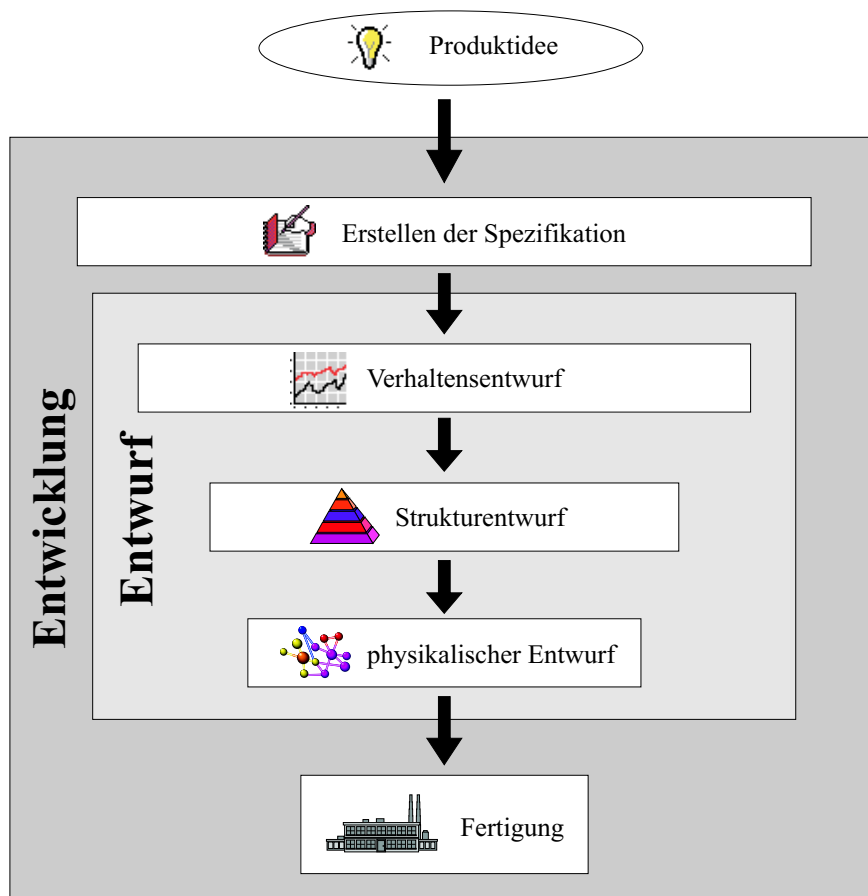


Abbildung 3.31: Entwicklung einer Mikrostruktur

Informationen liefert, welche für die Erstellung einer vollständigen Fertigungsspezifikation benötigt werden. Dabei ist zwischen den Abstraktionsebenen zu unterscheiden, auf denen man die Daten beschreibt. Der Verhaltensentwurf befindet sich auf einer hohen Abstraktionsebene, der physikalische Entwurf auf einer niedrigen, fertigungsnahen Ebene. Am Schluss dieser Prozesskette ist mit Hilfe der Fertigungsspezifikation die Produktion des Fertigungsgegenstandes anzustoßen.

Die Teilprozesse des Entwurfs haben einen ähnlichen internen Ablauf. Jeder Teilprozess besteht aus generierenden und überprüfenden Aktivitäten, die wie im vorherigen Kapitel beschrieben, durch Iterationen schrittweise versuchen, ein Ergebnis zu finden. Welche Funktionalität die generierenden und überprüfenden Aktivitäten jedes Teilprozesses in Abhängigkeit von L_{in}^V , L_{in}^S und L_{in}^P konkret realisieren, wird im einzelnen später untersucht. Zusätzlich bedarf es einer Untersuchung, wie die Teilprozesse miteinander kommunizieren. Hier wird eine bidirektionale Kommunikation gefordert, um beispielsweise Probleme beim

physikalischen Entwurf, die ihren Ursprung in vorgelagerten Teilprozessen haben, zu lösen.

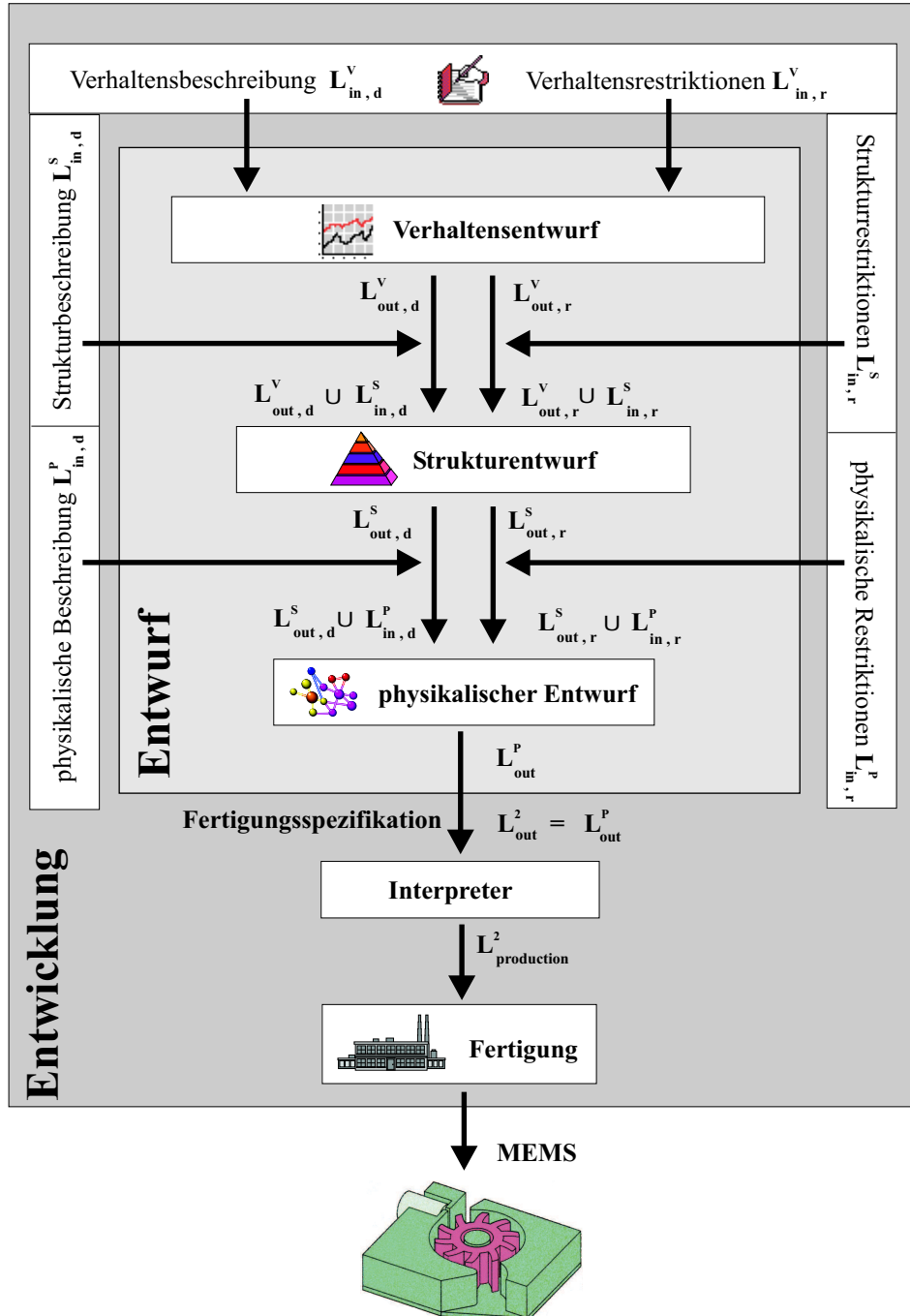


Abbildung 3.32: Spezifikation und Entwurfsprozess

Die Spezifikation L_{in}^{VSP} beeinflusst und steuert die Teilprozesse des Entwurfs. Abbildung 3.32 verdeutlicht, an welchen Stellen des Entwurfs die einzelnen Spezifikationen eingehen.

Die erste Stufe beim Entwurf ist der Verhaltensentwurf. Er erhält die Verhaltensbeschreibung und die Verhaltensrestriktionen als Eingabe. Er liefert als Ausgabe die vollständige⁵ Verhaltensspezifikation in Form von $L_{out,d}^V$ und $L_{out,r}^V$. In den sich anschließenden Strukturentwurf gehen die jetzt vollständige Verhaltensspezifikation und die Strukturspezifikation in Gestalt von $L_{out,d}^V \cup L_{in,d}^S$ und $L_{out,r}^V \cup L_{in,r}^S$ ein. Als Ergebnis entsteht die vollständige Strukturspezifikation⁶ ($L_{out,d}^S$ und $L_{out,r}^S$).

Sind Verhaltens- und Strukturentwurf beendet, ist eine idealisierte Mikrostruktur entstanden. Idealisiert daher, da noch nicht festgelegt ist, wie der Bauplan für den Entwurfsgegenstand aussieht und weil zu untersuchen ist, ob bestimmte funktionale und strukturelle Eigenschaften der **idealisierten Mikrostruktur** durch ein adäquates Herstellungsverfahren in die Praxis umzusetzen sind. Diese Fragen sind durch den physikalischen Entwurf zu beantworten.

Die letzte Stufe beim Entwurf startet mit der vollständigen Strukturspezifikation, die noch um die physikalische Spezifikation zu ergänzen ist. Die Verhaltensspezifikation ist nicht zwingend notwendig, da sie bereits implizit in der Strukturspezifikation enthalten ist. Nach erfolgreichem Abschluss des physikalischen Entwurfs liegt das Ergebnis in Form von $L_{out,d}^P$ (physikalische Beschreibung) und $L_{out,r}^P$ (physikalische Restriktionen) vor.

Dieses Ergebnis spiegelt auch das Resultat des Gesamtentwurfs wider. Die vollständige⁷ physikalische Spezifikation des Entwurfsgegenstandes entspricht der Fertigungsspezifikation (siehe Abbildung 3.32). Das in der Definition des Terminus Entwurfsprozess formulierte Ziel: eine geschlossene, korrekte und vollständige Beschreibung zu erstellen, die ausreicht, den Fertigungsgegenstand in einer der Spezifikation genügenden Ausführung zu fertigen, ist erreicht und der Entwurfsprozess damit beendet. Aus der Black-Box-Darstellung in Abbildung 3.2 auf Seite 16 ist eine White-Box-Darstellung entstanden.

Aus Abbildung 3.32 folgt, dass die physikalische Beschreibung der Fertigungsbeschreibung entspricht ($L_{out,d}^P = L_{out,d}^2$). Dies kann in Abhängigkeit von den

⁵Eine Verhaltensspezifikation ist genau dann vollständig, wenn sie alle notwendigen Eingabeinformationen für die Durchführung des sich anschließenden Strukturentwurfs enthält.

⁶Eine Strukturspezifikation ist genau dann vollständig, wenn sie alle notwendigen Eingabeinformationen für die Durchführung des sich anschließenden physikalischen Entwurfs enthält.

⁷Eine physikalische Spezifikation ist genau dann vollständig, wenn sie alle Informationen für die komplette Durchführung des Fertigungsprozesses enthält.

Anforderungen eines Unternehmens an die Fertigungsbeschreibung manchmal nicht so gewünscht sein. Hier ist gegebenenfalls noch ein „Interpreter“ einzusetzen, der $L_{out,d}^P$ in eine den unternehmensspezifischen Anforderungen entsprechende Fertigungsbeschreibung übersetzt. Die Anforderungen können sein, Arbeitspläne und Anweisungen für Verantwortliche der Produktion zu erstellen. Diese enthalten dann eine detaillierte Charakterisierung der Tätigkeiten, die bei einer Durchführung der Produktion zu erledigen sind (z.B. welche technischen Geräten auszuwählen sind und welche Einstellung vorzunehmen ist). In diesem Zusammenhang ist eine Verknüpfung von unternehmensspezifischen Daten wie vorhandene Ressourcen (Geräte, Materialien, Experten) für die Produktion von Mikrostrukturen und der physikalischen Beschreibung denkbar. Hierdurch ist es möglich eine Fertigungsbeschreibung zu erstellen, welche die Produktionsumgebung für Mikrostrukturen oder Mikrosystemen des Unternehmens berücksichtigt. Ein solcher „Interpreter“ kann als Softwarelösung realisiert werden, der als Eingabe L_{out}^2 erhält und als Resultat $L_{production}^2$ liefert. $L_{production}^2$ enthält dann zum Beispiel konkrete Steueranweisungen für die Produktionsmaschinen. Er ist über eine Konfigurationsdatei mit den unternehmensspezifischen Produktionsumgebungsdaten zu versorgen.

Gleichzeitig ist eine geräteunabhängige Spezifikationsbeschreibungssprache zu definieren. Sie ermöglicht es eine konkrete physikalische Spezifikation für einen Entwurfsgegenstand zu formulieren. Zur Zeit existiert keine Spezifikationsbeschreibungssprache für die Mikrotechnik, die als de facto Standard anzusehen wäre. Ein „Interpreter“ könnte daher beispielsweise, die am Institut für Mikrotechnik der Universität Siegen entwickelte physikalische Spezifikationsbeschreibungssprache **PDML** unterstützen. Die **Process Description Markup Language** ist eine auf XML basierende Sprache, die alle fertigungsrelevanten Aspekte der Mikrotechnik beinhaltet. Eine PDML-Sprachbeschreibung erfolgt in [Kle02].

Mit Abbildung 3.32 wurde ein Überblick über den Ablauf und die Informationsverarbeitung beim Entwurf von Mikrostrukturen und Mikrosystemen gegeben. Zu kritisieren ist der lineare Ablauf des Entwurfprozesses ohne Möglichkeiten zu besitzen, mit vorgelagerten Entwurfsphasen zu kommunizieren. Ein weiterer Kritikpunkt ist, dass die einzelnen Entwurfsphasen durch eine Black-Box repräsentiert wurden und die jeweiligen Aufgaben und Aktivitäten nicht transparent dargestellt sind.

Auf Basis dieser Erkenntnisse wird ein verbessertes Gesamtmodell für den Mikrostruktur-Entwurf aufgestellt. Zur Visualisierung der Aktivitäten beim Verhaltens-, Struktur- und physikalischen Entwurf kommt das Kreismodell zum

Einsatz. Es bildet den kleinsten gemeinsamen Nenner dieser drei Entwurfsphasen. Mit Hilfe des Kreismodells werden die Aufgaben in den drei Phasen des Entwurfsprozesses eindeutig identifiziert. Jede Phase ist ein inkrementeller und iterativer Vorgang. Sie besteht aus drei Aktivitäten, die mehrmals durchlaufen werden, bis ein vollständiges Ergebnis existiert.

Abbildung 3.33 zeigt das Kreismodell für den Entwurf von Mikrostrukturen und Mikrosystemen, in das eine Rückkopplung zu der jeweils vorgelagerten Entwurfsphase integriert wurde. Die Kommunikation zu der im Modell vorgelagerten Entwurfsphase ist somit explizit berücksichtigt. In Situationen, in denen kein Entwurfsergebnis in einer Entwurfsphase erzeugt werden kann, sind in den höher gelegenen Entwurfsphasen Anforderungen/Restriktionen zu entschärfen. Hierzu übergibt die aktive Entwurfsphase die Restriktionen, die verletzt wurden, über die Rückkopplung an die vorherige Entwurfsphase. Sie überarbeitet die entsprechende Entwurfsspezifikation und aktiviert dann wieder die nachfolgende Entwurfsphase.

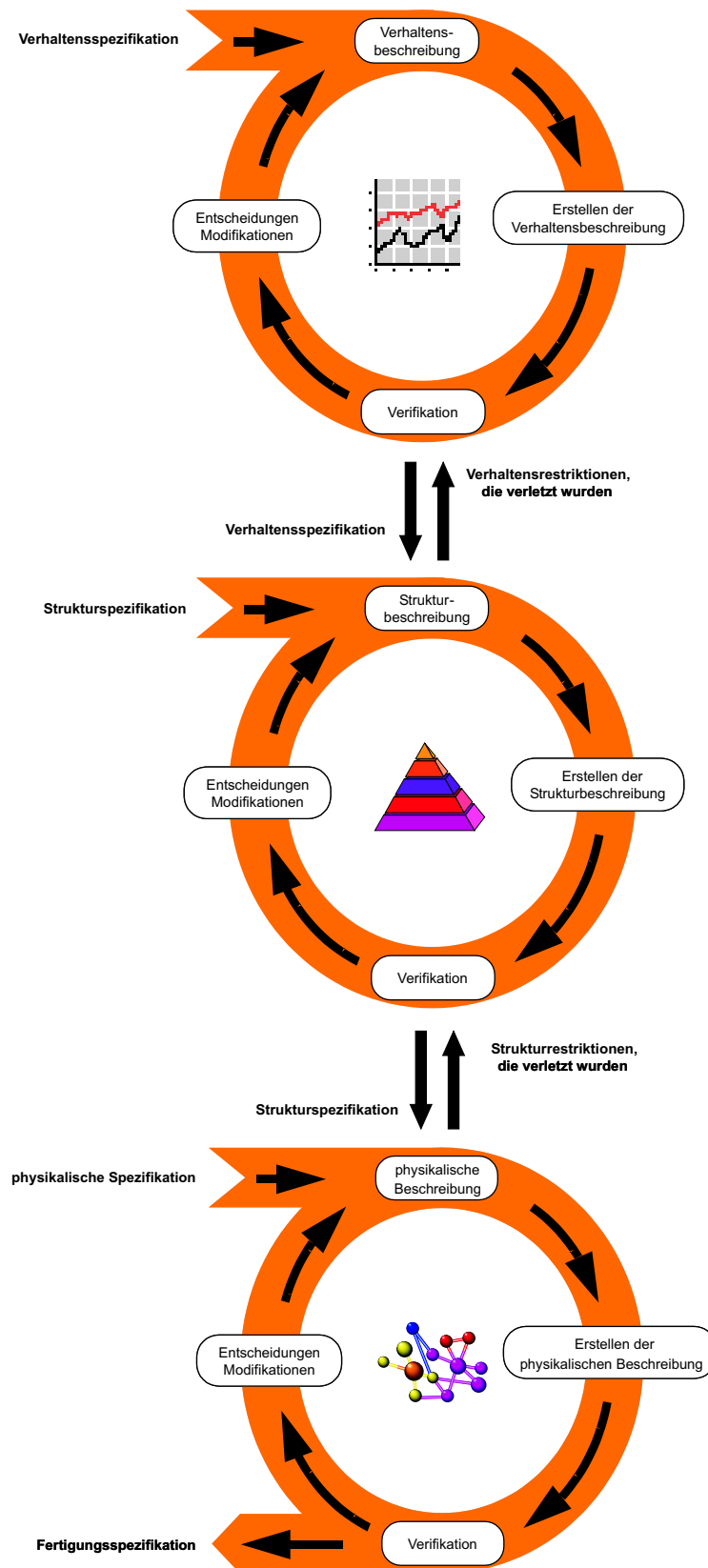


Abbildung 3.33: Gesamtmodell mit Rückkopplung

Kapitel 4

Design Flow in der Mikrotechnik

Ziel des letzten Kapitels war es, einen groben Überblick über die vorgelagerten Entwurfsphasen zu geben, um so ein besseres Verständnis für die Positionierung des physikalischen Entwurfs im Rahmen des Entwurfsprozesses zu entwickeln. In diesem Kapitel erfolgt eine detaillierte Untersuchung des physikalischen Entwurfs von Mikrostrukturen und Mikrosystemen. Eine neue Entwurfsmethodik wird vorgestellt, um Entwurfsschritte und -Zustände systematisch herzuleiten. Die Suche nach einer präzisen Definition für den Begriff Design Flow steht im Mittelpunkt der in diesem Kapitel beschriebenen Forschungsaktivitäten. Damit soll die Grundlage gelegt werden, um später eine verbesserte IT basierte Entwurfsautomatisierung in der Mikrotechnik voranzutreiben.

4.1 Kreismodell für den physikalischen Entwurf

Der physikalische Entwurfsprozess beschäftigt sich mit den fertigungsnahen Aspekten des Entwurfs und versucht den Bauplan für den Entwurfsgegenstand zu finden. Der Begriff Bauplan ist gleichzusetzen mit der vollständigen physikalischen Beschreibung, die festlegt, wie eine Produktion durchzuführen ist und dadurch implizit die exakten Abmessungen des Entwurfsgegenstandes beinhaltet. Sie enthält Informationen über die Materialien und Fertigungstechniken, die bei der Herstellung zu verwenden sind und beschreibt, auf welche Art und Weise diese Fertigungstechniken zu benutzen sind. Im Gegensatz zu der bisherigen grafisch/visuellen Sichtweise wird durch den physikalischen Entwurf eine maschinennahe/technische Sicht festgelegt. Es liegt ein sequentieller Prozess vor,

der mit Hilfe von Maschinen die Produktion des Entwurfsgegenstandes erlaubt.

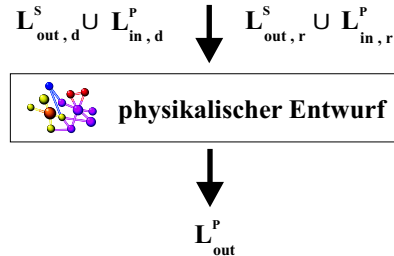


Abbildung 4.1: Physikalischer Entwurf

Abbildung 4.1 verdeutlicht die Ausgangsbedingungen beim physikalischen Entwurf mit Hilfe einer Black-Box-Darstellung. Zwei Spezifikationen dienen als Informationsquellen für den physikalischen Entwurf. Am Ende der Entwurfsphase liegt die vollständige physikalische Spezifikation als Resultat vor.

Im Folgenden werden die Aktivitäten beim physikalischen Entwurf genauer beleuchtet und die Vorgehensweise für die Erstellung einer Fertigungsbeschreibung wird konkreter erläutert. Im Gesamtmodell mit Rückkopplung ist der physikalische Entwurf bereits durch das Kreismodell dargestellt worden (siehe Abbildung 3.33 auf Seite 50). Das Kreismodell für den physikalischen Entwurf identifiziert drei Aktivitäten, welche solange zyklisch zu durchlaufen sind bis eine Entscheidung darüber zu treffen ist, ob eine vollständige physikalische Spezifikation erstellt werden kann. Folgende Aktivitäten sind im einzelnen von Bedeutung:

- Erstellen der physikalischen Beschreibung durch generierende Aktivitäten, die schrittweise das vorläufige Entwurfsergebnis erzeugen
- Verifikation der physikalischen Beschreibung durch prüfende Aktivitäten, die das vorläufige Entwurfsergebnis auf Korrektheit untersuchen
- Entscheidungen/Modifikationen bezüglich der physikalischen Beschreibung treffen oder festlegen durch Aktivitäten, die zum einen das weitere Vorgehen beim physikalischen Entwurf dokumentieren und bei Restriktionsverletzungen entsprechende Korrekturanweisungen erstellen

Jede Aktivität im Kreismodell ist durch ein Symbol skizziert. Die Pfeile zwischen den Symbolen verdeutlichen in welcher Reihenfolge die Aktivitäten zu erledigen sind. Als Eingabemenge für den physikalischen Entwurfsprozess dient die Strukturspezifikation L_{out}^S und die physikalische Spezifikation L_{in}^P . Der

Strukturentwurfsprozess liefert die vollständige Strukturspezifikation als Entwurfsteilergebnis. Sie enthält die Strukturbeschreibung $L_{out,d}^S$ mit der graphischen Darstellung vom Aufbau der Mikrostruktur sowie die Strukturrestriktionen $L_{out,r}^S$ (z.B. maximale Höhe der Mikrostruktur), die vom physikalischen Entwurfsprozess zu berücksichtigen sind.

Die physikalische Spezifikation besteht aus der physikalischen Beschreibung $L_{in,d}^P$, die festlegt, wie eine Fertigung des Mikrosystems oder der Mikrostruktur durchzuführen ist, und den physikalischen Restriktionen $L_{in,r}^P$, welche technologische, wirtschaftliche oder zeitliche Randbedingungen darstellen, die beim Entwurf zu beachten sind. Die physikalische Beschreibung besteht zu Beginn des Entwurfs aus einer ersten noch unvollständigen Version des Bauplans der Mikrostruktur. Sie enthält beispielsweise das Fertigungsverfahren und damit verbunden auch die Materialien sowie die Fertigungstechniken, welche für die Produktion zur Auswahl stehen.

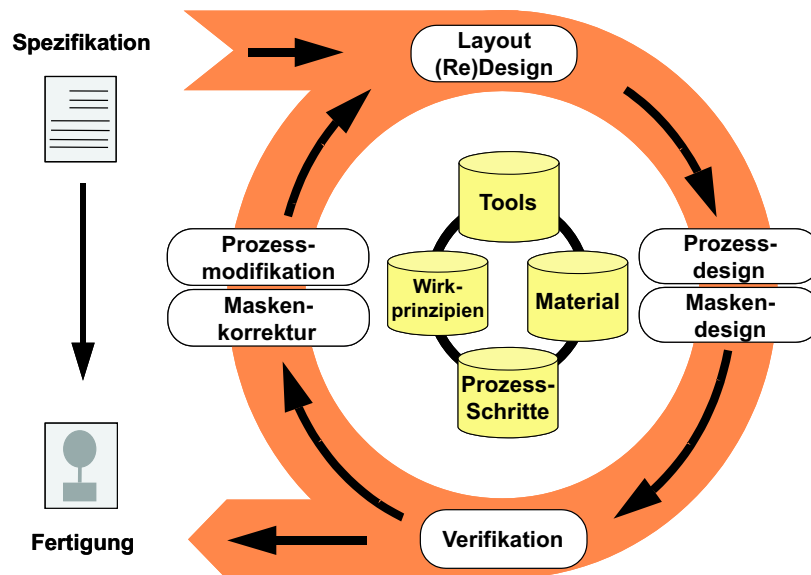


Abbildung 4.2: Erweitertes Kreismodell für den physikalischen Entwurf

Das Kreismodell für den physikalischen Entwurf von mikrotechnischen Bauteilen ist am Institut für Mikrosystemtechnik im Laufe der letzten Jahre schrittweise weiterentwickelt worden [Sch02b]. Die aktuelle Ausprägung des Modells ist von Hahn und Wagener in [HW03] veröffentlicht.

Abbildung 4.2 zeigt den letzten Entwicklungsstand des Kreismodells. Es wurde speziell auf den physikalischen Entwurf für die lithographiebasierte Mikrotechnik angepasst. Dem Maskenentwurf fällt bei der lithographiebasierten Mikro-

technik eine wichtige Rolle zu. Daher wurden explizit die Aktivitäten *Maskendesign* und *Maskenkorrektur* im Kreismodell zusätzlich benannt. Gleichzeitig wurde der Begriff *Layout* neu in das Modell aufgenommen, um damit zum Ausdruck zu bringen, dass auch ein Layout schrittweise (nebenläufig zu den Fertigungsschritten) durch Durchläufe im Kreismodell zu erstellen ist. Das Wort *physikalische Beschreibung* aus der Ursprungsversion des Kreismodells wurde durch den Begriff *Layout/Re(Design)* ersetzt. Hiermit sollen die bei jedem Iterationsschritt vorgenommenen Veränderungen an der physikalischen Beschreibung (d.h., Masken- und Fertigungsschrittänderungen) besser zum Ausdruck kommen. Die generierende Aktivität *Prozessdesign* legt den Fertigungsprozess, also eine Folge von Fertigungsschritten mit den entsprechenden Konfigurationen der Fertigungstechniken, Masken und Materialien, fest. Hierbei ist auf für die Mikrotechnik geeignete Materialien, Wirkprinzipien, vorkonfigurierte Fertigungsschritte und wenn vorhanden auch auf Tools zuzugreifen. Diese sind durch die Objekte in der Mitte des Kreismodells dargestellt. Eine ausführliche Beschreibung des Kreismodells für den physikalischen Entwurf in der Mikrotechnik erfolgt in [Hah99].

4.2 UML-Aktivitätsdiagramm

Um über das Kreismodell hinaus mehr Transparenz und eine genauere Darstellung des physikalischen Entwurfs zu ermöglichen, werden die Entwurfsaktivitäten mit Hilfe der **Unified Modeling Language (UML)** beschrieben. UML ist eine Notation und Sprache zur Spezifikation, Visualisierung und Dokumentation von Modellen für Softwaresysteme [Oes01], die aber auch geeignet ist, ein Modell für den physikalischen Entwurf zu beschreiben. Mit Hilfe von UML besteht die Möglichkeit, Subaktivitäten der drei zentralen Aktivitäten des Kreismodells zu dokumentieren. Das UML-Aktivitätsdiagramm für den physikalischen Entwurf ist in Abbildung 4.3 zu sehen.

Der physikalische Entwurf beginnt mit der Aktivität *Initiale Generierung der physikalischen Beschreibung*, welche die Strukturspezifikation sowie die noch unvollständige physikalische Spezifikation als Eingabeinformation erhält. Sie legt fest, dass die Fertigungsbeschreibung für eine Mikrostruktur in Abhängigkeit von der Strukturbeschreibung und dem gewählten Fertigungsverfahren zu bestimmen ist. Eine Analyse der Strukturbeschreibung ist zu Beginn notwendig, um zu erkennen, welche **Teilstrukturen** auf dem Substrat zu erzeugen sind.

Da es technisch nicht möglich ist eine Mikrostruktur in einem Schritt herzustellen, ist als nächstes eine Menge von Fertigungsschritten zu definieren. Sie

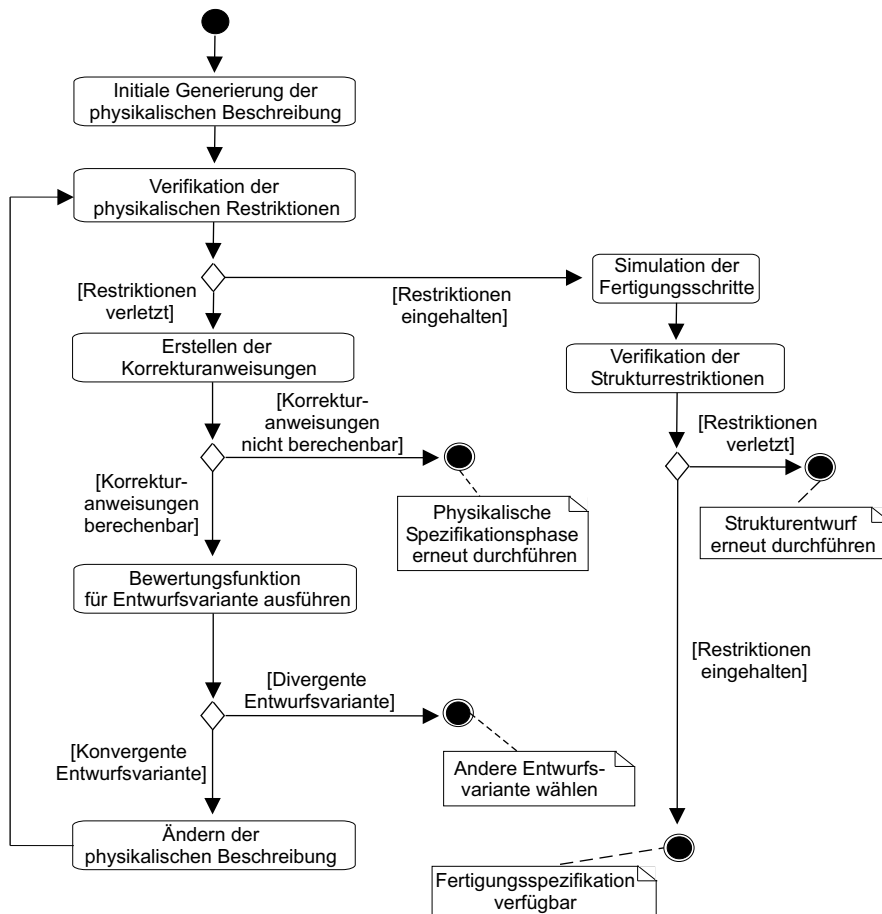


Abbildung 4.3: UML-Aktivitätsdiagramm für den physikalischen Entwurf

repräsentieren Teilaufgaben, die bei der Herstellung in einer bestimmten Reihenfolge auf dem Substrat durchzuführen sind und teilen den Gesamtprozess der Fertigung in n Schritte auf.

Zu beachten ist, dass manchmal Fertigungsschritte notwendig sind, um gewisse Vorarbeiten durchzuführen (z.B. das Aufbringen einer Opferschicht), bevor die eigentliche Teilstruktur im nächsten Fertigungsschritt produziert wird. Eine 1:1-Zuordnung von Teilstrukturen und Fertigungsschritten ist nicht immer möglich, um diesen Anforderungen gerecht zu werden, wird der Terminus **Fertigungsgruppe** eingeführt. Eine Fertigungsgruppe repräsentiert eine Menge von Fertigungsschritten, die genau eine Teilstruktur des Entwurfsgegenstandes produziert. Abbildung 4.4 verdeutlicht die Zerlegung des Entwurfsgegenstandes und die Generierung von Fertigungsschritten grafisch.

Die Reihenfolge FS_1 bis FS_e von Fertigungsschritten ist zu bestimmen. Sie legt fest, wann welcher Fertigungsschritt auf das Substrat anzuwenden ist. Für

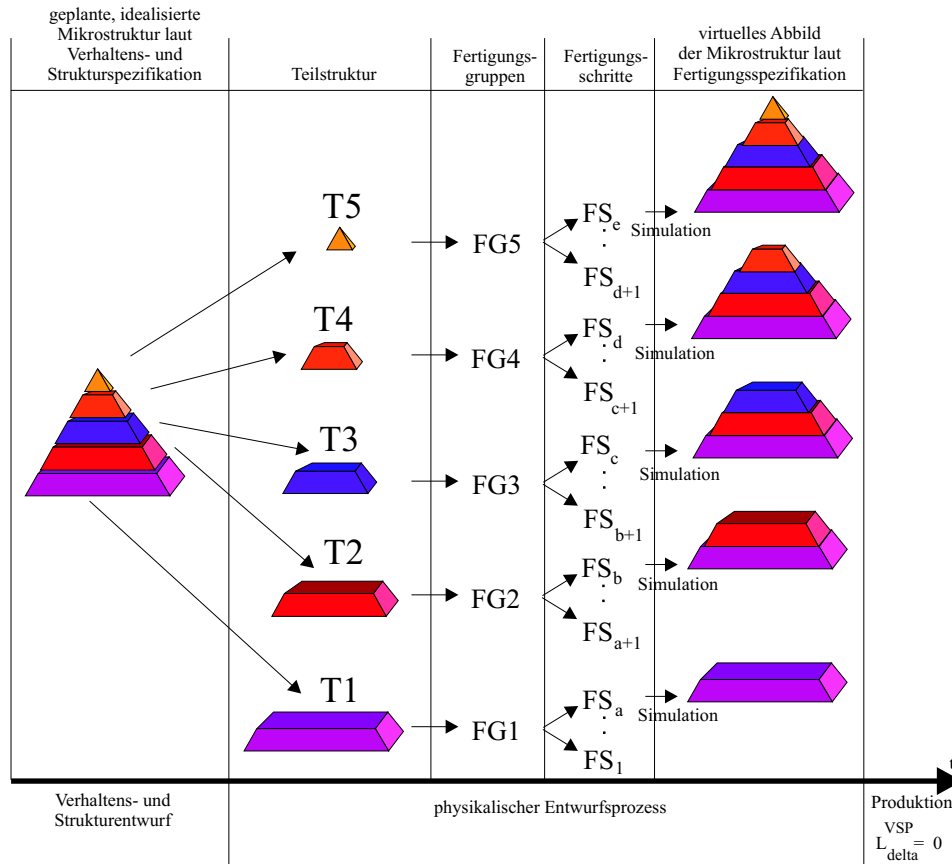


Abbildung 4.4: Zerlegung und Generierung

jeden Fertigungsschritt ist zu ermitteln, welche Fertigungstechnik, Materialien und Masken zu verwenden sind.

Zu jeder Fertigungstechnik gehört eine Menge von Fertigungsparametern. Beispielsweise gehört zu der Fertigungstechnik Lithographie eine Maske, die geometrische Informationen der Teilstruktur enthält und konkrete Parameter wie die Belichtungsdauer, den Wellenlängenbereich des Lichtes oder den Abstand der Maske zum Resist.

Für das Generieren der physikalischen Beschreibung wird eine Funktion f_{gen} gesucht, die als Ergebnis die vorläufige physikalische Beschreibung L_{check}^P liefert:

$$\boxed{\text{Initiale Generierung der physikalischen Beschreibung} \quad f_{gen}(L_{in}^P, L_{out}^S) \longrightarrow L_{check}^P}$$

Zur Zeit wird die Fertigungsbeschreibung durch den Experten weitgehend manuell erstellt. Mit seinem Expertenwissen legt er Fertigungsschritte und die dazugehörigen Parameter, Materialien und Fertigungstechniken fest. Dieser sehr

komplexe Vorgang gestaltet sich um so schwieriger, wenn Randbedingungen existieren, die beispielsweise fordern, dass ein gewisses Budget oder ein Produktionszeitraum nicht zu überschreiten sind.

Die Aktivität *Verifikation der physikalischen Beschreibung* im UML-Aktivitätsdiagramm erhält das vorläufige Entwurfsergebnis L_{check}^P und die physikalischen Restriktionen $L_{in,r}^P$ als Eingabe, um damit eine formale Prüfung durchzuführen, die als Resultat eine Liste von Restriktionsverletzungen liefert.

Um in der Mikroelektronik die Herstellbarkeit des Schaltungsentwurfs mit dem geplanten Fertigungsverfahren zu garantieren, werden Mikroelektronik-Layouts mit Hilfe der **Entwurfsregelprüfung** (**DRC = Design Rule Check**) auf die Einhaltung von technologiebedingten geometrischen Restriktionen überprüft [Brü93]. Anschließend wird die **Schaltungsextraktion** durchgeführt, die sich aus der **Bauteil-** und **Parameterextraktion** zusammensetzt. Die Parameterextraktion liefert elektrische Daten wie Kapazität, Transistorparameter oder Widerstände. Die Bauteilextraktion dient zur Identifizierung von Transistoren und den dazugehörigen Verbindungen. Damit wird aus dem Layout die zu realisierende Schaltung zurückgewonnen, um dann mit der Sollschaltung verglichen zu werden [Sch92]. Die Vorgehensweise beim physikalischen Schaltungsentwurf kann als Richtlinie für die Verifikation der physikalischen Beschreibung des Entwurfsgegenstandes dienen.

Bei der Verifikation in der Mikrotechnik ist zuerst auch eine Entwurfsregelprüfung mit Hilfe der physikalischen Restriktionen anzustoßen. Im Rahmen der Entwurfsregelprüfung ist zu untersuchen, inwieweit die Fertigungsschritte gewissen Randbedingungen genügen, die durch die Wahl des Fertigungsverfahrens bestimmt wurden. Dadurch ist die Verträglichkeit der einzelnen Fertigungsschritte bezüglich des Fertigungsverfahrens sichergestellt [BH97].

Die physikalischen Restriktionen sind in Form von **Fertigungsprozess-** und **Layoutregeln** beschrieben und sind vom Entwurfsgegenstand unabhängig, d.h. sie werden ausschließlich durch das eingesetzte Fertigungsverfahren und die damit verbundenen Fertigungstechniken bestimmt. Physikalische Restriktionen geben vor, was aus Sicht der Technologie heutzutage möglich ist, und prägen durch ihren Vorgabe-Charakter die Gestalt des Mikrosystems.

Fertigungsprozessregeln stellen sicher, dass die verwendeten Prozessparameter, Materialien und Maskenebenen eine in sich geschlossene, widerspruchsfreie Fertigungsschrittfolge bilden. Sie legen zudem die Reihenfolge, Konformitäten und Kompatibilitäten von Fertigungsschritten fest, indem sie bestimmen, welche Zusammenstellung von Fertigungsschritten erlaubt oder zu vermeiden ist.

Als Layoutregel werden die geometrischen Restriktionen bezeichnet, die das Maskenlayout betreffen. Sie bestehen sowohl aus den topologischen Regeln, welche die grundsätzliche Anordnung von Strukturelementen beschränken, als auch aus topografischen Regeln, welche quantitative Beziehungen wie Weiten und Abstände umfassen [BH96]. Layoutregeln legen fest, welche Maskenentwürfe technologische Gültigkeit besitzen und sind ausschließlich für lithografiebasierte Fertigungsverfahren zu verwenden.

Nachdem mit einem entsprechenden Algorithmus die Einhaltung von Fertigungsprozess- und Layoutregeln geprüft wurde, ist in Abhängigkeit vom Ergebnis die weitere Vorgehensweise zu entscheiden. Sind Regelverletzungen aufgetreten, werden die Abweichungen in Form von L_{delta}^P dokumentiert. Die Funktion f_{ver} beschreibt den Vorgang formal:

Verifikation der physikalischen Restriktionen	$f_{ver}(L_{check}^P, L_{in,r}^P) \longrightarrow L_{delta}^P$
--	--

L_{delta}^P beschreibt, welche Restriktion verletzt wurde und wie groß die Abweichung war. Diese Informationen werden an die Aktivität *Erstellen der Korrekturanweisungen* weitergeleitet und die Verifikation ist vorerst beendet.

Die Aktivität *Erstellen der Korrekturanweisungen* im UML-Aktivitätsdiagramm entscheidet zunächst, ob für $L_{delta}^P \neq \emptyset$ ein L_{korr}^P erstellt werden kann. Tritt der Fall ein, dass keine Korrekturanweisungen erzeugt werden können, wird L_{delta}^P und L_{check}^P an die physikalische Spezifikationsphase übergeben. Dort sind gegebenenfalls Restriktionen zu entschärfen oder ein anderes Fertigungsverfahren zu wählen. Im anderen Fall, werden entsprechende Korrekturanweisungen erzeugt. Hierzu wird die Funktion f_{mod} eingesetzt, die als Resultat (wenn berechenbar) L_{korr}^P liefert:

Erstellen der Korrekturanweisungen	$f_{mod}(L_{delta}^P, L_{check}^P) \longrightarrow L_{korr}^P$
---------------------------------------	--

L_{korr}^P wird an die Aktivität *Ändern der physikalischen Beschreibung* übergeben. Zuvor ist die Bewertungsfunktion an dieser Stelle einzusetzen, um die Qualität der Entwurfsvariante bezüglich der Konvergenz oder Divergenz zu beurteilen.

In der generierenden Aktivität *Ändern der physikalischen Beschreibung* im UML-Aktivitätsdiagramm wird mit Hilfe von f_{new} , L_{korr}^P und L_{check}^P ein neues vorläufiges Entwurfsergebnis $L_{check,x}^P$ erstellt:

Ändern der physikalischen Beschreibung	$f_{new}(L_{korr}^P, L_{check}^P) \longrightarrow L_{check,x}^P$
---	--

Die zuvor erläuterten Aktivitäten werden solange wiederholt bis keine Regelverletzungen mehr vorhanden sind und das aktuelle Entwurfsergebnis, bezogen auf das Fertigungsverfahren sowie die damit verbundenen physikalischen Restriktionen, korrekt ist (siehe Abbildung 4.3).

Nach der physikalischen Entwurfsregelprüfung erfolgt eine Simulation der Fertigungsschritte, um deren Auswirkungen auf das Substrat zu verifizieren. Wird die Simulation für alle Fertigungsschritte durchgeführt, d.h. der komplette Fertigungsprozess ist simuliert worden, liegt als Resultat ein virtuelles Abbild der zu produzierenden Mikrostruktur in Form von L_{check}^S vor. Hierfür wird eine Funktion f_{sim} benötigt, die mit Hilfe von L_{check}^P die Strukturbeschreibung L_{check}^S erzeugt. Die Funktion f_{sim} beschreibt den Vorgang formal:

Simulation der Fertigungsschritte	$f_{sim}(L_{check}^P) \longrightarrow L_{check}^S$
--------------------------------------	--

Die Strukturrestriktionen sind einzusetzen um zu prüfen, inwieweit das virtuelle Abbild der zu produzierenden Mikrostruktur L_{check}^S mit der idealisierten Mikrostruktur $L_{out,d}^S$ übereinstimmt. Die Verhaltensrestriktionen sind bereits implizit in der Strukturbeschreibung enthalten und sind daher an dieser Stelle nicht relevant. Ermittelt die Aktivität *Verifikation der Strukturrestriktionen* im UML-Aktivitätsdiagramm, dass die Sollvorgaben mit den Istwerten übereinstimmen, ist die physikalische Spezifikation vollständig und der physikalische Entwurf sowie der gesamte Entwurfsprozess beendet. Bei Differenzen zwischen den Soll- und Istwerten ist durch die Funktion f_{diff} die Datei L_{delta}^S mit den Abweichungen zu erstellen. Die Funktion f_{diff} beschreibt den Vorgang formal:

Verifikation der Strukturrestriktionen	$f_{diff}(L_{check}^S, L_{out,r}^S) \longrightarrow L_{delta}^S$
---	--

Für $L_{delta}^S \neq \emptyset$ ist der physikalische Entwurf beendet und der Strukturentwurf ist mit L_{delta}^S wieder zu aktivieren (siehe Abbildung 3.33: Gesamtmodell mit Rückkopplung). Seine Aufgabe ist es, eine modifizierte Strukturbeschreibung unter Beachtung der Deltas und der Verhaltensrestriktionen zu erzeugen und danach den physikalischen Entwurf erneut zu starten.

Stimmen die Sollvorgaben mit den Istwerten überein, ist die physikalische Spezifikation vollständig und identisch mit der gesuchten Fertigungsspezifikation. Der physikalische Entwurf sowie der gesamte Entwurfsprozess ist damit beendet.

```

00:   $L^P L_{in}^P = \{\text{physikalische Spezifikation}\}$  ;
01:   $L^P L_{check}^P = \{\text{vorläufige physikalische Beschreibung}\}$  ;
02:   $L^S L_{out}^S = \{\text{vollständige Strukturspezifikation}\}$  ;
03:  int bewertung[]; // Array für die Werte der Bewertungsfunktion
04:  int nummer = 0; // Zähler für die Entwurfsschritte
05:  entwurf = {konvergent, divergent}; // Datentyp definieren
06:  entwurf status; // Variable für die Konvergenz/Divergenz eines Prozesses
07:   $L_{check}^P = f_{gen}(L_{in}^P, L_{out}^S)$ ; // Initiales Erstellen der physikalischen Beschreibung
08:   $L_{delta}^P = f_{ver}(L_{check}^P, L_{in,r}^P)$ ; // Verifikation der physikalischen Beschreibung
09:  while (  $L_{delta}^P \neq \emptyset$  ) { // Vervollständigen der physikalischen Beschreibung
10:       $L_{korr}^P = f_{mod}(L_{delta}^P, L_{check}^P)$ ; // Korrekturanweisungen erstellen
11:      if (  $L_{korr}^P == \text{nicht berechenbar}$  ) { // Berechenbar?
12:          break Ende; // Aktueller Entwurfsprozess wird gestoppt
13:          // Änderungen in der physikalischen Spezifikation notwendig
14:          // Spezifikationsphase mit  $L_{delta}^P$  und  $L_{check}^P$  aktivieren
15:      }
16:      bewertung[nummer] =  $f(L_{korr}^P)$ ; // Bewertungsfunktion berechnen
17:      nummer = ++nummer; // Zähler erhöhen
18:      status = prüfe.entwurf(bewertung[]) // Bewertung der Variante
19:      if (status == divergent) // Nicht streng monoton fallend
20:          break Ende; // Andere Entwurfsvariante ausprobieren
21:       $L_{check}^P = f_{new}(L_{korr}^P, L_{check}^P)$ ; // Ändern der physikalischen Beschreibung
22:       $L_{delta}^P = f_{ver}(L_{check}^P, L_{in,r}^P)$ ; // Verifikation der physikalischen Beschreibung
23:  }
24:   $L_{check}^S = f_{sim}(L_{check}^P)$ ; // Simulation
25:   $L_{delta}^S = f_{diff}(L_{check}^S, L_{out,r}^S)$ ; // Soll/Istvergleich
26:  if (  $L_{delta}^S \neq \emptyset$  ) // Abweichungen?
27:      break Ende; // aktueller Entwurfsprozess wird gestoppt
28:      // Strukturrestriktionen sind zu entschärfen
29:      // Strukturentwurf mit  $L_{delta}^S$  aktivieren
30:  else
31:       $L_{out}^P = L_{check}^P$  ; // physikalische Spezifikation ist vollständig
32:  Ende:

```

Abbildung 4.5: Entwurfsmethodik für den physikalischen Entwurf

Im Mittelpunkt aller Aktivitäten steht der Experte. Er entscheidet, ob der physikalische Entwurf autonom fortzusetzen ist oder ob vorgelagerte Spezifikations- oder Entwurfsphasen zu konsolidieren sind. Der Experte legt fest, inwieweit

Differenzen im Rahmen von akzeptablen Fertigungstoleranzen liegen oder entscheidet, dass die Produktqualität nicht mehr ausreichend ist.

4.3 Entwurfsmethodik für den physikalischen Entwurf

Mit dem UML-Aktivitätsdiagramm sind die Aktivitäten beim Entwurfsprozess genauer und transparenter dargestellt worden als durch das Kreismodell. Um zum einen den Informationsfluss zwischen den Aktivitäten zu modellieren und zum anderen eine zur Entwurfsmethodik des Y-Modells vergleichbare Entwurfsmethodik zu erhalten, wird mit Hilfe einer fiktiven Programmiersprache der physikalische Entwurf beschrieben (siehe Abbildung 4.5).

Die Entwurfsmethodik legt eine Folge von Entwurfsschritten fest, die eine Fertigungsspezifikation L_{out}^P als Entwurfsergebnis liefert. Gleichzeitig beschreibt und steuert die Methodik den Datenstrom zur Erstellung der Fertigungsspezifikation beim Entwurfsprozess.

Zu Beginn des physikalischen Entwurfs ist ein Entwurfzustand durch ein Tupel identifiziert, das aus der Struktur- und physikalischen Spezifikation besteht:

$$\boxed{\begin{array}{l} \text{Initialer} \\ \text{Entwurfzustand} \end{array} \quad (L_{out}^S, L_{in}^P)}$$

In der vorläufigen physikalischen Spezifikation L_{in}^P ist mindestens das Fertigungsverfahren (z.B. LIGA-Technik) ausgewählt worden, mit dem der Fertigungsgegenstand später produziert wird. Für den konkreten Entwurf sind damit die Fertigungstechniken bekannt, die bei der Erstellung der Fertigungsspezifikation zu verwenden sind (siehe Abbildung 3.28 auf Seite 41).

Ein erster Entwurfsschritt ist durch die Entwurfsmethodik beschrieben durch:

$$\boxed{\begin{array}{l} \text{erster} \\ \text{Entwurfsschritt} \end{array} \quad f_{gen}(L_{in}^P, L_{out}^S) = L_{check}^P}$$

Als Funktionswert liegt mit L_{check}^P eine erste Version der Fertigungsbeschreibung vor.

$$\boxed{\begin{array}{l} \text{neuer} \\ \text{Entwurfzustand} \end{array} \quad L_{check}^P}$$

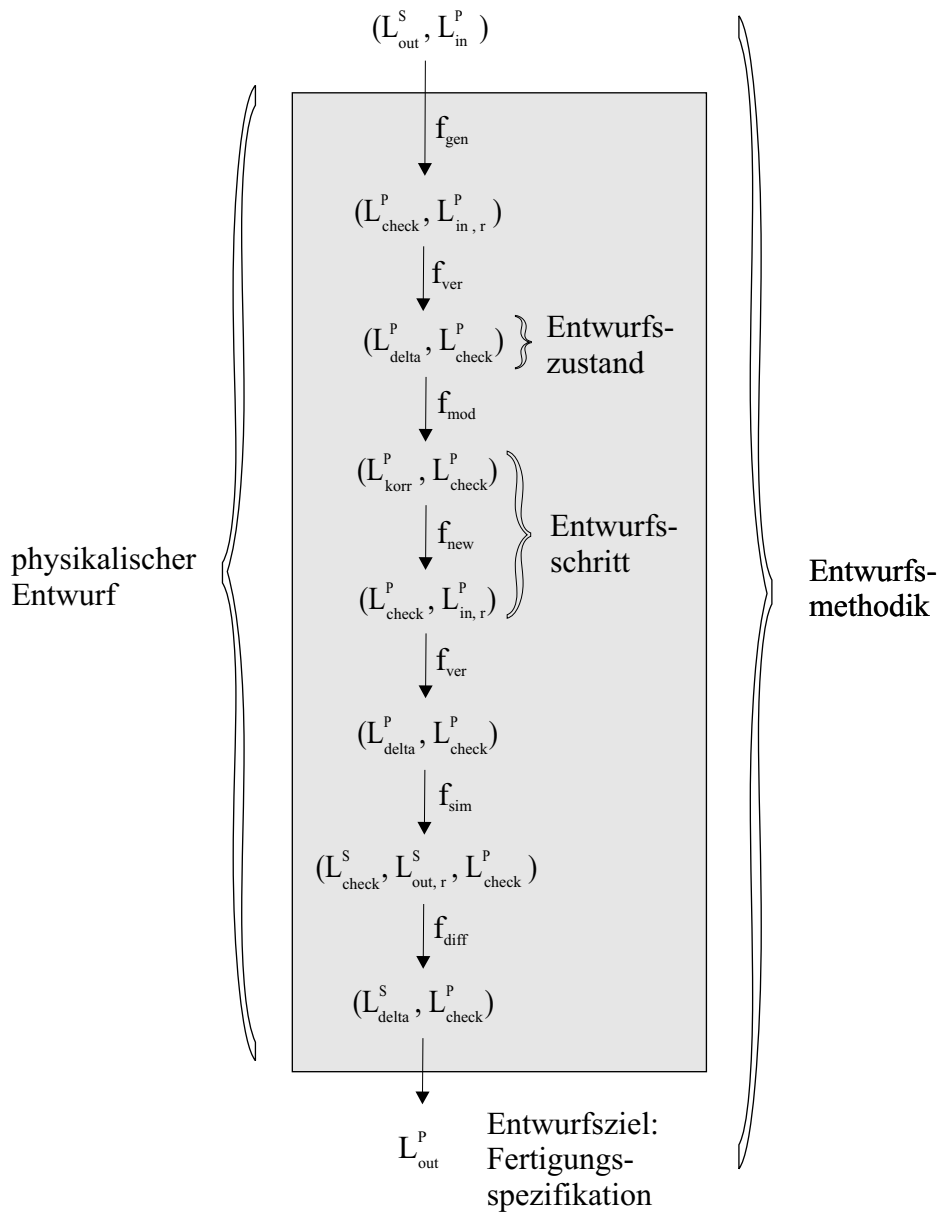


Abbildung 4.6: Ausprägung der Entwurfsmethodik

Der neue Entwurfzustand ist dann einer ersten Verifikation mit Hilfe der Funktion f_{ver} zu unterziehen. Die nächsten Entwurfsschritte leiten sich aus dem Algorithmus ab (siehe Abbildung 4.5).

Die Entwurfsmethodik greift in Abhängigkeit vom Entwurfzustand auf die Funktionen f_{gen} , f_{ver} , f_{sim} , f_{diff} , f_{mod} und f_{new} zu, um den nächsten Entwurfsschritt durchzuführen. Der Funktionswert der jeweiligen Funktion ist durch die Entwurfsmethodik eindeutig bestimmt. Zum Beispiel liefert die Abbildung f_{ver} die Abweichungen bezüglich der physikalischen Restriktionen in Form von

L_{delta}^P .

Abbildung 4.6 repräsentiert eine konkrete Ausprägung der Entwurfsmethodik unter der Annahme, dass nach zwei Verifikationsschritten keine Restriktionen mehr verletzt wurden, die Korrekturanweisungen berechenbar sind und eine konvergente Entwurfsvariante vorliegt. Die Entwurfzustände, Entwurfsschritte und das Entwurfsziel werden durch die Ausprägung der Entwurfsmethodik eindeutig beschrieben.

Die Herausforderung besteht darin, in Abhängigkeit vom Fertigungsverfahren eine geeignete Implementierung für die jeweilige Funktion zu finden. Gesucht wird quasi eine Methode im Sinne der objektorientierten Programmierung, welche als Rückgabewert, beispielsweise für die Abbildung f_{ver} , den Funktionswert L_{delta}^P liefert. Ein solche Methode implementiert die Abbildung, in dem es eine feste Folge von konkreten in sich abgeschlossenen **Entwurfsaufgaben (Design Tasks)** beschreibt. Zu jeder Entwurfsaufgabe ist eine Rolle zugeordnet, welche die Verhaltensweise und die Verantwortlichkeiten eines Experten festlegt (siehe Abbildung 4.7). Eine Folge von Entwurfsaufgaben, die auf dem Weg zum gesuchten Funktionswert zu erledigen sind, wird als **Task Flow** bezeichnet. Zur Durchführung dieser Entwurfsaufgaben greift der Experte auf Mikrotechnik **Design Services** zurück.

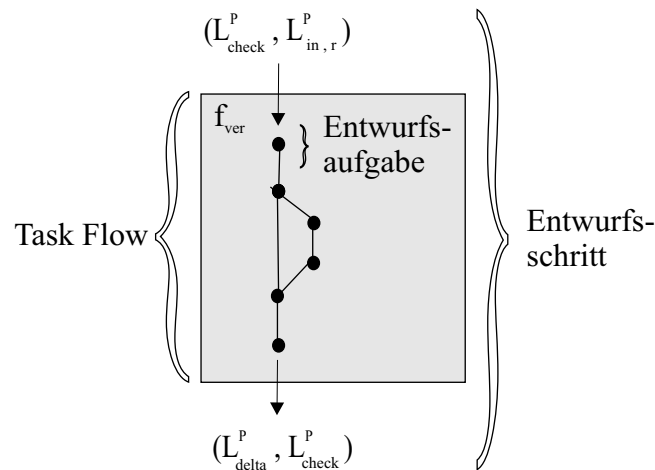


Abbildung 4.7: Beispiel für einen Entwurfsschritt und Task Flow

Unter Design Services sind Dienstleistungen im Umfeld der Mikrotechnik gemeint, die einen Experten bei der Durchführung der Entwurfsaufgabe unterstützen. Zum Beispiel repräsentieren schriftlich formulierte Arbeitsanweisungen, die dem Experten vorgeben, wie eine Entwurfsaufgabe zu erledigen ist, einen Design Service, genauso ein Entwurfswerkzeug, das die Entwurfsaufgabe

ohne Eingriff des Experten durchführt. Selbstverständlich sind auch klassische Dienstleistungen in Form von Beratung und Support durch menschliche Experten, als Design Services einzuordnen. An den Beispielen ist zu erkennen, dass verschiedene Arten der Implementierung einer Entwurfsaufgabe möglich sind.

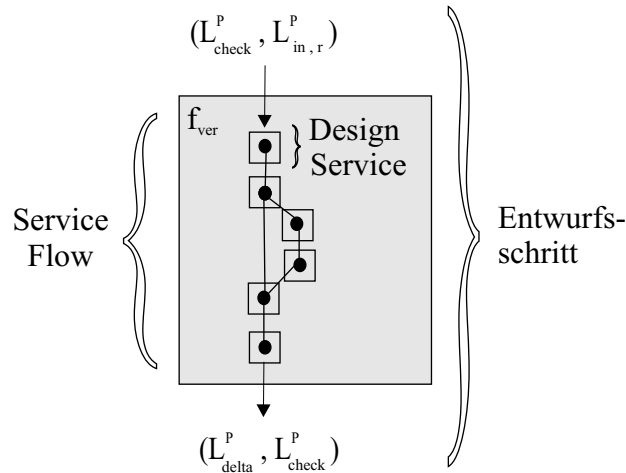


Abbildung 4.8: Beispiel für einen Service Flow

Ein Design Service repräsentiert eine neue Generation von Dienstleistung für den physikalischen Entwurf, die dem Experten bei der Bearbeitung einer Entwurfsaufgabe hilft. Um diese Funktion zu erfüllen, muss ein adäquater Design Service für die jeweilige Entwurfsaufgabe benannt werden. Der Experte soll sich nicht mehr der Arbeitsweise eines Tools unterordnen, wie es bei den bestehenden CAD Frameworks der Fall ist. Er erhält vom Design Service eine in Art, Umfang und Leistung unmittelbar seinen Bedürfnissen entsprechende Dienstleistung angeboten. Ein Design Service konzentriert sich auf die Unterstützung genau einer Entwurfsaufgabe und versucht hierfür die bestmögliche Unterstützung anzubieten. Im Laufe des Entwurfsablaufs werden daher verschiedene Design Services involviert. Ist allen Entwurfsaufgaben eines Task Flows jeweils genau ein Design Services zugeordnet, so wird diese Folge von Design Services als **Service Flow** bezeichnet.

Abbildung 4.8 zeigt einen Service Flow für einen Entwurfsschritt. Ein Design Service ist durch ein Rechteck dargestellt. Eine Entwurfsaufgabe ist durch einen schwarzen Punkt symbolisiert. Zu jeder Entwurfsaufgabe ist ein Design Service zugeordnet, der den Experten bei der Durchführung des Entwurfsprozesses unterstützt.

Die Grundlagen sind geschaffen worden, um den Terminus Design Flow zu definieren:

Definition: 20 *Eine explizite Festlegung aller Entwurfsschritte mit Hilfe der Entwurfsmethodik für ein Fertigungsverfahren durch Implementierung der Funktionen f_{gen} , f_{ver} , f_{sim} , f_{diff} , f_{mod} , f_{new} und zwar insbesondere durch die konkrete Auswahl von Design Services zur automatischen oder manuellen Implementierung aller Funktionen heißt **Design Flow**.* //

Ein Design Flow ist eindeutig beschrieben durch einen Task Flow, der Entwurfsschritte und Entwurfsaufgaben definiert, und einen Service Flow, der eine Folge von Design Services für die Entwurfsaufgaben festlegt. Er führt den Experten durch den Entwurf und bietet ihm dedizierte Unterstützung in Form von Arbeitsanweisungen oder vorkonfigurierten Werkzeugen an. Es wird sichergestellt, dass alle Arbeitseinheiten korrekt durchgeführt werden und die Ergebnisdokumente vollständig sind.

Ein wichtiger Aspekt bei der Erstellung von Design Flows sind die Design Services. Sie werden von der Mikrotechnik-Industrie angeboten und sind in intelligenter Form zu kombinieren, um einen unter dem Aspekt Zeit, Kosten und Qualität optimalen Design Flow zu definieren. Hierfür ist das Design Service Portfolio der Mikrotechnik-Industrie zu ermitteln, auf das während der MEMS-Produktentwicklung zugegriffen werden kann und wo jeder Design Service eindeutig bezüglich seiner zeitlichen Verfügbarkeit, der Leistungsfähigkeit und der verursachten Kosten beschrieben ist.

Abbildung 4.9 ist ein Beispiel für einen Design Flow. Die Entwurfsmethodik legt die Entwurfszustände und die Entwurfsschritte fest. Jede Abbildungsfunktion ist mit einem Rechteck markiert. So wird symbolisiert, dass Task- und Service Flow für jede Abbildungsfunktion spezifiziert sind. Die Aktivität *Verifikation der physikalischen Beschreibung* verdeutlicht explizit die Zuordnung von Entwurfsaufgaben zu Design Services.

4.4 Einsatzgebiet des Kreismodells

In der Mikroelektronik bleiben bei unterschiedlichen integrierten Schaltungen die eingesetzten Bauteile (Transistoren und Verbindungen) gleich. Bedingt dadurch, wird eine gleich bleibende standardisierte Folge von Fertigungsschritten auf das Resist angewandt. Die Konfiguration einzelner Fertigungsschritte und die Festlegung des Maskenlayouts sind die zentralen Aufgaben des Entwurfsprozesses. Das Y-Modell unterstützt diesen Prozess durch eine Entwurfsmethodik,

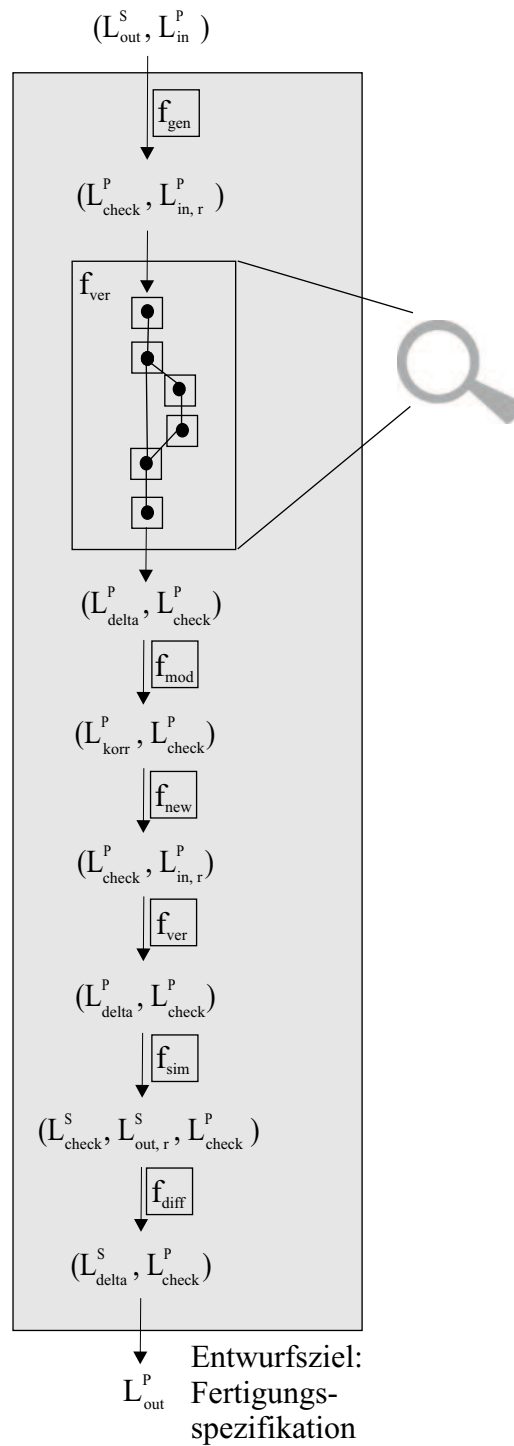


Abbildung 4.9: Beispiel für einen Design Flow

welche schrittweise die Prozessparameter und das Layout erstellt und so den Experten zum Entwurfsziel leitet.

Für die Mikrotechnik variieren die verwendeten Bauteile. Zum Beispiel sind mit Hilfe der OMM sowohl ein elektrostatisch angetriebener Mikromotor als auch auffaltbare Strukturen, so genannte Origamistrukturen, herzustellen [Men97]. In beiden Fällen werden unterschiedliche Folgen von Fertigungsschritten benutzt. Eine Standard-Folge von Fertigungsschritten für alle Entwürfe eines Fertigungsverfahrens, wie in der Mikroelektronik, existiert in der Mikrotechnik nicht. Das Y-Modell besitzt aus diesem Grund nur eine eingeschränkte Eignung, um solche mikrotechnischen Fertigungsschrittfolgen zu erstellen. Es konzentriert sich primär auf die Unterstützung der Konfiguration einzelner Fertigungsschritte als auf die Auswahl von Fertigungstechniken und die Festlegung, in welcher Reihenfolge sie anzuwenden sind.

Um eine genauere Positionierung des Kreismodell bei der Entwurfsunterstützung zu erzielen und gleichzeitig eine noch klarere Abgrenzung zum Y-Modell zu erreichen, werden die Fertigungstechniken der Mikroelektronik und Mikrotechnik zur Hilfe genommen.

Moderne Hochleistungstechnologien für ICs (z.B. Speicher und Prozessoren) sind heutzutage erheblich leistungsfähiger und komplexer als dies bei der Herstellung eines MEMS-Produktes¹ erforderlich ist. Fertigungstechniken in der Mikroelektronik sind in der Lage, Bauteile mit einer Auflösung von 130 Nanometer zu produzieren, wogegen sich in der Mikrotechnik MEMS-Produkte in der Größenordnung zwischen 1 mm und 1 μ m bewegen. Die Silizium-Oberflächen-Mikromechanik kann daher bei der Produktion von Mikrosystemen auf den hohen Entwicklungsstand der Fertigung in der Mikroelektronik zurückgreifen. Sie baut MEMS-Produkte von der Silizium-Oberfläche aus additiv nach oben auf, wodurch ihre Affinität zu CMOS-Fertigungstechniken deutlich wird [Men97].

Für Massen Anwendungen werden MEMS-Produkte häufig mit Hilfe von CMOS-Technologien gefertigt, indem die Fertigungstechniken der OMM so modifiziert werden, dass sie kompatibel zu CMOS-Fertigungstechniken der Mikroelektronik sind. Für die Produktion von Mikrosystemen stehen damit die vorhandenen CMOS-Fertigungskapazitäten der IC-Industrie zur Verfügung.

In der Praxis haben sich zwei unterschiedliche Varianten entwickelt, um die CMOS-Technologie für die Fertigung von MEMS-Produkten gewinnbringend einzusetzen [Lan98]:

- Die erste Variante nutzt einen **statischen CMOS-Fertigungsprozess**

¹Unter einem **MEMS-Produkt** ist der produzierte Fertigungsgegenstand zu verstehen, also das konkrete physikalische Stück Hardware, welches greifbar und für den praktischen Einsatz geeignet ist; daher keine Abstraktion oder Beschreibung dieses Produktes!

zur Herstellung von mikromechanischen Strukturen

- Die zweite Variante verwendet einen **dynamischen CMOS-Fertigungsprozess** zur Herstellung von mikromechanischen Strukturen

Beim statischen CMOS-Fertigungsprozess werden für die Herstellung einer mikromechanischen Struktur nur die Layer eingesetzt, welche ebenfalls für die Herstellung der CMOS-Schaltung selbst eingesetzt werden. Mit Hilfe der Layer wird ein IC-Standard-Prozess durchgeführt, d.h. eine feste, immer gleichbleibende Folge von Fertigungsschritten kommt auf dem Substrat zur Anwendung. Mit der Ergänzung, dass am Ende des IC-Standard-Prozesses ein Fertigungsschritt mit einer anisotropen Ätztechnik zusätzlich zum Einsatz kommt.

Der statische CMOS-Fertigungsprozess besitzt den Vorteil, dass eine Batch-Fertigung ohne Einschränkungen möglich ist. Der Nachteil bei der Produktion auf der Grundlage des statischen CMOS-Fertigungsprozesses, liegt in der geringen Strukturhöhe der MEMS-Produkte. Sie bewegt sich im Bereich weniger Mikrometer bis etwa 20 Mikrometer. MEMS-Produkte, die auf Basis von statischen CMOS-Fertigungsprozessen hergestellt werden, sind, bedingt durch ihre laterale Dimension von einigen hundert Mikrometern, somit noch als quasi-zweidimensionale Strukturen einzuordnen. Dadurch ist die Menge der mikromechanischen Strukturen, die mit diesem Prozess zu erstellen sind, relativ klein. Das Y-Modell ist als Entwurfsmodell für den statischen CMOS-Fertigungsprozess geeignet, da immer von einer festen Folge von Fertigungsschritten ausgegangen wird.

Ein dynamischer CMOS-Fertigungsprozess produziert als erstes die mikroelektronischen Bauteile auf dem Wafer. Danach erfolgt das Prozessieren mit einer vom Entwurfsgegenstand abhängigen Folge von Fertigungsschritten, die aus Maskierungs-, Ätz- oder Schichtabscheidungsvorgängen besteht. Hiermit werden die mikromechanischen Strukturen auf dem Substrat produziert. Zu beachten ist, dass die bestehenden mikroelektronischen Bauteile dadurch nicht beschädigt werden und ihre Funktionsfähigkeit erhalten bleibt.

Der Vorteil eines dynamischen CMOS-Fertigungsprozesses liegt in Möglichkeit, weitgehend beliebige mikromechanische Strukturen zu erstellen, die auch eine 3-Dimension besitzen. Dafür ist deutlich mehr Zeit in die Planung und Durchführung des Design Flows zu investieren, da zusätzliche Masken und Fertigungsschritte zu definieren sind. Kreismodell, UML-Aktivitätsdiagramm und die Entwurfsmethodik repräsentieren, genau für diesen Fall, die idealen Hilfsmittel. Sie sind in der Lage ein Layout und eine Folge von Fertigungsschritten zu finden sowie die benötigte Fertigungstechnik zu identifizieren und

zu konfigurieren. Dabei betrachten sie den Fertigungsprozess als eine mit dem Entwurf korrelierende Variable und beschreiben deshalb den Entwurfsprozess im Gegensatz zum Y-Modell, wo von einer unidirektionalen Entwurfsmethodik ausgegangen wird (siehe Abbildung 3.20 auf Seite 31), als iterativen Vorgang, in den der Fertigungsprozess involviert ist. So sind hinreichend flexible, mit der notwendigen Genauigkeit ausgestattete, dynamische CMOS-Fertigungsprozesse auf Grundlage des Kreismodells und der daraus abgeleiteten Entwurfsmethodik für die MEMS-Produkte zu generieren.

Die Einsatzgebiete für Kreismodell und Y-Modell sind eindeutig formuliert worden. Der weitere Fokus der Arbeit liegt auf dem Entwurf von „echten“ 3-dimensionalen Strukturen, daher bildet das Kreismodell und die Entwurfsmethodik die Basis, für weitere Forschungsaktivitäten bezüglich optimierter Entwurfsunterstützung in der Mikrotechnik.

4.5 Elemente beim Entwurf von Mikrosystemen

Für den Entwurfsprozess in der Mikrotechnik ist ein neues Entwurfsmodell entstanden. Ausgehend von einer einfachen Black-Box-Darstellung (siehe Abbildung 3.2 auf Seite 16) sind anfangs die notwendigen Eingabeinformationen für den Entwurfsprozess in Form von drei Spezifikationen identifiziert und beschrieben worden. Dies sind die Verhaltens-, Struktur- und physikalische Spezifikation, welche die Grundlage für die Durchführung des Entwurfsprozesses bilden. Das Kreismodell repräsentiert als Alternative zum Y-Modell einen neuen Weg zur Darstellung des Entwurfsprozesses.

Das Gesamtmodell mit Rückkopplung gibt einen Überblick über die Interaktionen zwischen den Entwurfsphasen und die grundlegenden Aktivitäten in den einzelnen Phasen. Auf dieser Basis ist eine Weiterentwicklung des Kreismodells speziell für den physikalischen Entwurf im Bereich der lithografiebasierten Mikrotechnik erfolgt. Das UML-Aktivitätsdiagramm ist eingesetzt worden, um die im Kreismodell enthaltenen Entwurfsaktivitäten detaillierter zu spezifizieren und die Vorgehensweise beim physikalischen Entwurf exakter zu dokumentieren. Mit Einführung der Entwurfsmethodik für den physikalischen Entwurf ist eine Möglichkeit geschaffen worden, Entwurfsschritte und -Zustände algorithmisch abzuleiten. So erhält der Experte eine erste Vorgabe, wie bei der Durchführung des Entwurfsprozesses vorzugehen ist. Mit Hilfe des Task- und Service Flows ist eine klare Trennung zwischen Entwurfsaufgaben und den dazugehörigen Dienstleistungen vorgestellt worden. Diese speziellen Dienstleistungen für den Mikrotechnik-Entwurf heißen Design Services. Sie begleiten quasi

den Experten mit Hilfe des Design Flows durch den Entwurfsprozess und bieten ihre Mitarbeit in Abhängigkeit von der intellektuellen Schwierigkeit der Entwurfsaufgabe und den Möglichkeiten der Formalisierung des dazugehörigen Wissens an.

Der Experte behält weiterhin die vollständige Kontrolle über den Ablauf des Entwurfsprozesses und trifft die entwurfsrelevanten Entscheidungen. Routinetätigkeiten, zeitintensive Prüfungen und andere zu automatisierende Tätigkeiten werden ihm abgenommen und an die Design Services ausgelagert.

Design Services sind von den Akteuren der Mikrotechnik-Industrie anzubieten und bilden das letzte Glied in der Kette von Elementen, die als Voraussetzung für einen systematischen Entwurf dienen. Abbildung 4.10 gibt abschließend einen Überblick über alle Elemente, die für einen systematischen Entwurf von Bedeutung sind:

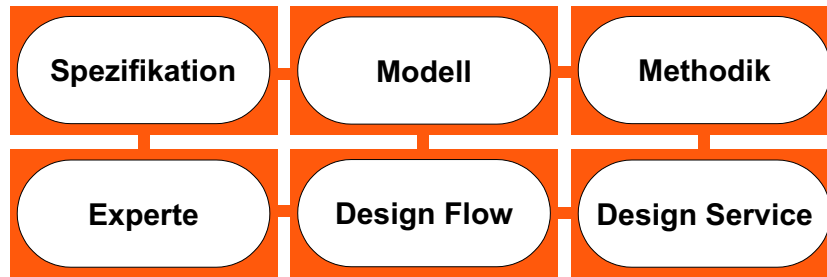


Abbildung 4.10: Elemente beim Entwurf

1. **Spezifikation:** Im Mittelpunkt des Entwurfsprozesses steht die physikalische Spezifikation L_{in}^P und das Ziel, hieraus eine für die Produktion ausreichende Fertigungsspezifikation zu generieren.
2. **Modell:** Das Kreismodell bildet die Grundlage für die Beschreibung des physikalischen Entwurfsprozesses. Es trennt die Aktivitäten *Generierung*, *Verifikation* und *Modifikation* voneinander.
3. **Methodik:** Die Entwurfsmethodik ist durch einen Algorithmus beschrieben worden. Sie beschreibt den Vorgang der Informationsgewinnung für die Spezifikation durch eine Folge von Entwurfszuständen und liefert als Ergebnis das Entwurfsziel: die Fertigungsspezifikation.
4. **Experte:** Der Experte führt den Entwurf mit Hilfe des Design Flows durch. Er trifft die entwurfsrelevanten Entscheidungen und ist für die termingerechte sowie kostenoptimierte Erstellung der Fertigungsspezifikation verantwortlich.

5. **Design Flow:** Der Design Flow besteht aus dem Task Flow und Service Flow, die eine Folge von Entwurfsaufgaben und Design Services enthalten. Er dient dem Experten als Instrument zur Durchführung des Entwurfs.
6. **Design Service:** Ein Design Service ist eine Dienstleistung zur Unterstützung des Experten bei der Bearbeitung einer konkreten Entwurfsaufgabe.

Die theoretischen Grundlagen für die Durchführung des Entwurfsprozesses in der Mikrotechnik sind vorgestellt worden. Auf diesen Grundlagen aufbauend werden in den folgenden Kapiteln Richtlinien für die Entwicklung von IT basierten Werkzeugen zur Unterstützung des Entwurfs abgeleitet.

Kapitel 5

Entwurfsszenarien

Das Kapitel betrachtet zunächst die Akteure der Mikrotechnik-Industrie, die beim Entwurfsprozess eines MEMS-Produktes beteiligt sind und erklärt, welche Abhängigkeiten zwischen den Akteuren existieren. Danach werden Entwurfsszenarien vorgestellt und analysiert, in welche die Akteure aktiv eingebunden sind. Mit Hilfe der Analyseergebnisse wird ein neues Soll-Entwurfsszenario mit dem dazugehörigen Design Services für kleine und mittelständische Unternehmen in der Mikrosystemtechnik konzipiert.

5.1 Akteure

Im Folgenden werden die beim Entwurfsprozess involvierten Akteure aus dem DA-Umfeld vorgestellt und ihre Tätigkeiten näher erläutert¹.

- **Design Team** ist ein Unternehmen oder häufig nur eine Gruppe innerhalb eines größeren Unternehmens, die sich ausschließlich mit der Durchführung eines konkreten Entwurfsprozesses für einen Entwurfsgegenstand beschäftigt. Sie benötigt hierfür einen Design Flow, der vorkonfigurierte Entwurfswerkzeuge und ein „Kochrezept“ für die Mitarbeiter enthält und in klar definierten Schritten zum Entwurfsziel führt. Die Mitarbeiter besitzen keine weit reichende Kompetenz in den Bereichen der Fertigungsverfahren und -techniken. Deshalb ist der Design Flow möglichst neutral

¹Zu beachten ist, dass die beschriebenen Akteure und Tätigkeiten zur Zeit nicht in der ganzen Mikrotechnik-Industrie in der gleichen Stärke ausgeprägt sind. Die identifizierten Akteurnamen und -beschreibungen sind, daher auch als Empfehlung anzusehen, wie eine zukünftige Organisationsstruktur beim Mikrotechnik-Entwurf aussehen könnte.

bezüglich des Herstellungsprozesses zu formulieren. Auf Fertigungskompetenz wird nur über Abstraktionen (z.B. Restriktionen) zugegriffen.

- **DA-Vendor** ist ein Spezialsoftwarehaus, das sich ausschließlich mit der Entwicklung von DA-Tools oder -Frameworks beschäftigt. Heute gibt es fünf dominierende Global Player für integrierte DA Frameworks: Cadence Design Systems, Coventor, IntelliSense Software, Synopsis und Mentor Graphics. Daneben existieren über zweihundert kleine meist sehr innovative Firmen, wie Sagantec, Silvaco, Dosis, Catena und PDF-Solutions/aiss, die den DA-Markt mit hoch spezialisierte DA-Tools versorgen. Die Kompetenz von DA Vendors liegt in der Erstellung von speziellen Algorithmen, die sie dem Experten in Form von IT basierten Werkzeugen anbieten.
- **DA-Support** ist eine Arbeitsgruppe, deren Aufgabe es ist, Design Flows für konkrete Entwurfsprozesse zu erstellen. Sie besitzen das Wissen um die richtigen Frameworks auszuwählen und zu konfigurieren. Ihr Integrations-Know-how erlaubt es, hoch spezialisierte DA-Tools in Frameworks einzubinden, um so für jede einzelne Entwurfsaufgabe die bestmögliche Lösung anzubieten. Gleichzeitig offerieren sie dem Design Team Unterstützungsleistungen bei der Durchführung des Design Flows.
- **Foundry** ist ein Unternehmen oder eine Abteilung innerhalb einer großen Firma, die eine MEMS- oder IC-Fertigungslinie betreibt und sich ausschließlich mit der Produktion, nicht aber mit dem Entwurf von MEMS-Produkten beschäftigt. Eine Foundry besitzt das Technologie Know-how, um eine Fabrikationsanlage einzurichten, instand zu halten und zu betreiben. Sie liefert Technologieinformationen in Form von Layout- und Fertigungsprozessregeln an das Design Team. Sie stellt sicher, dass der Fertigungsgegenstand alle physikalischen Restriktionen einhält, um ihn in der Fabrikationsanlage zu produzieren.

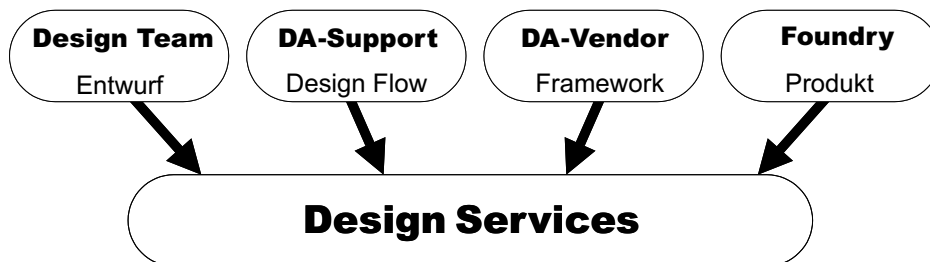


Abbildung 5.1: Akteure in der Mikrotechnik-Industrie

Abbildung 5.1 stellt die beschriebenen Akteure und ihre jeweiligen Arbeitsergebnisse dar:

- Das Design Team liefert den Entwurf oder genauer die Fertigungsspezifikation als Ergebnis
- Der DA-Support ist für die Erstellung des Design Flows verantwortlich
- Die DA-Vendors produzieren die benötigten Werkzeuge zum Beispiel in der Gestalt von Frameworks für die Entwurfsautomatisierung
- Die Foundry ist für die Herstellung des MEMS-Produktes verantwortlich.

Jeder Akteur liefert einen Beitrag zum Design Service Portfolio der Mikrotechnik-Industrie durch Dienstleistungen, die er für die Durchführung des Entwurfsprozesses anbietet. Beispiele für Dienstleistungen der Akteure sind die Algorithmen zur Entwurfsautomatisierung eines DA-Vendors oder das Entwurfswissen einer DA-Support Abteilung, aber auch die physikalischen Restriktionen für die Verifikation, welche eine Foundry offeriert.

Die Abbildung 5.1 dient als Schablone um die Beziehungen zwischen den Akteuren in der Mikrotechnik-Industrie in den nachfolgenden Abschnitten näher zu erläutern.

5.2 Mikroelektronik-Großindustrie

In den großen IC-Unternehmen wie Infineon, Intel und AMD liegt die komplette Entwicklung von Produkten in deren eigener Hand. Als einzige Ausnahme ist die Entwicklung von Design-Software zu nennen, die von außen zugekauft wird, gemeinsam mit umfassenden Software-Support. Zur Entwurfsautomatisierung kommen in der Regel Frameworks zum Einsatz. Die Unternehmen nutzen gewöhnlich mehrere Frameworks gleichzeitig (z.B. Cadence für High Level Design, Mentor für Low-Level und Analog-Design) und ergänzen diese um Software für Spezialanwendungen, die nicht in Frameworks enthalten ist (z.B. Sagantec für Technologiemigration). Wo der konkrete Entwurfsprozess es erfordert, wird spezielle Zusatzsoftware vom DA-Support entwickelt und in das jeweilige Framework integriert. Abbildung 5.2 stellt das Entwurfsszenario in der **Mikroelektronik-Großindustrie** dar.

DA-Support- und Design Team wie auch die In-House Foundry nutzen das Intranet zum Austausch von entwurfsrelevanten Daten. Externe Design Services

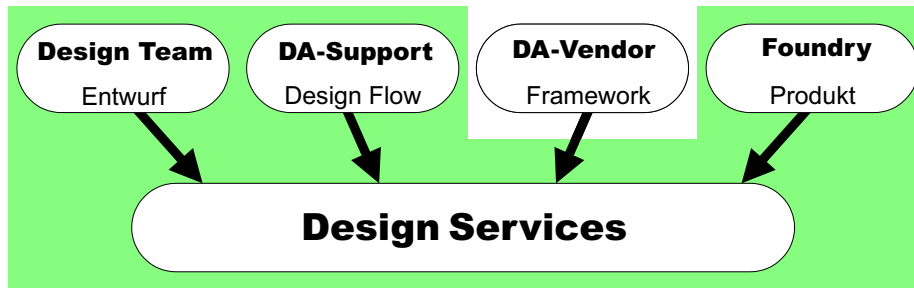


Abbildung 5.2: Szenario in der Mikroelektronik-Großindustrie

werden bis auf Zulieferungen der Tool Anbieter primär nicht genutzt. Das Szenario ist relativ unflexibel hinsichtlich der konkreten Design Flow Erstellung und der unterstützenden Fertigungsverfahren, da meist nur auf die in-House vorhandenen Design Services und Foundries Zugriff besteht.

Das Entwurfsszenario ist extrem teuer. Heute betragen die Kosten für Design-Software und DA-Support bereits ca. 40 % der Gesamtkosten für die Produktentwicklung. Beim Übergang auf 90nm-Technologien wird mit einem Anstieg dieses Faktors auf ca. 60 % gerechnet. Damit würde eine Schmerzgrenze erreicht, die einen Bedarf für ein neues Geschäftsmodell und somit auch für ein anderes Entwurfsszenario weckt [SM03].

5.3 Mikroelektronik-Design House

Für das Entwurfsszenario **Mikroelektronik-Design House** gilt, dass nur Design Flow Entwicklung und Durchführung in einem Haus liegen (siehe Abbildung 5.3).

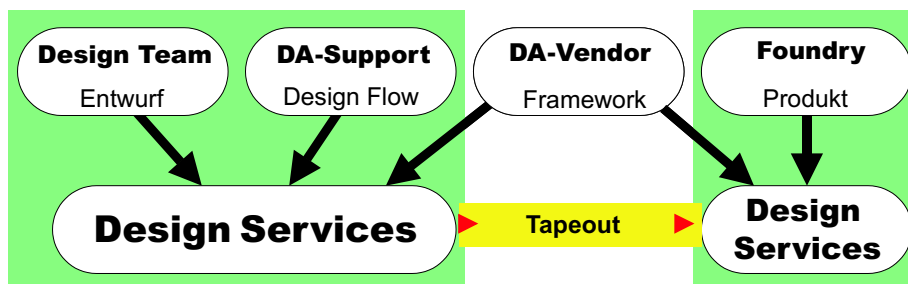


Abbildung 5.3: Mikroelektronik-Design House

Die Fertigung wird nach außen gegeben. Diese Szenario ist häufig im Bereich der anwendungsspezifischen Schaltungsentwicklung anzutreffen (z.B. Automotive, Mobilcom, Medical, etc) und bringt auch nicht-elektronische Komponenten

ins Spiel (z.B. Sensoren). Ein zentrales Problem des Entwurfsszenarios ist das Thema **Second Sourcing**. Entwürfe müssen daher so gestaltet werden, dass sie mit unterschiedlichen Technologien zu fertigen sind, bzw. leicht an neue Technologien anpassbar sind. Die DA-Vendors versorgen sowohl Design-House, als auch Foundry mit denselben Frameworks. Beide Institutionen stimmen sich bezüglich Design Flow Entwicklung im allgemeinen aber nicht ab.

Ist der Entwurfsprozess abgeschlossen sind Teile der Fertigungsspezifikation an eine Foundry zu übergeben. Eine Tape-out Schnittstelle ist zu definieren, um die fertigungsrelevanten Daten (z.B. Maskengeometrien) zwischen den beteiligten Unternehmen auszutauschen. Die ausgewählte Foundry nutzt ihr eigenes Entwurfsframework zum Produktfinishing. Nachdem Eingang einer Fertigungsspezifikation in die Foundry, ist ein erneuter Verifikationsvorgang erforderlich, der prüft, ob die gelieferten Daten für die Fabrikationsmaschinerie geeignet sind, und ob der Entwurf (im Bezug auf den Foundry Design Flow) korrekt war.

Grundsätzlich besteht bei dem Entwurfsszenario dieselbe Problematik hinsichtlich der Flexibilität und der Kosten für den DA-Support. Die Problematik wird verschärft um die Interfacing-Problematik, die durch das Tape-out entsteht. Sie umfasst Datenkonsistenz- und Intellectual Property-Sicherungsfragen.

5.4 Special Design Service

Eine spezielles Entwurfsszenario, das in der Praxis nur gelegentlich anzutreffen ist, ist in Abbildung 5.4 visualisiert.

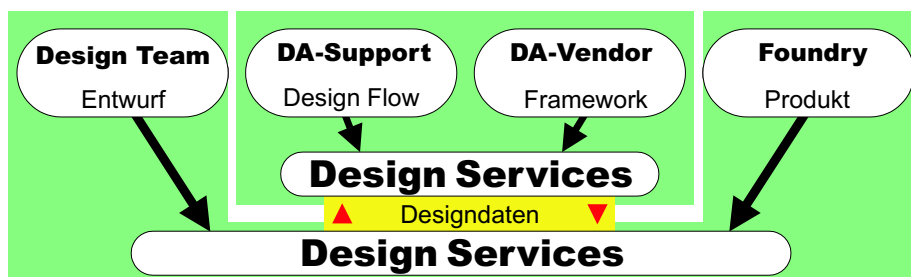


Abbildung 5.4: Special Design Service

Ein Unternehmen verfügt über Know-how und Werkzeuge, um einen konkreten, einzelnen Design Services anzubieten. Es realisiert damit keinen eigenständigen Design Flow, sondern stellt seinen Service als Teil eines beliebigen Design Flows zur Verfügung. Als Beispiel ist Rubicad (<http://www.rubicad.net/>) zu nennen,

wo eine Technologiemigration als Service angeboten wird. Hierfür ist wiederum vom DA-Support ein interner Design Flow zu erstellen, der als Ergebnis die durchgeführte Technologiemigration für eine Fertigungsspezifikation liefert. Das Design Team erhält einen Design Service angeboten, der als Eingabe eine Fertigungsspezifikation und den Migrationswunsch einfordert. Als Resultat liefert der Design Service die überarbeitete Fertigungsspezifikation in dem neuen Technologieformat.

Die Besonderheiten, Design Flow Kompetenz und Werkzeuge in einer Hand zu belassen, jedoch weder Entwurfs- noch Fertigungskompetenz zu besitzen, führt zu einer Verschärfung des Interfacing-Problems. Sowohl Framework zu Framework als auch Intellectual Property-bezogen sind spezielle Vorkehrungen zu treffen, um eine reibunglose Kommunikation zu erlauben.

Der Technologiemigrations-Service ist ein typischer Vertreter für einen unabhängigen Design Service, den die Mikrotechnik-Industrie zur Verfügung stellt.

5.5 Bewertung der Entwurfsszenarien

Für die Mikrotechnik-Industrie sind die bisher betrachteten Szenarien mit verschiedenen Nachteilen behaftet. In allen Szenarien kommen komplexe Frameworks zum Einsatz. In der Mikroelektronik hat sich gezeigt, dass der Ansatz des Frameworks mit vielen Schwierigkeiten verbunden ist. Frameworks besitzen eine starre monolithische Struktur, d.h., sie bestehen aus einem statischen Gebilde vieler Module, die eine Vielzahl von Funktionen zur Verfügung stellen. Daraus ergeben sich einige gravierende Nachteile dieser Systeme:

- hoher Ressourcenbedarf von Frameworks bezüglich Hardware. Frameworks sind daher nur lokal auf leistungsfähigen Servern verfügbar
- die Anpassung von Frameworks an wechselnde Anforderungen ist mit hohem Aufwand verbunden (z.B. neue Technologie Restriktionen einbinden)
- die Einarbeitungszeit der Anwender ist aufgrund der Komplexität und des Umfangs von Frameworks sehr hoch
- Migration von Frameworks in Richtung Internet basierter Lösungen ist durch ihre monolithische Struktur nur sehr eingeschränkt mit vertretbarem Aufwand realisierbar
- eine Reduktion und Anpassung der Funktionalitäten von Frameworks an die Bedürfnisse der Anwender ist nur schwer möglich

- Lizenz-, Betriebs-, Hardware- und Schulungs-Kosten beim Einsatz von Frameworks sind relativ hoch. Für KMU mit wenigen Entwürfen pro Jahr ist die Einführung von Frameworks meist nicht rentabel

Die Punkte werden von der aktuellen Umfrage zum Werkzeugeinsatz und Werkzeugbedarf beim Entwurf von Mikrosystemen des *Zentralverbandes Elektrotechnik- und Elektronikindustrie* bestätigt und lassen Frameworks in der heutigen Ausprägung insbesondere aus wirtschaftlichen Gesichtspunkten als nicht besonders geeignet für die Entwurfsunterstützung in der Mikrotechnik-Industrie erscheinen [LN04]. Wobei festzuhalten ist, dass einzelne Algorithmen, die in den Frameworks enthalten sind, sehr wohl hilfreich wären um den Entwurfsprozess zu automatisieren.

Die schwierige Situation der Mikrotechnik-Industrie ist auch im Positionspapier *Zukünftige Schwerpunkte für die Themen Entwurf - Simulation - Modellierung - Test* des Arbeitskreises MST-Entwurfstechnik beschrieben: „Unabhängig davon ähnelt die Abdeckung des Gebiets des MST-Entwurfs durch Werkzeuge immer noch einem Flickenteppich. Insbesondere KMU sind durch diese Situation überfordert, weil das breite Know-how über die Vielzahl der Werkzeuge und der erforderliche Vorrat an Werkzeugen nicht vorgehalten werden kann [Joh01].“ Der Handlungsbedarf ist damit klar und deutlich, so dass sich die Frage stellt, wie am Besten vorzugehen ist, um schnell ein attraktives Angebot für KMU anzubieten. Ausgangspunkt für die Überlegungen bilden die zuvor vorgestellten Szenarien.

Die ersten beiden Szenarien enthalten eine eigene DA-Support Abteilung, die für die Design Flow Entwicklung verantwortlich ist. Eine solche Abteilung verursacht hohe Personalkosten und ist von KMU nur schwer zu finanzieren. Wünschenswert wäre es für KMU zeitlich begrenzten Zugriff auf eine DA-Support Abteilung zu haben und dann nur für den in Anspruch genommenen Services zu zahlen. Auch die Möglichkeit zwischen unterschiedlichen DA-Support Abteilungen zu wählen, um unter dem Aspekt Kompetenz und Kosten die beste Wahl zu treffen, wäre ein Vorteil für KMU.

Die Foundries verursachen extrem hohe Kosten und sind einem permanenten technologischen Wandel unterworfen. In den meisten Fällen produzieren KMU ihre MEMS-Produkte nicht selber. Es gibt Ausnahmen im Bereich der HARMST-Technologien, wo einige KMU, wie zum Beispiel BARTELS Mikrotechnik oder MICROTEC als Nischenanbieter auch die Herstellung übernehmen.

KMU suchen für ihre Entwürfe (Fertigungsspezifikationen) günstige und flexible Foundries mit entsprechenden Kapazitäten zur Herstellung der geplanten MEMS-Produkte. Um ihren Kostenrahmen nicht zu überschreiten, benötigen die KMU dafür einen einheitlichen Dienst über alle Foundries, der es ihnen ermöglicht Kosten- Leistungs- und Technologievergleiche anzustellen. Dieser Dienst sollte an zentraler Stelle verfügbar sein und so den KMU einen einfachen Überblick über den geeignetsten Produzenten geben. Die Überprüfung, ob die Fertigungsspezifikation vollständig und korrekt ist, sollte nicht von dem Design Team und der Foundry durchgeführt werden. Es erscheint sinnvoll, diesen Dienst extern von der Foundry zur Verfügung zu stellen und in den Design Flow einzubauen. Dadurch ist der Migrationsaufwand beim Tape-out deutlich reduzierbar.

Um die Entwurfskosten zu senken, ist der Aspekt der Wiederverwendung von Design Flows zu überdenken. In den zwei dargestellten Szenarien werden Design Flows von dem jeweiligen Unternehmen erstellt und nur für sich selber genutzt. Die hohen Kosten der Design Flow Entwicklung könnten reduziert werden, wenn gewisse Design Flows zentral verfügbar wären und dann für einen fairen Preis den KMU temporär zur Verfügung ständen.

Zusammenfassend ist festzuhalten, dass im Mittelpunkt der bestehenden Entwurfsszenarien komplexe Frameworks stehen, die zu extrem hohen Kosten für den Entwurfsprozess führen. KMU können sich diese Kosten und den zusätzlichen Aufwand für Schulungen und Kompetenzerwerb im Entwurf nicht leisten.

5.6 Ein neues Entwurfsszenario

Die Nachteile der vorgestellten Entwurfsszenarien deuten darauf hin, dass für die KMU ein eigenes Entwurfsszenario zu entwickeln ist. Der hohe Kostendruck dem KMU unterliegen, bietet ihnen keine Möglichkeit bekannte Entwurfsszenarien einfach zu kopieren. Daraus ist die Idee entstanden, ein kostenoptimiertes auf die Bedürfnisse der KMU abgestimmtes Entwurfsszenario, zu definieren.

Abbildung 5.5 zeigt das neue **service-orientierte Entwurfsszenario (SES)** für KMU.

Die Design Services sind in drei Service-Klassen eingeteilt worden:

- **Manuelle Services** sind klassische Dienstleistungen, die in Form von Beratungsleistungen durch menschliche Experten zu erbringen sind. Die Berater erhalten Entwurfsaufgaben zugewiesen und lösen diese selbständig.

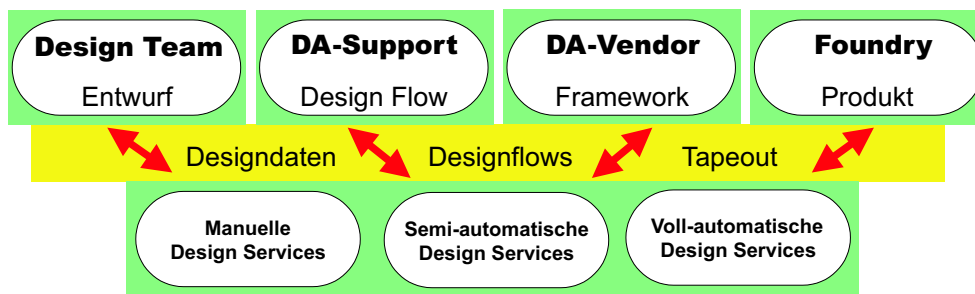


Abbildung 5.5: Neues Entwurfsszenario für KMU

Jeder der vier Akteure kann Berater sein und damit manuelle Services anbieten.

- **Semi-automatische Services** sind mit Hilfe von Entwurfswerkzeugen anzubieten und erfordern immer eine Interaktion mit dem Experten um die Entwurfsaufgabe zu erledigen. Semi-automatische Services werden über eine Kommunikationsinfrastruktur zur Verfügung gestellt. Der Akteur bedarf keiner lokalen Installation einer Software.
- **Voll-automatische Services** führen die Entwurfsaufgabe ohne Eingriff des Experten durch und liefern nach einer endlichen Zeit das geforderte Ergebnis und damit den neuen Entwurfzustand. Voll-automatische Services sind von Entwurfswerkzeugen über ein standardisiertes Interface anzubieten, an welches die Eingabeinformationen zu übergeben sind und worüber das Resultat zu liefern ist.

Das service-orientierte Entwurfsszenario enthält eine völlige Entkopplung aller beim Entwurf beteiligten Akteure. Hierdurch sind die Nachteile der bestehenden Entwurfsszenarien weitgehend beseitigt. Ein Design Team ist zum Beispiel flexibel bezüglich der temporären Auswahl eines Dienstleisters für den DA-Support. Die Voraussetzung hierfür ist, dass die bestehenden DA-Support Abteilungen ihre Dienstleistungen am freien Markt anbieten, um damit ihre Kosten zu senken, oder, dass hier ein neuer Markt von Dienstleistern entsteht, der entsprechende Design Services wie Tool Integration, Technologieauswahl oder Coaching für KMU offeriert. Das Szenario Special Design Service stellt ein Beispiel für einen solchen unabhängigen Service dar. Dazu gehört auch der Design Service Brokering: die Vermittlung von Zugriff auf vorhandene Design Flows, Werkzeuge und Foundries.

Ein weiterer Vorteil des service-orientierten Szenarios ist die Möglichkeit, Foun-

dries anhand der oben beschriebenen Merkmale zu vergleichen und gleichzeitig einen Design Service einzufordern, der die Verifikation der Fertigungsspezifikation direkt von der Foundry durchführen lässt.

DA-Frameworks sind innerhalb des neuen Entwurfsszenarios nur dann einzusetzen, wenn sie Design Services unterstützen. So besteht die Möglichkeit, über Design Services auf einzelne Algorithmen der Frameworks dediziert zuzugreifen ohne das Gesamtsystem lokal vorzuhalten und die damit verbundenen Nachteile einzukaufen. Design Service fähige DA-Frameworks können zum Beispiel von DA-Support Abteilungen betrieben werden. Sie können so den KMU für einen eingeschränkten Zeitraum zu deutlich geringeren Kosten ihre Funktionalität punktuell zur Verfügung stellen. Verschiedene Abrechnungsmodelle sind hier möglich wie zum Beispiel *pay-per-use*. Das neue Entwurfsszenario liefert einem Design Team, das eine konkrete Produktidee umsetzen möchte, die Möglichkeit, genau die dafür erforderlichen Dienstleistungen von den Akteuren am Markt einzukaufen, d.h. Entwurfswissen, Entwurfswerkzeuge und Foundry-Schnittstellen.

Um aus der Produktidee auf systematische Art und Weise ein MEMS-Produkt zu erzeugen, unterstützt das service-orientierte Entwurfsszenario folgende Vorgehensweise:

- **Auswahl der geeigneten Akteure**

Durch die Entkopplung der Akteure bedarf es vor Beginn der MEMS Produktentwicklung der Auswahl von geeigneten Kandidaten. Bevor die Akteure nach den Kriterien Verfügbarkeit, Kosten und Qualität ausgewählt werden, entscheidet das Design Team zuerst, welche Dienstleistungen es grundsätzlich extern am Markt einkaufen möchte und welche es selber erbringen kann.

- **Spezifikation erstellen**

Von den Akteuren ist eine Spezifikation zu erstellen. Aufbau und Inhalte der Spezifikation L_{in}^{VSP} für einen Entwurfsgegenstand sind in Abschnitt 3.7 beschrieben. Die Spezifikation L_{in}^{VSP} dient den zuvor gewählten Akteuren als Grundlage, um die Design Flow Entwicklung durchzuführen.

- **Design Flow entwickeln**

Um einen Design Flow zu erhalten, ist entweder mit Hilfe der Spezifikation ein bestehender Design Flow von den Akteuren auszuwählen und zu adaptieren oder es bedarf der Entwicklung eines neuen Design Flows. Eine neue Design Flow Entwicklung besteht aus folgenden drei elementaren Schritten:

1. Für den Entwurfsprozess sind alle Entwurfsschritte und ihre Reihenfolge mit Hilfe der in Abbildung 4.5 auf Seite 60 dargestellten Entwurfsmethodik festzulegen
2. Für jeden Entwurfsschritt sind die Entwurfsaufgaben zu beschreiben (was ist zu tun), d.h., eine feste Folge von konkreten in sich abgeschlossene Arbeitseinheiten ist zu definieren (Task Flow)
3. Auswahl und Zuordnung von Design Services der Mikrotechnik-Industrie zu Entwurfsaufgaben (wie ist es zu tun). Hierdurch wird explizit beschrieben, wie die Durchführung jeder einzelnen Entwurfsaufgabe realisiert werden soll (Service Flow)

Die Design Flow Entwicklung ist damit beendet.

- **Design Flow durchführen**

Ist die Design Flow Entwicklung abgeschlossen, kann mit der Durchführung begonnen werden. Ziel der Akteure ist es, alle Informationen zu generieren, um eine für die Herstellung ausreichende Fertigungsspezifikation zu erstellen. Im Gegensatz zum Entwurfsszenario in der Mikroelektronik-Großindustrie sind DA-Support und Foundry durch Akteure repräsentiert, die nur temporär für das Design Team die benötigten Design Services erbringen. Wie schon bei der Design Flow Erstellung gilt auch bei der Design Flow Durchführung, dass die Akteure quasi als Mitglieder eines Projektes fungieren, die mit einem klaren Ziel, einem vorgegebenen Budget und für einen bestimmten Zeitraum zusammenarbeiten. Der Vorteil ist, dass die Akteure nach einer **best-of-breed Strategie**² ausgewählt werden, um für dieses Projekt Design Flow Durchführung ein optimales Ergebnis unter dem Aspekt Zeit, Kosten und Qualität zu liefern. Wie schon erwähnt, steuern die großen DA-Vendors für bestehende Entwurfsszenarien fast ausschließlich Entwurfswerkzeuge in Form von DA-Frameworks bei. Um das service-orientierte Entwurfsszenario zu nutzen sind Anstrengungen zu unternehmen, um anstelle von komplexen Werkzeugen entsprechende Design Services für Entwurfsaufgaben anzubieten.

- **Herstellung initiieren**

Das Design Team nutzt einen Design Service der Foundry um die Fertigungsspezifikation auf produktionsrelevante Restriktionen zu überprüfen. Danach erfolgt die Übergabe der Fertigungsspezifikation an die Foundry.

²Die best-of-breed Strategie ermöglicht es, die jeweils besten Ressourcen für verschiedene Teilaufgaben zu kombinieren oder investitionsschonend bereits vorhandene Ressourcen einzusetzen.

Die Fertigungsspezifikation $L_{out,d}^P$ wird von der Foundry verwendet, um konkrete Steueranweisungen für die Fertigungsmaschinen abzuleiten. Die Herstellung des MEMS-Produktes kann jetzt beginnen.

Die Vorgehensweise beim service-orientierten Entwurfsszenario ist in Abbildung 5.6 dargestellt.

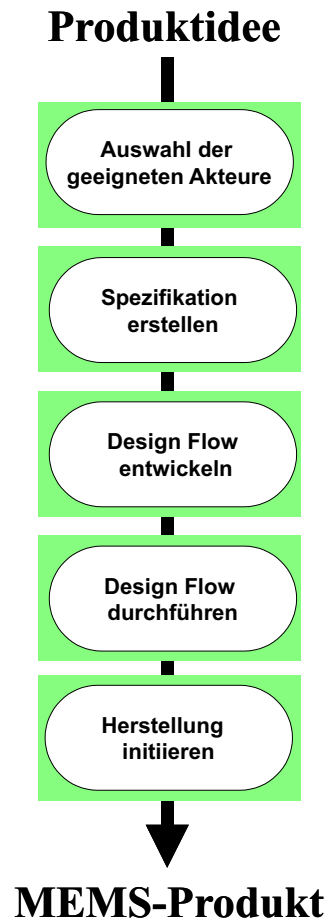


Abbildung 5.6: Vorgehensweise beim service-orientierten Entwurfsszenario

Das service-orientierte Entwurfsszenario stellt eine Alternative zu den bestehenden Entwurfsszenarien dar und ist unter dem Gesichtspunkt von Kostenreduktion beim Entwurfsprozess und hoher Flexibilität bei der Auswahl der Akteure entwickelt worden. Die nächste Aufgabe im Rahmen der Dissertation besteht darin, ein tragfähiges Realisierungskonzept für das service-orientierte Entwurfsszenario zu finden.

Kapitel 6

Design Services

Im letzten Kapitel wurde das Konzept des service-orientierten Entwurfsszenario vorgestellt. In diesem Kapitel geht es um die Realisierung dieses Szenarios mit Hilfe einer Softwarelösung. Sie soll die Eigenschaften und Vorgehensweise des service-orientierten-Szenarios unterstützen. Es werden verschiedene teilweise sehr innovative Software-Architekturen und Techniken verwendet, um das service-orientierte Szenario in der Praxis anzubieten. Dabei wird darauf geachtet, die bestehenden Entwurfswerkzeuge der Mikrotechnik soweit wie möglich zu berücksichtigen und in das neue Szenario zu integrieren. Am Ende entsteht eine Softwarelösung für die flexible Entwicklung von beliebigen Design Flows in der Mikrotechnik, die in dieser Form heute noch nicht von der DA-Industrie angeboten wird.

6.1 Anforderungen

Ziel ist es, eine Softwarelösung zur Unterstützung des service-orientierten Entwurfsszenario zu konzipieren. Hierfür sind die Anforderungen an die Softwarelösung zu analysieren und auszuwerten. Diese Arbeit wurde bereits im letzten Kapitel weitgehend erledigt. Einen Überblick und eine kurze Zusammenfassung der Anforderungen gibt die nachfolgende Aufstellung:

1. Eine Unterstützung von lose gekoppelten Akteuren beim Entwurfsprozess ist durch eine flexible Infrastruktur zu realisieren. Sie ist notwendig, da es im Rahmen des Entwurfsprozesses zu einer beliebigen Konstellation von Akteuren kommen kann. Diese müssen in der Lage sein, effizient miteinander zu kommunizieren und Informationen auszutauschen. Über diese Infrastruktur sind ebenso die Design Services anzubieten oder zu nutzen.

2. Eine Unterstützung von manuellen, semi- und voll-automatischen Design Services mit Hilfe von standardisierten Interfaces ist zu implementieren. Sie sollen sicherstellen, dass ein reibungsloser Informationsaustausch erfolgen kann, ohne zum Beispiel bei voll-automatischen Design Services eine aufwändige Schnittstellenspezifikation zu erarbeiten, die danach mit viel Aufwand zu realisieren ist.
3. Eine zentrale Stelle ist einzurichten, die einen Überblick der vorhandenen Design Services liefert, über die eine entsprechende Service-Auswahl zu treffen ist und die beschreibt, wie der jeweilige konkrete Service-Zugriff erfolgen kann.
4. Die Unterstützung einer systematischen Vorgehensweise beim Entwurfsprozess, hauptsächlich bei der Design Flow Entwicklung und Durchführung, ist software-technisch umzusetzen. Hier geht es darum, einen Task Flow mit dem dazugehörigen Service Flow zu finden, um danach den konkreten Entwurfsprozess auszuführen.

Für die oben genannten Punkte sind im weiteren Verlauf der Arbeit Software-Techniken und Architekturen zu suchen, welche die oben genannten Anforderungen adäquat in Leistungsmerkmale einer Softwarelösung umsetzen.

6.2 Eine flexible Infrastruktur

Bedingt durch die hohe Akzeptanz und das damit verbundene starke Wachstum des Internets zur globalen Kommunikationsplattform unserer Zeit, repräsentiert es die ideale Infrastruktur für das neue service-orientierte Entwurfsszenario. In Abhängigkeit vom Adressaten und im Hinblick auf Sicherheitsaspekte kann eine lose Kopplung von Akteuren und der Einsatz von Design Services in einem local area network, virtuell privat network oder wide area network erfolgen. Das **local area network (LAN)** wird im Web-Zeitalter als Intranet bezeichnet, das **virtuell privat network (VPN)** heißt jetzt Extranet und das **wide area network (WAN)** ist durch das Wort Internet charakterisiert.

Design Services sind über das Internet öffentlich allen Akteuren anzubieten. Ist eine geschlossene Benutzergruppe von Akteuren mit Services zu versorgen, so ist das Extranet als Kommunikationsnetzwerk zu wählen. Für reine Inhouse Services bietet sich das Intranet als geeignete Infrastruktur an.

Die Kommunikation und damit auch die Kopplung der Akteure ist ebenso über die drei möglichen Ausprägungen der Kommunikationsinfrastruktur ab-

zuwickeln. Das Internet, Extranet oder Intranet repräsentieren die passende Realisierungsstrategie für die SES Anforderungen, da eine hohe Verfügbarkeit der Infrastruktur garantiert ist, die Bandbreite in Abhängigkeit vom Bedarf gewählt werden kann und insbesondere die lose Kopplungen von Akteuren im Rahmen von Entwurfsprojekten Idealerweise unterstützt wird.

6.3 Manuelle Design Services

Beim manuellen Service erfolgt eine Mensch-Mensch Kommunikation, in dem ein Design Team für einen gewissen Zeitraum Beratungsleistungen von einem anderen Akteur eingekauft. Alternativ ist das Internet als Kommunikationsplattform zu nutzen. Über eine Videokonferenz kann das Gespräch mit einem Experten aufgenommen werden. Er berät und liefert Lösungen für Entwurfsaufgaben, die nicht durch semi- oder voll-automatische Services abzudecken sind. So sind punktuell menschliche Experten in einen Service Flow einzubinden. Dieser Markt für Beratungsleistungen via Internet muss sich in der Mikrotechnik noch entwickeln. In Anbetracht von steigenden Personalkosten auf der einen Seite und mangelnden Fachexperten auf der anderen Seite sind manuelle Services für KMU eine wirkliche Alternative, um sich temporär mit teurem Spezialwissen zu versorgen.

Es sind zahlreiche Videokonferenzlösungen am IT-Markt vorhanden, die das Internet als Infrastruktur nutzen und ausschließlich ein Web-Browser als Interface sowie eine Webcam als Zubehör für die Videokonferenz benötigen [Ehr03]. Die Kombination von klassischen Beratungsleistungen mit innovativen internet-basierten Videokonferenz-Lösungen ist ein interessanter Ansatz, um manuellen Services eine neue Qualität zu geben.

6.4 Semi-automatische Design Services

Beim semi-automatischen Design Service erfolgt eine Mensch-Maschine Kommunikation über ein Benutzerinterface. Als geeignetes Medium ist ein Web Browser zu verwenden. Er ist auf jedem Desktop Rechner vorhanden und offeriert den Service in Form von statischen und dynamischen Web-Seiten oder durch eine Applikation, die direkt im Browser läuft. Der Akteur kann über das Interface den Service nutzen und die Entwurfsaufgabe interaktiv erledigen. Jeder Service ist über eine **Unified Resource Location (URL)** direkt über das Internet, Intranet oder Extranet anzusprechen.

Mit `http://[firma]/services/[servicename]` kann zum Beispiel ein DA-Vendor einzelne Design Services für Entwurfsaufgaben anbieten. Die Granularität eines Services soll sich nach den Entwurfsaufgaben ausrichten, die sich aus Funktionen f_{gen} , f_{ver} , f_{sim} , f_{diff} , f_{mod} , und f_{new} ergeben. Konkrete Beispiele für semi-automatische Services in der Mikrotechnik werden später vorgestellt.

Die bestehenden Entwurfswerkzeuge unterstützen heutzutage semi-automatische Services in den meisten Fällen nicht. Der Blick auf DA-Tools, wie zum Beispiel IntelliSuite der Firma IntelliSense Software¹, spiegeln zum einen die Nachteile wider (siehe Abschnitt 5.5 auf Seite 78) und zum anderen zeigen sie, die erhebliche Defizite im Bereich der dringend benötigten Design Services auf. Der interaktive Zugriff über einen Browser auf einzelne Services ist meist nicht möglich. Um den Bedarf der KMU und damit verbunden den steigenden Marktdruck bezüglich besserer Design Flow Unterstützung zu befriedigen, sind Framework Anbieter aufgefordert hier kurzfristig Abhilfe zu schaffen.

6.4.1 Software-Architekturen

Um semi-automatische Services anzubieten, bedürfen Entwurfswerkzeuge einer bestimmten Software-Architektur. Unter einer **Software-Architektur** ist ein Konzept zur Beschreibung der Struktur eines Softwaresystems zu verstehen.

In diesem Abschnitt werden potentielle Software-Architekturen für Entwurfswerkzeuge vorgestellt, die semi-automatische Services unterstützen. Es werden solche Architekturen betrachtet, die bereits eine bedeutende Relevanz in der Industrie besitzen und ihre Praxistauglichkeit bereits unter Beweis gestellt haben. Ziel ist es, eine Architektur zu finden, die es erlaubt **service-orientierte Entwurfswerkzeuge**, d.h. solche die semi-automatische Services anbieten, zu entwerfen und Wege aufzuzeigen, wie bestehende DA-Tools ebenfalls mit überschaubarem Aufwand in diese Richtung migrieren können.

Die IT-Welt wurde lange Zeit durch zwei Systemarchitekturen dominiert, dies ist zum einen die Mainframe-Architektur und zum anderen die Client/Server-Architektur.

Die **Mainframe-Architektur**² hat ihre Ursprünge in den sechziger Jahren und wurde von der Firma IBM entwickelt. Banken, Versicherungen und der Handel setzen überwiegend zur Abwicklung ihrer Geschäftsprozesse Mainframe basierte Rechnersysteme ein. Sie betreiben mit diesen Systemen ihre über Jahrzehnte

¹<http://www.intellisensesoftware.com/>

²Die Mainframe-Architektur wird auch als **Centralized Computing Model** bezeichnet.

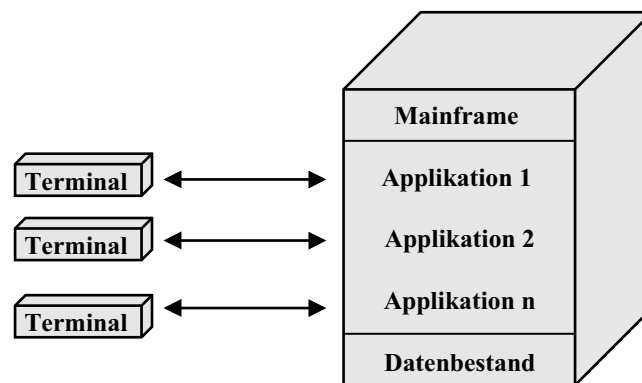


Abbildung 6.1: Beispiel für Centralized Computing Model

entwickelten Applikationen, die sich durch eine hohe Verfügbarkeit und ein relativ einfaches Softwaremanagement auszeichnen. Eine Mainframe-orientierte Applikation besteht aus einer starren Struktur, die alle Funktionen in sich vereint und ausschließlich auf dem Großrechner zur Ausführung kommt. Die Wartung und Weiterentwicklung einer solchen Anwendung ist relativ schwierig und zeitaufwendig. Die Leistungsfähigkeit der Applikationen ist kaum skalierbar. In der Regel sind größere Investitionen in die Mainframe-Hardware erforderlich, um eine Leistungssteigerung zu erreichen. Als weiterer Nachteil sind die hohen initialen Anschaffungskosten für die Hardware zu nennen und die sich daraus ergebenden Betriebskosten.

Die Rechenleistung stellt der Großrechner allen Anwendern zur Verfügung und dient dabei als Laufzeitumgebung für die Applikationen. Die Terminals dienen als Ein-/Ausgabegeräte und ermöglichen den Anwender Daten zu übermitteln oder Ergebnisse abzurufen (Abbildung 6.1).

Der logische Aufbau der Mainframe-Architektur ist in Abbildung 6.2 zu sehen.

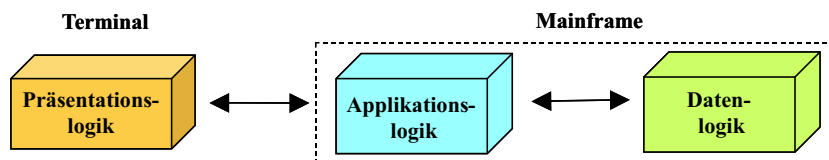


Abbildung 6.2: Logischer Aufbau der Mainframe-Architektur

Die **Präsentationslogik** dient zur Visualisierung von beliebigen Ein- und Ausgabenformationen einer Applikation. Beim Centralized Computing Model kommen ASCII Terminals zum Einsatz. Das Mainframe-System beinhaltet die Ap-

pplikationslogik und die Datenlogik. Die **Applikationslogik** ist verantwortlich für die Ausführung der Algorithmen und die Berechnung der Ergebnisse. Die **Datenlogik** speichert Informationen einer Applikation und erlaubt es ihr auf unterschiedliche Datenquellen zuzugreifen.

Die mangelnde Flexibilität bei der Funktionsaufteilung, hohe Hardware- und Betriebskosten sowie die langsamen Antwortzeiten von Mainframe-Systemen führten dazu, dass Ende der achtziger Jahre die **Client/Server-Architektur**³ entwickelt wurde. Sie besteht aus leistungsfähigen Arbeitsplatzrechnern den Client-Stationen und aus einer Reihe von Servern, die Basis-Dienste, wie z.B. Zugriff auf Datenbanken, Dateiablage, Drucken oder E-Mail zur Verfügung stellen. Die Clients übernehmen Darstellung und Berechnung der anwendungsbezogenen Informationen und die Server stellen dabei ihre Dienste netzweit zur Verfügung und sind für die gesamte Datenverwaltung verantwortlich (siehe Abbildung 6.3).

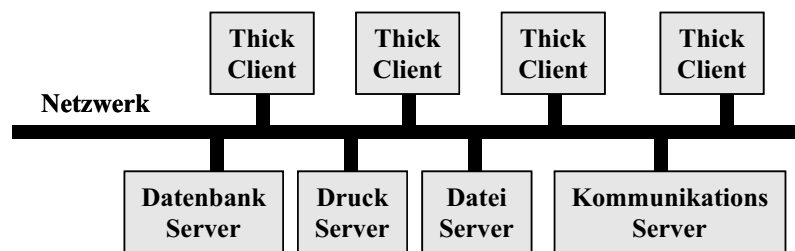


Abbildung 6.3: Beispiel für eine Client/Server-Architektur

Die Auf- und Verteilung der Funktionen auf die Clients und Server erlaubt ein höheres Maß an Skalierbarkeit und Flexibilität, wenn sich Umgebungsparameter, wie zum Beispiel Benutzerzahlen, ändern, ist durch hinzufügen einer neuen Client-Station kurzfristig ein weiterer Arbeitsplatz zu schaffen. Die Client/Server-Architektur reduziert im Vergleich zum Mainframe-System die Kosten pro Arbeitsplatz und rückte die Informationstechnologie finanziell in den Zugriffsbereich kleiner und mittelständischer Unternehmen.

Der Ansatz der Client/Server-Architektur besitzt einige Nachteile. Das Softwaremanagement, d.h. das erstmalige Einrichten von Anwendungen oder das Einspielen von neuen Versionen hat sich auf der Client-Seite als schwierig und sehr zeitintensiv herausgestellt. Hier sind vergleichbare Mainframe-Lösungen deutlich im Vorteil. Es hat sich in der Praxis herausgestellt, dass es problematisch ist, ein ausgewogenes Verhältnis zwischen Client und Server-Funktionalitäten herzustellen, was häufig zu sehr komplexen und großem Implementierungen von

³Die Client/Server-Architektur heißt auch **2-Schichten-Modell**.

Applikationen auf der Client-Seite führt, die dementsprechend als **Fat-** oder **Thick Clients** bezeichnet werden.

Der logische Aufbau der Client/Server-Architektur ist in Abbildung 6.4 zu sehen.

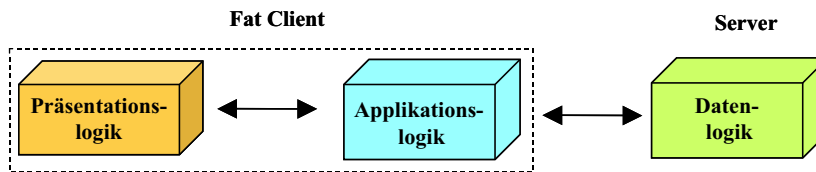


Abbildung 6.4: Logischer Aufbau der Client/Server-Architektur

Mainframe- und Client/Server-Architektur sind nicht geeignet, um als Zielarchitektur für service-orientierte Entwurfswerkzeuge zu fungieren. Es fehlt die grundsätzlich Fähigkeit über einen Web-Browser und mit Hilfe des Internets Informationen und/oder Applikationen anzubieten (Mainframe-Systeme werden von den gängigen Browsern nicht unterstützt, für Fat-Clients ist in Browsern keine Laufzeitumgebung vorhanden⁴). Für service-orientierte Entwurfswerkzeuge ist deshalb eine andere Architekturlösung zu suchen.

Bei der Suche nach einer Software-Architektur steht der Wunsch im Vordergrund die Applikationslogik aus den Client Stationen zu entfernen. Deswegen ist die Idee entstanden, den Fat Client durch einen **Thin Client** abzulösen. Ein Thin Client ist ausschließlich für die Darstellung der Präsentationslogik einer Anwendung verantwortlich. Ein typischer Vertreter für einen Thin Client ist ein PC mit einem Web-Browser, wobei der Browser für die Darstellung der Präsentationslogik verantwortlich ist und als multimediales Ausgabegerät fungiert. Thin Clients sind kostengünstig in der Beschaffung und minimieren den administrativen Overhead, da eine Applikation nicht mehr lokal zu installieren und zu verwalten ist. Die Applikationslogik wandert in den Bereich der einfacher administrierbaren Server. Den Bereich bezeichnet man als **Mittel-Schicht** (englisch: **Middle Tier**). Die Datenlogik wird in die **Daten-Schicht** (englisch: **Data Tier**) ausgelagert. Sie beinhaltet unterschiedliche Datenquellen

⁴Es besteht die Möglichkeit, durch den Einsatz von zusätzlicher Spezialsoftware, bestehende Fat-Clients im Browser zu betreiben. Hierbei wird der Fat-Client auf einem separaten Server-System aktiviert und seine Eingabe- und Ausgabeinformationen werden in den Browser umgelenkt. Für den Browser ist ein Plug-in zu installieren, das die Kommunikation mit dem Server-System übernimmt und für die korrekte Darstellung der Daten sorgt. Für diesen Ansatz ist teure kommerzielle Software zu kaufen, wie sie zum Beispiel das Unternehmen Citrix (<http://www.citrix.de/>) anbietet. Der Ansatz wird aus diesem Grund nicht weiter verfolgt.

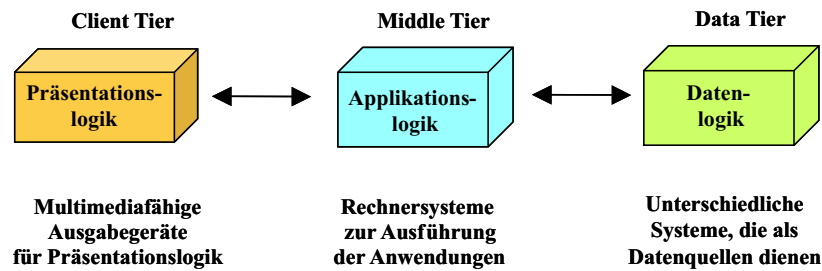


Abbildung 6.5: Logischer Aufbau der 3-Schichten-Architektur

wie Datenbanken, Flat Files oder andere Systeme, die über spezielle Adapter anzubinden sind. Zwischen der Client-, Mittel- und Datenschicht erfolgt über das Internet/Intranet eine ständige bidirektionale Kommunikation.

Ein **3-Schichten-Modell** oder **3-Schichten-Architektur** ist entstanden, die eine Anwendung in drei logische Bereiche aufteilt (siehe Abbildung 6.5):

Präsentationslogik, dient zur Visualisierung von beliebigen Informationen einer Applikation.

Applikationslogik, führt die Algorithmen aus und berechnet den Funktionswert also das Ergebnis einer Applikation.

Datenlogik, speichert Informationen einer Applikation und erlaubt es ihr auf unterschiedliche Datenquellen zuzugreifen.

Die 3-Schichten-Architektur kombiniert die Vorteile aus der Client/Server- und Mainframe-Architektur, wie einfaches Softwaremanagement, hohe Skalierbarkeit und große Flexibilität bezüglich der Funktionsaufteilung und repräsentiert keine neue Architektur, sondern ist ein Hybrid-Modell aus Mainframe- und Client/Server-Architektur [Sch99].

6.4.2 Software-Techniken

Für die Entwicklung von semi-automatischen Services wird die 3-Schichten-Architektur verwendet. Sie erfüllt die Anforderungen um semi-automatische Services zu realisieren, da dieser Service dem Akteur ohne lokale Software Installation anzubieten ist, ein Web-Browser auf jedem Rechnersystem existiert und die Kenntnis der URL den direkten Zugriff auf den Service erlaubt.

Abbildung 6.6 basiert auf der **Java 2 Platform Enterprise Edition (J2EE)** Spezifikation der Firma Sun Microsystems [Sun03]. Die Spezifikation legt eine

konkrete Ausprägung für die 3-Schichten-Architektur fest (siehe Abbildung 6.5) und ist ein de facto Standard, um komponenten-basierte Mehrschichtenapplikationen zu entwickeln.

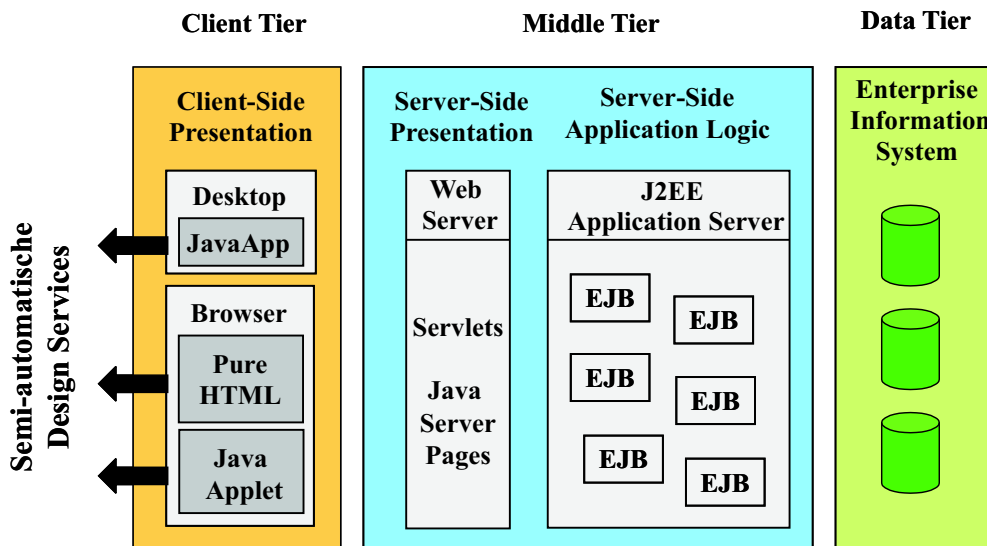


Abbildung 6.6: Beispiel für eine 3-Schichten-Architektur

Bis vor wenigen Jahren wurden in der Industrie ausschließlich objektorientiert entwickelte Applikationen als modern und innovativ angesehen. Die objektorientierte Softwareentwicklung hat es aber nicht geschafft den Grad der Wiederverwendbarkeit von Software signifikant zu steigern [JBR98], bis auf die wenigen Erfolge, die durch die Verwendung von Bibliotheken mit einer Sammlung von Standardklassen erzielt wurden. Insbesondere ist es der objektorientierten Softwareentwicklung nicht gelungen das seit Jahrzehnten bestehende zentrale Problem der Software-Technik zu lösen: Software nicht mehr in den Entwicklungsabteilungen der Unternehmen zu implementieren, sondern in deren Fertigungsabteilungen zu produzieren [McI68].

Die Methode der komponenten-orientierten Softwareentwicklung ist angetreten, um für diese Problematik eine bessere Lösung anzubieten. Als Vorbild für die komponenten-orientierte Softwareentwicklung dient das klassische Ingenieurwesen. Hier werden komplexe Systeme aus einfachen vorgefertigten Komponenten erstellt. Solche Komponenten repräsentieren wiederverwendbare Bauteile, mit denen durch intelligente Komposition wiederholt neue Systeme entstehen können.

Die Industrie verspricht sich von der Übertragung des Komponentenkonzeptes

auf die Softwareentwicklung eine deutliche Kostenreduktion, eine Steigerung der Qualität von Software, einfachere Wartbarkeit und einen großen Schritt in Richtung der standardisierten Fertigung von Software. Die **komponentenorientierte Softwareentwicklung** transferiert das Komponentenkonzept der Ingenieurdisziplin in die Software-Welt. Eine wichtige Frage hierbei ist, wie der Begriff der Software-Komponente zu definieren ist.

Brad Cox hatte in seinem Buch *Object-Oriented-Programming: An Evolutionary Approach* als erster die Idee der Software-Komponente beschrieben [Cox86]. Cox nannte Komponenten **Software ICs** analog zu den Hardwarekomponenten der Mikroelektronik. Dort bietet beispielsweise die Hauptplatine eines Personal Computers standardisierte und normierte Schnittstellen an, die mit Steckkarten bestückt werden. Jede Karte liefert eine Teilfunktionalität für das Gesamtsystem. Der Benutzer kann das System entsprechend seiner Anforderungen individuell ausstatten und so ein funktionierendes Gesamtsystem erstellen. Entspricht das System im Laufe der Zeit nicht mehr seinen Ansprüchen, so kann er zum Beispiel eine Grafikkarte durch eine leistungsstärkere Karte ersetzen.

Ein Softwaresystem ist als eine Art Hauptplatine anzusehen, dass über Schnittstellen Software ICs einbindet. Zum Beispiel ist ein Editor oder eine Rechtschreibkorrektur eine Ausprägung für ein Software IC, da hierdurch eine dedizierte Funktionalität offeriert wird. Entspricht der eingebundene Editor nicht mehr den Wünschen der Anwender, besteht die Möglichkeit ihn gegen einen anderen auszutauschen, wenn dieser bezüglich der Schnittstelle auf der Hauptplatine kompatibel ist.

Zur Zeit existiert keine allgemein gültige und einheitliche Definition des Terminus Software-Komponente. Eine hinreichend brauchbare und kurze Definition liefert Szyperki in [Szy02]:

Definition: 21 *A **Software component** is a unit of composition with contractually specified interfaces and explicit context dependencies only. A Software component can be deployed independently and is subject to composition by third parties.* //

Die Definition enthält alle relevanten Merkmale von Software-Komponenten.

Komponenten werden als Teil einer Komposition beschrieben. Um bestimmte Aufgaben zu erfüllen, interagieren sie mit anderen Komponenten über Schnittstellen. Eine Schnittstelle stellt den Interaktionspunkt für andere Komponenten dar. Aus technischer Sicht repräsentiert eine Schnittstelle eine Menge von Methoden, die andere Komponenten aufrufen können. Jede Methode einer Komponente ist mit einer einzuhaltenden Vor- bzw. zugesicherten Nachbedingung im

Sinne eines *programming by contract* ausgestattet [Mey97]. Die Implementierungsdetails der Methoden werden von der Komponente gekapselt und bleiben für die Außenwelt verborgen. So ist eine Komponente leicht durch eine neue oder verbesserte Komponente auszutauschen, wenn sie das gleiche Interface realisiert.

Die Eigenständigkeit und Autonomie einer Komponente erlaubt ihren Einsatz losgelöst von anderen eingesetzten Komponenten. Unternehmen können so eine beliebige Komponente auswählen und bedingt durch ihre explizite Wiederverwendbarkeit für verschiedene Aufgabenstellungen verwenden. Um ein Gefühl dafür zu entwickeln, was unter einer Software-Komponente zu verstehen ist, wird ein einfaches Beispiel aus der UNIX/LINUX-Welt vorgestellt [Gri98]. Dort werden Kommandozeilen-Befehle verwendet, die alle Eigenschaften einer Komponente besitzen und insbesondere immer wieder verwendbar sind. Die Befehlsfolge

cat telefondei — sort — uniq — more

erzeugt eine sortierte Liste mit Telefonnummern, die von Mehrfacheinträgen bereinigt ist und seitenweise auf dem Bildschirm angezeigt wird. Die LINUX-Kommandos *cat*, *sort*, *uniq* und *more* sind eigenständige, funktionale Einheiten, die eine einfache Kopplung über den UNIX/LINUX-Filtermechanismus ermöglichen um miteinander zu interagieren. Wobei die richtige Auswahl und Kombination dieser Software-Komponenten hier die Herausforderung darstellt.

Die Applikation, die als Ergebnis einer komponenten-orientierten Softwareentwicklung entsteht, wird als **Komponenten-Software** oder als **Componentware** bezeichnet. Componentware stellt den Mittelweg zwischen Individual- und Standardsoftware dar.

Abschließend ist noch zu erwähnen, dass Software-Komponenten eine Kontextabhängigkeit bezüglich ihrer Umgebung besitzen. Für Komponenten ist, daher neben dem Interface auch die Laufzeitumgebung zu spezifizieren, in der sie zum Einsatz kommen. Dies ist notwendig, da zur Zeit verschiedene Komponentenmodelle mit jeweils unterschiedlichen Laufzeitumgebungen in der IT-Welt nebeneinander zum Einsatz kommen.

Das in der J2EE Spezifikation enthaltene **Enterprise JavaBeans Komponentenmodell** ist auf der Mittel-Schicht angesiedelt und ermöglicht die Erstellung von serverseitigen Software-Komponenten in Java [Sun03]. Die erstellten Software-Komponenten werden gleichzeitig als **Enterprise Java Beans** - kurz

Beans oder **Enterprise Beans** - bezeichnet. Sie beinhalten die Applikationslogik, haben keine Benutzeroberfläche und benötigen einen **J2EE Applikationsserver** als Laufzeitumgebung [Sun04].

Ein Applikationsserver, der die J2EE Spezifikation unterstützt, wird als J2EE konformer Applikationsserver bezeichnet. Die **Java Virtual Maschine** bildet die Basis, auf der ein Applikationsserver arbeitet. Der Applikationsserver realisiert die Lastverteilungs- und Fehlertoleranzfunktionen sowie die Verwaltung der Server-Ressourcen. So ist sichergestellt, dass ohne allzu hohen Programmieraufwand skalierbare, robuste, leistungsfähige und einfach verwaltbare Software-Komponenten erstellt werden können.

Neben der Variante, dass im Browser ausschließlich HTML-Seiten zur Beschreibung der Benutzerschnittstelle verwendet werden und die komplette Applikationslogik auf die Mittel-Schicht ausgelagert wird, gibt es noch zwei andere Varianten (siehe Abbildung 6.6).

Zum einen besteht die Möglichkeit Applets einzusetzen und zum anderen können Applikationen, die auf der Java 2 Plattform basieren, Verwendung finden. Ein **Applet** ist ein Java Programm, das in einem Browser läuft und den Teil der Applikationslogik enthält, der aus Performance Sicht unkritisch ist und über HTTP/HTTPS mit der Mittel-Schicht kommuniziert. Eine Java 2 Plattform kompatible Applikation wird auf einem Arbeitsplatzrechner installiert und braucht zur Ausführung keinen Web Browser. Sie ist eine Client-Anwendung, die aus einer Präsentationsschicht besteht und einen Teil oder die ganze Applikationslogik abbildet. Die Kommunikation zwischen der Client-Anwendung und der Daten-Schicht und/oder der Mittelschicht kann über das Internet/Intranet erfolgen [Sun03].

In Abschnitt 5.6 auf Seite 80 wurde von einem semi-automatischen Service gefordert, dass keine lokale Installation eines Entwurfswerkzeuges nötig ist. Der Aufwand für Installation, Wartung und Betrieb solcher Entwurfswerkzeuge ist im Allgemeinen sehr hoch und widerspricht dem Prinzip, dass ein Service schnell und einfach zu nutzen ist. Auf der anderen Seite sprechen einige Argumente für solche lokalen Entwurfswerkzeuge:

- Sie unterstützen ein anspruchsvolles User Interface, wohingegen mit statischen und dynamischen Web-Seiten eher ein einfaches User Interface zu beschreiben ist
- Sie sind schnell über den Desktop zu starten und benötigen nicht unbedingt einen Web-Browser als Laufzeitumgebung. Dagegen ist ein Applet

bei jeder Nutzung wieder runter zu laden und seine Nutzung im Web-Browser ist für manche Akteure gewöhnungsbedürftig

- Sie sind auch offline zu betreiben und ihre Antwortzeiten sind dann unabhängig von der Verbindungsgeschwindigkeit

Um diese Vorteile zu nutzen und die genannten Nachteile zu eliminieren, ist von Sun Microsystems die **Java Web Start Technology** eingeführt worden. Sie ermöglicht es, Anwendungen, die auf der Java 2 Plattform basieren, auf beliebige Betriebssystem-Plattformen auszuliefern. Die Bereitstellung der Applikationen geschieht auf eine sehr einfache und sichere Art ohne von dem Akteur Installationswissen oder Kenntnisse bezüglich der Laufzeitumgebung zu verlangen. Aus diesem Grund sind semi-automatische Services, die auf Basis von Java 2 Plattform konformen Entwurfswerkzeugen angeboten und über die Java Web Start Technology installiert werden, als „echte“ Design Service anzusehen. Eine ausführliche Darstellung der Java Web Start Technology und Architektur erfolgt in [Sun02].

In der J2EE-Spezifikation erfolgt eine Aufteilung der Mittel-Schicht in die Teile *server-side presentation* und *server-side domain-logic* (siehe Abbildung 6.6). Im **server-side domain-logic** Teil sind die Enterprise Beans mit dem Applikationsserver als Ablaufplattform platziert.

Der Teil **server-side presentation** dient zur Aufbereitung von Informationen für die Präsentationsschicht. Hier sind Applets, reine HTML-Clients und Java Clients einsetzbar. Für die Präsentationsaufbereitung kommt ein Web-Server zum Einsatz, der sowohl statische als auch dynamische Web-Seiten bearbeitet und Anfragen mit Hilfe von HTTP als Web-Protokoll beantwortet. Um eine effiziente Verarbeitung von dynamischen Web-Seiten zu ermöglichen sind die so genannten **Servlets** eingeführt worden. Ein Servlet ist ein auf der Serverseitigen Präsentationsschicht ablaufendes Java-Programm, das HTTP-Requests eines Web-Browser-Clients entgegen nimmt, wie zum Beispiel die Benutzereingabe über eine HTML-Seite. Es überprüft und bewertet die Eingabe und leitet sie dann an ein EJB weiter, um dort den eigentlichen Bearbeitungsprozess zu aktivieren. Ebenso ist das Ergebnis einer EJB-Komponente von einem Servlet abzurufen, in eine HTML-Ausgabe umzuwandeln und dann via HTTP an den Web-Client zu senden.

Die Anforderung der Entwickler eine klare Aufteilung zwischen Anwendungsprogrammierung und Oberflächengestaltung zu erreichen, wurde mit der Einführung von **Java Server Pages (JSP)** erfüllt. JSP ermöglichen eine saubere Trennung der Rollen von Servlet- bzw. EJB-Entwickler einerseits und

Webseiten-Grafikdesignern andererseits. Die Entwicklung von Servlets erfordert umfangreiche Java Programmierkenntnisse, die oft bei Web-Designern nicht sehr ausgeprägt sind. Umgekehrt sind Software-Entwickler, nicht die geborenen Web-Designer. JSP erlauben es Web-Designer schnell HTML-Code zu erzeugen und bieten ihnen eine Reihe von vordefinierten Java Funktionen an, so genannte **Tag Libraries**. Der Web-Designer kann sehr einfach in seiner HTML-Seite diese Funktion aufrufen ohne sich mit Implementierungsdetails zu beschäftigen. Der Funktionsaufruf führt dann zu einem server-seitigen Servlet und EJB Aufruf.

Factors	Applets	XML/HTML-based clients	Java Web Start
User Interface	Moderate to sophisticated	Simple to moderate	Moderate to sophisticated
Offline support	No	No	Yes
UI response	Network independent	Network dependent	Network independent
Interactivity	Browser limited	Browser/markup limited	Open
First use response	Minutes	Seconds	Minutes
Subsequent use response	Minutes	Seconds	Seconds
Bandwidth usage	Variable	Fixed	Flexible
Lightweight client support	Limited	Open	Limited

Tabelle 6.1: Vergleich von Web-Präsentationstechniken [Sun01]

Um den Bedarf an semi-automatischen Design Services zu befriedigen, ist bei der Entwicklung von neuen Entwurfswerkzeugen die in Abbildung 6.6 vorgestellte Zielarchitektur zu berücksichtigen. Sie repräsentiert die gute Grundlage, um die für KMU dringend benötigten semi-automatischen Services anzubieten. Diese sind über drei verschiedene Mensch-Maschine orientierte Präsentationstechniken (pure HTML, Applet oder pure Java Applikation) anzubieten. Welche Präsentationstechnik für ein Entwurfswerkzeug zu wählen ist, hängt von verschiedenen Faktoren ab. Die Tabelle 6.1 gibt einen Überblick über die relevanten Faktoren. Die Tabelle kann als Entscheidungshilfe bei der Planung von neuen service-orientierten Entwurfswerkzeugen dienen.

6.4.3 Anwendungsbeispiele

Zwei konkrete Anwendungsbeispiele für semi-automatische Services sind am Institut für Mikrosystemtechnik der Universität Siegen in den letzten Monaten entwickelt worden. Zum einen wurde im Rahmen eines Kooperationsprojektes mit der Robert Bosch GmbH das Entwurfswerkzeug **PRINCE** (**PR**ocess

INformation and management CEnter) entwickelt [HPW03] und zum anderen ist als Ergebnis der Diplomarbeit von T. Schmidt [Sch04] eine Material- und Effektdatenbank für Simulation und Analyse in der Mikrosystemtechnik entstanden. Beide Werkzeuge basieren auf einer 3-Schichten-Architektur und sind in Java realisiert. Sie besitzen ein anspruchsvolles User Interface und bieten ihre Services über die Java Web Start Technology an.

PRINCE liefert Design Services für die Entwurfsaufgaben, die im Rahmen der Generierung der physikalischen Beschreibung zu erledigen sind und nimmt dem Experten das rein manuelle Erstellen der Fertigungsbeschreibung zu einem großen Teil ab. Die Services helfen ihm bei dem komplexen Vorgang die richtigen Parameter, Materialien und Fertigungstechniken für die einzelnen Fertigungsprozessschritte zu bestimmen. Folgende Services bietet PRINCE zur Zeit an:

- Wissenssuche: Der Service erlaubt, die konsistente Speicherung und Verwaltung von kompletten Fertigungsprozessen oder von einzelnen Fertigungsschritten inklusive der dazugehörigen Materialien, Parametern, Techniken und Verweisen auf die Masken in einer Wissensdatenbank
- Fertigungsprozess-Modellierung: Der Service ermöglicht, die grafische Eingabe von Fertigungsprozess-Schrittfolgen über ein Benutzerinterface
- Generierung von PDML-Dateien [Kle02]: Der Service bietet das Erzeugen von PDML-Dateien aus den grafisch modellierten Fertigungsprozess-Schrittfolgen an.

Weitere Informationen und eine Leistungsbeschreibung der Services sind in der Diplomarbeit von A. Wagener [Wag01] zu finden. Abbildung 6.7 zeigt Ausschnitte aus der PRINCE Benutzeroberfläche.

Das Interessante an der 3-Schichten-Architektur von PRINCE ist, dass neben dem bestehenden User Interface auf Basis der Java Web Start Technology ein neues zusätzliches Benutzerinterface relativ einfach zu realisieren wäre. Mit Hilfe von dynamischen HTML-Seiten ist beispielsweise nur der Service Wissenssuche über eine URL im Internet oder Intranet den Akteuren anzubieten.

Die Idee der Diplomarbeit von T. Schmidt war es, den Akteuren in der Mikrotechnik-Industrie einen Material-Datenbank-Service zugänglich zu machen, der die benötigten Materialeigenschaften wie Bruchfestigkeit und Dichte für den Fertigungsprozess liefert und ebenso die Wechselwirkungen der Materialien untereinander berücksichtigt. Zusätzlich ist er in der Lage, Effekte, die bei der Fertigung auftreten, wie Löslichkeit, Haftung und chemische Reaktionen, mit Hilfe

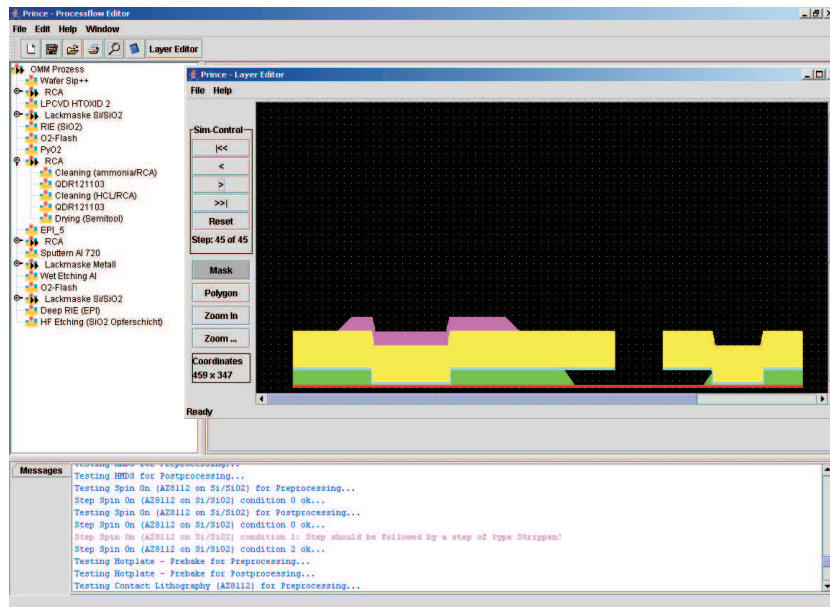


Abbildung 6.7: Beispiel PRINCE User Interface

von Berechnungsvorschriften zu verwalten und deren komplexe Eigenschaften zu modellieren.

Um die Qualität und Leistungsfähigkeit des Services unter Beweis zu stellen wurde ein Ätz-Simulation Service entwickelt, der zur Durchführung der Simulation den Material-Datenbank-Service nutzt. In Abbildung 6.8 ist ein Teil der Benutzeroberfläche des Simulations-Service zu sehen.

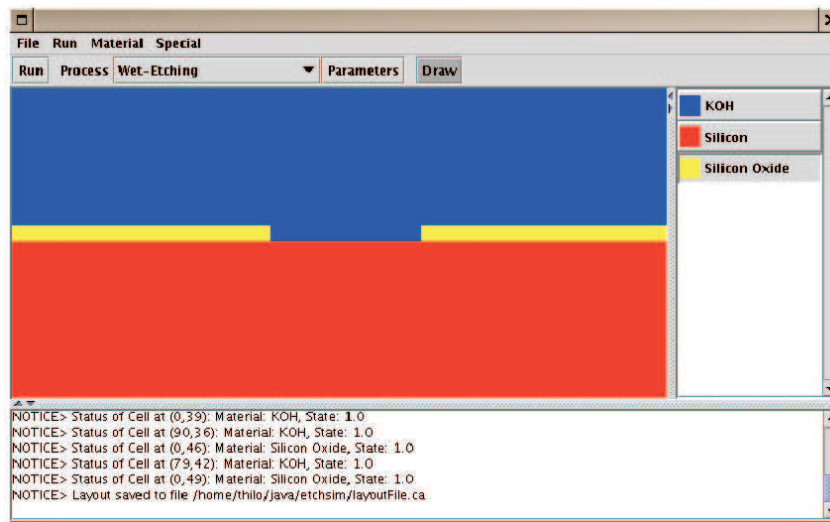


Abbildung 6.8: User Interface des Ätz-Simulations Service

Das zu simulierende Layout ist durch den großen Bereich in der Mitte des Bildes zu erkennen. Die Liste mit den im Layout eingesetzten Materialien ist im Auswahlfenster rechts zu sehen. Der obere Rand erlaubt die Selektion des Fertigungsprozessschrittes und das Starten der Simulation.

Aktuelle Arbeiten beschäftigen sich mit der Integration des PRINCE Service-Wissenssuche, um so auf vollständige Fertigungsprozesse zurückzugreifen und diese als Eingabe für den Simulations-Service zu verwenden. Zuvor ist die Leistungsfähigkeit des momentan sehr einfachen Simulations-Service zu erhöhen.

An diesen beiden Beispielen für semi-automatische Services wurde verdeutlicht, wie Entwurfsaufgaben schneller und einfacher zu erledigen wären. Die Entwicklungszeiten der Services, die im Bereich von 6 bis 36 Monaten lagen, verdeutlichen das hier ein neuer Weg existiert, um relativ schnell den Bedarf an service-orientierter Unterstützung in der Mikrotechnik-Industrie zu befriedigen.

6.5 Voll-automatische Design Services

Bei dem voll-automatischen Service existiert keine direkte Kommunikation mit dem Experten, sondern hier reduziert sich das Ganze auf eine reine Maschine-Maschine Kommunikation. Ein Service, der in dieser Form verfügbar ist, wird mit Hilfe eines Web Service realisiert. Aktuelle Entwurfswerkzeuge der Mikrotechnik bieten zur Zeit keine Web Service an, obwohl sie die ideale Technik darstellen, wie die nachfolgenden Ausführungen beweisen, um dedizierte Funktionen von bestehenden Frameworks zu extrahieren und als Service anzubieten.

6.5.1 Web Services

Ein **Web Service** basiert auf zwei ausgereiften und erprobten Technologien, deren Konzepte bereits vor fast vierzig Jahren entwickelt wurden. Diese sind zum einen **RPC (Remote Procedure Call)** und zum anderen **XML (eXtensible Markup Language)**. Bei einem Remote Procedure Call ruft ein Client eine Methode auf einem entfernten Server auf. Dadurch werden Daten zwischen zwei Systemen über ein Netzwerk übertragen. Der Nachteil von RPCs war damals ihre Plattform- und Sprachabhängigkeit.

XML als eine Sprache zur Beschreibung von Dokumenten, Metainformationen und Daten, die zwischen Rechnern zu transferieren sind, hat ihre Wurzeln in den Entwicklungslabors von IBM und der Graphic Communication Association,

die bereits Ende der sechziger Jahren die Grundlage für Auszeichnungssprachen gelegt haben.

Um den Web Service Cocktail fertig zu mixen, fällt neben dem entfernten Methodenaufruf und XML als einheitliches Datenformat noch ein Protokoll zur Rechnerkommunikation, das im Gegensatz zu **IIOP (Internet Inter-ORB Protocol)** oder **RMI (Remote Method Invocation)** einfach, plattformunabhängig und über das Internet funktioniert. Bevor dieses Protokoll vorgestellt wird, erfolgt zum besseren Verständnis die Motivation für die Einführung von Web Services in der IT-Industrie.

Um die Motivation für die Entwicklung von Web Services zu ergründen, bedarf es eines Blickes zurück in die Geschichte des Internets. Zu Beginn der kommerziellen Nutzung des Internets wurde es primär als neuer, einfacher und relativ billiger Vertriebskanal eingesetzt. Konsumenten wurden mit bunten Webseiten beworben und animiert, Warenbestellungen über das Internet aufzugeben. Dieser Bereich der betriebswirtschaftlichen Geschäftsabwicklung wird als **Business-to-Consumer (B2C)** bezeichnet, da eine Beziehung zwischen dem Anbieter des Geschäfts (Business) und dem Endverbraucher (Consumer) hergestellt wird.

Im B2C Bereich hat sich **HTTP (Hypertext Transfer Protocol)** als Standard Transportprotokoll etabliert, nicht aus Qualitätsgründen, sondern weil es das Web-Protokoll der ersten Stunde war, über das die Anbieter ihre statischen Webseiten im WWW transferierten. Um HTTP-Requests entgegenzunehmen sind die Ports 80 (HTTP) und 443 (HTTPS) in Firewalls von Unternehmen fast immer geöffnet. Im Gegensatz dazu konnten sich verbindungsorientierte und komfortabler nutzbare Internet/Intranet-Protokolle wie zum Beispiel RMI oder RMI-IIOP nicht durchsetzen, da die Firewalls für diese in der Regel nicht durchlässig waren. Ihr Einsatzbereich ist daher meist auf das Intranet und Extranet beschränkt.

Neben HTTP hat sich **HTML (HyperText Markup Language)** im B2C-Bereich als Standard zur Beschreibung von Webseiten heraus kristallisiert. Beide Technologien bilden zusammen eine gute Plattform zur Durchführung der Maschine-Mensch Kommunikation über einen Web-Browser. Was liegt näher als das so erfolgreiche Duo aus HTTP und HTML in den **Business-to-Business (B2B)** Bereich zu übertragen. Dort besteht auch die Herausforderung einen Standard für die elektronische Geschäftsabwicklung über das Internet zu etablieren. Die Internet basierte Abwicklung von betriebswirtschaftlichen Geschäftsprozessen über Unternehmensgrenzen hinweg ist zu realisieren, um dadurch Kosten einzusparen und die gesamte Durchführung zu beschleunigen.

Zwischen Unternehmen HTML-Dokumente über HTTP auszutauschen und dabei die Inhalte der Dokumente nicht anzuzeigen, sondern die relevanten Informationen maschinell zu extrahieren und danach automatisch weiter zu verarbeiten, hat sich als nicht praxistaugliches Vorgehen herausgestellt. HTML-Seiten enthalten normalerweise nicht nur Nutzdaten, sondern auch Informationen bezüglich der Präsentation, wie Schriftgröße, Positionierung von Grafiken und andere Layouthinweise. Insbesondere fehlen Meta-Informationen, welche die Semantik der Nutzdaten beschreiben. Aus diesem Grund ist HTML nicht das geeignete Format um Daten maschinell zu interpretieren und diese danach in irgendeiner Form weiterzuverwenden.

An Stelle von HTML ist daher XML ausgewählt worden, um Dokumente über HTTP zu übertragen. XML trennt strikt die Nutzdaten und die Beschreibung ihrer Bedeutung von einander. So bestand die Möglichkeit Daten zwischen Unternehmen zu verteilen und weiter zu bearbeiten. Als offener Punkt blieb die Frage übrig: Wie ist ein Entwurfsschritt von beliebigen Unternehmen durchzuführen? Zum Beispiel besteht der Wunsch der Firma A den Entwurfsschritt der Layoutverifikation von der Firma B durchführen zu lassen. Die Antwort auf die Frage ist eng mit der Geburtsstunde von Web Services verknüpft und lautet **SOAP (Simple Object Access Protocol)**. Mit SOAP ist der fehlende Bestandteil vorhanden, um einen Web Services Cocktail technisch zu mixen. SOAP ist das gesuchte Protokoll zur plattformunabhängigen Rechnerkommunikation, das auf HTTP aufsetzt und ohne Konfigurationsänderungen an den Firewall-Systemen funktioniert [Sch02a].

SOAP ist vom **World Wide Web Consortium (W3C)** neu entwickelt worden und basiert auf einer XML-Syntax⁵. Ein SOAP-Aufruf ist ein XML-Dokument, welches einen entfernten Methodenaufruf formuliert und mittels HTTP übertragen wird. Damit ist ein Web Service aus programmtechnischer Sicht nicht anderes als ein Funktionsaufruf. Wobei die eigentliche Berechnung des Funktionswertes nicht auf dem Rechner stattfindet, der den Aufruf der Funktion initiiert, sondern auf dem Server, der die Funktion (den Service) zur Verfügung stellt. Da das Internet als Trägermedium involviert ist, spricht man in der IT-Gemeinde nicht nur von einem Service, sondern von einem Web Service.

Mit SOAP ist von der W3C ein Standard formuliert worden, um im B2B-Bereich entfernte Methodenaufrufe zu formulieren, der schnell populär wurde und mittlerweile auch als Protokoll zwischen beliebigen Clients und Server zum Einsatz kommt. Hierbei legt SOAP lediglich die Formatierung der XML-Daten fest.

⁵Die SOAP Spezifikation ist unter <http://www.w3c.org/> erhältlich.

XML-Daten die gemäß dem SOAP-Format codiert sind, werden als **SOAP-Nachricht** bezeichnet. Die Übertragung einer SOAP-Nachricht und insbesondere die technische Umsetzung eines RPCs auf dem jeweiligen Zielrechner ist nicht Aufgabe von SOAP. Das ist ein entscheidender Vorteil von SOAP und somit auch von Web Services. Der Anbieter der Plattform, auf der ein RPC ausgeführt wird, muss sich um die Implementierungstechnik des Web Services kümmern. Der Konsument eines Web Services kann den eigenen Wissenshorizont auf seine Plattformumgebung beschränken und sich voll auf den Nutzen des Dienstes konzentrieren.

Abbildung 6.9 zeigt die Schritte, die notwendig sind, um einen Web Service durchzuführen und enthält die daran beteiligten Technologien.

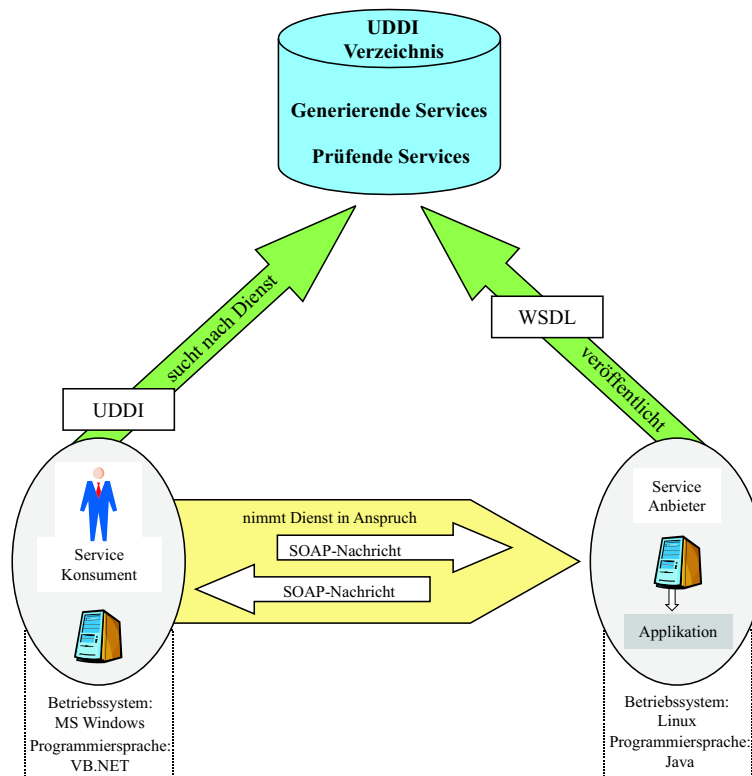


Abbildung 6.9: Web Service [Lan03]

Es gibt eine klare Trennung zwischen dem Service-Konsumenten und dem Service-Anbieter. Der Service-Konsument nimmt eine Dienstleistung oder einen Service eines Service-Anbieters in Anspruch. Dies geschieht, indem der Konsument einen entfernten Methodenaufruf auf dem Server ausführt. Der Service-Anbieter wird auch als Dienstleister bezeichnet, weil er eine dedizierte Dienstleistung der Öffentlichkeit zur Verfügung stellt. Eine Dienstleistung kann zum

Beispiel die Lieferung einer Fertigungsspezifikation sein, die gesucht und gefunden wurde, eine Transformation dieser in ein gewünschtes Ausgabeformat und Informationen über die erlaubte Nutzungsdauer der Spezifikation und die Nutzungsgebühr.

Um seine Dienstleistung eindeutig zu dokumentieren, benutzt der Service Anbieter die auf XML basierende **Web Service Definition Language (WSDL)**. Sie erlaubt es, in einem standardisierten Format die Dienstleistung zu beschreiben. Technisch gesehen ist durch WSDL definiert, welche Methoden eine Serverkomponente zum einen anbietet, und zum anderen werden für jede dieser Methoden die Übergabeparameter und Rückgabewerte festgelegt. Für einen Web Service der mit Hilfe von WSDL dokumentiert wurde, besteht der Bedarf der Veröffentlichung. Dies geschieht durch eine Eintragung in ein Verzeichnis, das für alle Konsumenten frei zugänglich ist. Konsumenten können so einen Überblick über die potentiell verfügbaren Web Services erhalten. Sie bekommen beispielsweise Informationen über Leistungsumfang der Dienstleistung, den Leistungsort, den Anbieter, die Verfügbarkeit, das Nutzungsrecht und die Nutzungskosten. Das Verzeichnis hat prinzipiell eine ähnliche Funktion wie die Gelben Seiten der Deutschen Telekom. Hier bieten Handwerker ihre Dienstleistung an und geben Auskunft über die Art der Tätigkeit.

Mit **Universal Description, Discovery and Integration (UDDI)** besteht die Möglichkeit einen universellen und globalen Verzeichnisdienst für Web Services bereitzustellen. Die WSDL-Informationen werden in einem UDDI-Verzeichnis abgelegt und erlauben es Web Services über eine festgelegte Taxonomie zu suchen.

WSDL ermöglicht eine vollständige Beschreibung des Dienstleistungs-Angebotes eines Service-Anbieters. Mit UDDI existiert ein Standard, um WSDL-Informationen den Konsumenten zugänglich zu machen. HTTP fungiert als Trägerprotokoll, um den Informationsaustausch zwischen Konsumenten, Dienstleister und UDDI-Verzeichnis zu ermöglichen.

Damit ist der technische Rahmen geschaffen worden, um im nächsten Abschnitt Einsatzmöglichkeiten für Web Services in der Mikrotechnik zu betrachten.

6.5.2 Anwendungsbeispiele

Konkrete Beispiele für den Einsatz von Web Services in der Mikrotechnik-Industrie sind momentan nicht bekannt, daher wird ein Überblick über die potentiellen Einsatzmöglichkeiten von Web Services gegeben. Er soll die Be-

deutung von Web Services für die Entwurfsunterstützung in der Mikrotechnik verdeutlichen.

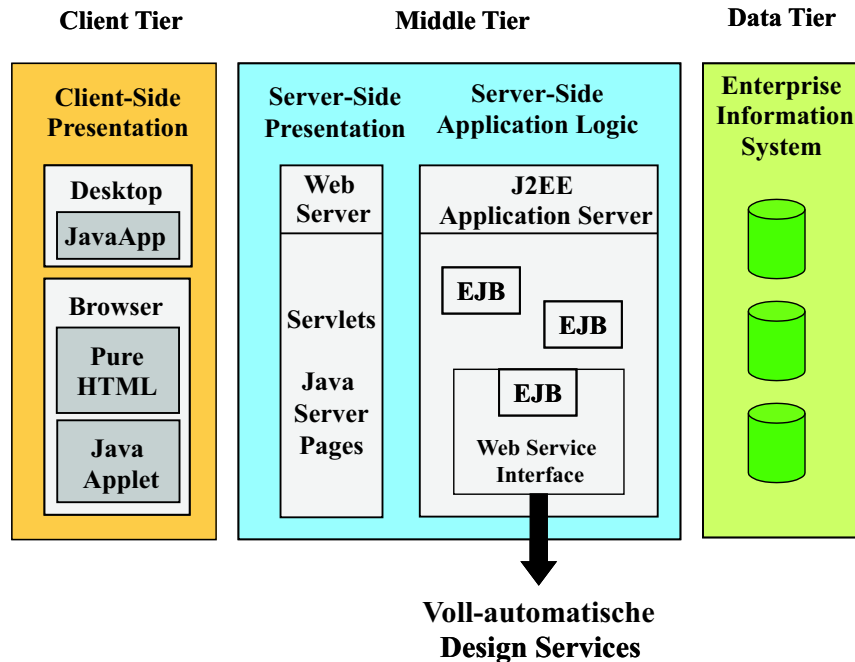


Abbildung 6.10: Web Services und 3-Schichten-Architektur

Ausgangspunkt für die Überlegungen von Einsatzmöglichkeiten sind die zuvor vorgestellten Anwendungsbeispiele für semi-automatischen Services in der Mikrotechnik-Industrie. Ihre 3-Schichten-Architektur bietet eine hervorragende Grundlage um Web Services effizient zur Verfügung zu stellen. Die Enterprise Java Beans können ihre Applikationslogik direkt über ein WSDL-Interface als Web Service anbieten (siehe Abbildung 6.10).

Der Zugriff auf den Service kann jetzt entweder über ein Mensch-Maschine- oder über ein Maschine-Maschine Interface geschehen. Die Herausforderung liegt in der Spezifikation des Service-Interfaces und damit in der Festlegung der WSDL-Datei. Ein Web Service Wissenssuche kann beispielsweise als Übergabeparameter die Suchanfrage erhalten und als Rückgabewert eine Liste von Fertigungsspezifikationen liefern, die nach Kriterien wie Fertigungsdauer und Kosten sortiert ist. Genauso liefert ein Web Service Material-Datenbank eine Liste von Materialien, die mit den gesuchten Materialeigenschaften übereinstimmen und zusätzlich nach Verfügbarkeitsgesichtspunkten sortiert ist.

Um zu zeigen, dass bestehende Frameworks oder Tools auch auf Web Services umgestellt werden können, soll das an der Universität Dortmund in den 90

Jahren entwickelte LIDO-System dienen [Hah99]. Hier besteht die Möglichkeit, die hochwertigen in C++ implementierten Algorithmen zur Prozess- und Layoutverifikation über Web Services zur Verfügung zu stellen. Hierfür ist ein Wrapper als Schale zu bauen und ein entsprechendes WSDL Service Interface ist zu spezifizieren. Jetzt ist der Dienst im UDDI-Verzeichnis anzumelden und kann dann über die Kommunikationsinfrastruktur den Akteuren zur Verfügung gestellt werden.

Eine Renaissance von bereits etwas in die Jahre gekommenen Tools ist möglich, indem die Teile über Web Services anzubieten sind, die für den DA-Markt heute noch Relevanz besitzen. Gleiches gilt für die Frameworks, die über ein gezieltes Angebot von Web Services ihren Nutzen neu definieren können. Zum Beispiel bietet es sich an, einen Ätz-Simulations-Service zu offerieren und eine Abrechnung auf *pay-per-use* Basis einzuführen. KMU könnten so in den Genuss von sonst nur schwer finanzierbaren Entwurfswerkzeugen kommen. Gleichzeitig besteht auch eine Chance für kleine DA-Spezialanbieter sich neben den großen Framework-Lieferanten zu positionieren und hochwertige Web Services für die Mikrotechnik-Industrie anzubieten.

6.6 Ein Medium für den zentralen Zugriff

Die globale Verfügbarkeit von Design Services ist durch den Einsatz des Internets als Kommunikationsinfrastruktur sichergestellt. Jeder Akteur erhält den gewünschten orts- und zeit-unabhängigen Zugriff auf Services. Um eine zentrale Stelle mit einen Überblick über die vorhandenen Services den Akteuren zur Verfügung zu stellen, wird die Einführung eines Internet-Portals vorgeschlagen. Die folgende Definition legt fest, was unter einem Internet-Portal zu verstehen ist:

Definition: 22 *„Portal is a term, generally synonymous with gateway, for a World Wide Web site that is or proposes to be a major starting site for users when they get connected to the Web or that users tend to visit as an anchor site. There are general portals and specialized or niche portals. Some major general portals include Yahoo, Excite, Netscape, Lycos, CNET, Microsoft Network, and America Online’s AOL.com. Examples of niche portals include Garden.com (for gardeners), Fool.com (for investors), and Search-Networking.com (for network administrators)“ [BN03]. //*

Zur Umsetzung der Anforderungen, die sich aus dem service-orientierten-Entwurfsszenario ableiten, ist ein Spezial-Portal einzuführen, das Design Service Portal:

Definition: 23 *Ein Design Service Portal ermöglicht Design Teams, DA Support Abteilungen, DA Vendors und Foundries einen zentralen, einheitlichen, zeit- und orts-unabhängigen Zugang zu Design Services.* //

Der URL Aufruf <http://www.design-service.org/> eines Web-Browsers aktiviert die Startseite des Design Service Portals. Das Portal legt ein einheitliches Layout, die Navigation und eine inhaltliche Struktur für die Gesamtheit der zum Portal gehörenden Web-Seiten fest. Das Portal ermöglicht die Suche und den Zugriff auf UDDI Verzeichnisse und damit auf Web Services. Es gibt einen Überblick über manuelle- und semi-automatischen Services in Form einer URL-Liste.

Die einen Akteure stellen Ihre Services über das Portal vor und die anderen nutzen diese für die Service Flow Erstellung (siehe Abbildung 6.11).

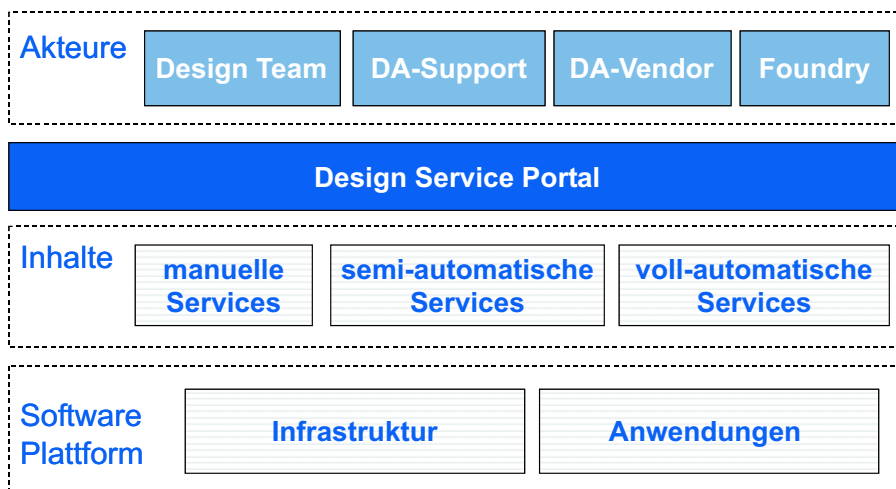


Abbildung 6.11: Design Service Portal

Das Portal ist als Marktplatz für Design Services anzusehen. Es verschafft den Akteuren einen Überblick über das aktuelle Angebot an Services. Jeder Service sollte eindeutig charakterisiert sein, indem seine Leitungsmerkmale, Nutzungskosten und Verfügbarkeit benannt werden. Es sollte die Möglichkeit bestehen, einzelne Services im beschränkten Rahmen zu testen. Um eine schnelle Zuordnung von Services zu Design Tasks vorzunehmen, ist eine Kategorisierung in

bestimmte Serviceklassen wünschenswert. Es bietet sich an, die in der Entwurfsmethodik vorgestellten Funktionen als Orientierungshilfe beim Aufbau eines Servicekatalogs zu verwenden. Neben der Ausrichtung als Marktplatz ist das Portal als Kollaborationsmedium einzusetzen, um bei der Suche nach Akteuren für ein Design Flow Entwicklungsprojekt zu assistieren.

Eine weitere Anforderung an das service-orientierte-Entwurfsszenario ist durch das Konzept des Portals erfüllt worden. Im nächsten Abschnitt wird die software-technische Umsetzung der Design Flow Entwicklung und Durchführung vorgestellt.

6.7 Design Flow Entwicklung und Durchführung

Die aktive Hilfe bei der Design Flow Entwicklung und Durchführung durch eine Softwarelösung ist das zentrale Thema dieses Abschnitts. Bei der Design Flow Entwicklung geht es primär um das Formulieren von präzisen Entwurfsaufgaben und die Suche nach geeigneten Design Services. Für diesen Vorgang ist ein DA-Tool zu verwenden, das die drei elementaren Schritte bei der Design Flow Entwicklung software-technisch unterstützt.

Die Hauptaufgabe einer Software für die Durchführung eines Design Flows liegt darin, den Akteur bei der Bearbeitung der Entwurfsschritte zu assistieren und am Ende die Fertigungsspezifikation zu ermitteln. Entscheidend ist das Abstraktionslevel auf dem der Entwurfsvorgang stattfindet. Ein Softwarelösung sollte als Anwendungsgruppe nicht nur Experten mit tiefgehendem Know-how im Bereich des physikalischen Entwurfs adressieren, sondern ebenfalls für Produktdesigner geeignet sein.

6.7.1 Design Flow Entwicklungsumgebung

Die **Design Flow Integrated Development Environment (IDE)** ist eine softwaregestützte Entwicklungsumgebung für Design Flows. Sie ermöglicht Design Support Abteilungen oder KMU einen Task Flow für einen Entwurfsgegenstand am Bildschirm grafisch zu modellieren. Für jeden Design Task sind die erforderlichen Eingabeinformationen, die eigentliche Aufgabenbeschreibung und das Ergebnis zu dokumentieren. Sie bilden die Grundlage für die sich anschließende Suche nach Web Services zur effizienten Task-Durchführung.

Die Design Tasks, wie sie bei der Entwurfsdurchführung zu bearbeiten sind, werden nachfolgend beispielhaft beschrieben. Sie werden noch recht grob charakterisiert und bedürfen, um als konkrete Entwurfsanleitung für einen Akteur

im Design Team zu dienen, einer genaueren Spezifikation. Um als Grundlage für die Herleitung von potentiellen benötigten Web Services zu dienen ist die Granularität der Design Task Beschreibung aber ausreichend. Die Einteilung der Design Tasks erfolgt unter Berücksichtigung der vorgestellten Entwurfsmethodik (siehe Abbildung 4.5 auf Seite 60):

- Realisierung der Abbildung $f_{gen}(L_{in}^P, L_{out}^S) \longrightarrow L_{check}^P$ durch generierende Design Tasks
 1. Unter Berücksichtigung des Fertigungsverfahrens ist die Strukturbeschreibung zu analysieren. Sie liefert Teilstrukturen, die auf dem Substrat zu produzieren sind
 2. Grobe Festlegung der Fertigungsgruppen und den dazugehörigen Fertigungsprozessschritten, wobei es primär um die Selektion der grundlegenden Fertigungstechnologie also Schichtabscheidung, -modifikation, -strukturierung und -abtragung geht
 3. Für jeden Fertigungsschritt ist eine exakte Beschreibung der verwendeten Materialien, Masken und Fertigungsparameter zu erstellen

Die Entwurfsunterstützung im Bereich der generierenden Tasks beschränkt sich bei den semi-automatischen Services auf PRINCE, wo ein bottom-up Entwurfsansatz realisiert wurde. Der Akteuer erhält eine grafische Eingabemöglichkeit für Fertigungsschritte und legt Werkstoffe, Masken und Fertigungsparameter überwiegend selber fest. Ein Service, der voll-automatisch mit Hilfe der Strukturbeschreibung und der physikalischen Beschreibung einen ersten Vorschlag für eine Fertigungsschrittfolge erstellt, ist nicht vorhanden. Erste Ansätze wurden hierzu von M. Lang in [Lan98] formuliert und mit dem Werkzeug FELLINI realisiert. Der Ansatz beschränkt sich auf den statischen CMOS-Fertigungsprozess, wo die Maskegeometrien für den IC-Entwurf um nicht-elektronische Mikrostrukturelemente erweitert wurde. „Echte“ dreidimensionale Strukturen sind mit dem Ansatz nicht zu realisieren, da die Einschränkung bezüglich der Prozessschritt-Auswahl zu groß ist. Es besteht an dieser Stelle deutlicher Forschungsbedarf voll-automatische Services zu entwickeln, die in der Lage sind die generierende Design Tasks abzudecken.

- Umsetzung der Abbildung $f_{ver}(L_{check}^P, L_{in,r}^P) \longrightarrow L_{delta}^P$ durch prüfende Design Tasks
 1. Verifikation der Korrektheit der einzelnen Maskegeometrien, Prozessparameter und Materialien für jeden einzelnen Fertigungsschritt

mit Hilfe von Fertigungsprozessregeln und Layoutregeln

2. Korrektheit der Reihenfolge, Konformitäten und Kompatibilitäten von Fertigungsschritten überprüfen durch den Einsatz von Fertigungsprozessregeln und Layoutregeln
3. Analyseergebnis erstellen und für die Weiterverarbeitung zur Verfügung stellen

In dem Bereich der prüfenden Tasks zur Layout- und Fertigungsprozess-Verifikation existiert eine relativ gute Abdeckung durch Werkzeuge wie sie von Mentor Graphics (<http://www.mentor.com/>) und Cadence Design Systems (<http://www.cadence.com/>) aber auch von Forschungsinstituten angeboten werden. Die Aktivitäten in diesem Bereich müssen sich auf das offerieren von Web Services konzentrieren. Die Öffnung der bestehenden Werkzeugen ist notwendig, um die bestehenden Verifikationsalgorithmen zu extrahieren und über Web Services am Markt anzubieten. Der Entwicklungsaufwand ist überschaubar, da kein neuer Code zu erstellen ist sondern eine Schale um die Algorithmen zu legen ist, die nach außen ein Web Service Interface anbietet.

- Design Tasks für die Prozessmodifikation und Maskenkorrektur, die helfen die Abbildung $f_{mod}(L_{delta}^P, L_{check}^P) \longrightarrow L_{korr}^P$ und $f_{new}(L_{korr}^P, L_{check}^P) \longrightarrow L_{check_x}^P$ zu implementieren
 1. Auswertung der Restriktionsverletzungen, die durch Fertigungsprozess- und Layoutregeln erkannt wurden
 2. Angebot von Lösungsvarianten für jede einzelne Restriktionsverletzung vorbereiten und eine Entscheidungsvorlage erstellen
 3. Umsetzung der Entscheidung bezüglich der Regelverletzung durch Modifikation der Prozessschritte und Korrektur von Maskengeometrien.

Services oder Werkzeuge zur Prozessmodifikation und Maskenkorrektur beschränkten sich meist auf das Darstellen von Restriktionsverletzungen. Die Entwicklung von Lösungsvarianten oder die voll-automatische Modifikation von Prozessschritten ist zur Zeit nicht als Service⁶ verfügbar. Hier ist zu untersuchen, ob Wissensdatenbanken dem Akteur Entscheidungshilfen bei Restriktionsverletzungen geben könnten.

⁶Das Unternehmen Rubicad bietet mit dem DA-Tool DECOR eine Möglichkeit an, Fehler in einem IC Layout größtenteils voll-automatisch zu korrigieren (<http://www.rubicad.net/>).

- Design Tasks für die Simulation der Fertigungsschritte und die Verifikation der Strukturrestriktionen beziehen sich auf die Abbildungen

$$f_{sim}(L_{check}^P) \longrightarrow L_{check}^S \quad \text{und} \quad f_{diff}(L_{check}^S, L_{out,r}^S) \longrightarrow L_{delta}^S$$

1. Simulation der kompletten Fertigungsschrittfolge
2. Darstellung der Simulationsergebnisse in Form eines Bildes des zu produzierenden Mikrosystems
3. Identifikation und Darstellung von Abweichungen zwischen Plan-Mikrosystem (ist in der Strukturbeschreibung enthalten) und Ist-Mikrosystem (ist durch die Simulation beschrieben) als Entscheidungsvorlage, ob die Fertigungsspezifikation an die Foundry zu liefern ist.

Beispielsweise bietet das Unternehmen Coventor das Werkzeug *MEMulator* an, das eine komplette Fertigungsschrittfolge in großen Teilen simulieren kann (<http://www.coventor.com/>). Die Funktionalitäten von *MEMulator* sind nicht als Web Services verfügbar, sondern als DA-Framework Lösung. Für KMU und auch für Design Support Abteilungen ist es interessant, auf die einzelnen Simulations-Funktionalitäten über Web Services zuzugreifen, um sie schnell in ihre Design Flows zu integrieren. Gleichzeitig kann so eine best-of-breed Strategie gefahren werden, und Simulations-Algorithmen verschiedener Hersteller können kombiniert zum Einsatz kommen.

Abweichungen zwischen Plan- und Ist-Mikrosystem sind über ein FEM-Modell grafisch darzustellen. Sind keine signifikanten strukturellen Abweichungen ersichtlich, kann die Simulation bestimmter Verhaltensmuster (z.B. Schwingungsverhalten) eine entgeltliche Entscheidung liefern, ob eine Fertigungsspezifikation an die Foundry zu übergeben ist. Produkthersteller wie das Unternehmen ANSYS (<http://www.ansys.com/>) sind in diesem Zusammenhang potentielle Wunsch-Kandidaten für die Einführung von Web Services.

Die obigen Design Tasks sind unter der Prämisse formuliert worden, dass ein Design Team mit der Durchführung des Design Flows beauftragt wird. Design Teams bevorzugen einen top-down Entwurf, der auf einem relativ hohen Abstraktionsniveau die Prozessdurchführung ermöglicht und das Fertigungswissen weitgehend über Services zur Verfügung stellt. Der Arbeitsschwerpunkt von Design Teams liegt primär in der Bearbeitung und der Leistungsoptimierung des dreidimensionalen Entwurfsgegenstandes und der Suche nach einem bezüglich der Restriktionen im Toleranzbereich liegenden Fertigungsgegenstandes. Genau das ist die Zielgruppe von Anwendern für das service-orientierte Entwurfsszenario, die im Rahmen dieser Dissertation im Mittelpunkt stehen. Es geht darum,

für diese Anwender eine adäquate Entwurfsunterstützung zu finden, welche den Komplexitätsgrad des physikalischen Entwurfs deutlich reduziert

Zwei Klassen von Design Teams werden durch das service-orientierte Entwurfs-szenario besonders gut unterstützt. Zu einem sind das Design Teams, die in kleinen und mittelständischen Unternehmen sowie Forschungseinrichtungen aktiv sind. Sie erhalten einen Design Flow mit über das Internet eingebundenen Design Services. Zum anderen geht es um Design Teams in Großunternehmen, die über das Intranet von der DA-Support Abteilung einen Web Service basierten Design Flow zur Verfügung gestellt bekommen.

Es stellt sich die Frage, ob die Aufteilung in eine nach der top-down und der bottom-up Strategie agierenden Entwurfsgruppe gerechtfertigt und sinnvoll ist. Das Design Team auf der einen Seite, das die funktionalen Leistungsmerkmale des Entwurfsgegenstandes im Vordergrund sieht und fertigungsrelevante Aspekte, wie welche Ätztechnik einzusetzen ist, so weit wie möglich ausblenden möchte. Die andere Seite mit der Design Support Abteilung, die umfangreiches Expertenwissen im Bereich physikalischer-technischer Entwurf besitzt und die Fähigkeit hat Design Flows zu erstellen. Sie greift auf DA-Vendors und Foundries zu, um mit Hilfe von Services den Design Flow zu beschleunigen und die Entwurfs- und Integrationskosten zu senken.

In der Software-Industrie ist diese Trennung bereits vollzogen und wird gelebt in Form von zahlreichen Off-shore Aktivitäten der Unternehmen. Die eine Gruppe der Mitarbeiter spezifiziert und legt die Architektur sowie das GUI fest, die andere Gruppe ist ausschließlich für das Codieren zuständig. Die Trennung ist in der Software-Industrie zum einen aus Kostengesichtspunkten und zum anderen aus Gründen der immer komplexer werdenden IT-Welt vollzogen worden, die eine Spezialisierung auf IT-Fokusbereiche zwingend erfordert um Qualitätsstandards zu halten. Eine ähnliche Evolution zeichnet sich in der Mikroelektronik ab [SM03]: „Diese große Spannweite der Anforderungen könnte das Gros der Entwickler an ihre Grenzen bringen und zu einer Aufteilung in zwei Gruppen führen: Die einen, die auf der oberen Ebene Chips entwerfen, ähnlich wie bei der Softwareentwicklung, die anderen, die diesen Entwurf auf der physikalisch-technischen Ebene umsetzen.“ Die getroffene Rolleneinteilung bezüglich Design Team, DA Support, DA Vendor und Foundry ist mit Blick auf die Tendenzen in der Software Industrie und der Mikroelektronik als bestätigt anzusehen.

Abbildung 6.12 zeigt schematisch den Aufbau der Design Flow IDE.

Die Benutzeroberfläche der Applikation besteht aus einer **Menüleiste**, dem **Arbeitsfenster** und aus drei **Auswahlfenstern**. In den Auswahlfenstern ste-

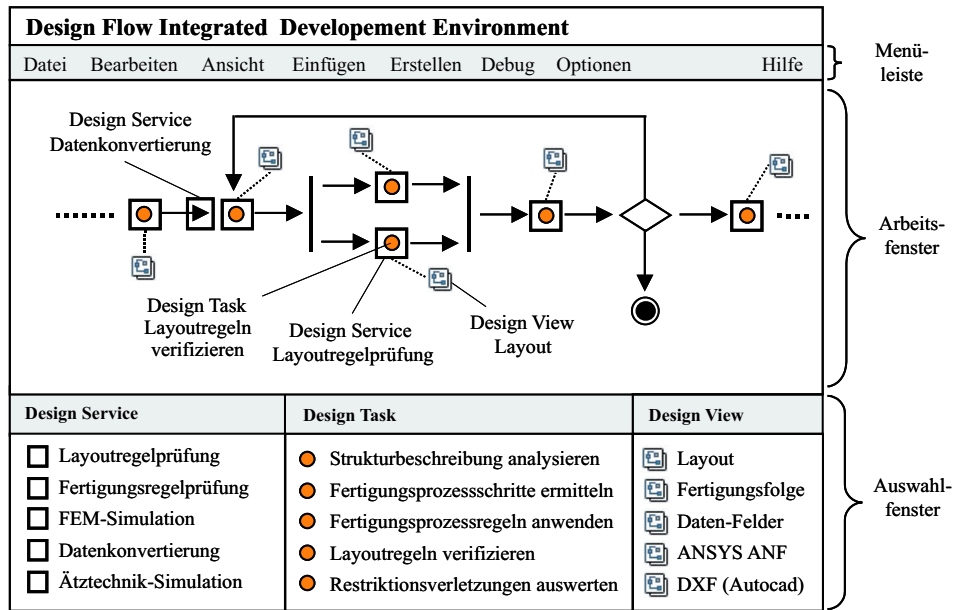


Abbildung 6.12: Design Flow Entwicklung

hen **Design Services (DS)**, **Design Tasks (DT)** und **Design Views (DV)** zur Verfügung. Sie können durch einen Mausklick selektiert werden und sind danach im Arbeitsfenster zu positionieren.

Ein Design View Element erlaubt es, Daten grafisch zu visualisieren oder einzugeben. Das Ergebnis eines Web Service Aufrufs ist durch ein Design View Element dem Akteur auf dem Bildschirm anzuzeigen. Beispiele für Design View Elemente sind Layoutanzeigen oder die Darstellung von FEM Berechnungsergebnissen. Die Design View Elemente sind durch Software-Komponenten zu realisieren, die bereits überwiegend am IT-Markt angeboten werden.

Unter <http://www.componentsource.com/> werden momentan über 9.000 Komponenten für verschiedene Entwicklungs-Plattformen in unterschiedlichen Programmiersprachen Teilweise als OpenSource oder zum Kauf angeboten. Hier greift das vorgestellte Konzept der komponenten-orientierten Softwareentwicklung.

Design Task Elemente enthalten die konkrete Aufgabenbeschreibung. Sie sind von der Support Abteilung am Anfang zu definieren. Als Richtlinie ist die in dieser Arbeit vorgestellte Entwurfsmethodik zu verwenden. Beispiele für Design Tasks wurden in diesem Abschnitt vorgestellt.

Design Services liefern über das Internet/Intranet die gewünschten Methoden zur Realisierung der Design Tasks. Das Portal dient als Quelle für die Suche

nach geeigneten Services, die dann in die Design Flow IDE importiert werden können.

Über die Menüleiste stehen dem Akteur Befehle zur Steuerung der Design Flow IDE zur Verfügung. Sie enthält acht Menüpunkte:

- *Datei*: enthält Befehle zum Neu anlegen, Öffnen und Speichern von Design Flow Dateien sowie den Befehl zum Beenden der Applikation. Das Importieren von Dateien mit Beschreibungen von Design Services, Design Tasks und Design Views für die Anzeige in den Auswahlfenstern erfolgt auch über diesen Menüpunkt.
- *Bearbeiten*: enthält Befehle zum Ausschneiden, Einfügen, Kopieren und Löschen von Elementen im Arbeitsfenster
- *Ansicht*: enthält Befehle zur Vergrößerung und Verkleinerung der Design Flow Zeichnung im Arbeitsfenster. Zusätzlich sind Befehle vorhanden, um ein Lineal oder Führungslinien oder ein Raster einzublenden
- *Einfügen*: enthält eine Liste mit UML-Elementen um ein Aktivitätsdiagramm zu spezifizieren und bietet Befehle an, um einzelne UML-Elemente auszuwählen und im Arbeitsfenster zu platzieren. Auf diese Art ist die Design Flow Zeichnung schrittweise aufzubauen
- *Erstellen*: enthält Befehle, um den Generierungsprozess für den Design Flow Assistenten zu starten und zu stoppen. In Abhängigkeit von der gewählten Web Präsentationstechnik entsteht ein Applet, ein Java Anwendung oder ein Client, der auf einer Reihe von verbundenen XML/HTML-Seiten basiert
- *Debug*: enthält Befehle, um die Konsistenz und Vollständigkeit der Design Flow Zeichnung zu prüfen. Unvollständige und fehlerhafte Teile der Zeichnung werden rot markiert
- *Optionen*: enthält Befehle, um die Web Präsentationstechnik einzustellen und zur Konfiguration von Design Services, Design Tasks sowie Design Views
- *Hilfe*: enthält Hilfen zur Benutzung der Applikation.

Über den Menüpunkt *Datei* beginnt der Akteur mit dem Anlegen eines Design Flows für einen Entwurfsgegenstand. Alternativ kann er auf bestehende Design Flows zugreifen und diese modifizieren oder komplett neue erstellen.

Das Arbeitsfenster dient der Eingabe des Task und Service Flows. Der Akteur zeichnet mit Hilfe von drag-and-drop Techniken im Arbeitsfenster einen Design Flow und legt auf diese Weise den Steuerfluss, die Entwurfsaufgaben, Services und die Art der Visualisierung von Entwurfsdaten fest. Jedem Design Task Element ist ein Design Service und ein Design View Element zuzuordnen. Hierzu nutzt der Akteur die Elemente in den Auswahlfenstern. Der Steuerfluss ist durch die UML-Elemente zu spezifizieren.

Der Menüpunkt *Debug* erlaubt es den geplanten Design Flow auf Fehler zu überprüfen. Der Menüpunkt *Erstellen* startet den Algorithmus, um am Ende der Entwicklungsphase den Design Flow in eine Web-Applikation zu gießen.

Am IT-Markt existiert mit dem **Bea Weblogic Workshop** bereits eine IDE, die über eine grafische Benutzeroberfläche das dynamische Bauen von Komponenten-Software erlaubt (<http://www.bea.com/>). Die Bereitstellung der produzierten Applikation ist auf die Ablaufumgebung des Bea Weblogic Applikationsserver reduziert. Die Abhängigkeit führt dazu, dass teure kommerzielle Software zum Einsatz kommen muss, was für viele KMU nicht akzeptabel wäre.

Um die Entwicklung von Java Web Applikationen und Services zu fördern, ist von BEA Systems und der Apache Software Foundation das Open Source Projekt namens **Beehive** ins Leben gerufen worden. Beehive basiert überwiegend auf dem Bea Weblogic Workshop Quellcode und ist ab Sommer 2004 kostenfrei verfügbar. Als Ablaufumgebung können beliebige Applikationsserver (z.B. **JBoss**) Verwendung finden. Beehive könnte eine gute Grundlage bilden, um die oben beschriebene Design Flow IDE zu implementieren.

Mit Beehive und JBoss wäre für Entwicklung und Durchführung des Design Flows eine Open Source basierte Umgebung vorhanden. KMU bräuchten keine Investitionen in Lizenzen zu tätigen. Forschungsinstitute besitzen die Chance auf dieser Open Source Plattform attraktive Design Service zu realisieren.

6.7.2 Design Flow Assistent

Der **Design Flow Assistent** ist eine Applikation, die den Akteur zielgerichtet durch den Entwurfsprozess führt und die Erstellung der korrekten und vollständigen Fertigungsspezifikation sicherstellt [SSBP01] [SSBP00] [BHP⁺99]. Die Kontrolle und Steuerung der Entwurfschrittfolge liegt zwar in der Hand des Assistenten, die wichtigen Entwurfsentscheidungen sind aber weiterhin vom Design Team zu treffen. Der Assistent ist als Komponenten-Software realisiert [BS99] [SB98], d.h. er besteht aus Komponenten, die in Form von Web Services

eingebunden sind und das Domänenwissen liefern, einer Steuer- und Kontroll-Komponente, die den Datenfluss beeinflusst, sowie aus Komponenten, die für die Visualisierung der Information zuständig sind. Der Quellcode, um diese Komponenten interagieren zu lassen und zu einer logischen Einheit zu verbinden, wird von der IDE bei der Assistenten-Generierung automatisch produziert. Die DA-Support-Abteilung ist somit in die Lage, dem Design Team einen anwendungsspezifischen Design Flow zu liefern, den sie zuvor mit der IDE entwickelt hat.

Abbildung 6.13 skizziert die Benutzeroberfläche des Design Flow Assistenten.

Sie besteht aus folgenden Teilen:

- *Menüleiste*
- *Ein- und Ausgabefenster*
- *Design Flow-Anzeige*
- *Service-Fenster*
- *Task-Fenster*
- *Informationen zur Spezifikation*

Die Menüleiste erlaubt dem Akteur Befehle zur Steuerung des Design Flow Assistenten auszuführen. Der Menüpunkt *Datei* bietet Befehle zum neu Anlegen, Öffnen und Speichern von Fertigungsspezifikationen an. Als Dateiformat ist die bereits erwähnte Spezifikationsbeschreibungssprache PDML zu verwenden.

Weitere Menüpunkte sind kontextsensitiv anzubieten, d.h. in Abhängigkeit vom Entwurfschritt ist ein Menüpunkt dynamisch ein- oder auszublenden. Zum Beispiel ist für die grafische Erstellung und Bearbeitung von Fertigungsschritten, wie sie in PRINCE angeboten wird, ein Menüpunkt *Bearbeiten* notwendig, der Befehle zum Ausschneiden, Einfügen, Kopieren und Löschen von Fertigungskomponenten anbietet.

Geht es um die Visualisierung von Simulationsergebnissen eines Fertigungsprozesses, ist ein Menüpunkt *Ansicht* einzublenden, der Zoom- und Rotationsbefehle enthält, um den Fertigungsgegenstand aus allen Perspektiven zu betrachten, gegebenenfalls auch einen Befehl, der Abweichungen zwischen Soll- und Ist-Mikrosystem farblich heraus arbeitet.

Ein Menüpunkt *Optionen* erlaubt es, den gerade aktiven Web Service im Rahmen der vorgegebenen Möglichkeiten zu konfigurieren. Abschließend sollte der

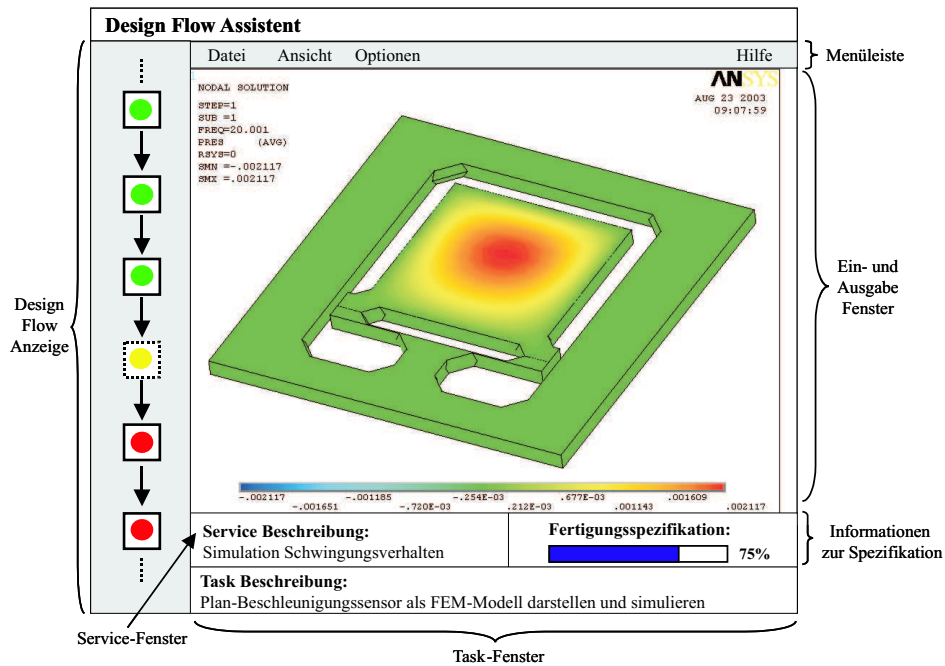


Abbildung 6.13: Design Flow Durchführung

Menüpunkt *Hilfe* allgemeine Informationen zum Assistenten enthalten und kontextsensitiv Benutzungshinweise zum jeweiligen Service.

Das *Ein- und Ausgabefenster* dient als Schnittstelle, um die Informationen an die Web Services zu übergeben oder das Resultat eines entfernten Methodenaufrufes anzuzeigen. Die aktuelle Ausprägung des Ein- und Ausgabefensters ist durch eine Design View Komponente realisiert. Sie variiert in Abhängigkeit vom gerade aktiven Web Service. Zum Beispiel kann für einen Web Services als Eingabe ein Reihe von Datenfeldern notwendig sein und als Ausgabe ist eine DXF-Datei zu visualisieren.

Die *Design Flow Anzeige* gibt dem Akteur einem Überblick über die Entwurfsaufgaben, die im Rahmen des Entwurfsprozesses zu erledigen sind und mit Hilfe der IDE zuvor spezifiziert wurden. Die einzelnen Design Tasks sind in Form von Punkten dargestellt. Grüne Punkte kennzeichnen bereits erledigte Design Tasks, gelbe Punkte sind noch nicht vollständig bearbeitet oder werden gerade bearbeitet und rote Punkte sind noch gar nicht bearbeitet worden. Im Sinne des Kreismodells und der Entwurfsmethodik sind gewisse Design Tasks zyklisch mehrmals zu bearbeiten.

Das *Task-Fenster* liefert dem Akteur die Aufgabenbeschreibung, welche von ihm im Rahmen des Entwurfsschrittes zu erledigen ist. Gleichzeitig ist im *Service-*

Fenster die Leistungsbeschreibung des entsprechenden Web Services anzuzeigen. Ein Mausklick in das Task- oder Service Fenster öffnet ein neues Fenster mit einer detaillierteren Aufgaben- oder Leistungsbeschreibung.

Bei der Bearbeitung des Design Flows ist für den Akteur der momentane Fertigstellungsgrad der Spezifikation wichtig. Er ermöglicht eine grobe Restaufwandsabschätzung und damit eine Aussage, wann die Arbeiten an der Fertigungsspezifikation voraussichtlich beendet sind. Das Fenster *Informationen zur Spezifikation* zeigt grafisch den prozentualen Fertigstellungsgrad der Spezifikation an. Der Akteur kann mit einem Mausklick in das Fenster sich den aktuellen Stand der Fertigungsspezifikation (Quellcode der PDML-Datei) in einem separaten Fenster anzeigen lassen.

Das Beispiel eines Beschleunigungssensors in Abbildung 6.13 zeigt, wie mit einem FEM-Service die Einhaltung von Struktur- oder Verhaltensrestriktionen zu überprüfen ist. Der Akteur erhält dazu eine Service- und eine Task Beschreibung eingeblendet. Das Simulationsergebnis ist im Ein- und Ausgabefenster als dreidimensionale Grafik zu sehen. Ein Balken zeigt an, dass 75 % Prozent der Fertigungsspezifikation für den Beschleunigungssensor bereits erstellt wurde. Ist der Entwurfsprozess beendet, sorgt ein Web Service der Foundry für das Erzeugen einer Chargenkarte oder anderer maschinennaher Fertigungsinformationen.

6.8 Service-orientiertes Entwurfsszenario

Mit der Design Flow IDE und dem Assistenten sind DA-Tools konzipiert worden, um eine systematische Vorgehensweise bei der Design Flow Entwicklung und Durchführung zu ermöglichen. Dieser Abschnitt ordnet beide DA-Tools in ein Realisierungskonzept für das service-orientierte Entwurfsszenario ein. Darüber hinaus wird eine grafische Darstellung des service-orientierten Entwurfsszenarios vorgestellt. Ziel ist es ein Verständnis dafür zu entwickeln, wie mit Hilfe des SES die Entwurfsprozessautomatisierung verbessert werden kann.

6.8.1 Realisierungskonzept

In diesem Kapitel ist schrittweise ein Konzept erarbeitet worden, dass alle Anforderungen an ein service-orientiertes Entwurfsszenario erfüllt. Das Konzept legt für jedes Leistungsmerkmal des Entwurfsszenarios eine entsprechende technische Umsetzung fest:

- Für die lose Kopplung von Akteuren den Einsatz des Internets als Kommunikationsinfrastruktur
- Manuelle-, semi-automatische und voll-automatische Design Services nutzen Java Applikationen, Web Browser und Web Services zur technischen Umsetzung
- Überblick und Zugriff auf das Serviceangebot der DA-Industrie ist durch die Schaffung eines Web-Portals ermöglicht worden
- die systematische Vorgehensweise beim Entwurf ist mit der Bereitstellung von Design Flow IDE und dem Assistenten gewährleistet. Beide Anwendungen verkörpern zeitgemäße Ansätze, um die Produktivitätslücke zwischen Entwurf und Fertigung zu schließen.

Abbildung 6.14 zeigt Akteure, Applikationen und Infrastruktur des service-orientierten Entwurfsszenario im Überblick.

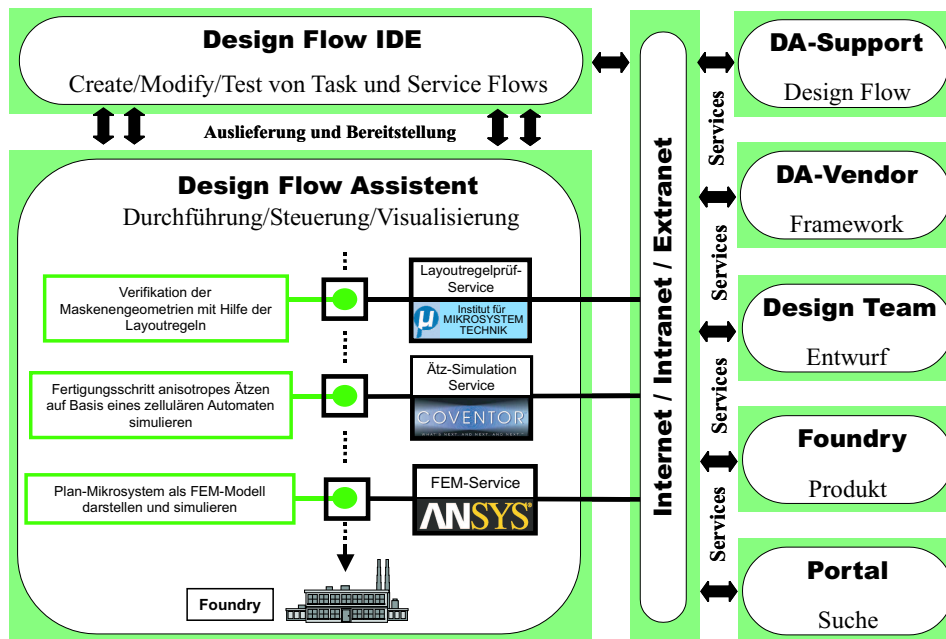


Abbildung 6.14: Realisierungskonzept für das service-orientierte Entwurfsszenario

Auf der rechten Seite sind die Akteure zu erkennen. DA-Support, DA-Vendor, Design Team und Foundry repräsentieren jeweils eine Klasse von Akteuren, die Dienstleistungen über Design Services am DA-Markt anbieten oder nutzen. Als reiner Service-Konsument tritt das Design Team auf. Es kauft sich Design

Services vom DA-Markt ein, um die Durchführung eines Entwurfsprozesses für einen Entwurfsgegenstand zu beschleunigen. Die DA-Vendors sind als Service-Anbieter einzuordnen, die über Web Services ihre Algorithmen zur Entwurfsautomatisierung als Dienstleistung verkaufen. Ebenso die Foundries, die über Web Services eine Layoutregelprüfung anderen Akteuren vor der finalen Fertigstellung der Spezifikation ermöglichen. Der DA-Support ist zum einen ein Service Konsument, wenn er Services eines DA-Vendors nutzt oder zum anderen ein Service-Anbieter gegenüber einem Design Team, für den er einen Design Flow entwickelt. Das Portal stellt einen virtuellen Akteur dar, dessen Aufgabe es ist, Suchdienste anzubieten, die es erlauben Design Services zu finden. Demzufolge reiht sich dieser Akteur in die Klasse der Service-Anbieter ein.

Zwischen den Akteuren und dem Design Flow Assistent sind Internet, Intranet und Extranet als Kommunikationsvarianten eingezeichnet. Sie dienen zum Austausch der fertigungsrelevanten Informationen während der MEMS-Produktentwicklung. Bedingt durch das hohe zu bewegendes Datenvolumen, insbesondere bei den Maskenlayouts, ist auf eine ausreichende Bandbreite in der Betriebsumgebung zu achten. Sicherheitsaspekte sind bei In-house Anwendungen zu vernachlässigen. In diesem Fall versorgt die DA-Support-Abteilung verschiedene Design Teams eines Unternehmens mit Assistenten. Bei KMU, die von unterschiedlichen Akteuren Dienstleistungen über das Internet beziehen, sind Verschlüsselungstechniken zu verwenden und gegebenenfalls ist ein Login-Mechanismus einzusetzen, der den Zugang zum Portal oder einzelnen Diensten regelt.

Auf der linken Seite der Abbildung 6.14 sind die DA-Applikationen positioniert. Die Design Flow IDE ist für das Erstellen, Ändern und Testen von Task- und Service Flows verantwortlich. Sie erzeugt den Design Flow Assistenten und liefert ihn anschließend über eine URL-Adresse oder über die Web Start Technology an das Design Team aus. Der Assistent ist für die Durchführung der einzelnen Entwurfsschritte verantwortlich und steuert die systematische Erstellung der Fertigungsspezifikation. Er visualisiert die Ein- und Ausgabeinformationen der einzelnen Web Service Aufrufe. Drei Beispiele für Web Service Einsätze sind zum besseren Verständnis in der Abbildung eingezeichnet:

1. *Service zur Layoutregelprüfung*: kann z.B. von Forschungsinstituten als Dienst-Anbieter offeriert werden
2. *Service zur Ätz-Simulation*: kann durch einen großen DA-Vendor realisiert werden, der durch einen Wrapper um sein Framework diese Funktion liefert

3. *Service zur FEM-Simulation*: kann vom Marktführer ANSYS als Web Services freigeschaltet werden

Die Services dienen zur Entwurfsautomatisierung der Design Tasks, die durch grüne Punkte und die jeweiligen Aufgabenbeschreibungen dargestellt sind.

Während der Entwicklung und der Durchführung des Design Flows greifen die DA-Applikationen über die Kommunikationsinfrastruktur auf Design Services zu. Ein Design Assistent bindet erst zur Laufzeit einen großen Teil seiner Methoden ein, indem er Web Services aufruft. Die Verfügbarkeit und die Performance der Dienste ist daher von entscheidender Bedeutung. Daraus leitet sich die Frage ab, wie die Infrastruktur (Hardware, Netzstruktur, etc.) zur Umsetzung des service-orientierten Entwurfsszenario aussieht. Nachfolgende Arbeiten sollten sich mit diesem Thema auseinandersetzen.

Ein Anwendungsfall, der mit dem service-orientiertem Entwurfsszenario umgesetzt werden kann, soll das Zusammenspiel zwischen Akteuren, Applikationen und Infrastruktur verdeutlichen. Ein Design Team erhält den Auftrag, für einen Entwurfsgegenstand eine Fertigungsspezifikation zu erstellen. Es ist zu entscheiden, welches DA-Support Team in das Projekt zu involvieren ist. Hierzu ist das Design Service Portal als Grundlage für die Suche nach einer geeigneten DA-Support Abteilung einzusetzen. Eine Ausschreibung ist mit Hilfe der Entwurfsspezifikation über das Portal durchzuführen. Nach der Identifikation des geeigneten DA-Support Teams folgt die Design Flow Entwicklung auf Basis der IDE. Jetzt werden DA-Vendor und DA-Foundry eingebunden. Sie beliefern das DA-Support Team mit den aus Sicht des Task Flows notwendigen Services. In regelmäßigen Abständen sind vom DA-Support Software-Versionsstände des Assistenten an das Design Team zu übergeben. Die eingesetzten Web-Technologien erlauben ein schnelles und unkompliziertes Ausliefern von solchen vorläufigen Versionen des Assistenten. Das Projekt ist beendet, wenn eine Abnahme durch das Design Team erteilt wurde und der Assistent in den Produktiveinsatz übergeht.

Dieser Anwendungsfall ist nur ein Beispiel dafür, wie das Szenario zu verwenden ist, um eine bessere Entwurfsunterstützung für KMU zu ermöglichen. Andere Anwendungsfälle können unterschiedliche DA-Support Teams gleichzeitig bei der Design Flow Entwicklung einbinden oder das Design Team kann die IDE selber nutzen, um sich einen Design Flow zu bauen. Selbstverständlich kann ein DA-Vendor auch die Rolle des DA-Supports mit übernehmen und so als Full Service Provider auftreten. Eine interessanter Anwendungsfall liegt vor, wenn ein DA-Vendor und eine DA-Foundry kooperieren würden und so entweder dem

Design Team oder dem DA-Support sowohl Services für den Entwurf und die Fertigung anbieten könnten.

Als nächster Schritt ist eine Referenzimplementierung des Entwurfsszenario im Rahmen eines Entwicklungsprojektes durchzuführen. Ein vergleichbares Entwicklungsprojekt ist gerade von dem Softwareanbieter SAP gestartet worden. Die SAP hat eine Suite von Softwareprodukten zur Abwicklung von kaufmännischen Prozessen im Portfolio. Die Produkte basieren auf der Client/Server-Architektur. In den nächsten zwei Jahren wird die komplette Produktfamilie auf eine 3-Schichten-Architektur umgestellt. Erste Produkte laufen bereits auf einem Applikationsserver und sind nach komponenten-orientierten Prinzipien realisiert worden.

Die SAP unterliegt der Herausforderung, dass bei Kunden die Geschäftsprozesse analog zu den Entwurfsprozessen in der Mikrotechnik sich häufig ändern. Die Client/Server basierten SAP-Produkte mit starren Geschäftsprozessen stehen in der Kritik der Kunden, da die Aufwände für Dienstleistungen, um die Konfiguration und Integration in die bestehende IT-Landschaft vorzunehmen, die meisten Budgetgrenzen immer häufiger überschreiten. Eine vergleichbare Situation finden man in der Mikrotechnik mit Blick auf die DA-Frameworks. Ziel der SAP ist es eine **Enterprise-Service-Architektur** einzuführen, die dem service-orientierten Entwurfsszenario sehr ähnlich ist [SAP04]. Sie besteht ausschließlich aus Web Service fähigen 3-Schichten-Applikationen, die Geschäftsprozessschritte aus den Produkten extrahieren und über das Internet oder Intranet anbieten (analog zu den zukünftigen DA-Werkzeugen).

Der Kunde oder die SAP-Partnerfirmen, die Dienstleistungen rund um die Produkte offerieren, sind jetzt in der Situation, flexibel auf Änderungen in den Geschäftsprozessen reagieren zu können. Dafür nutzen sie ein Tool der SAP, das es erlaubt einen neuen Geschäftsprozess (Task-Flow) mit Hilfe von Web Services (Service-Flow) zu modellieren. Das Tool ist vergleichbar mit der Design Flow IDE und erlaubt es automatisch eine Applikation (Assistent) zu erzeugen, die dann den neuen Geschäftsprozess vollständig IT gestützt abwickelt. Die SAP setzt für die Umsetzung der Enterprise-Service-Architektur mehrere tausend Entwickler ein. Ein vergleichbares Projekt zur Realisierung des service-orientierten Entwurfsszenario sollte in der Open Source Community platziert werden, um die benötigten Entwicklungskapazitäten zu aktivieren und dadurch eine zeitnahe Fertigstellung zu ermöglichen.

6.9 Design Flow Management

Nachdem ein Realisierungskonzept für das service-orientierte Entwurfsszenario existiert, soll abschließend eine neue Disziplin für die Mikrotechnik präsentiert werden. Sie basiert auf dem SES und kanalisiert alle Aktivitäten, die bei der Design Flow Suche und Anwendung vorkommen. Auf Grundlage der neuen Disziplin erfolgt eine Optimierung des SES, in dem Technologiestandards aus der Mikroelektronik integriert werden.

6.9.1 Daten- und Applikationsintegration

Alle Aktivitäten, welche sich mit der systematischen Erstellung, Durchführung und Verwaltung von anwendungsspezifischen Design Flows in der Mikrotechnik beschäftigen, werden unter dem Begriff **Design Flow Management** subsummiert. Um die Disziplin des Design Flow Managements in der Praxis zu betreiben, ist das service-orientierte Entwurfsszenario einzusetzen. Hiermit sind Design Flows zu entwickeln und anschließend über Web-Technologien relativ einfach an die Nutzer zu verteilen.

Um das volle Potential des Design Flow Managements auszuschöpfen, bedarf es neben dem SES noch einer Technologie, die das Problem des Informationsmanagements in der Mikrotechnik löst. Hier geht es um die Frage, wie Datenkonsistenz sicherzustellen ist oder wie ein einheitliches Datenmodell für den effizienten Zugriff und Austausch von Mikrotechnik-Entwurfsdaten aussieht. Eine Antwort auf die Fragen könnte die Mikroelektronik liefern, wo sich zur Zeit ein Standard für das Informationsmanagement heraus kristallisiert.

In der Mikroelektronik wächst die Größe von IC-Layoutentwürfen permanent [SM03]. DA-Tools legen diese Layoutentwürfe heutzutage überwiegend in einem Dateisystem ab. Datenbanken kommen relativ selten zum Einsatz und sind, wenn vorhanden, nicht auf einem Standard basierend entwickelt worden.

Die Datenmenge von IC-Layoutentwürfen hat in den letzten Jahren ein Volumen angenommen, dass die Speicherung innerhalb eines Dateisystems aus Performance-Gründen fraglich erscheinen lässt. Die Anwender müssen lange Lade- und Speicherzeiten von Layoutdateien in Kauf nehmen, was ein effizientes Bearbeiten von Layoutinformationen deutlich erschwert.

Zusätzlich existieren zahlreiche Layoutdatenformate in der Mikroelektronik, die beim Einsatz verschiedener DA-Tools oft eine zeitintensive Datenkonvertierung erfordern. Dies führt insgesamt dazu, dass immer mehr Zeit für die Design Flow Durchführung benötigt wird (siehe Abbildung 6.15).

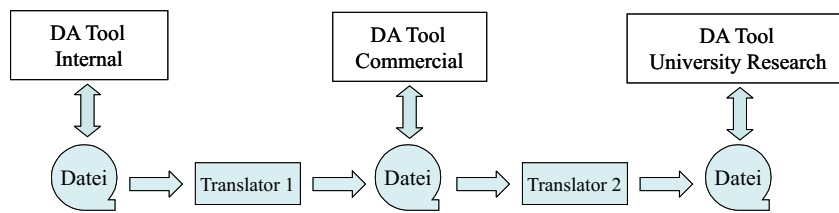


Abbildung 6.15: Datei basierter Datenaustausch

Hieraus ist die Idee entstanden, eine Plattform zu schaffen, die eine einfache Daten-Interoperabilität beim Entwurf von komplexen digitalen und analogen ICs erlaubt. Das Unternehmen Cadence Design System hat die Idee aufgegriffen und auf der Design Automation Conference 2001 hierfür eine erste prototypenhafte Implementierung vorgestellt, die aus einer speziellen Datenbank und einem **Application Programming Interface (API)** für DA-Tools bestand. Mittlerweile ist die Plattform kontinuierlich weiterentwickelt worden und unter dem Begriff **Open Access (OA)** Technologie zusammengefasst worden (<http://www.cadence.com/>).

Die OA Datenbank erlaubt es alle entwurfsrelevanten IC-Informationen über ein API zu verwalten. Sie ist optimiert bezüglich Zugriffszeit und Speichernutzung für große IC-Layouts. Sie repräsentiert eine auf die Bedürfnisse der Mikroelektronik abgestimmte Lösung, um die Daten-Interoperabilität zwischen DA-Tools zu verbessern und zu beschleunigen. Um die OA API zu nutzen, sind bestehende DA-Tools zu modifizieren. Sie müssen ihre Daten nicht mehr in Dateien ablegen, sondern über das API in der OA Datenbank speichern. Das hierfür benötigte Datenmodell ist durch die *OpenAccess Initiative* vorgeben (<http://www.openeda.org/>). DA-Tools, die noch nicht die OA API unterstützen, können ihre IC-Informationen über einen Transformator in die Datenbank ablegen. Zur Zeit existiert ein Trend in der Mikroelektronik-Gemeinde OA zu nutzen. Namhafte Unternehmen wie zum Beispiel Hewlett Packard führen OA als Standard ein. Selbstverständlich nur schrittweise, da bestehende interne DA-Tools zu modifizieren sind und das mit Aufwendungen verbunden ist.

Die OA-Datenbank in der aktuellen Ausprägung besteht aus drei Teilen:

- Design-Database, enthält z.B. IC-Layoutinformationen
- Technology-Database, enthält z.B. Layoutregeln der Foundries
- Library-Database, enthält Administrative Informationen wie z.B. Zugriffsrechte oder organisatorische Informationen

Durch die API-Funktionen ist festgelegt, in welchem Format Daten in die Datenbank eingestellt oder ausgelesen werden. Weitere Informationen zum Thema OA sind unter <http://www.si2.org/> zu finden.

Abbildung 6.16 zeigt die Elemente der OA Technologie und verdeutlicht die Zugriffsmöglichkeiten auf die Datenbank.

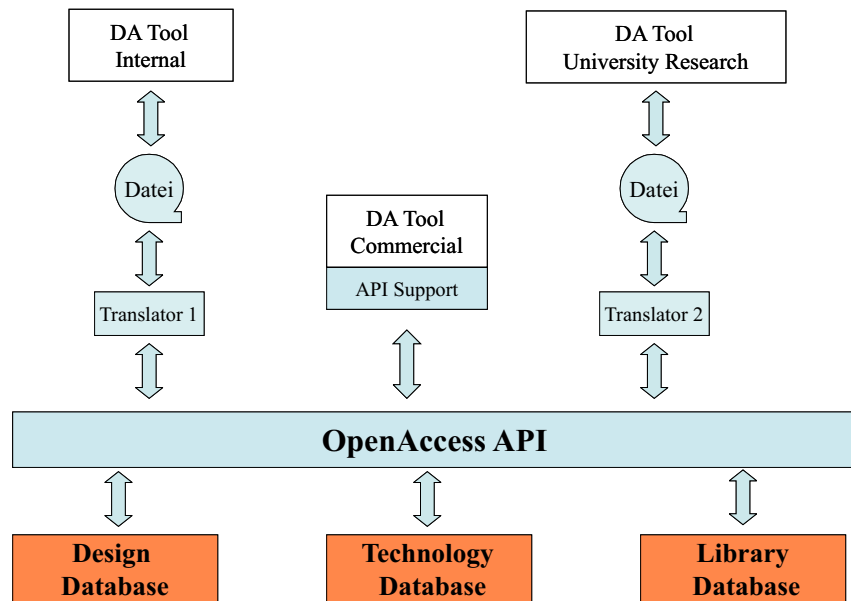


Abbildung 6.16: Open Access Technologie

Wie schon erwähnt, liegt der momentane Fokus der OA Technologie in der Verbesserung und Beschleunigung der Daten-Interoperabilität in der Mikroelektronik. Spezielle Anforderungen der Mikrotechnik wurden weder beim Datenbankentwurf noch bei der API Spezifikation berücksichtigt. Die Herausforderung in der Mikrotechnik liegt grundsätzlich in der Definition von De-facto-Standards zur Beschreibung der Semantiken von Datenstrukturen für Fertigungsprozesse, Materialien oder Maskengeometrien. Es bedarf einer für alle Akteure offenen MEMS-Community, welche die Semantik von Datenstrukturen für die Mikrotechnik spezifiziert. Eine solche Spezifikation ermöglicht dann in Kombination mit Web Services das service-orientierte Entwurfsszenario mit Leben zu füllen, da Service Konsument und Anbieter den Inhalt einer SOAP-Nachricht verstehen und weiterverarbeiten können.

Als Realisierungsmedium bietet sich die OA Technologie an. Neben speziellen mikrotechnischen Erweiterungen der Datenbanken kann über die API die Semantik und der Aufbau von Datenstrukturen festgelegt werden. Es ist zu untersuchen, inwieweit PDML als Orientierungshilfe bei dieser Aufgabe dienlich

sein kann. Ein erster Schritt wäre, dass beispielsweise ein Forschungsinstitut in die OA Community aktiv wird, um mikrotechnische Aspekte bei der Weiterentwicklung zu forcieren.

Für KMU ist die OA Technologie mit großen Herausforderungen verbunden. Sie hilft zwar auf der einen Seite die Daten-Interoperabilität zu optimieren, auf der anderen Seite sind es aber wieder Frameworks und DA-Spezial-Tools, die KMU kaufen, installieren und betreiben müssen. Diese DA-Tools werden an eine OA-Datenbank gehängt und starr über die C++ API angebunden. Der Zugriff auf einzelne dedizierte DA-Algorithmen ist weiterhin nicht möglich und damit auch kein dynamisches und flexibles Erstellen von Design Flows. Deshalb ist neben der Daten-Interoperabilität die Applikations-Interoperabilität für KMU entscheidend, um die benötigte Flexibilität bei der Design Flow Gestaltung zu garantieren. Hier bietet das service-orientierte Entwurfsszenario einen guten Ansatz, um als sinnvolle Ergänzung und Erweiterung der OA Technologie zu dienen. Da es den Gedanken Software, und insbesondere die Funktionen einer Software als Service anzusehen berücksichtigt, und so einen Weg für die Applikations-Interoperabilität mit Hilfe von Web Services aufzeigt.

Eine Kombination aus OA- und SES-Konzept ist in der Lage, den KMU und vielleicht auch der Großindustrie eine durchgängige Entwurfsautomatisierung zu offerieren, die ihren aktuellen Anforderungen entspricht. Abbildung 6.17 skizziert, wie eine solche Kombination aussehen könnte.

Die Kombination aus OA und SES besteht aus drei operativen Ebenen:

- **Design Flow Management:** eine neue Abstraktionsebene beim Entwurf in der Mikrotechnik, die KMU eine hohe Flexibilität bei der Entwurfsautomatisierung und eine Reduktion ihrer Kosten ermöglicht
- **Applikationsmanagement:** diese Ebene enthält neue moderne DA-Tools oder bestehende DA-Applikation, die über Services ihre Leistungen dem Design Flow Management anbieten. Installation, Konfiguration und der Betrieb von Applikationen bleibt somit für die Akteure verborgen
- **Informationsmanagement:** hat die Aufgabe die Verwaltung der Informationen beim Mikrotechnik-Entwurf durchzuführen. Es stellt die Datenkonsistenz sicher und bietet ein einheitliches Datenmodell an, sowie den effizienten Zugriff auf die Mikrotechnik Entwurfsdaten

Um mit den Assistenten zu kommunizieren, sind in der OA Technologie Web Services als zusätzliche Schnittstelle einzuführen. Die Darstellung der Web Ser-

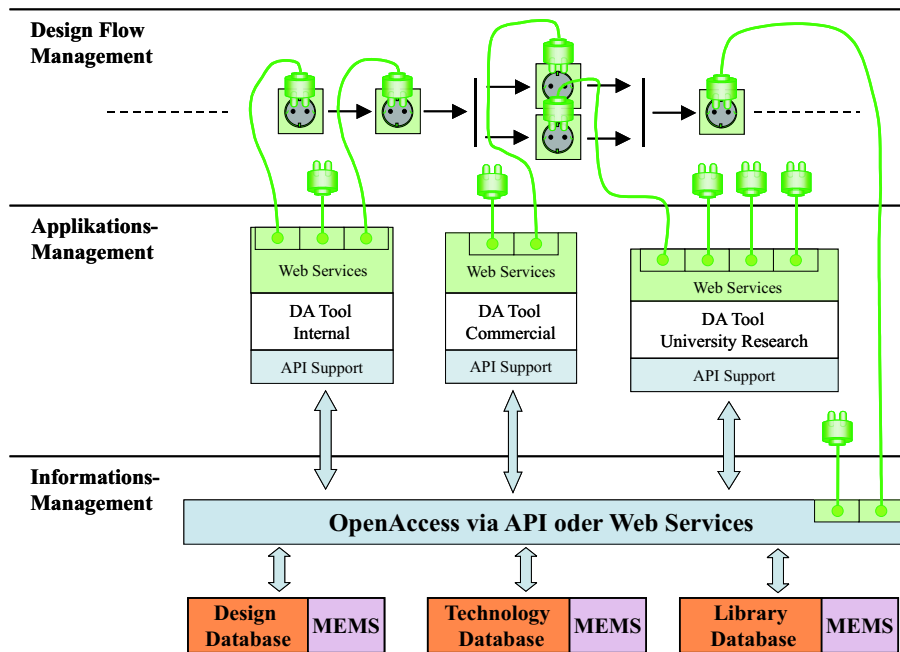


Abbildung 6.17: OA und SES

vices nach dem **Stecker/Steckdose Prinzip** verdeutlicht, dass die Interoperabilität zwischen DA-Tools vorhanden ist. Man kann zu jedem Task (Steckdose) einen Service (Stecker) zuordnen. Um einen Design Flow festzulegen, muss zusätzlich der Inhalt eines Web Services den Anforderungen der Entwurfsaufgabe entsprechen. Das bedeutet, wenn man den Bedarf hat Musik zu hören und drei Stecker zur Auswahl hat, wobei der eine zu einem Fernseher gehört, der andere zu einer Waschmaschine und der letzte zu einer Stereoanlage, so besteht die Aufgabe jetzt darin den richtigen auszuwählen und dann in die genormte Steckdose zu stecken. Genau vor dieser Herausforderung steht der Entwurfsexperte, wenn er zu einem Task einen Service zuordnen will.

Die Vorteile der Design Flow Management Disziplin sind unter qualitativen und quantitativen Aspekten nachfolgend zusammengefasst:

- unterstützt die anwendungsspezifische Entwicklung von Design Flows mit Hilfe der IDE und berücksichtigt damit die starke Abhängigkeit zwischen Entwurf und Fertigungsprozess in der Mikrotechnik
- die Design Flow Durchführung erfolgt mit einem Assistenten nach der best-of-breed Strategie. Er erlaubt es erstmalig, eine an den Bedürfnissen der Anwender (Task Flow) ausgerichtete Entwurfsautomatisierung (Service Flow) anzubieten

- Reduzierung von Integrationskosten, Support Aufwendungen und Betriebskosten durch Web Service basierte Komponenten-Software. Sie vermeidet zum einen teure lokale DA-Framework Installationen und zum anderen löst sie die kostenintensiven, herstellerabhängigen APIs der DA-Vendors ab
- führt die durch Open Access realisierte Datenintegration und die durch das service-orientierte Entwurfsszenario angebotene Applikationintegration zusammen. Durch den kombinierten Einsatz von OA und SES sind Design Flows deutlich schneller durchzuführen

Die Vorteile verdeutlichen, warum es sich lohnt, die Disziplin des Design Flow Management in der Mikrotechnik-Industrie einzuführen.

Kapitel 7

Zusammenfassung und Ausblick

Die Disziplin des Design Flow Managements unterstützt die KMU bei dem Ziel, MEMS-Produkte kostengünstiger, schneller und mit einer besseren Qualität am Markt anzubieten. Das service-orientierte Entwurfsszenario repräsentiert die operative Ebene, auf der mit Design Services, Design Flow IDE und Design Flow Assistenten die reale Umsetzung voranzutreiben ist. Die theoretischen Grundlagen für das service-orientierte Entwurfsszenario sind durch das Kreismodell gelegt worden. Abbildung 7.1 zeigt das Zusammenspiel zwischen Design Flow Management, Entwurfsszenario und Kreismodell.

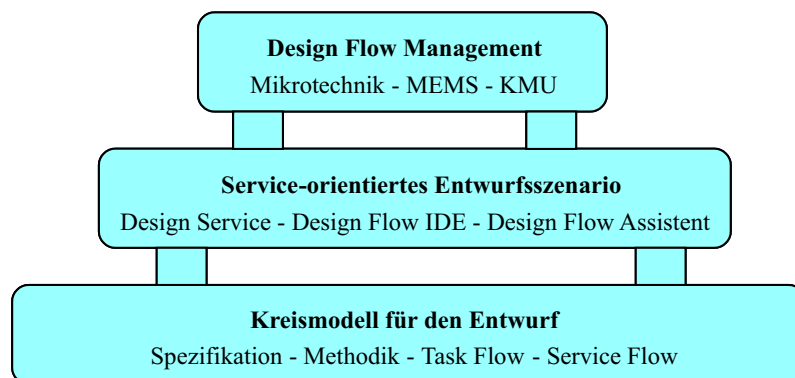


Abbildung 7.1: Relationen zwischen Design Flow Management, Entwurfsszenario und Kreismodell

Die hohe Flexibilität bei der Design Flow Gestaltung kombiniert mit niedrigen Integrationskosten und der Möglichkeit auch KMU mit hochwertigen Services

arbeiten zu lassen, sind die herausragenden Merkmale des neuen Entwurfsszenarios für die Mikrotechnik, welches im Rahmen dieser Arbeit erstmals vorgestellt wurde.

Die strikte Trennung von Applikationen zur Design Flow Entwicklung und Design Flow Durchführung und das damit verbundene, quasi automatische Generieren von Assistenten ist ein weiteres Innovationsmerkmal der Arbeit. Die Assistenten-Architektur enthält mit dem 3-Schichten-Modell, der Auszeichnungssprache XML, dem Applikationserver, dem Portal- und Komponentenansatz sowie der J2EE Spezifikation de facto IT Standards für die Implementierung und den Betrieb von Applikationen, wie sie in dieser Art heute weitgehend noch nicht in der DA-Industrie zum Einsatz kommen.

Die dynamische Nutzung von im Internet oder Intranet verfügbarem Entwurfs- und Fertigungswissen durch die Web Service Technologie erlaubt es auch, bestehende DA-Entwurfswerkzeuge in das neue Entwurfsszenario relativ einfach einzubinden. So ist gewährleistet, dass vorhandenes Wissen weiterverwendet wird und Sicherheitsaspekte durch den Einsatz von Protokollen wie dem **Secure Sockets Layer** Protokoll Berücksichtigung finden.

Das Kreismodell und die Methodik für den physikalischen Entwurf repräsentieren die theoretische Grundlage für das service-orientierte Entwurfsszenario. Sie sind aus dem Mangel an geeigneten Entwurfsmodellen für Mikrosysteme entwickelt worden und wurden speziell auf die Entwurfsanforderungen in der Mikrotechnik abgestimmt. Dabei wurden zum ersten Mal Task- und Service Flow vorgestellt und anhand von Beispielen erläutert.

Für die Akteure im Design Team ist eine top-down Vorgehensweise beim Entwurfsprozess beschrieben worden, die den Entwurfsgegenstand in den Mittelpunkt der Entwurfsaktivitäten stellt. Das service-orientierte Entwurfsszenario unterstützt diese Vorgehensweise mit dem Ziel, mittelfristig Mikrosysteme von Produktmanagern entwerfen zu lassen und die physikalischen Herausforderungen in die Hände der Fertigungsexperten zu übergeben. Diese Vorgehensweise ist nicht neu in der Mikrotechnik, wurde aber bisher noch nicht in Kombination mit einem dafür geeignetem Realisierungskonzept vorgestellt.

Für KMU ist das service-orientierte Entwurfsszenario eine große Chance, ihren Wunsch nach einem nutzungsbezogenen Abrechnungsmodell zu erfüllen. Sie können auf ihre Bedürfnisse abgestimmte Design Flows einsetzen und zugleich über Web Services die erprobten, leistungsstarken und mit großen Aufwand in vielen Tausenden von Mannjahren entwickelten Algorithmen der DA-Vendors in Anspruch nehmen. Das bedeutet für die KMU eine deutliche Reduzierung

des **TCO** (**T**otal **C**ost of **O**wnership) und somit ein weiteres Argument für Einsatz des in dieser Arbeit vorgestellten Entwurfsszenario.

Die DA Vendors sind aufgefordert, zukünftig ihre Werkzeuge mit Hilfe von Web Services gezielt zu öffnen und den Gedanken zu verinnerlichen, Software als einen Service anzusehen, insbesondere die Anbieter von Frameworks haben hier dringenden Handlungsbedarf. Es besteht sonst die Gefahr, dass die Produktivitätslücke zwischen Entwurf und Fertigung noch größer wird. Dies könnte den Siegeszug der nicht-elektronischen Mikrotechnik und somit auch von Mikrosystemen deutlich verzögern.

Weiterführende Arbeiten müssen sich mit der konkreten Umsetzung des serviceorientierten Entwurfsszenario beschäftigen. Anhand eines ersten Prototypen ist der Mehrwert des hier vorgestellten Realisierungskonzeptes deutlich zu machen. Damit wäre ein weiterer Schritt getan, um eine neue Qualität bei der Entwurfsautomatisierung in der nicht-elektronischen Mikrotechnik zu erzielen.

Abbildungsverzeichnis

1.1	Minisonde für die drahtlose Überwachung von Blutwerten. Die Silikonhülle enthält Antennenspule (links) und Mikrochip. Der Durchmesser beträgt ohne Haltebeinchen 2.3 Millimeter [SSR03]	2
1.2	Mikroelektronische Komponente und Drucksensor auf einem Siliziumsubstrat. Größe 5,7 mal 1,5 Millimeter [SSR03]	3
2.1	Mikrosystem	9
2.2	Airbag-Auslösesystem [Hah99]	10
2.3	Mikropumpe mit einer Größe von 6 mm x 10 mm [MHG ⁺ 01]	10
2.4	Prinzip der Erzeugung von Mikrostrukturen durch die Silizium-Mikromechanik [Büt94]	11
3.1	Entwurfsentscheidung	15
3.2	Entwurfsprozess	16
3.3	Spezifikationsphase	16
3.4	Formalisierungsprozess	17
3.5	Restriktionen	17
3.6	Einsatz von formalen Sprachen	18
3.7	Beispiel: Produktidee Hausbau	19
3.8	Beispiel: Spezifikationsphase	20
3.9	Beispiel: Entwurfsprozess	21
3.10	Makroskopisches Modell des Entwurfsprozesses nach [Ram89]	22
3.11	Verfeinertes Modell des Entwurfsprozesses nach [Ram89]	23

3.12	Entwurfsprozess-Variante mit Fertigungsspezifikation	24
3.13	Entwurfsprozess-Variante ohne Fertigungsspezifikation	24
3.14	Konvergenter Entwurfsprozess	25
3.15	Divergenter Entwurfsprozess	25
3.16	Relative Fehlerbeseitigungskosten bei der Entwicklung	27
3.17	Kreismodell	28
3.18	Spezifikationserstellung	29
3.19	Y-Modell	30
3.20	Beispiel für eine Entwurfsmethodik	31
3.21	Miniaturisierte Chipkühlung [Büt94]	33
3.22	Spezifikationsphase 1	35
3.23	Charakterisierende Variablen	35
3.24	Wärmeaufnahmefähigkeit der Chipkühlung	36
3.25	Spezifikationsphase 2	38
3.26	Form der Chipkühlung	39
3.27	Spezifikationsphase 3	40
3.28	Entscheidungsbaum	41
3.29	Beispiel für eine Mikrostruktur [Men97]	42
3.30	Aufbau der Spezifikation	43
3.31	Entwicklung einer Mikrostruktur	45
3.32	Spezifikation und Entwurfsprozess	46
3.33	Gesamtmodell mit Rückkopplung	50
4.1	Physikalischer Entwurf	52
4.2	Erweitertes Kreismodell für den physikalischen Entwurf	53
4.3	UML-Aktivitätsdiagramm für den physikalischen Entwurf	55
4.4	Zerlegung und Generierung	56
4.5	Entwurfsmethodik für den physikalischen Entwurf	60

4.6	Ausprägung der Entwurfsmethodik	62
4.7	Beispiel für einen Entwurfsschritt und Task Flow	63
4.8	Beispiel für einen Service Flow	64
4.9	Beispiel für einen Design Flow	66
4.10	Elemente beim Entwurf	70
5.1	Akteure in der Mikrotechnik-Industrie	74
5.2	Szenario in der Mikroelektronik-Großindustrie	76
5.3	Mikroelektronik-Design House	76
5.4	Special Design Service	77
5.5	Neues Entwurfsszenario für KMU	81
5.6	Vorgehensweise beim service-orientierten Entwurfsszenario	84
6.1	Beispiel für Centralized Computing Model	89
6.2	Logischer Aufbau der Mainframe-Architektur	89
6.3	Beispiel für eine Client/Server-Architektur	90
6.4	Logischer Aufbau der Client/Server-Architektur	91
6.5	Logischer Aufbau der 3-Schichten-Architektur	92
6.6	Beispiel für eine 3-Schichten-Architektur	93
6.7	Beispiel PRINCE User Interface	100
6.8	User Interface des Ätz-Simulations Service	100
6.9	Web Service [Lan03]	104
6.10	Web Services und 3-Schichten-Architektur	106
6.11	Design Service Portal	108
6.12	Design Flow Entwicklung	114
6.13	Design Flow Durchführung	118
6.14	Realisierungskonzept für das service-orientierte Entwurfsszenario	120
6.15	Datei basierter Datenaustausch	125
6.16	Open Access Technologie	126

6.17	OA und SES	128
7.1	Relationen zwischen Design Flow Management, Entwurfsszenario und Kreismodell	131

Literaturverzeichnis

- [BH96] BRÜCK, R. und K. HAHN: *Geometrische Entwurfsregelprüfung für lithographiebasierte Mikrotechnologien*. In: *Proc. GI FG 3.5.6 Fachtagung*, Frankfurt, Juni 1996.
- [BH97] BRÜCK, R. and K. HAHN: *An approach to layout and process verification for microsystem physical design*. Microsystem Technologies, Volume 02/97:82–90, 1997.
- [BHP⁺99] BRÜCK, R., K. HAHN, A. PRIEBE, C. SCHNEIDER, and C. SCHUMER: *Component based distributed design tools for microsystem-technologies*. In *Proc. Sensor 99*, Nürnberg, 1999.
- [BN03] BLISS, G. and C. NOWLIN: *Portal: Internet technology terms*; http://whatis.techtarget.com/definition/0,,sid9_gci212810,00.html. whatis.com, 2003.
- [Bot04] BOTTHOF, A.: *Was ist Mikrosystemtechnik? VDI/VDE Innovation + Technik GmbH*. <http://www.mstonline.de/>, 2004.
- [Brü93] BRÜCK, R.: *Entwurfswerkzeuge für VLSI-Layout: Methoden und Algorithmen für den rechnergestützten Entwurf von VLSI-Layout*. Hanser Verlag, 1993.
- [Brü96] BRÜCK, R.: *Der fertigungsnahe Entwurf von Mikrosystemen - Modelle, Methoden und Werkzeuge*. Universität Dortmund, Habilitation, 1996.
- [BRS01] BRÜCK, R., N. RIZVI und A. SCHMIDT: *Angewandte Mikrotechnik*. Fachbuchverlag Leipzig, 2001.
- [BS95] BRÜCK, R. and C. SCHUMER: *An object-oriented approach to physical microcomponent design*. Forschungsbericht 598, Universität Dortmund, Fachbereich Informatik, November 1995.

- [BS99] BRÜCK, R. and C. SCHUMER: *Internet mems design tools based on component technology*. In *Proc. Design, Test and Microfabrication of MEMS/MOEMS*, pages 316–325, Paris, 1999.
- [Büt94] BÜTTGENBACH, S.: *Mikromechanik: Einführung in Technologie und Anwendungen*. Teubner Studienbücher: Angewandte Physik, 2., durchges. Auflage, 1994.
- [Cox86] COX, B.: *Object-Oriented Programming: An Evolutionary Approach*. Addison-Wesley, 1986.
- [Ehr01] EHRFELD, W.: *Handbuch Mikrotechnik*. Fachbuchverlag Leipzig, 2001.
- [Ehr03] EHRMANN, S.: *Schau mir in die Augen*. c't Magazin für Computertechnik, Ausgabe 22, Seiten 106–121, 2003.
- [Gri98] GRIFFEL, F.: *Componentware: Konzepte und Techniken eines Softwareparadigmas*. dpunkt-Verlag, Heidelberg, 1998.
- [Hah99] HAHN, K.: *Methoden und Werkzeuge zur fertigungsnahen Entwurfsverifikation in der Mikrotechnik*. Doktorarbeit, Universität Siegen, Fachbereich Elektrotechnik und Informatik, 1999.
- [HPW03] HAHN, K., J. POPP, and A. WAGENER: *Prince - process information and management center*. In *Proceedings MICRO SYSTEM Technologies*, Munich, 2003.
- [HW03] HAHN, K. und A. WAGENER: *Eine Entwurfsmethodik für die Mikrosystemtechnik und Post-CMOS*. In: *Proceedings Austrochip*, Linz, Oktober 2003.
- [JBR98] JACOBSON, I., G. BOOCH, and J. RUMBAUGH: *The Objectory Software Development Process*. Addison-Wesley, 1998.
- [Joh01] JOHN, W. (Herausgeber): *Zukünftige Schwerpunkte für die Themen Entwurf - Simulation - Modellierung - Test*, Berlin, Dezember 2001. Arbeitskreis MST-Entwurfstechnik im Förderschwerpunkt Mikrosystemtechnik 2000+ (MST für HighTech-Produkte Made in Germany).
- [Kle02] KLEINERT, A.: *Entwicklung einer Prozessbeschreibungssprache für die Mikrosystemtechnik auf Basis von XML*. Diplomarbeit, Universität Siegen, Fachbereich Elektrotechnik und Informatik, 2002.

- [Lan98] LANG, M.: *Entwurfsmethoden und Rapid Prototyping integrierter Mikrosysteme*. Doktorarbeit, Technische Universität Darmstadt, 1998.
- [Lan03] LANGNER, T.: *Web Services mit Java: Neuentwicklung und Refactoring in der Praxis*. Markt+Technik Verlag, München/Germany, 2003.
- [LN04] LENTZER, A. und R. NEUL: *Ergebnisse der Umfrage zum Werkzeugeinsatz und Werkzeugbedarf beim Entwurf von Mikrosystemen*. Zentralverband Elektrotechnik- und Elektronikindustrie e.V., 2004.
- [Mad01] MADOU, M. J.: *Fundamentals of Microfabrication: the science of miniaturization*. CRC Press, 2nd edition, 2001.
- [McI68] MCILROY, M.D.: *Mass produced software components*. In NAUR, P. and B. RANDELL (editors): *Software Engineering: Report on a Conference by the NATO Science Committee*, pages 138–155. NATO Scientific Affairs Division, Brüssel, 1968.
- [Men97] MENZ, W.: *Mikrosystemtechnik für Ingenieure*. 2. erw. Aufl. - Weinheim: VCH Verlagsgesellschaft, 1997.
- [Mey97] MEYER, B.: *Object-Oriented Software Construction*. Prentice Hall International, Mai 1997.
- [MHG⁺01] MAILLEFER, D., H. VAN LINTEL, S. GAMPER, B. FREHNER, P. BALMER, and PH. RENAUD: *A high-performance silicon micropump for disposable drug delivery systems*. In *Proceedings of the MEMS conference*, pages 413–417, 2001.
- [Oes01] OESTEREICH, B.: *Objektorientierte Softwareentwicklung: Analyse und Design mit der Unified modeling language*. Oldenbourg Wissenschaftsverlag GmbH, München, 5. Auflage, 2001.
- [Pet82] PETERSEN, K.E.: *Silicon as a mechanical material*. Proc. IEEE, Vol. 70, pages 420–457, 1982.
- [Ram89] RAMMIG, F. J.: *Systematischer Entwurf digitaler Systeme: Von der System- bis zur Gatter-Ebene*. Teubner Verlag, Stuttgart, 1989.
- [SAP04] SAP AG: *Composite application framework: A robust environment for the design of composite applications*. <http://www.sap.com/>, 2004.

- [SB98] SCHUMER, C. und R. BRÜCK: *INTERLIDO - Web basierte Werkzeuge für den Mikrostrukturentwurf*. In: BRÜCK, R. (Herausgeber): *Workshop Multimedia und Mikrotechnik, Band 1*, Seiten 82–101. Fachbereich Elektrotechnik und Informatik der Universität Siegen, Institut für Rechnerstrukturen, 1998.
- [Sch92] SCHUMER, C.: *Extraktion von semantischen Informationen aus IC-Layouts*. Diplomarbeit, Universität Dortmund, Fachbereich Informatik, 1992.
- [Sch99] SCHUMER, C.: *Einsatz von Multi-Tier-Applikationen zur Unterstützung des Mikrostrukturentwurfs*. In: BRÜCK, R. (Herausgeber): *Workshop Multimedia und Mikrotechnik, Band 2*, Seiten 55–66. Fachbereich Elektrotechnik und Informatik der Universität Siegen, Shaker Verlag, 1999.
- [Sch02a] SCHUMER, C.: *Web Services*. In: BRÜCK, R. (Herausgeber): *Workshop Multimedia und Mikrotechnik, Band 4*, Seiten 119–132. Fachbereich Elektrotechnik und Informatik der Universität Siegen, Shaker Verlag, 2002.
- [Sch02b] SCHUMER, C.: *Workflow für den fertigungsnahen Entwurf von Mikrostrukturen*. In: BRÜCK, R. (Herausgeber): *Workshop Multimedia und Mikrotechnik, Band 3*, Seiten 87–102. Fachbereich Elektrotechnik und Informatik der Universität Siegen, Shaker Verlag, 2002.
- [Sch04] SCHMIDT, T.: *Konzeption und Implementierung einer Material- und Effektdatenbank für Simulation und Analyse in der Mikrosystemtechnik*. Diplomarbeit, Universität Siegen, Fachbereich Elektrotechnik und Informatik, 2004.
- [SM03] STIELER, W. und A. MEYER: *Erfinders treuer Helfer*. c't Magazin für Computertechnik, Ausgabe 7, Seiten 86–88, 2003.
- [SSBP00] SCHUMER, C., C. SCHNEIDER, R. BRÜCK, and J. POPP: *Design tool in 3-tier component-architecture for adaptive mems-design*. In *Proceedings of Micromechanics Europe Workshop*, Uppsala, Sweden, 2000.
- [SSBP01] SCHUMER, C., C. SCHNEIDER, R. BRÜCK, and J. POPP: *Component-based assistants for mems design tools*. In *Proceedings of MEMS Design, Fabrication, Characterization and Packaging*, pages 45–53, Edinburgh, April 2001.

- [SSR03] SCHNAKENBERG, U. und T. SCHMITZ-RODE: *Minisonde funkt Daten aus der Blutbahn*. <http://www.rwth-aachen.de/>, 2003.
- [Sun01] SUN MICROSYSTEMS INC.: *Java web start to the rescue*. <http://java.sun.com/>, 2001.
- [Sun02] SUN MICROSYSTEMS INC.: *Java web start architektur*. <http://java.sun.com/>, 2002.
- [Sun03] SUN MICROSYSTEMS INC.: *Java 2 platform enterprise edition specification, v1.4*. <http://java.sun.com/>, 2003.
- [Sun04] SUN MICROSYSTEMS INC.: *Java system application server platform edition 8*. <http://www.sun.com/>, 2004.
- [Szy02] SZYPERSKI, C.: *Component Software*. Addison Wesley, 2nd edition, 2002.
- [Tsc99] TSCHULENA, G.: *Mikrosystemtechnik: Grundlagen, Praxis, Trends*. Hüthig, Heidelberg, 1999.
- [Wag01] WAGENER, A.: *System- und Anforderungsanalyse für ein Prozessdesign-Werkzeug in der Mikrosystemtechnik*. Diplomarbeit, Universität Siegen, Fachbereich Elektrotechnik und Informatik, 2001.
- [WT85] WALKER, R.A. and D.E. THOMAS: *A model for design and representation*. In *Proc. 23rd Design Automation Conference*, pages 453–459, 1985.