

Korkiat Sedchaicharn

## **Eine rechnergestützte Methode zur Festlegung der Produktarchitektur mit integrierter Berücksichtigung von Funktion und Gestalt**

A computer-aided method for defining the product  
architecture with consideration of function and  
geometry

Band 45

Systeme ■ Methoden ■ Prozesse

Hrsg.: o. Prof. Dr.-Ing. Dr. h.c. A. Albers



Forschungsberichte

Korkiat Sedchaicharn

**Eine rechnergestützte Methode zur Festlegung  
der Produktarchitektur mit integrierter  
Berücksichtigung von Funktion und Gestalt**

A computer-aided method for defining the  
product architecture with consideration of  
function and geometry

Band 45

Herausgeber: o. Prof. Dr.-Ing. Dr. h.c. A. Albers

Copyright: IPEK Institut für Produktentwicklung, 2010  
Karlsruher Institut für Technologie (KIT)  
Universität des Landes Baden-Württemberg und  
nationales Forschungszentrum in der Helmholtz-Gemeinschaft

Alle Rechte vorbehalten

ISSN 1615-8113

# **Eine rechnergestützte Methode zur Festlegung der Produktarchitektur mit integrierter Berücksichtigung von Funktion und Gestalt**

Zur Erlangung des akademischen Grades eines  
**Doktors der Ingenieurwissenschaften**  
der Fakultät für Maschinenbau  
Karlsruher Institut für Technologie (KIT)

genehmigte  
**Dissertation**  
von

Dipl.-Ing. Korkiat Sedchaicharn  
aus Bangkok/Thailand

Tag der mündlichen Prüfung: 24. Juni 2010  
Hauptreferent: o. Prof. Dr.-Ing. Dr. h.c. A. Albers  
Korreferent: Prof. Dr. -Ing. G. Lanza



## **Vorwort des Herausgebers**

Wissen ist einer der entscheidenden Faktoren in den Volkswirtschaften unserer Zeit. Der Unternehmenserfolg wird in der Zukunft mehr denn je davon abhängen, wie schnell ein Unternehmen neues Wissen aufnehmen, zugänglich machen und verwerten kann. Die Aufgabe eines Universitätsinstitutes ist es, hier einen wesentlichen Beitrag zu leisten. In den Forschungsarbeiten wird ständig Wissen generiert. Dieses kann aber nur wirksam und für die Gemeinschaft nutzbar werden, wenn es in geeigneter Form kommuniziert wird. Diese Schriftenreihe dient als eine Plattform zum Transfer und macht damit das Wissenspotenzial aus aktuellen Forschungsarbeiten am IPEK Institut für Produktentwicklung Karlsruhe (ehemals: Institut für Maschinenkonstruktionslehre und Kraftfahrzeugbau) verfügbar.

Die Forschungsfelder des Institutes sind die methodische Entwicklung und das Entwicklungsmanagement, die rechnergestützte Optimierung von hochbelasteten Strukturen und Systemen, die Antriebstechnik mit einem Schwerpunkt auf den Gebieten Antriebsstrang-Engineering und Tribologie von Lager- und Funktionsreibsystemen, die Mikrosystemtechnik mit dem Focus auf die zugehörigen Entwicklungsprozesse sowie die Mechatronik. Die Forschungsberichte werden aus allen diesen Gebieten Beiträge zur wissenschaftlichen Fortentwicklung des Wissens und der zugehörigen Anwendung – sowohl den auf diesen Gebieten tätigen Forschern als auch ganz besonders der anwendenden Industrie – zur Verfügung stellen. Ziel ist es, qualifizierte Beiträge zum Produktentwicklungsprozess zu leisten.

Albert Albers



## **Vorwort zu Band 45**

Moderne Produkte, insbesondere in Form von mechatronischen Systemen, zeigen mittlerweile eine erhebliche Komplexität. Eine Vielzahl von Teilsystemen und Funktionen müssen in eine Produktlösung integriert werden. Hierbei sind unterschiedliche Vorgehensmodelle denkbar. Ein besonderer Aspekt für den Aufbau von Produkten ist die Produktarchitektur. Die Idee einer gezielten Entwicklung und Festlegung der Produktarchitektur ist Gegenstand vieler Forschungsarbeiten auf dem Gebiet der Produktentwicklungsmethodik. Man möchte von Anfang an durch eine geschickt gewählte Produktarchitektur möglichst wenige Interaktionen von Teilsystemen und Abhängigkeiten zwischen diesen erreichen. Verschiedene wissenschaftliche Ansätze dazu wurden erarbeitet. Bisher wird aber die Definition der Produktarchitektur insbesondere in der frühen Phase der Produktentstehung sehr stiefmütterlich behandelt. Der Grund liegt in einer oftmals wenig zugänglichen Vorgehensweise, die den natürlichen konstruktiven Arbeitsprinzipien der Produktentwickler nicht entspricht. An dieser Stelle setzt die Arbeit von Herrn Dr.-Ing. Korkiat Sedchaicharn an. Er hat sich zum Ziel gesetzt, durch die Entwicklung einer neuen methodischen Vorgehensweise auf der Basis des von Albers entwickelten Contact & Channel-Models (C&C-M), die Festlegung der Produktarchitektur von Anfang an unter Berücksichtigung auch der Gestaltinformation in den Produktentstehungsprozess zu integrieren. Die Methode fügt die Informationen aus dem Zielsystem und dem oft in Prinzip-Skizzen vorliegenden prinzipiellen Aufbauten und Konzepten durch einen strukturierten Prozess zusammen und leitet daraus Vorschläge für eine optimierte Produktarchitektur ab. Hierzu wird eine C&C-MODEL-Dependency-Matrix als Lösungselement vorgeschlagen. Diese Matrix wird dann in einem weiteren Prozessschritt systematisch bewertet und anschließend mit den Lösungsinformationen aus dem Zielsystem und den Kompatibilitätsbeziehungen zusammengebracht. So soll eine strukturierte und optimierte Produktarchitektur bereits in der frühen Phase der Produktentwicklung simultan mit der Entwicklung des Produktes aufgebaut werden.

Albert Albers



## Kurzfassung

Eine Produktarchitektur ist ein Schema der Beziehungen zwischen der Funktionsstruktur eines Produktes und seiner physischen Baustruktur. Die richtige Produktarchitektur ist ein wesentlicher Erfolgsfaktor eines Produktes und damit auch eines Unternehmens. Dennoch wird bei der Festlegung von Produktarchitekturen oft nicht ausreichend sorgfältig analysiert und nicht systematisch, nach quantitativen Kriterien und rechnerunterstützt vorgegangen. Manchmal wird die Produktarchitektur einer vorherigen Produktversion oder eines Konkurrenzprodukts übernommen. Häufig wird die Produktarchitektur im Produktentwicklungsprojekt eines komplexen Systems nicht ausreichend früh berücksichtigt oder sie wird nur intuitiv festgelegt, sodass die Gestaltung des Produktes zumindest vorerst auf einer intuitiv gewonnenen Architektur basiert. Eine solche frühzeitig intuitiv definierte Architektur ist aufgrund des zusätzlichen Zeitaufwands, in den sich anschließenden Entwicklungsphasen bzw. nach Konstruktion des Produktes, nur noch schwer zu ändern.

Der Stand der Forschung zur Festlegung der Produktarchitektur wird in dieser vorliegenden Arbeit erfasst. Die bisherigen Forschungsarbeiten in diesem Bereich werden in zwei Gruppen eingeteilt: in intuitive Ansätze (inklusive Funktions- und Gestaltinformationen) und quantitative Ansätze (auf der Grundlage der Integrationsanalyse von Funktionsinformationen). Diese bereits entwickelten Methoden existieren bislang unabhängig voneinander und besitzen jeweils ihre eigenen Vor- und Nachteile. Das Ziel dieser vorliegenden Forschungsarbeit ist die Entwicklung einer neuen Methode, die die Probleme der bisherigen Methoden eliminiert und die Vorteile der vorhandenen Ansätze verbindet.

Die vorliegende Arbeit beschreibt eine rechnergestützte Methode zur Festlegung der Produktarchitektur unter Berücksichtigung von Funktion und Gestalt. In dieser Arbeit wird ein Konzept zur Lösung des Problems mithilfe

- des Contact and Channel Models (C&C-M),
- der Dependency-Matrix,
- genetischer Algorithmen

und dessen Implementierung beschrieben.

Nach der Auswahl einer Prinziplösung, z. B. in Form einer Skizze, wird diese mit C&C-M Elementen und funktionalen Beschreibungen ergänzt. Das C&C-M verknüpft dabei Funktions- und Gestaltinformation miteinander. Danach lässt sich die Produktarchitektur durch eine Dependency-Matrix darstellen und mit Hilfe eines genetischen Algorithmus optimieren. Das Ergebnis der Methode ist eine Architektur, als Basis für das weitere Vorgehen, die die relevanten Ziele und Randbedingungen aus dem Zielsystem bestmöglich erfüllt. Die Produktarchitekturfestlegung eines Roboterunterarms, einer Bohrmaschine und eines humanoiden Roboterthorax mithilfe dieser Methode dienen als Beispiele.



## **Abstract**

Product architecture is a scheme by which the function of the product is allocated to physical components. Defining a product architecture is one important decision during product development, nevertheless the defining is often not systematic and existing product architectures are not carefully analyzed. In this step, the product architecture is often developed intuitively without any analysis, meaning that the design process is carried out based on the predefined architecture. Consequently, the obtained results are difficult to be used, or modified once the design is finished.

The state of the art of the defining the product architecture is described in this work. The previous researches in this research area can be divided into two groups; intuitive approach with both function and geometry information and quantitative approach based on integration analysis of function information, which are independent from each other and have their own advantages and disadvantages. The goal of this research is to develop a new method that eliminates the problems of previous methods and combines the advantages of both groups of existing approaches.

This research focuses on a computational method for defining the product architecture with consideration of function and geometry. A concept to solve the research problem with

- Contact and Channel Model (C&C-M),
- Dependency matrix,
- genetic algorithm

and the implementation are described in this work.

After a principle solution has been selected, the system is modeled with C&C-M elements. The function and geometry information are linked together in this step. An integration analysis using a Dependency matrix can be performed in parallel with the use of a search algorithm. The analysis result is a suggestion for an optimal architecture. This helps designers to decide the product architecture prior to the launch of embodiment design. This method is implemented in the development of a robot forearm for the humanoid robot ARMAR III, an electric drill and the thorax of the humanoid robot.



## Danksagung

Meinem Doktorvater Herrn Prof. Dr.-Ing. Dr. h. c. Albert Albers möchte ich herzlich für die Möglichkeit danken, diese Arbeit im IPEK-Team verfassen zu dürfen. Des Weiteren bin ich dankbar für seine vielen wertvollen Anregungen, Diskussionen und selbstverständlich die Betreuung der vorliegenden Arbeit.

Großer Dank geht auch an meinen Gruppenleiter Herrn Dipl.-Ing. Christian Sauter für seine große Unterstützung, die angenehmen Diskussionen und Ratschläge und auch das Korrekturlesen.

Ein großes Dankeschön gebührt auch dem gesamten IPEK-Team für das angenehme Arbeitsklima; besonders danken möchte ich der Forschungsgruppe EMM „Entwicklungsmethodik und Management“ für die Diskussionen und Anregungen, die meine Arbeit geprägt haben. Besonders hervorheben möchte ich die Mitarbeiter der Arbeitsgruppe Robotik (Herr Dr.-Ing. Sven Brudniok, Herr Dr.-Ing. Jens Ottnad und Herr Dipl.-Ing. Christian Sauter) und die Kollegen der SFB 588. Vielen Dank für die tolle Zusammenarbeit beim Aufbau des humanoiden Roboters ARMAR III. Dabei habe ich viel gelernt und auch riesen Spaß gehabt.

Für die sprachliche Korrektur möchte ich mich des Weiteren bei Herrn Steffen Maisch herzlich bedanken.

Bei der Königlichen thailändischen Regierung möchte ich mich für die finanzielle Unterstützung durch das Stipendium bedanken.

Ich danke meinen Eltern und auch alle Freunden, die mir immer wieder die Kraft zum Durchhalten in schwierigen Situationen gegeben haben.

Korkiat Sedchaicharn



# Inhalt

<b>1</b>	<b>Einleitung</b> .....	<b>1</b>
1.1	Aufbau der Arbeit .....	1
1.2	Motivation .....	2
<b>2</b>	<b>Grundlagen und Stand der Forschung</b> .....	<b>5</b>
2.1	Das integrierte Produktentstehungsmodell (iPeM) .....	5
2.2	Begriff „Phase“ in Produktentstehungsprozessen .....	11
2.3	Produktarchitektur .....	15
2.3.1	Begriffserklärungen.....	15
2.3.2	Typen von Produktarchitekturen.....	16
2.3.3	Modularitätsgrad .....	22
2.3.4	Vorteile der Produktarchitekturtypen .....	23
2.3.5	Festlegung der Produktarchitektur .....	24
2.4	Funktionen und Funktionsstrukturen.....	25
2.4.1	Funktionsbegriff .....	25
2.4.2	Funktionale Gliederung.....	28
2.5	Contact and Channel Model (C&C-M) .....	29
2.6	Dependency-Matrix .....	33
2.7	Genetische Algorithmen (GA) .....	38
2.8	Stand der Forschung – Produktarchitekturfestlegung .....	46
2.8.1	VDI-Richtlinie 2221 .....	47
2.8.2	Drei-Heuristiken-Methode nach STONE .....	49
2.8.3	Produktarchitekturfestlegung nach ALLEN und CARLSON-SKALAK.....	51
2.8.4	Produktarchitekturfestlegung nach VAN WIE.....	52
2.8.5	Produktarchitekturfestlegung nach HÖLTTÄ ET AL. ....	57
2.8.6	Produktarchitekturfestlegung nach ULRICH und EPPINGER .....	58
2.8.7	Zusammenfassung und Fazit zum Stand der Forschung.....	60
2.9	Problemstellung .....	62
<b>3</b>	<b>Zielsetzung der Arbeit</b> .....	<b>64</b>
<b>4</b>	<b>Lösungsansatz</b> .....	<b>65</b>
4.1	Lösungsansatz .....	65
4.2	Methodisches Vorgehen .....	66
4.3	Methodeneinsatz für Umkonstruktion .....	69
<b>5</b>	<b>Methodisches Vorgehen</b> .....	<b>71</b>
5.1	Erstellung des C&C-M und der C&C-M Dependency-Matrix .....	71
5.1.1	Erstellung der Funktionsstruktur zu einer Prinzipskizze .....	73
5.1.2	Verknüpfung von Funktionsinformation und Gestaltinformation.....	78
5.1.3	Übertragung der C&C-M Elemente in CDM .....	79
5.2	Bewertung der C&C-M Dependency-Matrix.....	80

5.2.1	Bestimmung der Kompatibilitätsbeziehungen .....	81
5.2.2	Gewichtungen der Beziehungen definieren.....	85
5.2.3	Bewertung der Beziehungen der Leitstützstrukturen der Tragstruktur.....	87
5.3	Optimierung der C&C-M Dependency-Matrix .....	88
5.3.1	Suche nach maximalem Wert der Zielfunktion .....	89
5.3.2	Unterstützungswerkzeug .....	98
<b>6</b>	<b>Anwendungsbeispiele der Methode .....</b>	<b>101</b>
6.1	Produktarchitekturfestlegung eines Roboterunterarms .....	101
6.2	Produktarchitekturfestlegung einer Bohrmaschine .....	110
6.3	Produktarchitekturfestlegung für einen humanoiden Roboterthorax .....	115
<b>7</b>	<b>Zusammenfassung und Ausblick .....</b>	<b>124</b>
<b>8</b>	<b>Literaturverzeichnis .....</b>	<b>127</b>
<b>9</b>	<b>Stichwortverzeichnis.....</b>	<b>133</b>



# Nomenklatur

<b>Zeichen und Abkürzungen</b>	<b>Beschreibung</b>
$B_{LSS_{ij}}$	Beziehungszahl zwischen $LSS_i$ und $LSS_j$
C&C-M	Contact and Channel Model
CAD	Computer Aided Design
CDM	Contact and Channel Model Dependency-Matrix
DSM	Design Structure Matrix
DW	Daten- und Wissensbank
ED	Editoren
$F_{WFP_{ij}}$	Funktionale Beziehung zwischen $LSS_i$ und $LSS_j$
GA	Genetischer Algorithmus
GS	Gesamtstruktur
$i$	Zeilennummer in der Matrix
IP	Integrierte Produktentwicklung
iPeM	Integrierte Produktentstehungsmodell
$j$	Spaltennummer in der Matrix
$K_{LSS_{ij}}$	Kompatibilitätsbeziehung zwischen $LSS_i$ und $LSS_j$
LSS	Leitstützstruktur
$m$	Anzahl der Ziele zur Erstellung der Produktarchitektur
MFD	Modular Function Deployment
MIM	Module Indikation Matrix
$m_k$	Index der letzten Leitstützstruktur im k-ten Modul
$m_p$	Index der letzten Leitstützstruktur im p-ten Modul
$n_k$	Index der ersten Leitstützstruktur im k-ten Modul
$N_{LSS}$	Anzahl der Leitstützstrukturen
$N_m$	Modulanzahl
$N_{m,opt}$	Optimale Modulanzahl
$n_p$	Index der ersten Leitstützstruktur im p-ten Modul
OM	Operationsmethoden
PM	Produktmodelle
QFD	Quality Function Deployment
RS	Reststruktur
TS	Tragstruktur
VDI	Verein Deutscher Ingenieure
WF	Wirkfläche

WFP	Wirflächenpaar
$Z$	Zielfunktionswert
$Z_{integral}$	Zielfunktionswert für völlige integrale Produktarchitektur
$Z_{max}$	Maximaler Wert der Zielfunktion
$Z_{modular}$	Zielfunktionswert für völlig modulare Produktarchitektur
$Z_{max} _{N_m}$	Maximaler Zielfunktionswert für eine bestimmte Modulanzahl
$Z_{min} _{N_m}$	Minimaler Zielfunktionswert für eine bestimmte Modulanzahl



# 1 Einleitung

## 1.1 Aufbau der Arbeit

Die vorliegende Arbeit lässt sich in fünf Ebenen gliedern: Untersuchungsziel, Methode zur Festlegung von Produktarchitekturen, Methodenbeschreibung, Validierung und Zusammenfassung (siehe Abbildung 1-1).

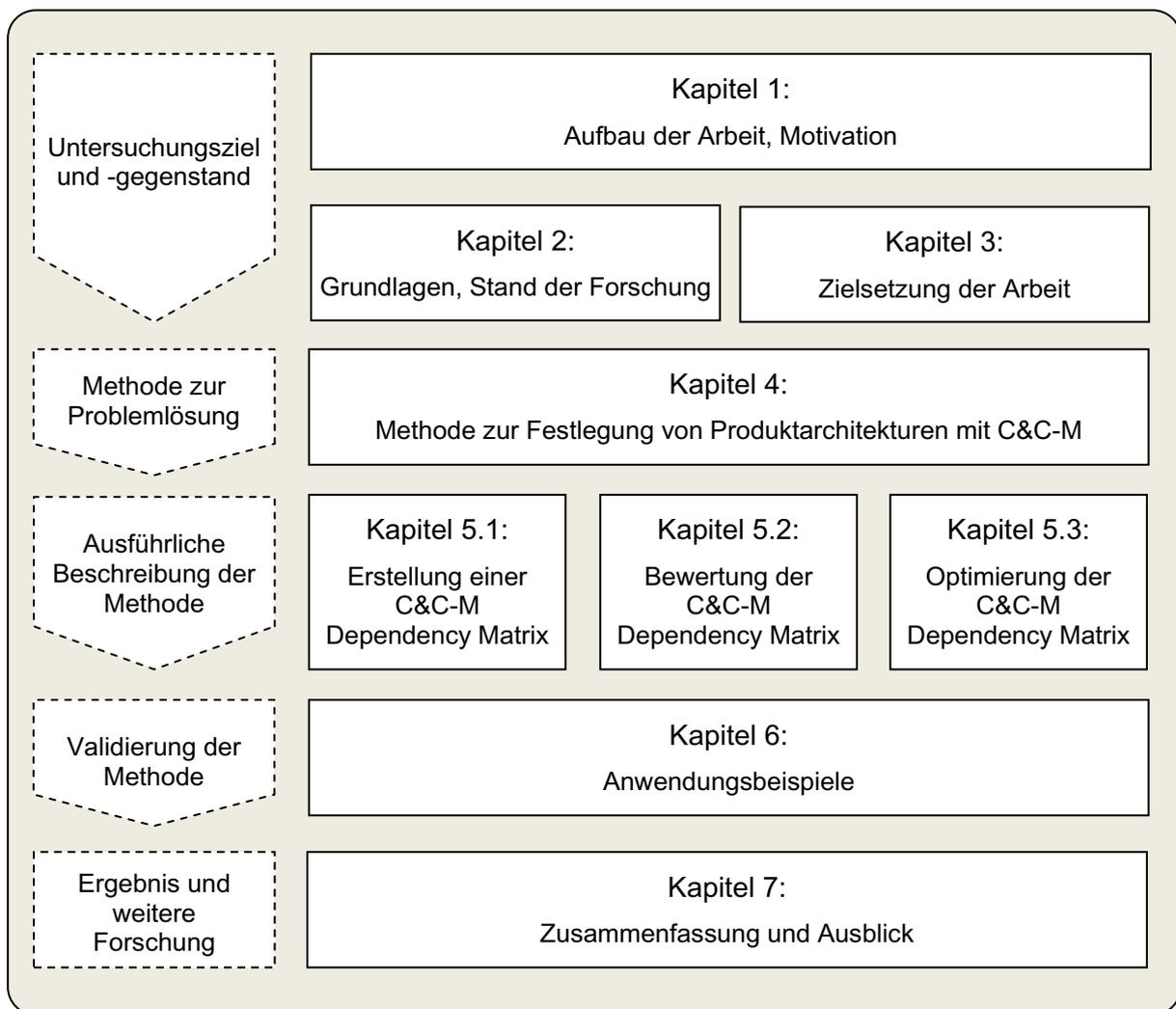


Abbildung 1-1. Struktur der Arbeit

Kapitel 1 enthält die Einleitung der Arbeit und gibt einen Einblick in die Bedeutung des Forschungsthemas. Auf diese Grundlagen und den Stand der Forschung wird in Kapitel 2 näher eingegangen. In diesem Kapitel werden verschiedene Methoden, mit denen Produktarchitekturen erstellt werden, dargestellt. Die Zielsetzung der Arbeit und der wissenschaftliche Fortschritt werden in Kapitel 3 vorgestellt.

In Kapitel 4 wird eine Methode zur Festlegung von Produktarchitekturen mit integrierter Berücksichtigung von Funktion und Gestalt mithilfe des Contact and Channel Models (C&C-M)<sup>1</sup> aufgezeigt. Das gesamte Vorgehen bei der Methode wird beschrieben, um einen Überblick zu geben, wie die Methode durchzuführen ist.

Die ausführliche Beschreibung der Methoden wird durch folgende drei Kapitel ergänzt: In Kapitel 5.1 wird gezeigt, wie eine C&C-M Dependency-Matrix (CDM) von einer Prinzipskizze, einer Funktionsbeschreibung und einem C&C-M abgeleitet wird. Diese CDM stellt eine andere Sicht der Prinzipskizze hinsichtlich der Systemintegrationen dar. Das Verfahren zur Bewertung der CDM, das von den Zielen im Zielsystem abhängig ist, wird in Kapitel 5.2 beschrieben. In Kapitel 5.3 wird ein Suchalgorithmus zur Optimierung der CDM eingesetzt. Die Lösung der von der CDM abgeleiteten Zielfunktion mit einem genetischen Algorithmus ist vorteilhaft, weil sie analytisch noch nicht lösbar ist. Auch in Kapitel 5.3 werden einige rechnerbasierte Werkzeuge zur Unterstützung der Methode gezeigt, damit die Anwendung des Ansatzes einfacher wird. Die Methode wird begleitend mit der Produktarchitekturfestlegung für ein Kreuzgelenk veranschaulicht.

Die Validierung der Methode wird in Kapitel 6 dargestellt. Die Festlegung der Produktarchitektur eines Roboterunterarms, einer Bohrmaschine und eines humanoiden Roboterthorax dienen als Beispiele.

Im letzten Kapitel werden die Inhalte der Arbeit zusammengefasst. Die Vor- und Nachteile der Methode und sinnvolle weitere Forschungsvorhaben werden in diesem Kapitel diskutiert.

### **1.2 Motivation**

Die Produktarchitektur ist ein wesentlicher Erfolgsfaktor für die Produkte eines Unternehmens und damit für das ganze Unternehmen. Dennoch wird bei der Festlegung von Produktarchitekturen oft nicht ausreichend sorgfältig analysiert und nicht systematisch, nach quantitativen Kriterien und rechnerunterstützt vorgegangen. Die Produktarchitektur ist ein zentrales Objekt in einem Unternehmen, da dort die Produktentwicklung sowie die Produktion und Organisation, die wichtige Elemente im Unternehmen sind, miteinander vernetzt werden (siehe Abbildung 1-2). Eine Änderung der Produktarchitektur kann die Produktfunktionalität, Produktplanung, Technologie, Organisationsstruktur und Kostenstruktur oder sogar die Unternehmensstrategie, beeinflussen.<sup>2</sup> Etwa 80 % der Kosten, 50 % der Qualität,

---

<sup>1</sup> Albers et al. 2005

<sup>2</sup> Vgl. Schuh/Schwenk 2001, S. 78

50 % der Zeit, 80 % der geschäftlichen Komplexität durch Produkt- und Prozessdesign, welche direkt auf die Produktarchitektur bezogen sind, können einfach variiert werden.<sup>3</sup> Aufgrund ihrer vielseitigen Verknüpfung und starken Auswirkung ist die Festlegung der Produktarchitektur eine wichtige Entscheidung im Produktentwicklungsprozess. Wenn sich ein Produkt bereits auf dem Markt befindet, ist es sehr schwierig, nachträgliche Änderungen vorzunehmen.

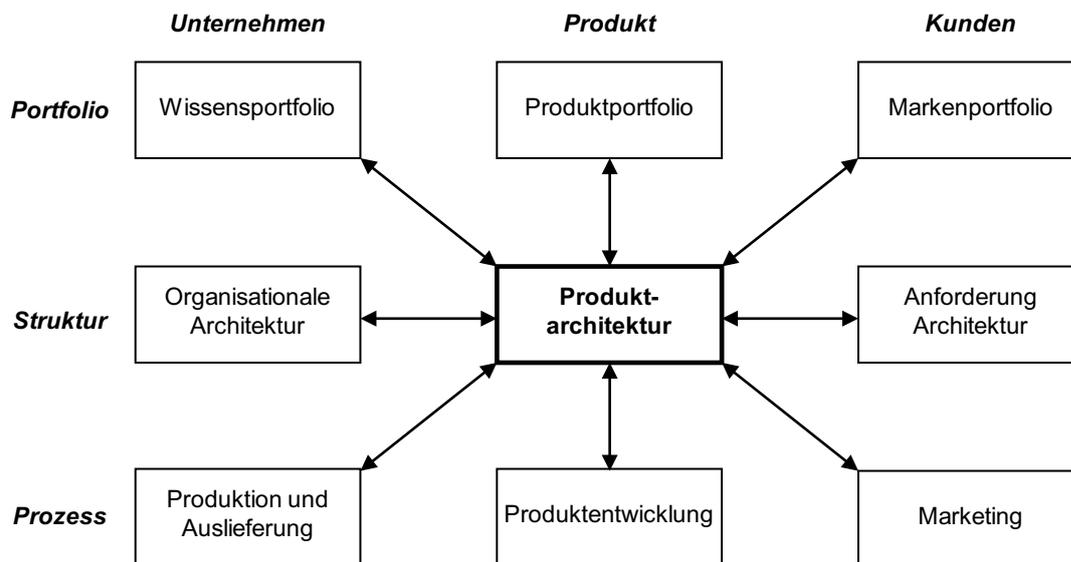


Abbildung 1-2. Beziehung der Produktarchitektur mit acht wichtigen Domänen in Unternehmen (modifiziert von YASSINE und WISSMANN<sup>4</sup>)

ANDERSON<sup>5</sup> postuliert, dass sich Entwickler oft nicht systematisch und sorgfältig mit der Festlegung von Produktarchitekturen beschäftigen, manchmal sogar die Produktarchitektur einer vorherigen Version oder eines Konkurrenzprodukts übernommen wird; besonders häufig sei dies bei der Entwicklung komplexer Systeme der Fall, in dem die funktionalen Anforderungen aus Zielsystemen meistens erste Priorität haben, obwohl die Festlegung der Produktarchitektur so bedeutungsvoll ist. Es ist nicht einfach für einen Entwickler, intuitiv und ohne Unterstützung durch eine systematische Methode und ein Analysewerkzeug eine optimale Produktarchitektur zu gestalten.<sup>6</sup>

Vor allem in den Aktivitäten zu Beginn eines Entwicklungsprojektes werden wichtige Entscheidungen getroffen, die für die Produktion und die spätere Nutzung unmittelbar kosten- und qualitätsrelevant sind. Wenn sich später Veränderungen der

<sup>3</sup> Vgl. Child et al. 1991, S. 78

<sup>4</sup> Vgl. Yassine/Wissman 2007, S. 122

<sup>5</sup> Vgl. Anderson 2004, S. 79

<sup>6</sup> Vgl. Albers et al. 2008b, S. 1

Produktarchitektur als notwendig erweisen, ist es zeit- und kostenaufwendig, diese umzusetzen (siehe Abbildung 1-3). Eine Analyse der Produktarchitektur erst gegen Ende des Entwicklungsprojektes birgt ein großes Risiko; Änderungen an der Produktarchitektur verursachen dann zusätzliche Kosten und sind zeitaufwendig – im ungünstigsten Fall sind sie gar nicht mehr möglich.

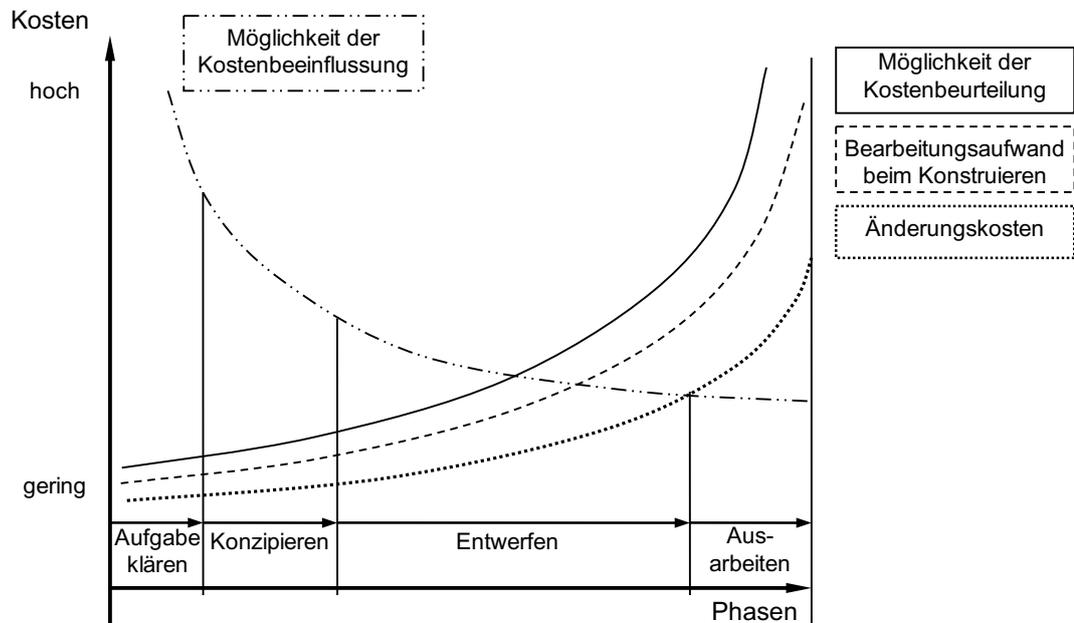


Abbildung 1-3. Phasen des konstruktiven Entwurfsprozesses und Kosten nach EHRENSPIEL<sup>7</sup>

Tatsächlich wird bereits seit Langem in diesem Bereich geforscht. Dabei geht es jedoch häufig um die Produktarchitekturanalyse eines vorherigen oder bestehenden Produkts. Das Ergebnis einer solchen Analyse ist zwar belastbar, kann jedoch meist nicht mehr in das aktuelle Produkt einfließen da die Gestaltung bereits abgeschlossen oder sehr weit fortgeschritten ist und Änderungen an so etwas Grundlegendem wie der Produktarchitektur meist sehr Zeit- und Kosten-intensiv sind.<sup>8</sup>

Bei diesem Vorgehen werden die Vorteile einer optimalen Produktarchitektur nicht ausgenutzt. Um solche Situationen zu vermeiden, muss die Festlegung der Produktarchitektur frühzeitig in Produktentwicklungsprojekte integriert werden.

<sup>7</sup> Ehrlenspiel 1995

<sup>8</sup> Vgl. Anderson 2004, S. 79

---

## 2 Grundlagen und Stand der Forschung

Alle wesentlichen Grundlagen für diese Arbeit und der Stand der Forschung für die Produktarchitekturfestlegung werden in diesem Abschnitt vorgestellt.

Die klassischen Methoden, die auf sequenziellen Schritten basieren, sind zwar wichtig und hilfreich für die Entwicklung, leider entsprechen sie nicht den realen Abläufen.<sup>9</sup> Ein integriertes Produktentstehungsmodell (iPeM), das auf Aktivitäten der Produktentstehung und Aktivitäten der Problemlösung basiert, kann diese Entwicklungsabläufe besser darstellen. Danach wird die Unklarheit des Begriffs „Phasengrenze“, auf dem viele Produktentwicklungsmethoden basieren, bzw. des Begriffs „Phase“ diskutiert.

Anschließend wird eine kurze Übersicht über den Begriff, Typen, Vorteile und Festlegung der Produktarchitektur geben. Die Grundlagen einer Funktion, Funktionsstruktur, das Contact and Channel Model (C&C-M), C&C-M Dependency-Matrix (CDM) werden als nächster Punkt in den Kapiteln 2.4, 2.5 und 2.6 behandelt. Danach wird der Algorithmus zur Optimierung der Architektur untersucht. In diesem Fall wird der genetische Algorithmus angewendet.

Der Stand der Forschung wird durch 6 unterschiedliche Methoden im Kapitel 2.8 beschrieben. In diesem Abschnitt werden sie auch miteinander verglichen. Es wird der weitere Forschungsbedarf aufgezeigt und dadurch die Problemstellung formuliert.

### 2.1 Das integrierte Produktentstehungsmodell (iPeM)

Das integrierte Produktentstehungsmodell (iPeM) wurde von ALBERS<sup>10</sup> entwickelt. Der Systemansatz des iPeM wird durch drei Systeme (Zielsystem, Handlungssystem, Objektsystem, kurz: ZHO-Systeme) beschrieben. Das Zielsystem beschreibt die vorgedachten und geplanten Eigenschaften des Objektsystems und des Produktes.<sup>11</sup> Es beschreibt die Ziele, deren Wechselwirkungen und Randbedingungen, die für die Entstehung eines Objektsystems notwendig sind, sowie deren Begründung. Das Zielsystem beschreibt auf Basis des bestehenden Zustands des Objektsystems den zukünftig gewollten Zustand. Das Zielsystem wird mit Fortschreiten der Produktentstehung kontinuierlich erweitert und konkretisiert. Eine richtige,

---

<sup>9</sup> Vgl. Albers/Meboldt 2007; Vgl. Albers 2010, S. 10

<sup>10</sup> Vgl. Albers 2010, S. 11

<sup>11</sup> Vgl. Albers 2010, S. 3; Deigendesch 2009, S. 33

konsistente und vollständige Definition des Zielsystems bietet die Grundlage einer erfolgreichen Produktentstehung.

Das Zielsystem steht dem Objektsystem gegenüber. Das Objektsystem enthält alle Objekte, die Ergebnis der Zielerreichung sind. Das Objektsystem ist vollständig sobald der bestehende Zustand dem geplanten entspricht. Im Objektsystem repräsentierte Objekte müssen nicht unbedingt physisch manifestiert sein, sondern können auch Software oder Dienstleistungen sein. Zudem besteht das Objektsystem nicht nur aus dem endgültigen Produkt, sondern auch aus auf dem Weg der Zielerreichung benötigten Objekten, wie Dokumentationen, Prüfberichte, Prototypen, technischen Zeichnungen, CAD-Modellen etc. Die zentrale Endgröße des Objektsystems und des Systems der Produktentstehung sind die Produkte.

Zwischen dem Zielsystem und dem Objektsystem besteht keine direkte Wechselwirkung. Die Wechselwirkung zwischen diesen zwei Systemen kann lediglich durch das Handlungssystem erfolgen. Das Handlungssystem ist ein soziotechnisches System, das sich aus strukturierten Aktivitäten, Methoden und Prozessen sowie aus zur Realisierung notwendigen Ressourcen, wie Personal, Maschinen, Budget, etc., zusammensetzt. Nach ALBERS<sup>12</sup> setzt sich ein Handlungssystem aus fünf Sub-Systemen zusammen. Diese sind: die Aktivitäten der Produktentstehung, die Aktivitäten der Problemlösung, das Ressourcen-System, die Implementierungsmodelle und die Anwendungsmodelle im Phasenmodell (siehe Abbildung 2-1). Das Handlungssystem generiert sowohl Ziel- als auch Objektsystem.

Zu Beginn der Produktentstehung sind sowohl das Ziel- als auch das Objektsystem noch unbestimmt und vollständig offen. Beide Systeme beeinflussen sich gegenseitig und wirken in der Produktentstehung wechselseitig durch das Handlungssystem aufeinander ein.

Das heißt, dass die Eigenschaften des Objektsystems zu den Eigenschaften im Zielsystem passen müssen und umgekehrt. Dadurch wird die Kontingenz des Objektssystems reduziert, und es entsteht eine Ordnung, aus der am Ende das Produkt hervorgeht. Die zwei Aktivitätensysteme, die für das Handlungssystem wesentlich sind, sind: Aktivitäten der Produktentstehung und Aktivitäten der Problemlösung (siehe Abbildung 2-1).

Die Aktivitäten der Produktentstehung beschreiben keinen sequenziellen Ablauf des Entwicklungsprozesses. Dieser wird erst durch die Anwendung bestimmt. ALBERS<sup>13</sup> stellt die Hypothese auf, dass jeder Prozess der Entstehung eines bestimmten

---

<sup>12</sup> Vgl. Albers 2010, S. 6

<sup>13</sup> Vgl. Albers 2010, S. 4

Produktes immer einzigartig und individuell ist. Die Aktivitäten müssen somit nicht sequenziell durchlaufen werden.

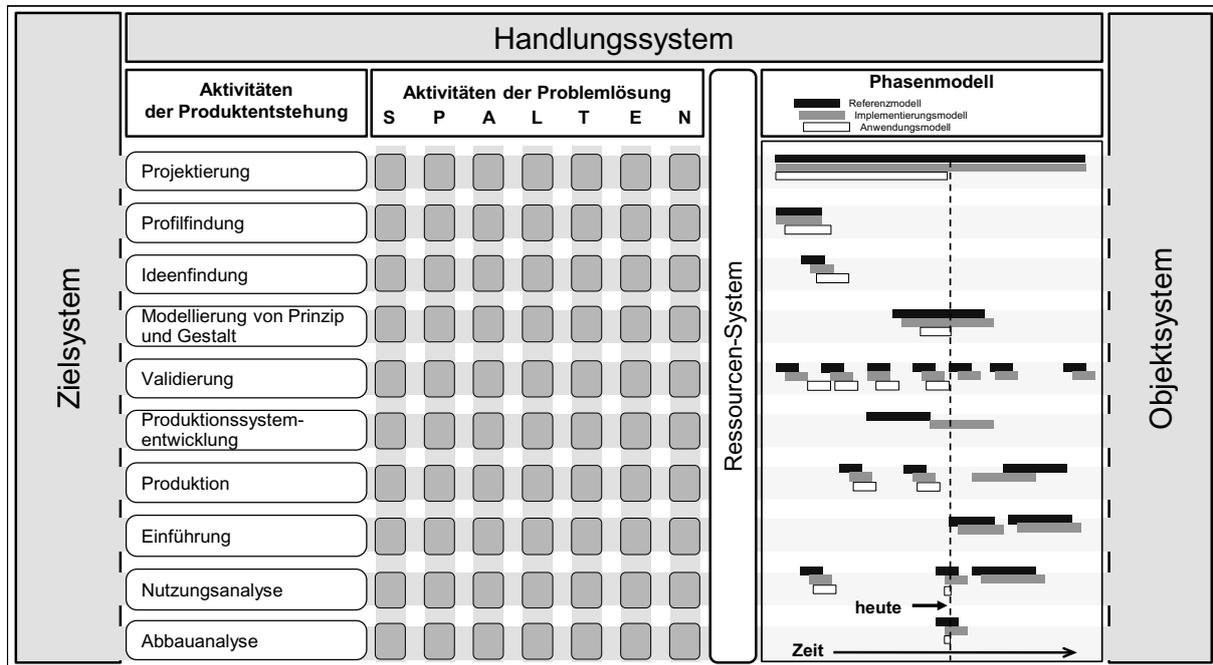


Abbildung 2-1. Das integrierte Produktentstehungsmodell (iPeM)<sup>14</sup>

Eine Aktivität wird somit wie folgt nach ALBERS<sup>15</sup> definiert: Aktivitäten sind die Ausführungseinheit des Handlungssystems. Sie setzen sich aus Aktionen, ausführenden Ressourcen, zu benutzenden Ressourcen und logischer Abhängigkeit zusammen. Dabei sind sie jedoch nicht als chronologische Reihung einer sequenziell abzuarbeitenden Abfolge von Prozessschritten zu verstehen. Aktivitäten sind vernetzte Aktionen und übergeordnete Ausführungseinheiten des Handlungssystems. Größere Zusammenhänge zwischen Aktivitäten, in einen zeitlichen Kontext gebracht, bilden einen Prozess. Aktionen führen einen Zustand in einen anderen über.

Die klassischen Produktentwicklungsansätze versuchen, das Prozessmodell auf der Ebene von Phasen und Methoden zu strukturieren, allerdings zeigt es sich immer wieder, dass reale Abläufe sich nicht allein durch Phasen und Methoden beschreiben lassen. Die einzelnen Aktivitäten lassen sich nicht vollständig differenzieren; dies zeigt sich deutlich an den Problemen der Anwendung der ursprünglichen Phasenmodelle (siehe Abschnitt 2.2). Das Phasenmodell visualisiert den Projektstand. Dabei umfasst das Phasenmodell drei Stufen: (1) Referenzmodell, (2)

<sup>14</sup> Vgl. Albers 2010, S. 11

<sup>15</sup> Vgl. Albers 2010, S. 4; Oerding 2009, S. 106

Implementierungsmodell und (3) Anwendungsmodell. Das Referenzmodell basiert auf den Erfahrungen vergangener Entstehungsprozesse eines Anwendungsbereichs und dient zur planerischen Orientierung. Ist ein Referenzmodell für einen Anwendungsbereich vorhanden, lässt sich aus diesem ein Implementierungsmodell ableiten, das fallspezifisch die Aktivitäten der Produktentstehung zeitlich verknüpft und terminlich fixiert. Es dient der inhaltlichen und zeitlichen Planung eines Entwicklungsprojektes. Die Kontrolle und Dokumentation der konkreten Projektdurchführung führt zum Anwendungsmodell. Es dient der Projektverlaufsdokumentation, dem übergeordneten Wissensmanagement und im Abgleich mit dem Referenzmodell zur kontinuierlichen Verbesserung von Referenzmodell und Entstehungsprozess.<sup>16</sup>

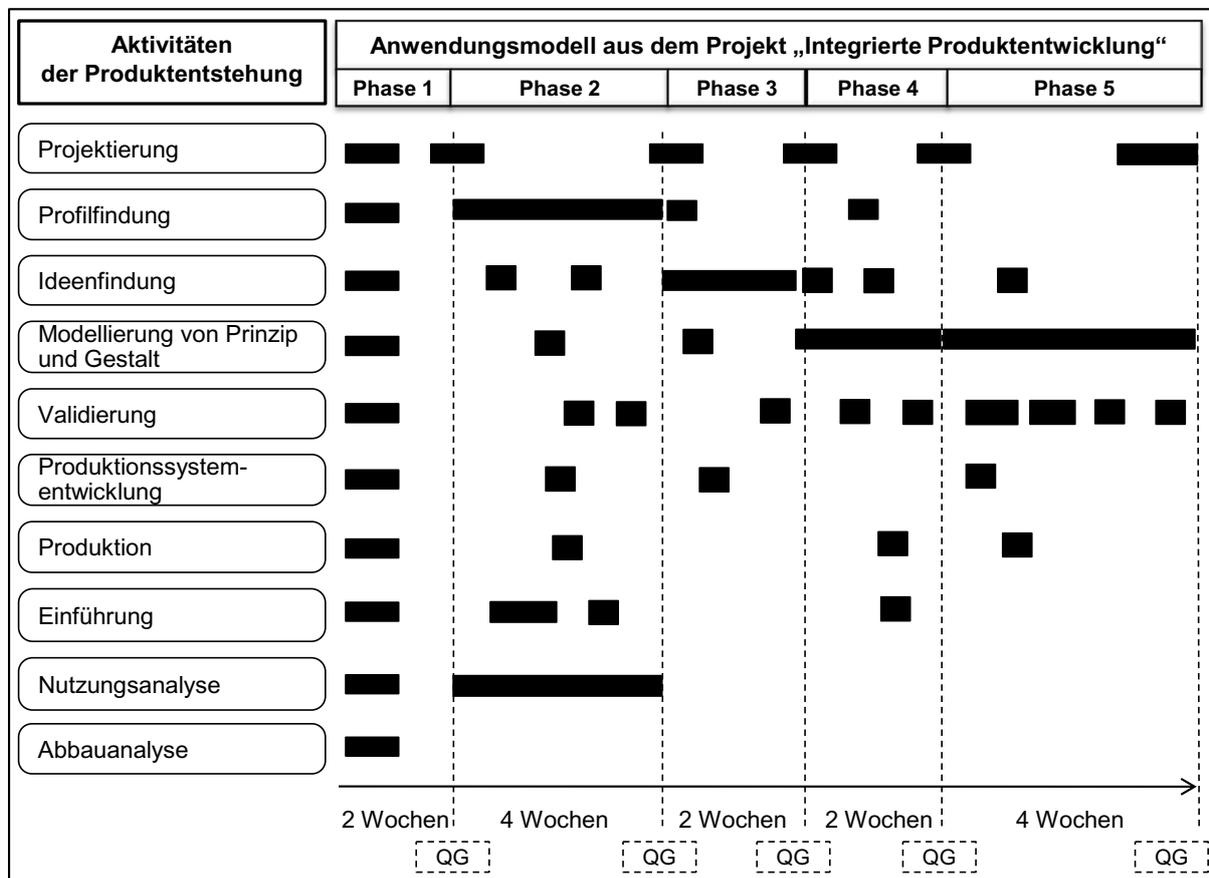


Abbildung 2-2. Ein Beispiel für ein Anwendungsmodell des Integrierte Produktentwicklung Projektes

In der Produktentwicklung finden viele Aktivitäten parallel statt. Zum Beispiel, in der Entwicklung von hoch-integrierten mechatronischen Systemen<sup>17</sup> finden die

<sup>16</sup> Vgl. Meboldt 2008; Vgl. Albers 2010; Vgl. Deigendesch 2009

<sup>17</sup> Vgl. Brudniok 2008,

Aktivitäten der „Modellierung von Prinzip und Gestalt“ immer mit den Aktivitäten der „Validierung“ statt. Durch diesen aktivitätsbasierten Ansatz von ALBERS wird das Problem der Phasentrennung überwunden. Ein weiteres Beispiel wird in Abbildung 2-2 gezeigt.

Das Anwendungsmodell in Abbildung 2-2 zeigt einen tatsächlichen Projektverlauf aus der Hauptfachveranstaltung Integrierte Produktentwicklung (IP)<sup>18</sup> am IPEK – Institut für Produktentwicklung. Im Rahmen dieser Lehrveranstaltung bearbeiten ca. 42 Studierende in Teams zu je 6 Personen während eines Semesters eine komplexe Innovationsaufgabe in enger Kooperation mit einem Industriepartner. Das Modell in Abbildung 2-2 zeigt den parallelen Ablauf von verschiedenen Aktivitäten wie sie reale Entwicklungsprojekte aufweisen. Es ist in diesem Anwendungsmodell deutlich zu erkennen, dass im Projektverlauf von Anfang bis zum Ende des Entwicklungsprojektes sehr viele unterschiedliche Aktivitäten zeitgleich durchgeführt werden. Durch die Visualisierung der Aktivitäten zusammen mit den Phasen (im Phasenmodell) wird anschaulich dargestellt welche Aktivitäten im Verlauf eines Produktentstehungsprojektes in den Projektphasen stattfinden.

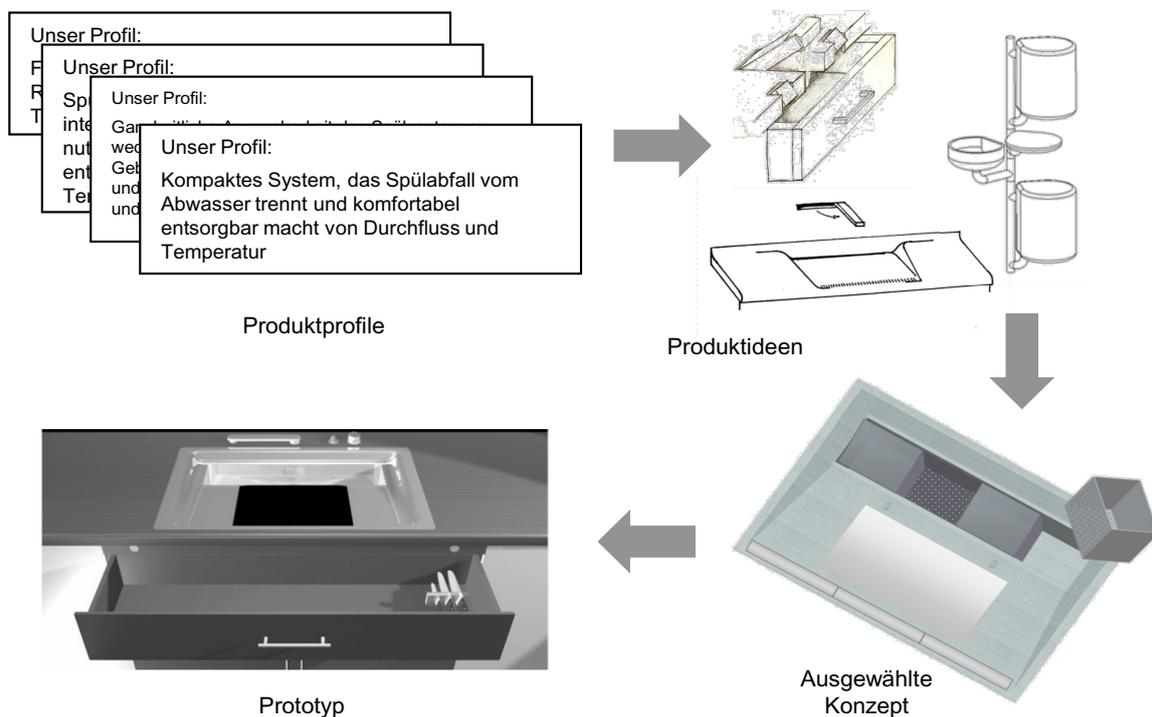


Abbildung 2-3. Beispiele für die Produktprofile, Produktideen, Produktkonzept, Produktprototyp des IP- Projektes (Kooperation mit Firma Blanco)

<sup>18</sup> Albers et al. 2009a

Der Projektablauf in dieser Lehrveranstaltung ist in die folgenden fünf Projektphasen unterteilt: Recherchephase (Phase 1), Profilphase (Phase 2), Ideenphase (Phase 3), Konzeptphase (Phase 4) und Prototypenphase (Phase 5). Die Ergebnisse der Phasen zwei bis fünf werden als „Profil“, „Idee“, „Konzept“ und „Prototyp“ bezeichnet werden. Nach ALBERS<sup>19</sup> werden solche Zwischenergebnisse wie folgt verstanden (siehe Beispiele in Abbildung 2-3):

Ein Produktprofil beschreibt zukünftige Potenziale und Bedarfssituationen am Markt. Dabei wird die Sicht des Kunden berücksichtigt, die Wettbewerbssituation sowie Aspekte des eigenen Unternehmens wie z.B. strategische Unternehmensziele, geplanter Anbieternutzen, Kernkompetenzen oder Markenimage. Ziel ist es, eine Marktlücke, beziehungsweise ein Marktsegment für sich auszumachen, in dem gezielt nach technischen Lösungen gesucht werden kann, um diese möglichst erfolgreich ausfüllen zu können.

Eine Produktidee ist der Kern einer möglichen Lösung für die Realisierung des Produktprofils. Die Beschreibung der Produktidee sollte nur die wesentlichen Eigenschaften und Charakterzüge des neuen Produkts beschreiben. Die Beschreibung der Produktidee kann mit Hilfe einer groben und abstrakten Darstellung der Gestalt, z. B. in einer Freihandskizze erfolgen oder einfach durch eine verbale Beschreibung der Produktidee.

Der weitere Projektverlauf bis zur endgültigen Gestaltfestlegung wird in zwei Projektphasen unterteilt, die Konzeptphase und die Prototypenphase. Die dominante Aktivität der Produktentstehung in diesen beiden Projektphasen ist die „Modellierung von Prinzip und Gestalt“. Dabei kann die Modellierung beider Aspekte nicht getrennt voneinander erfolgen sondern aufgrund gegenseitiger Abhängigkeit immer nur zusammen. Am Ende des ersten Abschnittes stellen die Studierenden mehrere Konzepte basierend auf der ausgewählten Produktidee vor, von denen eines für die endgültige Ausarbeitung ausgewählt wird. Ein Konzept ist hierbei ein noch nicht vollständig definiertes Modell des späteren Produkts das bereits wesentliche Anteile an Gestaltinformation enthält, allerdings noch nicht mit voller Detaillierung und Konkretisierung. Auch wenn die Gestalt hierbei noch nicht endgültig festgelegt ist, können Konzepte zu diesem Zeitpunkt bereits gut mit einer Skizze oder einem CAD-Modell visualisiert werden. Die in dieser Arbeit entwickelte Methode ist hauptsächlich für die Anwendung in der Aktivität der Modellierung von Prinzip und Gestalt geeignet, wobei eine Prinzipskizze<sup>20</sup> als eine Darstellung des Konzepts verwendet wird.

---

<sup>19</sup> Vgl. Meboldt 2008, S. 164 ff.

<sup>20</sup> Siehe Definition Prinzipskizze in Abschnitt 5.1.1

## 2.2 Begriff „Phase“ in Produktentstehungsprozessen

Der Begriff „Phase“ wird generell zur Einteilung der Prozessschritte von Produktentstehungsprozessen (beispielsweise in der VDI-Richtlinie 2221, PAHL/BEITZ oder bei ULRICH und EPPINGER<sup>21</sup>) angewendet. Nach der VDI-Richtlinie 2221 wird das gesamte Vorgehen in vier Phasen eingeteilt, wobei die Bezeichnungen der Phasen und die Phasengrenzen für jede Branche spezifisch sind. In den Branchen Verfahrenstechnik und Feinwerktechnik werden die Entwicklungsprozesse in drei Phasen aufgeteilt. Bei verfahrenstechnischen Fragestellungen wird die Verfahrensentwicklung in drei Konkretisierungsphasen (Vorstudie, Hauptstudie, Detailstudie) eingeteilt. In der Feinwerktechnik lässt sich der gesamte Ablauf gliedern in: konzipieren, entwerfen, ausarbeiten (siehe Abbildung 2-4).

Beim Maschinenbau lässt sich der Entwicklungsprozess zum Beispiel nach der VDI-Richtlinie 2221 in vier Hauptphasen (Aufgaben erklären, konzipieren, entwerfen, ausarbeiten) unterteilen<sup>22</sup> (siehe Abbildung 2-4), jedoch kann auch eine andere Gliederung erfolgen. Zum Beispiel werden nach ULRICH und EPPINGER Produktentstehungsprozesse in sechs Phasen (vgl. Abbildung 2-5) aufgeteilt.

Die Konzeptphase (auch: Konzipierungsphase) wird im Allgemeinen als Entwicklungsprozess bezeichnet, jedoch ist die Tätigkeiten in dieser Phase in jedem Entwicklungsprozess sehr unterschiedlich. Die Konzeptphase betrifft z. B. bei der Feinwerktechnik die Aufgabenklärung. Die Konzeptphase bei ULRICH und EPPINGER (Concept-Development-Phase) beschäftigt sich nicht mit den gleichen Problemschwerpunkten bei der VDI-Richtlinie 2221 (Abbildung 2-4), da der Entwicklungsprozess von ULRICH und EPPINGER eine Phase spezifisch für die Produktarchitekturanalyse enthält, nämlich die „System-Level-Design-Phase“<sup>23</sup> (siehe Phase 2 im Abbildung 2-5), die in der VDI-Richtlinie 2221 allerdings in der Konzeptphase integriert ist.

---

<sup>21</sup> Ulrich und Eppinger 2008, S. 15

<sup>22</sup> Vgl. VDI-Richtlinie 2221 1993, S. 26

<sup>23</sup> Vgl. Ulrich/Eppinger 2008, S. 15, „System-level design phase“ includes the definition of the product architecture and the decomposition of the product into subsystems and components. The final assembly scheme for the production system is usually defined during this phase as well. The output of this phase usually includes a geometric layout of the product, a function specification of each of product's subsystems, and a preliminary process flow diagram for the final assembly process.

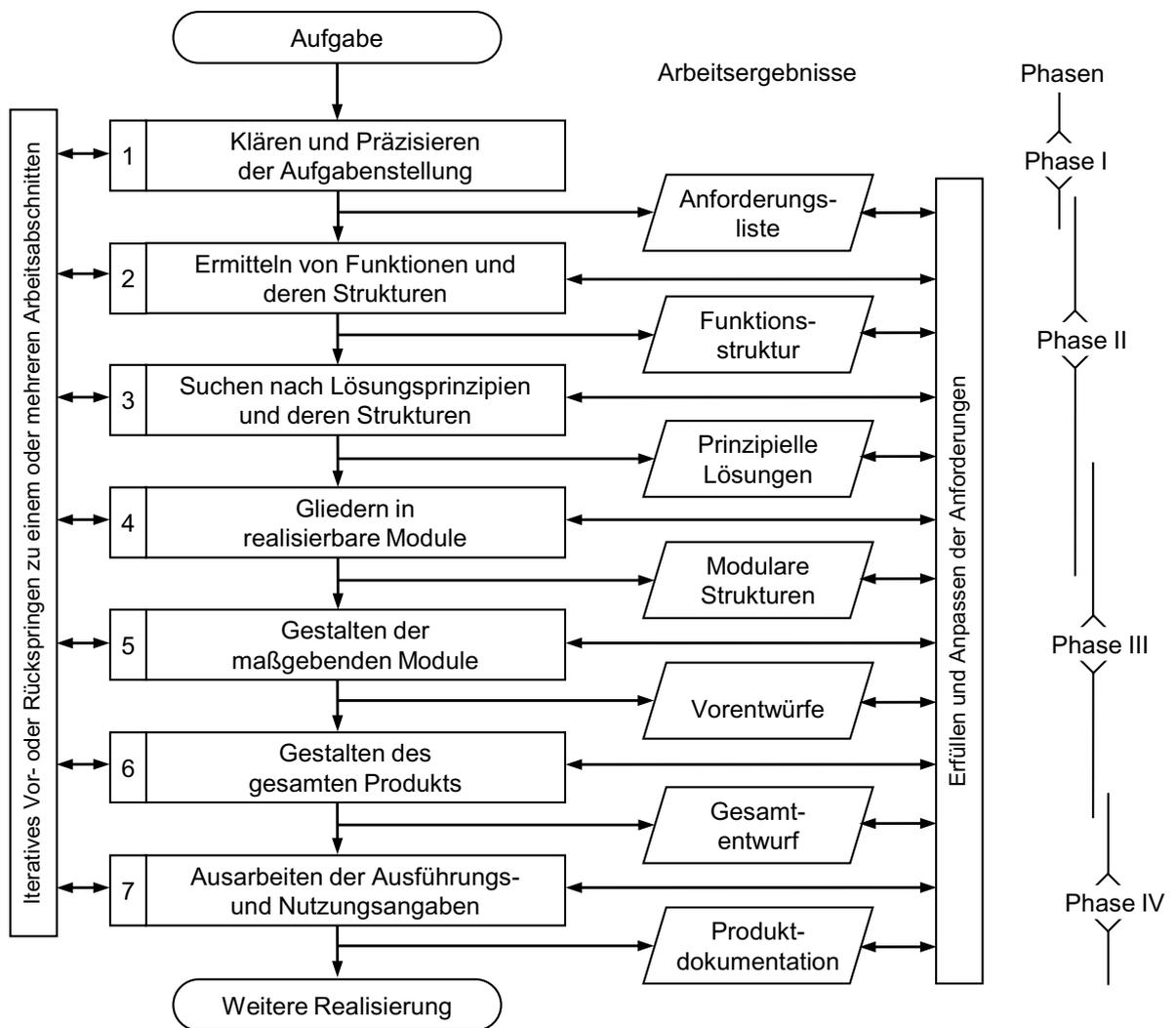
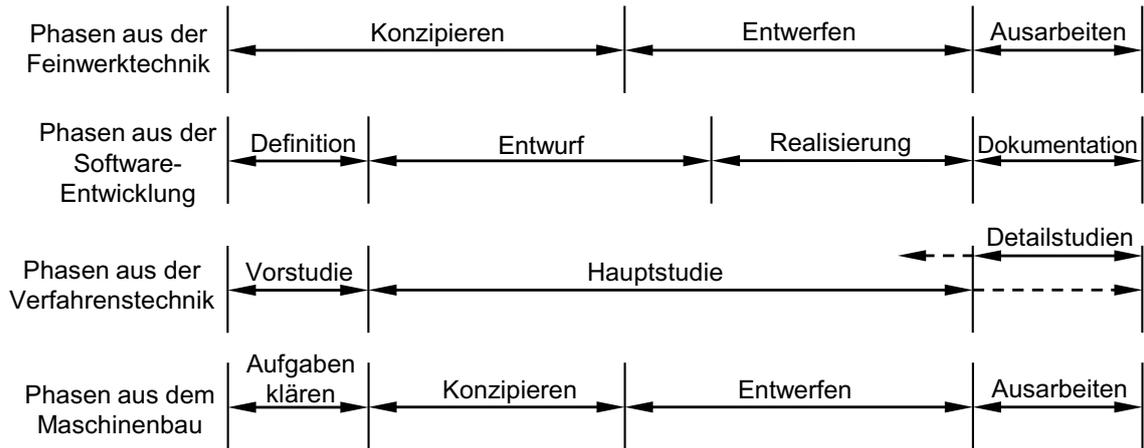


Abbildung 2-4. Unterschiedliche Phasen aus verschiedenen Branchen nach VDI-Richtlinie 2221<sup>24</sup>

<sup>24</sup> Vgl. VDI-Richtlinie 2221 1993, S. 9, 26, 28, 31

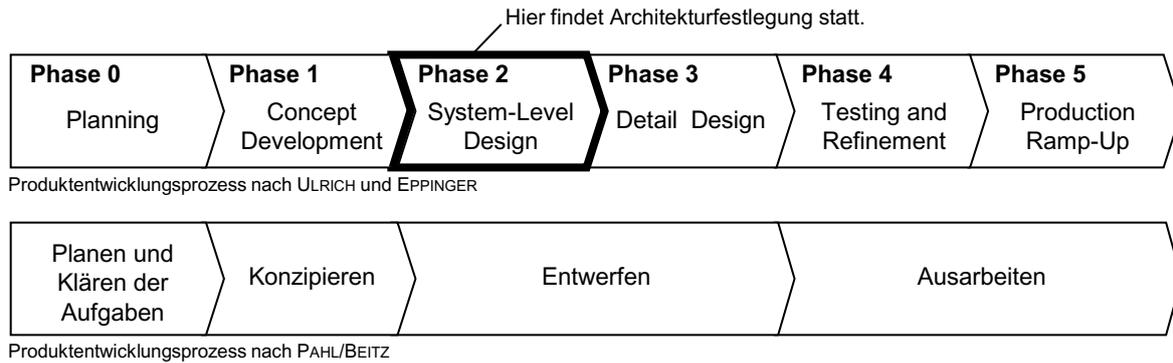


Abbildung 2-5. Vergleich zwischen den zwei Phasenaufteilungen klassischer Produktentstehungsprozessen (oben nach ULRICH und EPPINGER<sup>25</sup>, unten nach PAHL ET AL.<sup>26</sup>)

Im Maschinenbau ist die Trennung zwischen den Phasen ziemlich vage. PAHL ET AL. argumentieren, dass eine sehr scharfe Trennung der Hauptphasen nicht immer möglich ist, weil z. B. im Vorgriff Gestaltungsuntersuchungen bereits beim Konzipieren nötig sind oder beim Entwerfen bereits sehr detaillierte, fertigungstechnische Festlegungen getroffen werden müssen.<sup>27</sup>

Außerdem kann der Begriff „Phase“ nicht klar trennen, welche Tätigkeiten bei jeder Phase aktiv sind. Zum Beispiel beschäftigt sich die Suche nach einer Prinzipskizze in der Konzipierungsphase unvermeidbar schon mit dem Gestalten in der Entwurfsphase und der Validierungsmöglichkeit in der Ausarbeitungsphase, jedoch mit niedrigem Detaillierungsgrad.

ALBERS<sup>28</sup> argumentiert, dass diese grobe Phase im Entwicklungsprozess ein managementorientiertes Element ist. Dieses Element ist besser geeignet für die Planung als für die Konstruktionsmethodik. Er schlägt daher vor, den Phasenbegriff aus der Entwicklungsebene aufzugeben und er definiert den Produktentstehungsprozess als Abfolge vernetzter Aktivitäten. Die genauen Aktivitäten, die für vorliegende Arbeit relevant sind, werden durch das integrierte Produktentstehungsmodell (iPeM) im Kapitel 2.1 genauer dargestellt. Die Aktivitäten können nämlich im Gegensatz zu vagen Phasen auch parallel ablaufen.

### Frühe Phase der Produktentwicklung

Die „frühe Phase der Produktentwicklung“ (im Englischen: „early phase of product development“) wird häufig in verschiedenen Kontexten bei Produktentstehungsprozessen sowohl in der Forschung als auch in der Praxis

<sup>25</sup> Vgl. Ulrich/Eppinger 2008, S. 22

<sup>26</sup> Vgl. Pahl et al. 2003, S. 170

<sup>27</sup> Vgl. Pahl et al. 2003, S. 169

<sup>28</sup> Vgl. Albers/Meboldt 2007, S. 8

benutzt. Jedoch ist ihr Verständnis sehr unterschiedlich. Schon der Begriff „Phase“ besitzt keine scharfe Grenze und dafür, wann die „frühe“ Phase abgeschlossen ist, gibt es bislang keine einheitliche Definition.

Die „frühe Phase“ der Front-End-Ansätze,<sup>29</sup> die sehr stark von betriebswirtschaftlichen und marketingorientierten Aspekten getrieben ist, konzentriert sich auf die Produktstrategie und das Produktportfolio. In dieser Phase werden darüber hinaus die Produktideen erzeugt, Produktkonzepte erarbeitet, ausgewählt und verfeinert.

Außerdem ist der Begriff „frühe Phase“ sehr unscharf und sehr stark vom Produkt abhängig. In der Automobilindustrie versteht sich der Begriff „frühe Phase“ im Sinne einer Vorentwicklungsphase.<sup>30</sup> Dort wird der Entwicklungsprozess traditionell in drei Phasen – nämlich Forschung, Vorentwicklungsphase und Serienentwicklungsphase – getrennt. In der Vorentwicklungsphase wird ein gesamter Produktentstehungsprozess durchgeführt – von der Projektierungsaktivität (z. B. Erstellung des Lastenheftes) über die Projektablaufplanung bis hin zur Produktionsaktivität (z. B. Analyse der Fertigungsprozesse, Herstellung von Prototypen). Einige Forschungsarbeiten wie z. B. die von DONG und WHITNEY<sup>31</sup> sowie STRAWBRIDGE ET AL.<sup>32</sup> benutzen den Begriff „Frühe Phase“ (Engl.: „early stage“) lediglich konkret in Bezug auf die Aktivitäten der Ideenfindung und Modellierung von Prinzip und Gestalt des Entwicklungsprojektes, ohne die Produktstrategie oder das Produktportfolio einzuschließen. Außerdem gibt es viele Beiträge, die den Begriff der „frühen Phase“ verwenden, ohne eine klare Erklärung zu geben.

Die vorliegende Arbeit beschäftigt sich konkret mit dem Zeitraum, an dem die Prinzipskizze<sup>33</sup> erstellt wird. Dieser Zeitraum liegt oft in der Projektphase die auch als „Frühe Phase“ bezeichnet wird; er ist aber nicht gleichbedeutend mit ihr.

Um Missverständnisse wegen den unterschiedlich unscharfen Grenzen der Phasen und verschiedenen Definitionen der „Frühen Phase“ zu vermeiden, wird in vorliegender Untersuchung absichtlich nicht auf die „Phase“ referenziert.

---

<sup>29</sup> Kahn et al. 2003, S. 193

<sup>30</sup> Vgl. Braess und Seiffert 2007, S. 734

<sup>31</sup> Vgl. Dong und Whitney 2001, S. 1

<sup>32</sup> Vgl. Strawbridge et al. 2002, S. 2

<sup>33</sup> Siehe Definition Prinzipskizze in Abschnitt 5.1.1

## 2.3 Produktarchitektur

### 2.3.1 Begriffserklärungen

Im Allgemeinen beschreibt die „Produktarchitektur“ die Art und Weise, wie Bauteile oder Baugruppen zusammengebaut werden. Sie wird üblicherweise durch eine hierarchische Komponentenstruktur oder Produktstruktur dargestellt. SCHUH und SCHWENK<sup>34</sup> definieren die Produktstruktur als eine strukturierte Zusammensetzung des Produktes aus seinen Komponenten. Baugruppen und Einzelteile führen dabei zu Strukturstufen, indem sie Komponenten in tieferer Ebene in der Produktstruktur zusammenfassen.

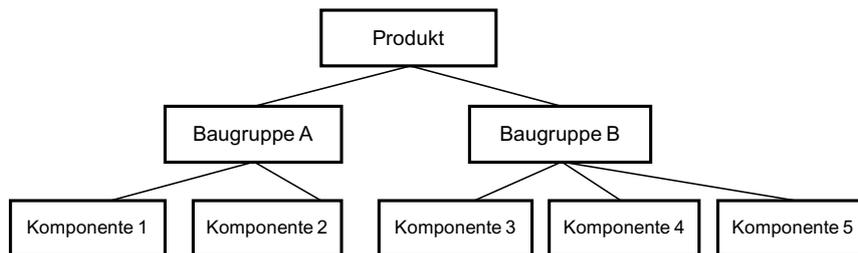


Abbildung 2-6. Produktstruktur: Hierarchische Komponentenstruktur

Der Begriff lässt sich im Systems Engineering etwas anders formulieren. Das Verständnis des Begriffs „Produktarchitektur“ wird mit Produktstruktur und Funktionsstruktur verbunden. DIETER<sup>35</sup> beschreibt die Produktarchitektur als eine Anordnung der physischen Elemente zum Erzielen einer Funktion. ULRICH<sup>36</sup> definiert sie als „ein Schema der Beziehungen zwischen der Funktionsstruktur eines Produktes und seiner physischen Baustruktur“.

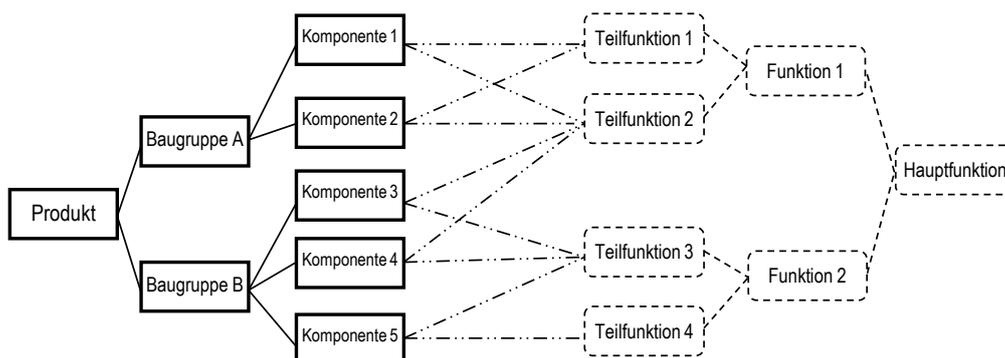


Abbildung 2-7. Beispiel der Produktstruktur und Funktionsstruktur und ihre Verknüpfungen

<sup>34</sup> Vgl. Schuh/Schwenk 2001, S. 73

<sup>35</sup> Vgl. Dieter 2000, S. 213

<sup>36</sup> Vgl. Ulrich 1995, S. 420

In diesem Bereich interessiert man sich nicht nur für die physischen Beziehungen der Produktkomponenten, sondern auch für die Funktionen und den Zusammenhang zwischen Funktionen und Produktkomponenten. Dies zeigt eine andere Perspektive der Produktarchitektur. Es ist nicht immer so, dass ein Produkt, das in mehreren Baugruppen gestaltet wird, in einer modularen Produktarchitektur erfasst wird. Es hängt auch davon ab, welche Produktfunktionen von welchen Komponenten erfüllt werden. Die konkreten Informationen werden im nächsten Abschnitt beschrieben.

Produktstruktur und Produktbauweise sind ähnliche Begriffe und werden manchmal im Sinne von Produktarchitektur verwendet, zum Beispiel in der Arbeit von RAPP<sup>37</sup>. Der Begriff „Bauweise“ kann je nach Fachdisziplin unterschiedliche Bedeutungen besitzen. Man kann ihn als eine Funktionsbauweise, eine reine hierarchische Struktur oder eine Bauform verstehen. Ganz anders ist dieser Begriff in der Werkstofftechnik zu benutzen, und zwar zur Beschreibung der Form und Eigenschaften des Materials wie zum Beispiel der Sandwichbauweise, Passiven Bauweise, Aktiven Bauweise.<sup>38</sup>

### 2.3.2 Typen von Produktarchitekturen

Wie schon im letzten Abschnitt beschrieben, werden unter der Produktarchitektur im Allgemeinen zwei Aspekte verstanden:

1. Aspekt: Hierarchische Komponentenstruktur,
2. Aspekt: Hierarchische Komponentenstruktur mit Berücksichtigung von Funktionen.

Zum Beispiel wird beim ersten Aspekt die Produktarchitektur allein durch die Anordnung oder Gliederung der Komponenten im Produkt entschieden. Die Architektur von komplexen Systemen wie Maschinen, Geräten und Apparaten wird häufig durch eine physikalische Hierarchie gegliedert. Zwei Haupttypen von Produktarchitekturen werden unterschieden:

- modulare Architekturen
- integrale Architekturen.

Falls ein Produkt aus mehreren diskreten Baugruppen besteht, wird die Architektur als modulare Architektur bezeichnet.

---

<sup>37</sup> Vgl. Rapp 1999, S. 9

<sup>38</sup> Vgl. Flemming et al. 1996, S. 211

Statt „Baugruppe“ wird in der Praxis bei den modularen Produktarchitekturen der Begriff „Modul“<sup>39</sup> verwendet.

Unter dem zweiten Aspekt lässt sich die Produktarchitektur ebenfalls in die modularen und integralen Produktarchitekturen aufteilen, wobei dazu die Funktionsgliederung betrachtet wird. Die vom ersten Aspekt Produktarchitektur kann möglicherweise in diesem Aspekt wegen den zugefügten Funktionsinformationen anders beurteilt werden. Zum Beispiel wird, wenn man nur die Komponenten in der Abbildung 2-7 betrachtet, das Produkt in eine modulare Produktarchitektur mit zwei unabhängigen Modulen (A und B) aufgebaut. Aber wenn die Funktionsgliederung verknüpft wird, sind die beiden Module nicht mehr unabhängig voneinander, weil die Teilfunktion 2 beide Module miteinander verbindet.

Falls die Funktionstrennung genauso wichtig wie eine Trennung physikalischer Elemente ist, kann man eine solche Architektur nicht völlig zu einer modularen Produktarchitektur zählen. Das Beispiel vom ULRICH<sup>40</sup> zeigt zwei Autoanhänger gleicher Funktion, aber völlig unterschiedlicher Produktarchitektur. Durch die Verknüpfungen zwischen Komponenten und Funktionenelementen wird eine funktionale Abhängigkeit der Komponenten dargestellt. Je mehr Eins-zu-eins-Beziehungen das Produkt hat, desto modularer ist es. Die Abbildung 2-8 zeigt die modulare Produktarchitektur (oben) und integrale Produktarchitektur (unten) der zwei Autoanhänger.

Außerdem spielen die Verbindungseigenschaften der Elemente besonders bei der modularen Produktarchitektur eine große Rolle. Es muss die Möglichkeit bestehen, die Elemente in der modularen Produktarchitektur auszubauen, ohne dass dies einen großen Einfluss auf andere Bauteile hat.

---

<sup>39</sup> Vgl. VDI-Richtlinie 2206 2004, S. 18: Modularisierung bedeutet die Bildung einer Produktstruktur mit Modulen, bei der die Beziehungen zwischen den Modulen geringer ausgeprägt sind als die Beziehungen innerhalb der Module. Damit lassen sich Wechselwirkungen zwischen Modulen auf ein Minimum reduzieren. Zwischen den Modulen sind Schnittstellen zu bilden, um die Kompatibilität auch bei unterschiedlicher Innovationsdynamik sicherzustellen, z. B. Hardwarekomponenten unterschiedlicher Produktgenerationen, Softwareupdate bei unveränderter Hardware etc.

<sup>40</sup> Ulrich 1995, S. 421

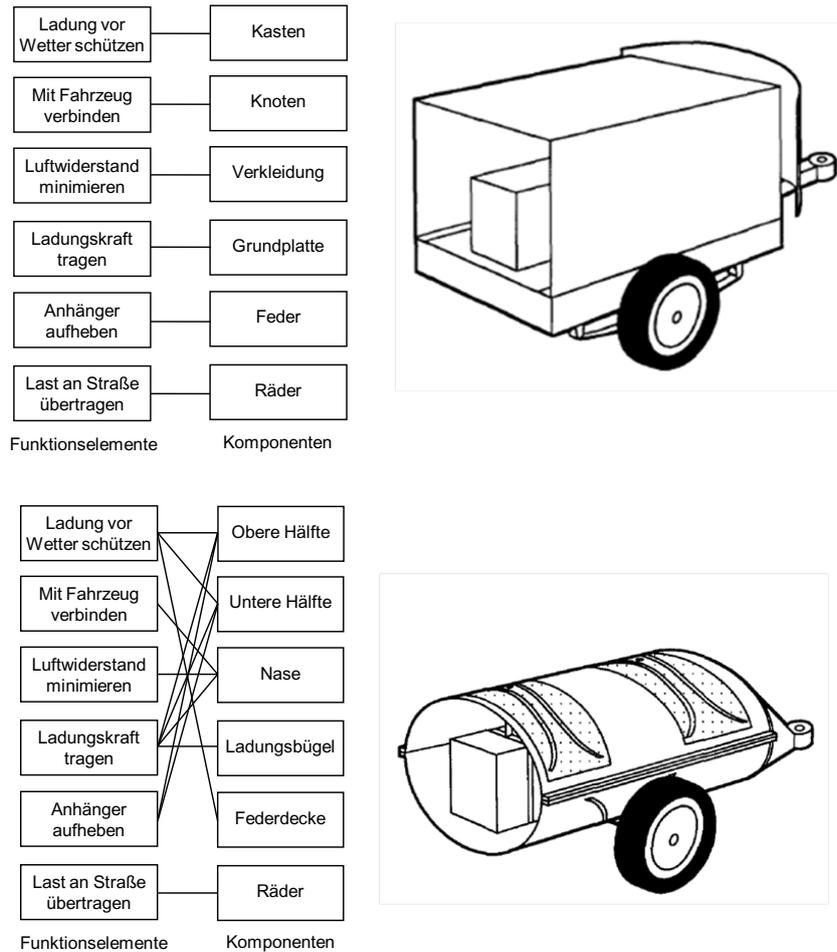


Abbildung 2-8. Mapping zwischen Funktionen und Bauteilen eines Autoanhängers nach ULRICH<sup>41</sup>

In ähnlicher Weise stellt GÖPFERT<sup>42</sup> die Klassifizierung der Produktarchitektur durch die funktionale und physische Unabhängigkeit dar. Funktionale Unabhängigkeit bedeutet, dass sie genau eine Teilfunktion erfüllt, und physische Unabhängigkeit bedeutet die Trennbarkeit von Komponenten. Vier verschiedene Typen, nämlich:

- die modulare Produktarchitektur,
- die integrale Produktarchitektur,
- die funktional-modulare Produktarchitektur und
- die physisch-modulare Produktarchitektur,

lassen sich durch die funktionale und physische Unabhängigkeit beurteilen (vgl. Abbildung 2-9). Die modulare Produktarchitektur besteht aus Komponenten, die sowohl funktional als auch physisch relativ unabhängige, abgeschlossene Einheiten

<sup>41</sup> Vgl. Ulrich 1995, S. 421 f.

<sup>42</sup> Vgl. Göpfert 1999, S. 116

bilden. Im Gegensatz dazu weisen die Komponenten der integralen Produktarchitektur sehr starke funktionale und physische Abhängigkeiten auf. Die funktionale-modulare Produktarchitektur hat zwar funktional unabhängige Komponenten, jedoch gibt es keine oder nur eine geringe physische Trennung. Die physisch-modulare Produktarchitektur weist physisch voneinander unabhängige Bereiche auf, hat aber sehr viele funktionale Abhängigkeiten zwischen den Komponenten.

Die Modularität wird auch in der Arbeit von GÖPFERT definiert. Sie wird durch die funktionale und physische Unabhängigkeit beurteilt. Je mehr ein Produkt beide Unabhängigkeiten aufweist, desto höher ist seine Modularität (siehe Abbildung 2-9).

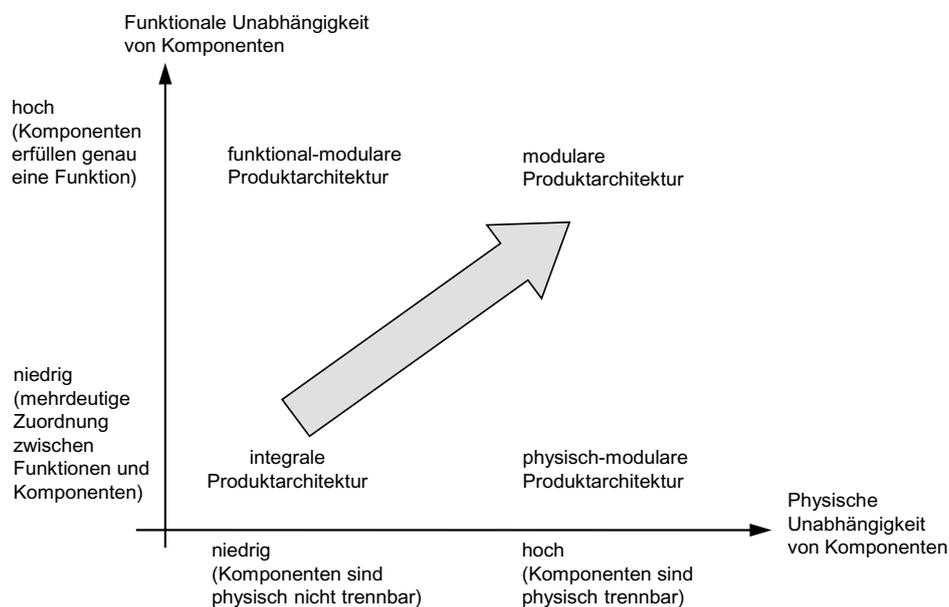


Abbildung 2-9. Klassifizierung von Produktarchitektur anhand der funktionalen und physischen Unabhängigkeit ihrer Komponenten nach GÖPFERT<sup>43</sup>

Die vier wesentlichen Faktoren zur Unterscheidung der Produktarchitektur lassen sich damit wie folgt zusammenfassen:

- Gliederung der physikalischen Komponenten,
- Gliederung der funktionalen Elemente,
- Beziehungen zwischen funktionalen Elementen und physikalischen Komponenten,
- Verbindungen zwischen den physikalischen Komponenten

<sup>43</sup> Göpfert 1999, S. 115

Beispiele für die modulare Produktarchitektur sind ein Desktop-PC, eine Spiegelreflexkamera, eine Audioanlage aus Hi-Fi-Bausteinen oder ein PKW. Beispiele für die integrale Produktarchitektur hingegen sind ein Notebook, eine Kompaktkamera, ein Walkman und ein Rennwagen.

In den meisten Fällen ist eine Produktarchitektur nicht vollständig modular oder integral. Die Entscheidung fällt manchmal schwer oder es ist unklar, zu welchem Architekturtyp die Produktarchitektur gehören soll. Aus diesem Grund wurden Werkzeuge zur Bewertung, wie modular oder integral die Produktarchitektur ist (Grad der Modularität), entwickelt. Zu diesen Werkzeugen gehören zum Beispiel die Design-Structure-Matrix (DSM)<sup>44</sup> oder die Module-Indication-Matrix (MIM) des Modular-Function-Deployment (MFD)<sup>45</sup>. Die DSM ist ein bekanntes Werkzeug und es wird auch in dieser vorliegenden Forschungsarbeit verwendet. Nähere Informationen zur DSM werden in Kapitel 2.6 gegeben. Die Methode des MFD wurde von ERIXON<sup>46</sup> an der Königlichen Technischen Hochschule (KTH) in Stockholm entwickelt. Bei dieser Methode handelt sich um eine systematische Vorgehensweise zur Entwicklung modularer Produktarchitekturen, die auf dem Prinzip der „Modultreiber“ aufgebaut ist. Diese Methode lässt sich in fünf Schritte gliedern: (1) Klärung der Kundenanforderungen, (2) Gliederung der Funktionen sowie Funktionsträger des Produktes, (3) Identifizieren potenzieller Module mithilfe der MIM, (4) Bewertung der MIM und (5) Optimierung der einzelnen Module. Zuerst müssen die Kunden- bzw. Marktanforderungen in der Produktarchitektur im Rahmen der Produktentstehung berücksichtigt werden. ERIXON verwendet hierfür die Methode des Quality-Function-Deployment (QFD) zur systematischen Umsetzung von Kundenanforderungen in technische Merkmale. Er integriert die Modularität als Produktmerkmal in das QFD und erhält dadurch einen ersten Überblick, ob diese dazu beiträgt, die Kundenwünsche zu erfüllen.<sup>47</sup> Die Ergebnisse des QFD werden in Funktionen abgebildet und so weit detailliert, dass den Teilfunktionen alternative technische Lösungen zugewiesen werden können. Zu diesem Zeitpunkt werden die gegliederten Funktionen und dazugehörigen Funktionsträger noch nicht zusammen gruppiert. Der dritte Schritt ist der Kern des MFD. ERIXON entwickelt generische Modultreiber, die für eine Realisierung in einem separaten Modul sprechen und durch unternehmensspezifische Modultreiber ergänzt werden können.<sup>48</sup> Die zwölf Modultreiber, die sich am Produktlebenszyklus orientieren sind:

---

<sup>44</sup> Vgl. Browning 2001, S. 292

<sup>45</sup> Erixon 1998

<sup>46</sup> Erixon 1998

<sup>47</sup> Vgl. Erixon 1998, S. 66; Vgl. Mayer 2007, S. 127

<sup>48</sup> Mayer 2007, S. 127

Mehrfachverwendung, technologische Dynamik, Produktplanung, technische Spezifikationen, Styling, Vereinheitlichung, spezifische Prozesse, unabhängige Prüfung, separate Liefereinheit, Service und Wartung, Aufrüstbarkeit und Recycling. Die Funktionsträger werden den Modultreibern in der MIM gegenübergestellt (siehe Abbildung 2-10) und bezüglich ihrer Modultauglichkeit bewertet, mit folgender Bewertungsskala: „9“ für sehr starke Treiber, „3“ für mittelstarke Treiber und „1“ für schwache Treiber. Die Summe der Modultreiber jedes Funktionsträgers zeigt, wie geeignet jeder Funktionsträger ist, für die Gestaltung als eigenständiges Modul. Hierbei können die Funktionsträger mit als „schwach“ bewerteten Modultreibern in einem gemeinsamen Modul gruppiert werden. Dieser Schritt der Optimierung der Gruppierung von Funktionsträgern muss intuitiv erfolgen. Daher werden für die Anwendung erfahrene Konstrukteure benötigt, weil z. B. räumliche Randbedingungen und die Bedeutungen der Funktionsinteraktionen beachtet werden müssen. Es ist anzumerken, dass das MFD bzw. die MIM sehr wenig auf die Beziehungen zwischen den Systemelementen eingehen, die wesentlich für die in dieser Arbeit entwickelte Methode sind. Daher werden MFD bzw. MIM hier nicht als Unterstützungswerkzeuge eingesetzt.

Function carrier		Module driver																									
		Fan	Noise absorbent, fan	Electric motor	Damper	Noise absorbent, motor	Chassis	Bag	Filter	Triistor+knob	Switch+knob	Housing	Wire+contact	Grip	Rear wheel	Front wheel	Accessories	Bumper	Cover	Indicator	Seal, cover	O-ring	Wire collector	Bag lock	Brake+knob		
Design and Development	Carry-over	●	●						●	○	○	○	○							●			●	○	○		
	Technology push						●	●																			
	Product Planning																										
Variance	Diff. specification	○	○	○					○	○	○	○	○														
	Styling								●	●	●	●	●	○					●						●		
Manuf.	Common unit	○	○	○	●	●	●	●	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○		
	Process/Org.	●	●				●	●				●															
Quality	Separate testing		●							○																	
Purchase	Black-box engineer.							●	●		●																
After sales	Service/maint.		○					○	○	○																	
	Upgrading							●																			
	Recycling		●			●					●												○				
● = 9 ○ = 3 ○ = 1		Weight of Driver vertically summarised		22	4	43	9	9	27	27	32	34	18	27	16	9	4	18	10	9	9	18	9	9	19	9	15
		Module candidates		√	√			√	√	√	√	√	√											√			

Abbildung 2-10. Module-Indication-Matrix (MIM) für einen Staubsauger nach ERIXON<sup>49</sup>

<sup>49</sup> Erixon 1998

### 2.3.3 Modularitätsgrad

Wie im Kapitel 2.3.2 beschrieben, lässt sich eine Produktarchitektur in zwei Typen der Produktarchitektur gestalten, nämlich die modulare Produktarchitektur und die integrale Produktarchitektur. In der Realität gibt es kaum eine Produktarchitektur, die ausschließlich integral oder modular ist.<sup>50</sup>

Die Modularität zeigt, inwieweit eine Produktarchitektur in modulare Produktarchitektur gestaltet wird. Der quantitative Wert der Modularität wird als Modularitätsgrad bezeichnet. Die Auswertung einer Modularität erfolgt auf verschiedene Weise, zum Beispiel über den der Singular-Value-Modularity-Index (SMI)<sup>51</sup> oder durch das Funktion/Komponenten Verhältnis<sup>52</sup>. In dieser Arbeit wird lediglich die Auswertung mit der Dependency-Matrix betrachtet, die auf der Arbeit von GUO und GERSHENSON<sup>53</sup> basiert, da diese Matrix die Interaktionen und Systemelemente, die die Kernsystemelemente dieser zu entwickelten Methode sind, anschaulich darstellen kann.

Die Matrixelemente, die innerhalb der Module bezeichnet werden, werden als positiv bewertet, und die, die außerhalb der Module liegen, sind dann negativ. In vorliegender Arbeit wird für die Bewertung zur Gruppierung der Matrixelemente oder die Zielfunktion ( $Z$ ) das Bewertungskonzept von GUO und GERSHENSON leicht modifiziert. Die Zielfunktion in dieser Arbeit lässt sich dann folgendermaßen formulieren:

$$Z = \frac{\sum_{k=1}^{N_m} \frac{\sum_{i=n_k}^{m_k} \sum_{j=n_k}^{m_k} B_{ij}}{(m_k - n_k + 1)^2}}{N_m} - 2 \times \frac{\sum_{k=1}^{N_m-1} \sum_{p=k+1}^{N_m} \frac{\sum_{i=n_k}^{m_k} \sum_{j=n_p}^{m_p} B_{ij}}{(m_p - n_p + 1)(m_k - n_k + 1)}}{N_m} \quad (2.1)$$

wobei  $n_k, n_p$  der Index des ersten Matrixelements im k-ten und p-ten Modul,  
 $m_k, m_p$  der Index des letzten Matrixelements im k-ten und p-ten Modul,  
 $N_m$  die gesamte Anzahl der Module im System,  
 $B_{ij}$  die Bewertung der Beziehungen jedes Paares der Matrixelemente und  
 $i, j$  Zeile und Spalte in der Matrix sind.

<sup>50</sup> Vgl. Ulrich 1995, S. 424

<sup>51</sup> Hölttä et al. 2005, S. 5

<sup>52</sup> Ulrich 1995, S. 432

<sup>53</sup> Guo/Gershenson 2004, S. 264

	A	B	C	D	E	F	G	H
Elemente A		1						1
Elemente B	1						1	
Elemente C					1			1
Elemente D							1	
Elemente E			1			1		
Elemente F					1		1	
Elemente G		1		1		1		1
Elemente H	1		1				1	

Abbildung 2-11. Beispiel einer Gruppierung der Dependency-Matrix

In Abbildung 2-11 wird die Architektur in drei Modulen verteilt, die Elemente A, B und C gehören zum 1. Modul. Die Elemente D, E und F zählen zum zweiten Modul und G und H zum dritten Modul. Die schattierten Felder in der Matrix stellen die Interaktionen innerhalb der Module dar. Der Zielfunktionswert ist in diesem Fall gleich

$$Z = \frac{\left[ \frac{2}{9} + \frac{2}{9} + \frac{2}{4} \right] - 2 \times \left[ \frac{1}{9} + \frac{3}{6} + \frac{2}{6} \right]}{3}$$

für diese Anordnung. Diese Berechnung der Zielfunktion wird in dieser vorliegenden Forschung angewendet.

### 2.3.4 Vorteile der Produktarchitekturtypen

In Bezug auf die zwei Produktarchitekturtypen kann man nicht genau sagen, welcher Produktarchitekturtyp besser oder schlechter ist, da jeder Vorteile für die gegebenen Randbedingungen bietet. Die Frage ist, welches Produkt unter den gegebenen Randbedingungen und Zielen die Erfordernisse besser erfüllt, und wie hoch die Modularität des Produkts für eine optimale Produktarchitektur sein soll. Diese Randbedingungen und das Zielsystem betreffen sowohl die Produkthanforderungen, zum Beispiel die Frage, wie leicht das Produkt werden soll, als auch die Prozessbedingungen durch den Produktlebenszyklus (beispielsweise optimale Fertigung). Außerdem können auch organisatorische Aspekte für die optimale Aufgabenverteilung eine wichtige Rolle spielen. In diesem Abschnitt werden daher die allgemeinen Vorteile der Architekturtypen angeführt.

Generell besitzt die modulare Produktarchitektur Vorteile gegenüber der integralen Produktarchitektur – ohne die Betrachtung der Stückzahl:

- Verkürzung der Produktentwicklungszeit durch parallele Entwicklung,
- Verkürzung der Produktentwicklungszeit durch Plattformstrategie,
- Erleichterung der Anpassung an Kundenwünsche und Upgrades aufgrund der modularen standardisierten Bauteile,

- Vereinfachung der Wartung durch Standardbauteile,
- Vereinfachung der Wiederverwendung der Komponenten,
- Vereinfachung zur Beschaffung von Produktvarianten,
- Vereinfachung der Aufgabenverteilung
- Einfachere und billigere Herstellung.

Die integrale Produktarchitektur bietet hingegen Vorteile für:

- kleinere sowie,
- leichtere Produkte,
- höhere Leistungsdichte,
- besseren Wirkungsgrad und
- verhindert Produktpiraterie.

Um die generellen Vorteile beider Architekturen genauer darzustellen, soll der Vergleich zwischen einem Desktop-PC und einem Notebook herangezogen werden. Der PC besitzt Vorteile vor allem in Bezug auf parallele Entwicklung, Wartung, Upgrade und Kosten; das Notebook hingegen bringt mehr Leistung pro Gewicht, eine bessere Mobilität und eine Energieeinsparung.

### 2.3.5 Festlegung der Produktarchitektur

Die Festlegung der Produktarchitektur ist eine entscheidende Aufgabe in der Produktentwicklung.<sup>54</sup> ANDERSON<sup>55</sup> postuliert, dass die Bedeutung der Festlegung Produktarchitektur häufig unterschätzt wird, da oft angenommen wird, dass die Produktarchitektur der Architektur eines vorherigen Produkts oder der eines Konkurrenzprodukts gleichen soll. HUTHWAITE<sup>56</sup> argumentierte, dass wenn ein Produkt erst einmal auf dem Markt ist, die Produktarchitektur sehr schwer zu ändern ist, ohne dass dies wesentliche Auswirkungen auf die Fertigungs- und Wartungsfähigkeit hätte. ANDERSON<sup>57</sup> ist der Meinung, dass eine suboptimale Produktarchitektur häufig beginnt, mit dem scheinbar harmlosen Schritt zur Bildung eines Funktionsprototyps „Nur um zu schauen, ob es funktioniert.“. Funktionsprototypen werden normalerweise konstruiert, um die Realisierbarkeit zu überprüfen, und sie werden mit den verfügbaren Materialien auf dem

---

<sup>54</sup> Vgl. Ulrich/Eppinger 2008, S.167

<sup>55</sup> Vgl. Anderson 2004, S. 79

<sup>56</sup> Vgl. Huthwaite 2007, S. 179

<sup>57</sup> Vgl. Anderson 2004, S. 79

zweckdienlichsten Weg gebaut. Die Produktarchitektur wird häufig in diesem Zeitraum leider nicht beachtet, da angenommen wird, dass die Optimierung auch noch später durchgeführt werden kann. Aber wenn ein Prototyp funktionsfähig ist, gibt es eine große Versuchung, ihn schnell in die Produktion zu geben. Das Ergebnis ist, dass Firmen Produkte mit der Produktarchitektur der Prototypen beibehalten und sie so produzieren. So verpassen Firmen die große Chance, die Vorteile einer optimalen Architektur zu nutzen.<sup>58</sup>

LINDSTEDT und BURENIUS<sup>59</sup> beschreiben, dass die Produktarchitektur wie ein unsichtbarer Gegenstand ist, der das Projekt entweder von Problemen wegführt oder in die Probleme hineinführt. Eine schlechte Produktarchitektur kann eine Ursache für Probleme, Kosten und andere Schwierigkeiten sein. Nach LINDSTEDT und BUDENIUS hat die Produktarchitektur Einfluss auf die folgenden Aspekte:

- Kundennutzen
- Produktkomplexität
- Produktion, Vertrieb und Wartung
- Benötigte Kompetenzen, Methoden und Werkzeuge im Produktentstehungsprozess

Daher kommt der Festlegung der Produktarchitektur eine große Bedeutung zu.

## **2.4 Funktionen und Funktionsstrukturen**

### **2.4.1 Funktionsbegriff**

Der Begriff „Funktion“ wird in der Konstruktionswissenschaft von verschiedenen Autoren unterschiedlich beschrieben. In der Arbeit vom Matthiesen<sup>60</sup> werden wichtige Begriffsbeschreibungen der „Funktion“ gesammelt die in der folgenden Tabelle 2-1 aufgeführt sind. Die Funktionsbegriffe unterscheiden sich teilweise deutlich. Zum Beispiel bei der Formulierung: Nach EHRENSPIEL und BIRKHOFER soll die Formulierung von Funktionen lösungsneutral erfolgen, bei KOLLER ist diese Forderung nicht gegeben. Aufgrund der Vielfalt der Definitionen des Funktionsbegriffs entsteht häufig Unklarheit, welche Definition bei der Verwendung dieses Begriffs in einer bestimmten Situation oder einem bestimmten Kontext zugrunde gelegt wird.

---

<sup>58</sup> Vgl. Anderson 2004, S. 79

<sup>59</sup> Vgl. Lindstedt/Burenium 2003, S. 554

<sup>60</sup> Matthiesen 2002

Quelle	Begriffsbeschreibung
RODENACKER <sup>61</sup>	Allgemeiner Wirkzusammenhang oder die Abhängigkeiten zwischen den Eingangs-, Ausgangs- und Zustandsgrößen eines Systems.
KOLLER <sup>62</sup>	Tätigkeit oder Fähigkeit technischer Gebilde. Qualitative und/oder quantitative (gesetzmäßige) Beschreibung oder Tätigkeit eines technischen Gebildes; beispielsweise „Kraft (500 N) leiten“.
PAHL und BEITZ <sup>63</sup>	Allgemeiner und gewollte Zusammenhang zwischen Eingang und Ausgang eines Systems mit dem Ziel, eine Aufgabe zu erfüllen.
EHRLENSPIEL <sup>64</sup>	Eine Funktion im Sinne der Konstruktionsmethodik ist die lösungsneutrale Formulierung des gewollten (geplanten, bestimmungsgemäßen) Zwecks eines Produktes. Sie drückt die Zustandsänderung (Eigenschaftsänderung) eines Objektes (Umsatzprodukt) aus, die durch den Funktionsträger bewirkt wird.
BIRKHOFFER <sup>65</sup>	Eine Funktion ist eine lösungsneutrale Beschreibung als Operation des gewollten Zusammenhangs zwischen Ein- und Ausgangsgröße.
VDI RICHTLINIE 2221 <sup>66</sup>	Lösungsneutral beschriebene Beziehung zwischen Eingangs-, Ausgangs- und Zustandsgrößen eines Systems.

Tabelle 2-1. Der Begriff „Funktion“ in der Konstruktionswissenschaft (nach Matthiesen<sup>67</sup>)

Im weiteren Verlauf der vorliegenden Arbeit und in der darin vorgestellten Methode wird der Begriff „Funktion“ entsprechend der folgenden Definition verwendet:

Eine Funktion ist eine qualitative oder quantitative verbale Beschreibung der Eigenschaften, die ein technisches System für die Erfüllung eines Zwecks aufweisen muss.

Funktionen werden oft über die Beziehung zwischen Eingangs- und Ausgangsgrößen oder Zustandsänderung eines Systems ausgedrückt.<sup>68</sup> Die Funktionsformulierung kann mit unterschiedlichem Detailierungsgrad erfolgen. Sie hängt vom jeweiligen System (und vom Zweck der Funktionsformulierung/Modellerstellung) ab. Es kann ausreichend sein, Funktion

<sup>61</sup> Rodenacker 1991

<sup>62</sup> Koller 1998

<sup>63</sup> Pahl/Beitz 1993

<sup>64</sup> Ehrlenspiel 1995

<sup>65</sup> Birkhofer 2000

<sup>66</sup> VDI Richtlinie 2221 2223

<sup>67</sup> Nach Matthiesen 2002, S. 37

<sup>68</sup> Pahl/Beitz 1993; Birkhofer 2000; VDI Richtlinie 2221 2223

einfach in Form eines „Funktionsflusses“ zu formulieren, mit einem „Verb“ für jede Funktion – z. B. „Energie übertragen“, „Stoff speichern“, „Information wandeln“. Mehrere Beispiele für Funktionsformulierungen werden in Abschnitt 6 vorgestellt.

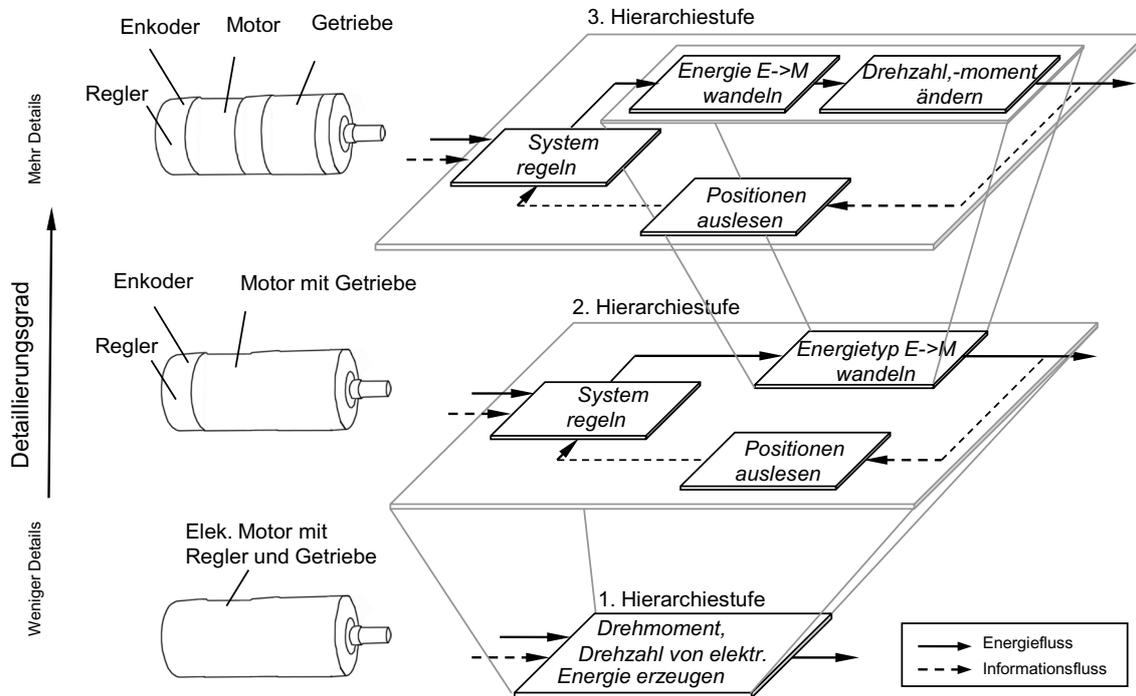


Abbildung 2-12. Die Funktionsstruktur zur Systembeschreibung in unterschiedlichen Detaillierungsgraden

Der Detaillierungsgrad einer Funktionsstruktur hängt mit der Tiefe der Funktionshierarchiestufen zusammen. Je tiefer die Funktionshierarchie betrachtet wird, desto mehr Details und eine höhere Anzahl an Funktionen in der Funktionsstruktur findet man. Dieser Detaillierungsgrad hängt davon ab, wie detailliert man die Funktionsstruktur des Systems kennt. Zum Beispiel hat die Funktionsstruktur bei der Ideenfindungsaktivität, wo das Zielsystem noch grob und abstrakt beschrieben ist, keinen hohen Detaillierungsgrad. Aus anderer Sicht hängt der Detaillierungsgrad auch davon ab, für wie genaue Details des Systems man sich interessiert oder wie tief man sich das System vorstellt. Das trifft besonders für die Funktionsstruktur zur Beschreibung eines bestehenden Systems zu. Zum Beispiel kann man die Funktionsbeschreibung eines Servomotors in verschiedenen Detaillierungsgraden darstellen (Abbildung 2-12). Zur Erstellung der Funktionsstruktur muss zunächst entschieden werden, welcher Detaillierungsgrad optimal dafür ist.

Zu vielen Zwecken, beispielsweise zur Aufgabenerklärung (durch Gesamtfunktion), Ideensuche (niedrige Details, abstrakte Formulierung) oder Realsystembeschreibung (viele Details, konkrete Formulierung), wird die Funktionsstruktur in unterschiedlichen

Abstraktionsgraden der Funktionsformulierung und Detaillierungsgraden der Funktionsstruktur bedient (Abbildung 2-12).

### 2.4.2 Funktionale Gliederung

In der VDI-Richtlinie 2803 werden die Funktionen, wie in Abbildung 2-13 beschrieben, gegliedert. Diese Gliederung und Bedeutung wird im Allgemeinen in der Produktentwicklung angewendet.

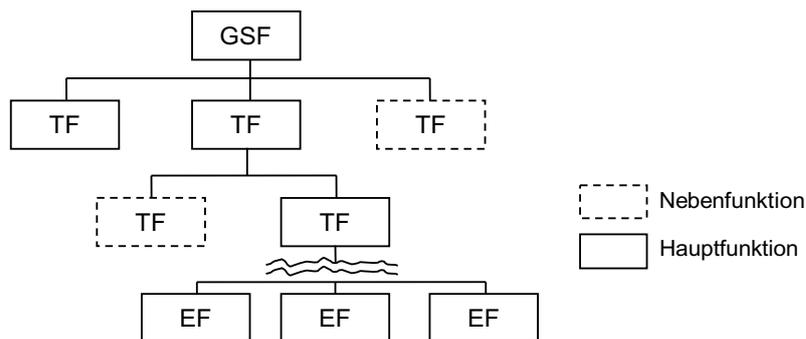


Abbildung 2-13. Gliederung der Funktion, modifiziert von der VDI-Richtlinie 2803<sup>69</sup>

Die Gesamtfunktion (GSF) ist eine Funktion, die die Aufgabe in ihrer Gesamtheit erfasst. Die Teilfunktion (TF) ist eine Funktion, die eine Teilaufgabe erfasst. Die Hauptfunktion ist eine Funktion, die unmittelbar der Gesamtfunktion dient. Die Nebenfunktion ist eine Teilfunktion, die die Hauptfunktion unterstützt und daher nur mittelbar der Gesamtfunktion dient (Einordnung je nach Betrachtungsebene unterschiedlich). Die Elementarfunktion (EF) ist eine Funktion, die sich nicht weiter gliedern lässt und allgemein anwendbar ist.

#### Funktionsarten bei der Baukastensystematik

Es gibt noch eine andere Funktionsgliederung, die in der Baukastensystematik eingesetzt wird.<sup>70</sup> Statt in Gesamt- und Teilfunktionen sowie Haupt- und Nebenfunktionen wird sie je nach Funktion der Bausteinarten (Grundbausteine, Hilfsbausteine, Sonderbausteine, Anpassungsbausteine und Nichtbausteine), die in der Baukastensystematik benutzt werden, aufgeteilt. Die Unterteilung in fünf unterschiedliche Teilfunktionen ist in der folgenden Abbildung dargestellt:

<sup>69</sup> Vgl. VDI-Richtlinie 2803 1996, S. 3

<sup>70</sup> Pahl et al. 2003, S. 635

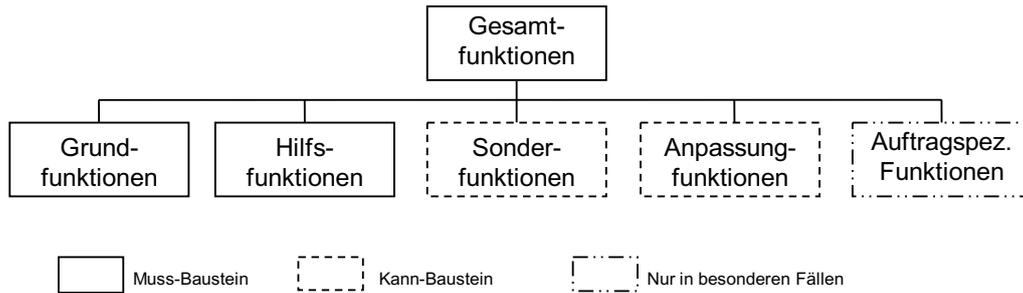


Abbildung 2-14. Funktionsarten bei Baukastensystemen, modifiziert von PAHL ET AL.<sup>71</sup>

*Grundfunktionen* (GS) sind in einem System grundlegend, immer wiederkehrend und unerlässlich. Sie sind grundsätzlich nicht variabel. Eine Grundfunktion kann zur Erfüllung von Gesamtfunktionsvarianten allein auftreten oder mit anderen Funktionen verknüpft werden (Muss-Baustein). *Hilfsfunktionen* (HF) sind verbindend und anschließend und werden durch Hilfsbausteine erfüllt, die sich im Allgemeinen als Verbindungs- und Anschlusselemente darstellen (Muss-Baustein). *Sonderfunktionen* sind besondere, ergänzende, aufgabenspezifische Teilfunktionen, die nicht in allen Gesamtfunktionsvarianten wiederkehren müssen (Kann-Baustein). *Anpassungsfunktionen* sind zum Anpassen an andere Systeme und Randbedingungen notwendig (Kann-Baustein). *Auftragsspezifische Funktionen* sind Funktionen, die nicht im Baukastensystem vorgesehen sind. Sie kommen trotz sorgfältiger Entwicklung eines Baukastensystems immer wieder vor.

Die Unterscheidung von verschiedenen Funktionstypen und die dazugehörigen Definitionen werden in der in dieser Arbeit vorgestellten Methode hauptsächlich aus der Baukastensystematik nach PAHL ET AL.<sup>72</sup> übernommen. In dieser Arbeit werden lediglich die Muss-Bausteine des Systems betrachtet. Die Grundfunktion selbst und die Interaktionen zwischen den Grundfunktionen sind notwendige Eingangsgrößen für die Integrationsanalyse. Die anderen Funktionstypen und ihre Verknüpfungen mit den Grundfunktionen werden beim jetzigen Stand der Forschung zur Produktarchitekturfestlegung noch nicht betrachtet.

## 2.5 Contact and Channel Model (C&C-M)

Am IPEK - Institut für Produktentwicklung Karlsruhe wird das Elementmodell C&C-M von ALBERS und Mitarbeiter<sup>73</sup> entwickelt. Das Modell hat das Ziel, sowohl die Analyse

<sup>71</sup> Vgl. Pahl et al. 2003, S. 636

<sup>72</sup> Vgl. Pahl et al. 2003, S. 636

<sup>73</sup> Albers/Matthiesen 2002

der Funktion technischer Systeme als auch die Synthese neuer Systeme bei vorgegebener Funktion zu unterstützen.<sup>74</sup>

Das C&C-M ist ein Modell, das nicht nur die Funktionsinformation, sondern zusätzlich auch die Gestaltinformation betrachtet.<sup>75</sup> Die Funktion beschreibt das Verhalten des Systems. Dagegen stellt die Gestalt die Geometrie des Systems im konkreten Betrachtungsniveau vor. Beide werden durch das C&C-M verknüpft. Durch diese Informationsverknüpfung können zwei unterschiedliche Informationen in einem Modell dargestellt werden. Dadurch können beide Produktinformationen – Gestalt- und Funktionsinformation bzw. ihre Beziehungen zueinander – auf einmal erfasst werden.

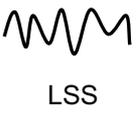
Grundelemente	Symbol / Abkürzung	Definition
Wirkflächen	 WF	Wirkflächen (WF) sind feste Oberflächen von Körpern oder generalisierte Grenzflächen von Flüssigkeiten, Gasen oder Feldern, die dauernd oder zeitweise im Kontakt zu einer weiteren Wirkfläche stehen und am Energie-, Stoff- und Informationsaustausch des technischen Systems beteiligt sind.
Wirkflächenpaar	 WFP	Wirkflächenpaare (WFP) werden aus genau zwei Wirkflächen gebildet, die zeitweise ganz oder teilweise, in Kontakt stehen und zwischen denen Energie, Stoff und Information übertragen wird.
Leitstützstruktur	 LSS	Unter der Leitstützstruktur (LSS) versteht man Volumina von Körpern, Flüssigkeiten, Gasen oder felddurchsetzten Räumen, die zwei Wirkflächenpaare verbinden und dauernd oder zeitweise eine Leitung von Energie, Stoff oder Information zwischen den Wirkflächenpaaren eines Körpers, einer Flüssigkeit, eines Gases oder eines Feldes ermöglichen.
Tragstruktur	TS	Tragstruktur ist die Menge aller möglichen Leitstützstruktur.
Reststruktur	RS	Reststruktur sind Volumina von Körpern, Flüssigkeiten, Gasen oder felderfüllte Räume, die nie Tragstruktur werden.
Gesamtstruktur	GS	Die Menge aller möglichen Leitstützstrukturen, als auch die Reststruktur

Tabelle 2-2. Symbole und Definitionen der Grundelemente des C&C-M nach ALBERS<sup>76</sup>

<sup>74</sup> Vgl. Albers/Matthiesen 2002, S. 55 ff.

<sup>75</sup> Vgl. Albers et al. 2005, S. 1

<sup>76</sup> Vgl. Albers/Matthiesen 2002, S. 58; Vgl. Enkler 2010

Im C&C-M werden Systeme hauptsächlich durch zwei Arten von Grundelementen, „Wirkflächenpaaren“ (WFP), die aus zwei Wirkflächen (WF) bestehen, und „Leitstützstrukturen“ (LSS) modelliert. Andere Grundelemente, wie z. B. die Reststruktur (RS) und die Begrenzungsfläche (BF), sind keine Funktionsträger und damit nicht notwendig für die funktionale Modellierung aber dennoch hilfreich bei der Systembetrachtung. Tabelle 2-2 gibt einen Überblick über die Definitionen von WF, WFP, LSS, TS, RS und GS.

Ein Begriff „Gesamtstruktur“ (GS) wird von ALBERS<sup>77</sup> ergänzt. Diese Ergänzung umfasst gemäß Gleichung (2.2) und (2.3) sowohl die Tragstruktur, die Menge aller möglichen Leitstützstrukturen, als auch die Reststruktur:

$$GS = TS + RS \quad (2.2)$$

$$GS = \sum_{t=t_1}^{t_n} LSS(t) + RS \quad (2.3)$$

wobei  $t$  die Zeit und  $n$  die Anzahl der LSS sind.

Normalerweise besteht ein System im C&C-M aus mehreren WFP und LSS (siehe folgende Hypothese). Die Anzahl der WFP und LSS hängt von der Systemgröße, Systemkomplexität und vom Systembetrachtungsniveau hinsichtlich Detaillierungsgrad und Abstraktionsgrad ab. Die Unterscheidung der Elemente erfolgt durch die Nummerierung der Grundelemente wie z. B. WFP<sub>1</sub>, LSS<sub>1</sub> und WF<sub>11</sub>, wie sie in Abbildung 2-16 zu sehen ist.

ALBERS<sup>78</sup> veröffentlichte drei Hypothesen zur Modellierung von Systemen mittels C&C-M. Die Modellierung basiert im Wesentlichen auf folgender Hypothese II.

#### **Hypothese:**

Die Funktion eines technischen Systems oder eines technischen Teilsystems wird grundsätzlich über mindestens zwei Wirkflächenpaare und sie verbindende Leitstützstrukturen verwirklicht. Ein System kann seine Funktion nur in Interaktion mit seiner Umgebung erfüllen. Ein System, welches keine Wirkflächenpaare mit seiner Umgebung bildet, erfüllt keine Funktion.<sup>79</sup>

<sup>77</sup> Vgl. Enkler 2010

<sup>78</sup> Vgl. Albers/Matthiesen 2002, S. 58

<sup>79</sup> Vgl. Albers 2010, S. 6

Diese Hypothese zeigt das Verständnis einer Funktion im C&C-M im Zusammenhang mit Wirkflächenpaaren und Leitstützstrukturen und wirkt als Haupthypothese für die Erstellung des C&C-M.

Folgende Darstellung ist ein Beispiel für ein C&C-M. Die Funktionsinformation (links) „Elektrische in mechanische Energie konvertieren“ beschreibt ein System mit abstraktem Betrachtungsniveau. Es fehlt jedoch die komplette Information zur Form oder Gestalt. Die Form oder Gestalt ohne Funktionsinformation des Systems in der Gestaltinformation wird dagegen rechts gezeigt.

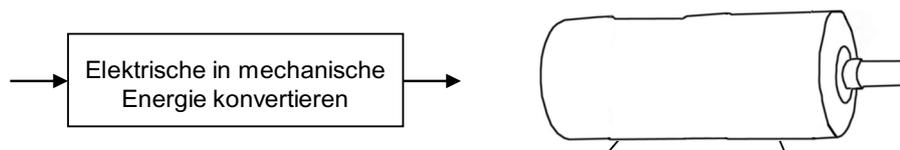


Abbildung 2-15. Beispiel der Funktionsinformation (links) und Gestaltinformation eines elektrischen Motors (rechts)

Zur Modellierung mittels C&C-M verwendet man die vorher gezeigte Hypothese. Zu Beginn der Modellierung werden die notwendigen LSS und WFP entsprechend der zu erfüllenden Funktion definiert. Danach werden die Orte aller LSS und WFP angegeben und verknüpft. Ihre Orte werden normalerweise auf einer Skizze oder einer anderen Repräsentation der Form des Systems angegeben, und damit wird die Struktur zur Funktionserfüllung, die Ein- und Ausgangsorte für Stoff, Energie, Information und Gestaltinformation miteinander verbunden.

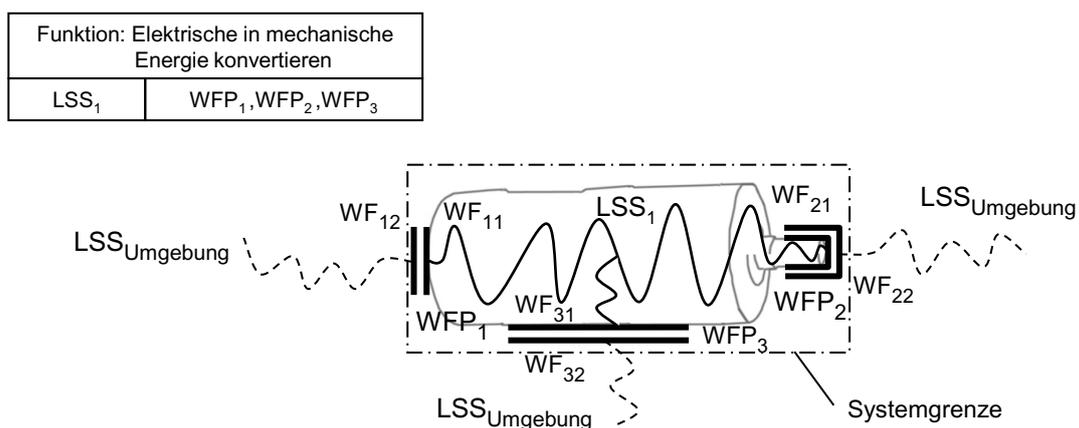


Abbildung 2-16. Ein Beispiel des C&C-M eines E-Motors auf höchster Abstraktionsebene (3 WFP und 2 LSS)

Die Funktions- und Gestaltinformation werden im C&C-M gleichzeitig präsentiert, und dadurch versteht der Entwickler leichter, welche Bauteile/Baugruppen für entsprechende Funktionen notwendig sind. Gleichzeitig wird eindeutig klar, wo die Funktion im System realisiert wird.

Durch den in der Hypothese auf Seite 31 formulierten Modellierungsgrundsatz bietet das C&C-M einen wesentlichen Vorteil bei der systematischen Funktionsanalyse gegenüber anderen Modellen.

## 2.6 Dependency-Matrix

Eine Dependency-Matrix wird als ein Hauptanalysewerkzeug dieser vorliegenden Arbeit angewendet. Die genaue Verwendung der Dependency-Matrix wird in den Kapiteln 5.1 bis Kapitel 6 erläutert. Die wichtigen Grundlagen und einige Beispiele werden in diesem Abschnitt vorgestellt.

Die Dependency-Matrix, die auch als Design-Structure-Matrix (DSM) bekannt ist, wird zur Darstellung und als Analysewerkzeug für die Systemmodellierung, besonders zur Analyse, Zerlegung und Integration, verwendet.<sup>80</sup> Die Dependency-Matrix ist das Dach des House of Quality im Quality Function Deployment (QFD).<sup>81</sup>

Es gibt zwei Typen von Dependency Matrizen, nämlich statische und zeitbasierte (siehe Abbildung 2-17). Die Reihenfolgen der Zeilen und Spalten der zeitbasierten Dependency Matrices zeigen einen Zeitfluss. Dieser Typ der Matrizen wird typischerweise durch Sequencing-Algorithmen analysiert und lässt sich häufig für die Optimierung von Vorgängen anwenden, was allerdings kein Schwerpunkt dieser zu entwickelten Methode mit C&C-M ist. Diese Methode fokussiert auf die statische komponenten-basierte Dependency-Matrix.

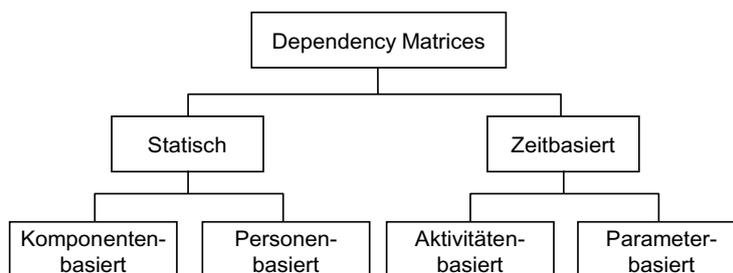


Abbildung 2-17. Typen der Dependency Matrizen nach BROWNING<sup>82</sup>

Die statische Dependency-Matrix ist eine quadratische Matrix mit identischen Zeilen- und Spaltenbezeichnungen. Die Beziehungen zwischen den Komponenten des Systems werden kompakt durch eine Dependency-Matrix dargestellt. Daher ist sie sehr gut geeignet für die Integrationsanalyse.

<sup>80</sup> Vgl. Browning 2001, S. 293

<sup>81</sup> Vgl. Lindemann et al. 2008, S. 55

<sup>82</sup> Vgl. Browning 2001, S. 293

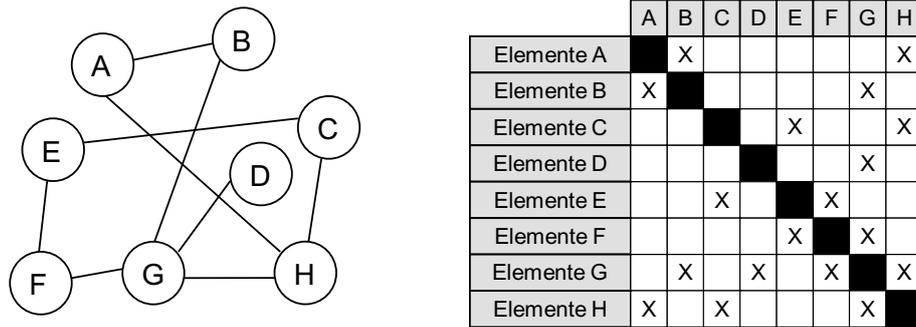


Abbildung 2-18. Beispiel einer Dependency-Matrix als Alternative zur Systemdarstellung, Links: Graph-Darstellung, Rechts: Dependency-Matrix

Ein Beispiel für eine Dependency-Matrix wird in Abbildung 2-18 gezeigt. Die Elemente auf der Diagonalen der Matrix sind schwarz markiert. Sie haben keine wesentliche Bedeutung. Die mit „X“ markierten Außerdiagonalelemente zeigen die Abhängigkeit oder die Beziehung von Elementen. Diese Markierung lässt sich in unterschiedlicher Weise bewerten. PIMMLER und EPPINGER<sup>83</sup> schlagen ein Bewertungsschema der Interaktionen mit den Werten „-2, -1, 0, 1, 2“ vor (Bedeutungen in Tabelle 2-3, Beispiel in Abbildung 2-19).

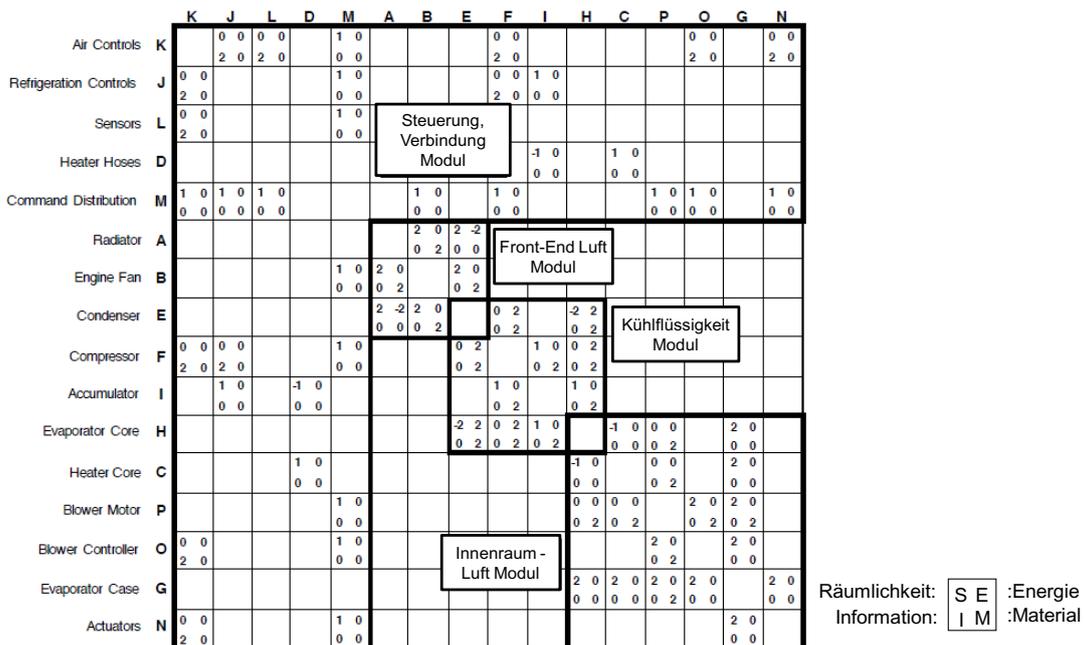


Abbildung 2-19. Beispiel für eine Dependency-Matrix eines Klimasteuerungssystems nach PIMMLER und EPPINGER<sup>84</sup>

<sup>83</sup> Vgl. Pimmler/Eppinger 1994, S. 4

<sup>84</sup> Pimmler/Eppinger 1994, S. 7

Werte der Interaktionen	Bedeutungen
Benötigt (+2)	Notwendig für die Funktionalität
Erwünscht (+1)	Vorteilhaft, aber nicht unbedingt notwendig für die Funktionalität
Gleichgültig (0)	Keinen Einfluss auf die Funktionalität
Nicht erwünscht (-1)	Negative Auswirkungen, die aber nicht die Funktionalität verhindern
Unzuträglich (-2)	Die Interaktion muss verhindert werden, um die Funktionalität zu erreichen

Tabelle 2-3. Bewertungsschema der Interaktionen nach PIMMLER und EPPINGER<sup>85</sup>

MCCORD und EPPINGER<sup>86</sup> bewerten die Beziehungen zwischen den Entwicklungsabteilungen in der Matrix als „hoch“, „mittel“ und „niedrig“ (siehe Abbildung 2-20).

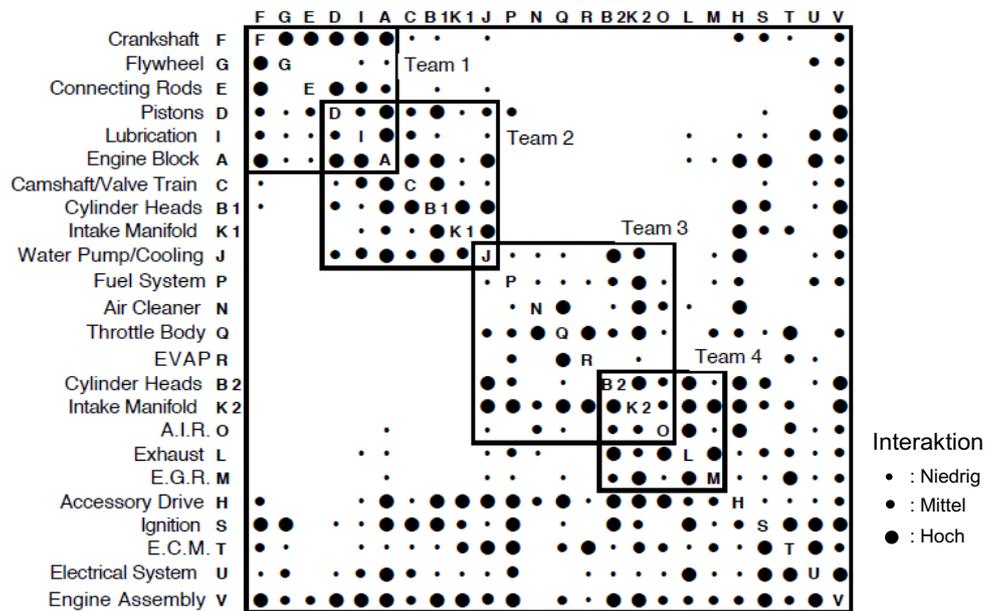


Abbildung 2-20. Beispiel für eine Dependency-Matrix der Beziehungen zwischen Entwicklungsteams eines Motorentwicklungsprojektes nach MCCORD und EPPINGER<sup>87</sup>

SOSALE<sup>88</sup> trennt die Beziehungen in Interaktionen, die von 1 bis 10 skaliert werden, und in Wichtigkeitsfaktoren. SHARMAN und YASSINE<sup>89</sup> (siehe Abbildung 2-21 und Abbildung 2-22) definieren ein Bewertungsschema über die Anzahl der Interaktionsflüsse zwischen den Elementen von 0 (keine wichtige Interaktion) bis 3 (drei oder mehr wichtige Interaktionen).

<sup>85</sup> Vgl. Pimmler/Eppinger 1994, S. 4

<sup>86</sup> Vgl. McCord/Eppinger 1993, S. 20

<sup>87</sup> McCord/Eppinger 1993

<sup>88</sup> Vgl. Sosale et al. 1997, S. 199

<sup>89</sup> Vgl. Sharman/Yassine 2003, S. 39

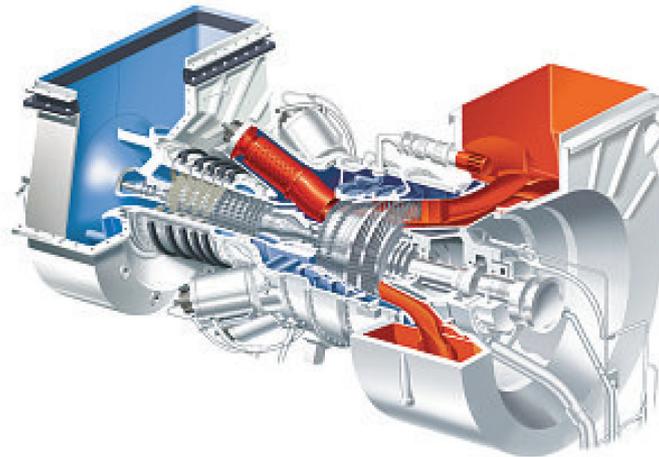
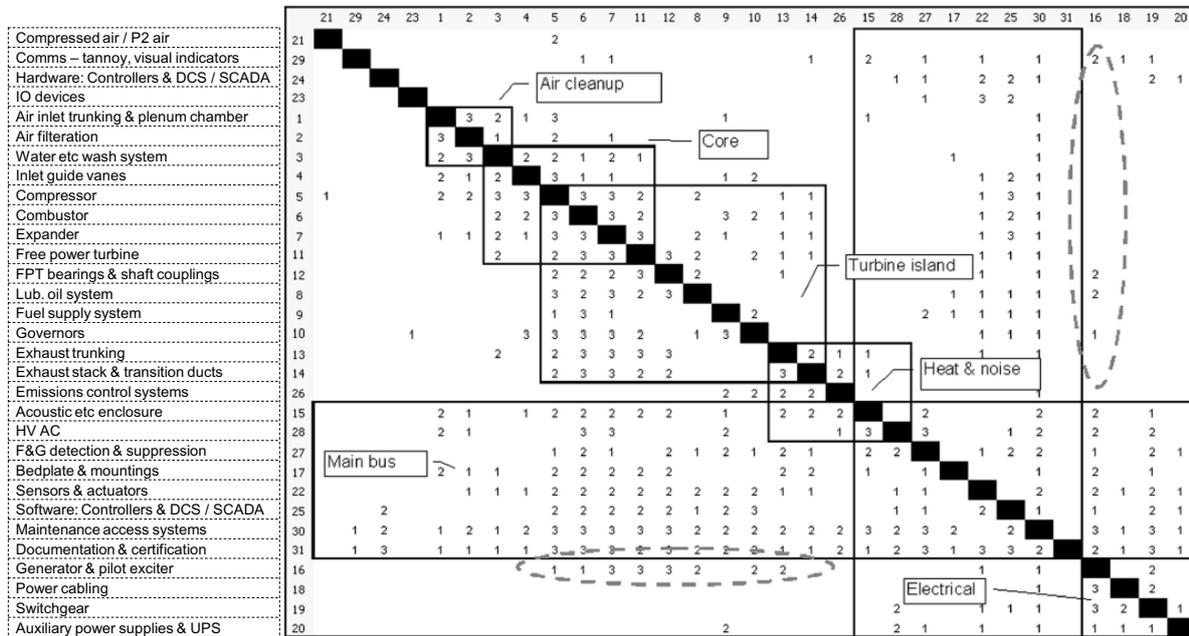


Abbildung 2-21. 3D-Schnittansicht eines Gasturbine-Generators<sup>90</sup>



Wichtigkeiten der Interaktion, 1: wenig wichtig, 2: wichtig, 3: sehr wichtig

Abbildung 2-22. Beispiel für eine Dependency-Matrix eines Gasturbine-Generators nach Sharman und Yassine<sup>91</sup>

Die statische Dependency-Matrix stellt die vorhandenen Systemelemente dar, wie z. B. die Komponenten einer Produktarchitektur oder Gruppen/Personen in einer Organisation. Statische Dependency-Matrizen werden in der Regel durch Clustering-Algorithmen analysiert. Zur Gruppierung der Matrixelemente wurden bereits viele Algorithmen eingesetzt.

<sup>90</sup> Siemens SGT-400 Gasturbinen

<sup>91</sup> Sharman/Yassine 2003, S. 53

HÖLTTÄ ET AL.<sup>92</sup> und LI ET AL.<sup>93</sup> setzen die Tree-basierte Clusteranalyse<sup>94</sup> ein, welche auf der Analyse der Ähnlichkeiten von Merkmalen der Systemelemente basiert (siehe Abbildung 2-35). Die Ähnlichkeiten/Unähnlichkeiten aller Systemelemente werden analysiert, bewertet und dann in einer Dependency-Matrix dargestellt. Durch alle Werte der Ähnlichkeiten in der Matrix wird ein Dendrogramm (Beispiel in Abbildung 2-35) erstellt. Damit sieht man die Gruppierung aller Module, die der Produkthierarchie entspricht.

KUSIAK und HUANG<sup>95</sup> benutzen ein sogenanntes Fuzzy-neuronales Netz für die Clusteranalyse, in der die umfangreichen Informationen der Systemelemente und Regeln zur Gruppierung immer zuerst für die Analyse angegeben werden müssen. EPPINGER<sup>96</sup> beschreibt, dass er zur Festlegung der optimalen Gruppierung eine systemspezifische heuristische Clusteranalyse mit Excel Makro und C-Programm benutzt.

Die naturanalogen Verfahren werden ebenso zur Gruppierung der Matrixelemente angewendet. WHITFIELD ET AL.<sup>97</sup>, LI ET AL.<sup>98</sup> und ROGERS ET AL.<sup>99</sup> verwenden genetische Algorithmen (GA – Detail im Abschnitt 2.7), um eine optimale Gruppierung der Matrixelemente zu erhalten. Ein GA-ähnlicher Suchalgorithmus, der Simulierte-Abkühlungs-Algorithmus (Engl.: „*simulated annealing algorithm*“), der auf der Simulation eines in der Natur vorkommenden Prozesses des Abkühlens von Stoffen basiert, wird in der Arbeit von SOSALE ET AL.<sup>100</sup> eingesetzt.

In vielen praktischen und theoretischen Bereichen der Produktentwicklung wird die Dependency-Matrix eingesetzt. Zum Beispiel wird sie von SOSA ET AL.<sup>101</sup> für die Systemintegrationsanalyse einer Flugzeugturbine verwendet. SOSALE ET AL.<sup>102</sup> modularisiert einen Generator mithilfe der Dependency-Matrix. HÖLTTÄ ET AL.<sup>103</sup> analysiert die Produktarchitektur eines Mobiltelefons. In der Automobilindustrie beschreiben KRISHNAMACHARI und PAPALAMBROS<sup>104</sup>, BROWNING<sup>105</sup>, EPPINGER<sup>106</sup>,

---

<sup>92</sup> Hölttä et al. 2003

<sup>93</sup> Li et al. 2003

<sup>94</sup> Romesburg 1984

<sup>95</sup> Kusiak/Huang 1996

<sup>96</sup> Vgl. Eppinger 1997, S. 201

<sup>97</sup> Whitfield et al. 2001

<sup>98</sup> Li et al. 2003

<sup>99</sup> Rogers et al. 2006

<sup>100</sup> Sosale et al. 1997

<sup>101</sup> Sosa et al. 2000

<sup>102</sup> Sosale et al. 1997

<sup>103</sup> Hölttä et al. 2005

<sup>104</sup> Krishnamachari/Papalambros 1996

KELLER ET AL.<sup>107</sup> den Einsatz der Dependency-Matrix zur Produktanalyse. Allerdings besteht der Nachteil solcher Methoden darin, dass sie an bereits existierenden Produkten angewendet werden. Die Ergebnisse dieser Methoden sind meist von hoher Qualität da beim Zeitpunkt der Analyse die Gestalt, und damit die Informationsbasis für die Analyse, bereits größtenteils festgelegt ist. Allerdings gestaltet sich die Rückführung der Ergebnisse in den Produktentwicklungsprozess meist schwierig, da Änderungen an fast fertig konstruierten Produkten sehr zeit- und kostenaufwendig sind.

Dieser Abschnitt zeigt, wie die Dependency-Matrix-Methode ständig weiterentwickelt wurde, um allgemeine Systeme möglichst einfach und rechnerbasiert analysieren zu können. Aufgrund der klaren Darstellung der Interaktionen und der Möglichkeit zur Clusteranalyse wird die Dependency-Matrix in der vorliegenden Arbeit eingesetzt.

## 2.7 Genetische Algorithmen (GA)

Genetische Algorithmen (GA)<sup>108</sup> bilden eine Untergruppe der evolutionären Algorithmen, die stochastische Suchverfahren sind. Diese Verfahren sind an die Prinzipien der natürlichen Evolution angelehnt, die gleichzeitig mit einer Vielzahl von potenziellen Lösungen arbeiten.

Zum Verständnis genetischer Algorithmen werden im Folgenden einige wichtige Begriffe erklärt.

Name	Bedeutung
Chromosom	String, Zeichen- oder Zahlenkette, nicht notwendigerweise mit fester Länge
Gen	Charakter, Feature, Zeichen, Variable
Population	Menge bzw. Familie von Chromosomen
Fitness	Güte einer durch ein Chromosom kodierten Lösung
Generation	Population zu einem Zeitpunkt
Reproduktion	Erzeugen von Chromosomen aus einem oder mehreren Chromosomen

Tabelle 2-4. Begriffserklärung bei genetischen Algorithmen<sup>109</sup>

<sup>105</sup> Browning 1997

<sup>106</sup> Eppinger 1997

<sup>107</sup> Keller et al. 2005

<sup>108</sup> Holland 1975

<sup>109</sup> Gerdes et al. 2004

Das Prinzip „der Stärkere überlebt“ wird auf diese Lösungen angewandt, um immer bessere Lösungen zu erhalten, die am Ende zu einer optimierten Lösung führen. Es gibt auch noch viele andere numerische Suchverfahren, beispielsweise das Gradientenverfahren<sup>110</sup>, das Newton Verfahren<sup>111</sup> oder der Simulierte-Abkühlungs-Algorithmus<sup>112</sup> (Engl.: „*simulated annealing algorithm*“). Die wichtigen Unterschiede zwischen den evolutionären Algorithmen (auch GA) und den anderen Verfahren sind z. B.:<sup>113</sup>

- Evolutionäre Algorithmen suchen in einer Population von Punkten parallel, nicht nur von einem einzelnen Punkt aus.
- Evolutionäre Algorithmen benötigen keine Ableitungen der Zielfunktionen oder andere Hilfsinformationen. Nur der Zielfunktionswert wird als Grundlage für die Suche verwendet.
- Evolutionäre Algorithmen verwenden Wahrscheinlichkeitsregeln, keine deterministischen Regeln.
- Evolutionäre Algorithmen können auf Probleme der verschiedensten Repräsentationen (kontinuierliche, diskrete Variablen) angewendet werden.
- Evolutionäre Algorithmen arbeiten sowohl im kontinuierlichen als auch für diskontinuierlichen und multimodalen Funktionsraum.
- Evolutionäre Algorithmen sind im Allgemeinen einfach und flexibel anzuwenden, da es für die Festlegung der Zielfunktion keine Einschränkungen gibt.

Aufgrund dieser Besonderheiten im Vergleich zu anderen Verfahren sind die evolutionären Algorithmen sehr geeignet für stark nichtlineare oder diskontinuierliche Funktionen.<sup>114</sup>

Genetische Algorithmen werden zur Suche von optimalen Lösungen in verschiedenen Bereichen eingesetzt. Die ersten Anwendungen von genetischen Algorithmen sind aus den ausgehenden 60er und beginnenden 70er Jahren zu finden<sup>115</sup>.

---

<sup>110</sup> Meister 2005

<sup>111</sup> Deuffhard/Hohmann 2002

<sup>112</sup> Feldmann 1999, S. 130

<sup>113</sup> Nach Pohlheim 2000, S. 11

<sup>114</sup> Vgl. Pohlheim 2000, S. 11

<sup>115</sup> Vgl. Schöneburg et al. 1993, S. 257

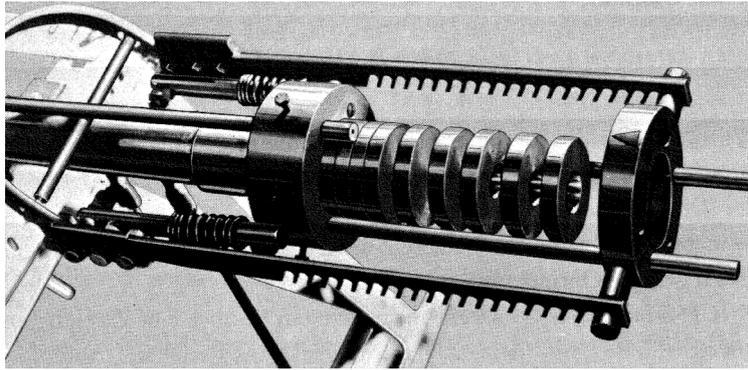


Abbildung 2-23. Versuchsaufbau – segmentierte Düse nach SCHWEFEL<sup>116</sup>

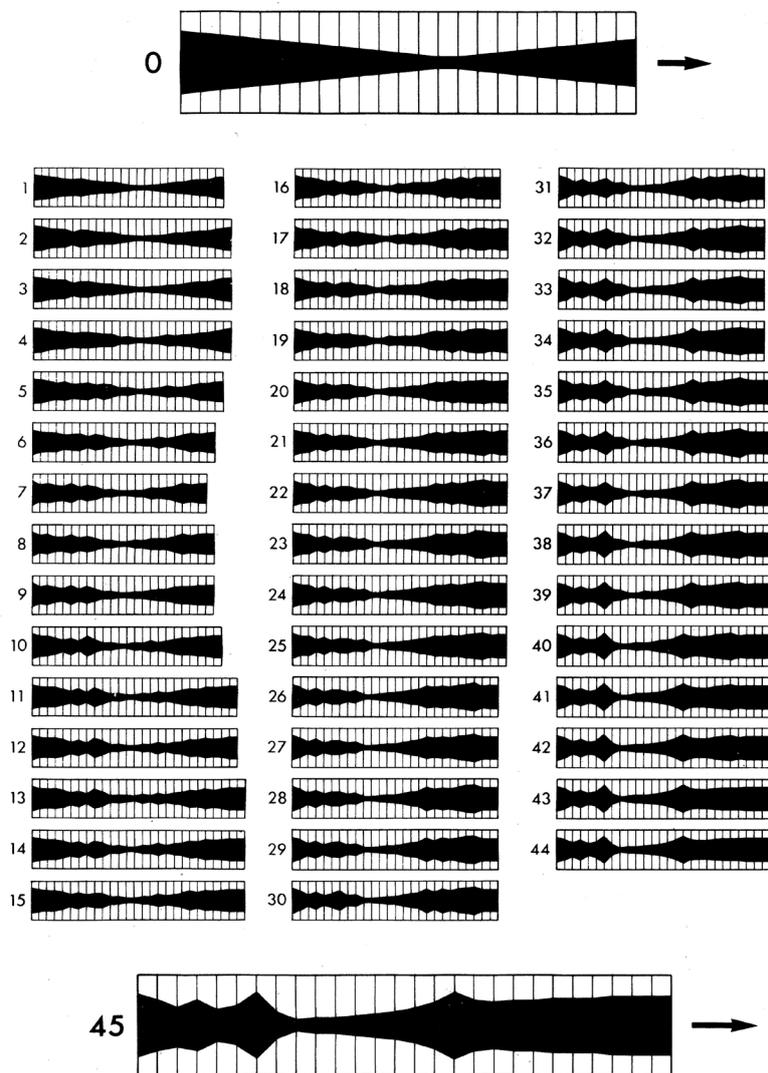


Abbildung 2-24. Schematische Darstellung der Veränderung des Querschnitts der Zweiphasen-Überschalldüse während der Optimierung mittels der Evolutionsstrategie<sup>117</sup> (von der Anfangsform 0 zur Optimalform 45)

<sup>116</sup> Rechenberg 1973, S. 34

Eines der ersten erfolgreichen Beispiele für die Anwendung eines genetischen Algorithmus mit wirtschaftlicher Relevanz war die Optimierung einer Überschalldüse für ein Raumfahrzeug durch SCHWEFEL<sup>118</sup> im Jahr 1968 an der TU Berlin. Die Düse wurde gleichzeitig von Flüssigkeit und Dampf durchströmt. Aufgrund der sich dadurch einstellenden komplexen Strömung konnte keine analytische Optimierung durchgeführt werden. Die Idee von SCHWEFEL war, die rotationssymmetrische Düsenform aus beliebig austauschbaren Segmenten zusammenzusetzen.

Insgesamt standen 330 Segmente mit passend abgestuften konischen Innenbohrungen zur Verfügung. Damit ließen sich theoretisch mehr als  $10^{60}$  verschiedene Düsenformen ohne Sprünge in der Kontur zusammenstellen. Abbildung 2-23 zeigt den Versuchsaufbau.

Da eine analytische Lösung nicht möglich war, wurde in dieser Untersuchung für die Optimierung der Düsenform die Evolutionsstrategie eingesetzt. Dabei wurde die Anordnung der Düsensegmente entsprechend der Evolutionsstrategie bis hin zur endgültigen Optimalform variiert. Das Ergebnis der Optimierung war eine völlig unerwartete Form der Düse (Form 45 in Abbildung 2-24). Der Wirkungsgrad der konischen Ausgangsform (Form 0 in Abbildung 2-24) betrug 55%. Die nach der Evolutionsstrategie entwickelte Optimalform besitzt einen Wirkungsgrad von nahezu 80%.<sup>119</sup>

Heute liegen erfolgreiche Implementierungen der GA in großer Zahl vor, und eine zunehmende Zahl von Veröffentlichungen und Konferenzen zeigt den Einsatz in fachübergreifenden Anwendungen.<sup>120</sup> So werden GA in verschiedenen Bereichen der Produktentstehung eingesetzt z. B. für die Suche nach der optimalen Form von Linsen<sup>121</sup>, Adaptive Verkehrsleittechnik<sup>122</sup>, Parameteridentifikation eines Dieselmotormodells<sup>123</sup>, Optimierung der Parameter von Reglern (Fahrzeuglenkung)<sup>124</sup>, Steuerung eines komplexen Systems (Gewächshausklima)<sup>125</sup>.

---

<sup>117</sup> Rechenberg 1973, S. 35

<sup>118</sup> Vgl. Schöneburg et al. 1993, S. 257

<sup>119</sup> Vgl. Rechenberg 1973, S. 36

<sup>120</sup> Vgl. Butteltmann/Lohmann 2004

<sup>121</sup> Schöneburg et al., S. 368

<sup>122</sup> Vgl. Schöneburg et al. 1993, S. 366 ff.

<sup>123</sup> Pohlheim 2000, S. 204 ff.

<sup>124</sup> Pohlheim 2000, S. 226 ff.

<sup>125</sup> Pohlheim 2000, S. 239 ff.

### Einfacher genetischer Algorithmus

Die Struktur eines einfachen evolutionären Algorithmus wird in Abbildung 2-25 gezeigt. Sie ist eine einfache grundlegende Struktur aus den grundsätzlichen Bestandteilen der evolutionären Algorithmen.

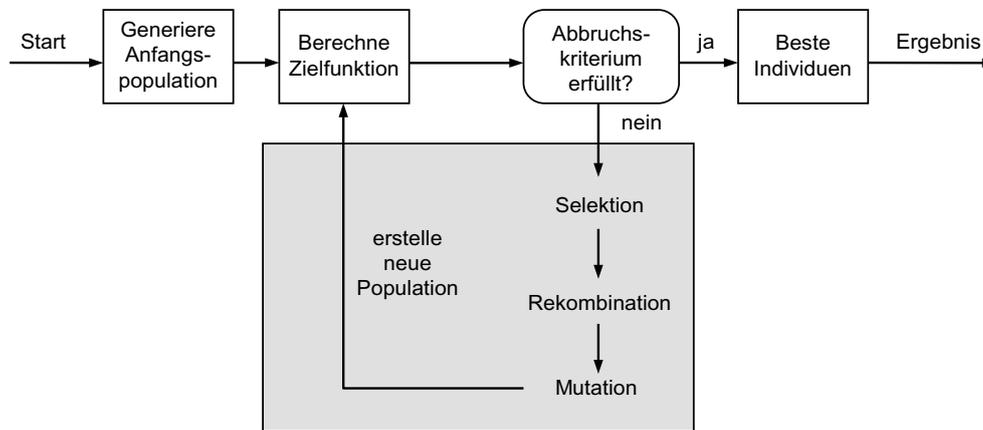


Abbildung 2-25. Struktur eines einfachen evolutionären Algorithmus<sup>126</sup>

Der Algorithmus beginnt mit der Initialisierung. Die Anfangspopulation wird normalerweise zufällig initialisiert, wobei auch problemspezifische Verfahren angewendet werden können. Durch die Zielfunktion werden alle Elemente der erstellten Anfangspopulation bewertet. Nun wird die Population dahingehend überprüft, ob das Abbruchkriterium in dieser Generation erfüllt ist. Wenn das Abbruchkriterium noch nicht erreicht ist, wird eine neue Generation durch Selektions-, Rekombinations- und Mutationsoperation erstellt. Die Individuen (Eltern) für die Erstellung von neuen Individuen (Nachkommen) werden nach deren Fitness ausgewählt.

Zu Beginn einer Initialisierungsphase lässt sich eine Suche über mehrere Generationen durchführen. Eine Anzahl neuer Lösungen wird in jeder Generation durch die Auswahl von Individuen entsprechend ihrer Fitness produziert.

Um eine neue Generation von Individuen zu generieren, werden jeweils Individuen aus der vorhandenen Population abhängig von ihrer Fitness selektiert, aus ihnen durch Variation zwei neue Individuen erzeugt, und in die neue Population eingefügt. Man erhält so den folgenden simplen Basis-Algorithmus, der auch als kanonischer GA bezeichnet wird<sup>127</sup>:

<sup>126</sup> Pohlheim 2000

<sup>127</sup> Buttelmann/Lohmann 2004, S. 152

**Prozedur Genetischer Algorithmus****Beginn**

t = 0

Generiere  $P(t)$  zufälligBerechne Fitnesswerte für  $P(t)$ **Solange** (nicht Abbruchkriterium erfüllt)**Beginn**

t = t+1

Bilden der neuen Population  $P(t)$ :

Selektion

Rekombination

Mutation

Berechne Fitnesswerte für  $P(t)$ **Ende****Ende**

, wobei  $P(t) = \{a_1(t), \dots, a_n(t)\}$ ,  $t$  = Iterationsschritt,  $a_i(t)$  = eine mögliche Lösung.

Durch die Anwendung evolutionärer Reproduktionsoperatoren werden neue Individuen erzeugt. Diese Individuen lassen sich in die Population einfügen und daraus entsteht eine neue Population.

Kodierung und Initialisierung

Um eine Population von Chromosomen erstellen zu können, müssen die Längen der einzelnen Gene in Bits berechnet und diese zur Gesamtlänge des Chromosoms aufsummiert werden.<sup>128</sup> Üblicherweise werden drei Kodierungen in dem genetischen Algorithmen benutzt: die binäre, ganzzahlige und reelle Kodierung. Die Auswahl der Kodierung hängt sehr stark von der Zielfunktion ab. GOLDBERG<sup>129</sup> hat eine Grundregel, das „Principle of minimal alphabets“, zur Auswahl der Kodierung vorgeschlagen: *Das Alphabet der Kodierung sollte so klein wie möglich gewählt werden, wie es einem natürlichen Ausdruck der Lösung entgegenkommt.* Die Anwendung einer ungeeigneten Kodierung kann zu Problemen mit der Rechenzeit führen, indem sie die Suche nach Lösungen erfordert, die nicht den Suchraum betreffen. Dagegen bekommt man wahrscheinlich keine gute Lösung durch die Anwendung eines zu kleinen Alphabets.

Die Anfangspopulationen werden normalerweise durch eine zufällige Initialisierung der Individuen innerhalb des Suchraums erstellt. Es besteht auch die Möglichkeit

---

<sup>128</sup> Huwyler/Weyland 2004

<sup>129</sup> Goldberg 1989

einer nicht-zufälligen Initialisierung, z. B. einer Optimierung mit einigen bekannten guten Lösungen, was jedoch im Normalfall nicht notwendig ist.

### Selektion

Die Selektion ist ein Auswahlverfahren für Individuen bei der Erstellung von Nachkommen für die nächste Generation. Das bekannteste Verfahren zur Auswahl von Individuen ist die Rouletterad-Selektion, ein stochastisches Verfahren. Die Individuen des Selektionspools lassen sich einzelnen Abschnitten zuordnen. Jede Abschnittsgröße ist eine Fitness eines Chromosoms. Eine Zahl wird jetzt zufällig erzeugt. Das Individuum, das der Zufallszahl entspricht, wird ausgewählt. Der Vorgang wird wiederholt, bis die Anzahl der Individuen in der neuen Population ausreicht.

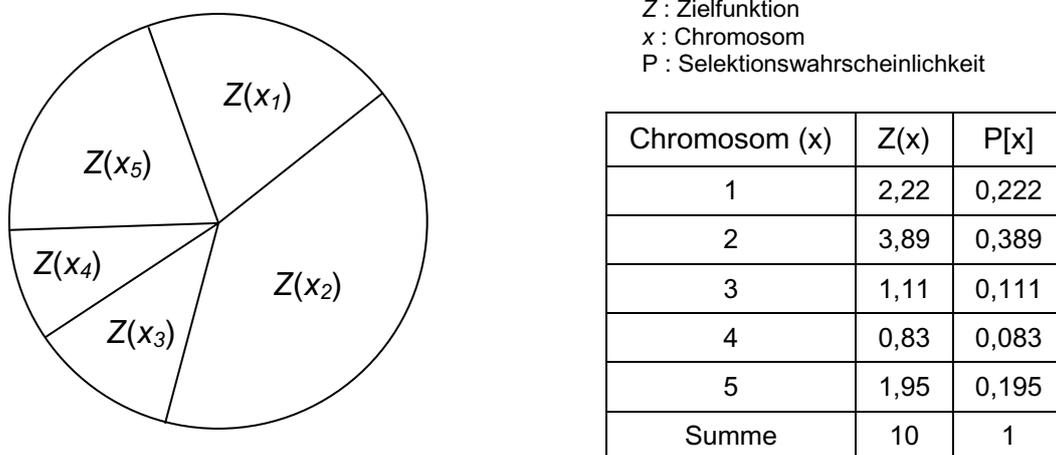


Abbildung 2-26. Ein einfaches Beispiel für eine Rouletterad-Selektion

Die Wahrscheinlichkeit, dass ein bestimmtes Chromosoms ausgewählt wird, wird durch die folgende Formel beschrieben:

$$P[x_a] = \frac{Z(x_a)}{\sum_{i=1}^k Z(x_i)} \quad (2.4)$$

Andere Selektionsverfahren sind z. B. das „Stochastic Universal Sampling“<sup>130</sup> oder die Turnirselektion<sup>131</sup>.

### Rekombination

Nachkommen werden durch Rekombination von den ausgewählten Eltern produziert. Die Informationen der Eltern werden miteinander kombiniert. Es gibt viele

<sup>130</sup> Baker 1987

<sup>131</sup> Goldberg/Deb 1991

unterschiedliche Rekombinationsverfahren, die von der Repräsentation der Variablen in den Individuen (reell, ganzzahlig, binär) und dem Einsatzzweck abhängen. Für die vorliegende Arbeit ist die diskrete Rekombination<sup>132</sup> (uniform crossover) interessant. Sie kann auf alle Repräsentationen (reell, ganzzahlig, binär) angewandt werden. Bei der diskreten Rekombination werden die Variablen zwischen Elternindividuen zur Erstellung der Nachkommen ausgetauscht. Es wird für jede Position der Variablen entschieden, von welchem Elternteil der Variablenwert übernommen wird. Abbildung 2-27 zeigt ein Beispiel für eine diskrete Rekombination.

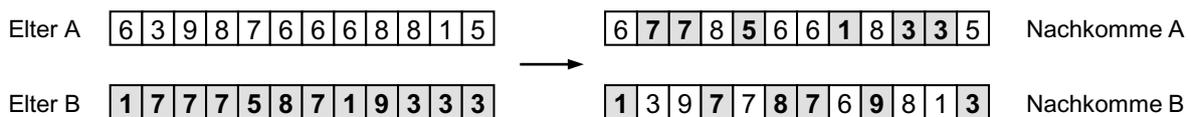


Abbildung 2-27. Beispiel der diskreten Rekombination

### Mutation

Die Mutation sorgt dafür, dass der genetische Algorithmus nicht vorzeitig in einer lokalen Lösung abbricht. Die lokale Lösung kann die Population dominieren und im Extremfall kann es dazu führen, dass alle Individuen gleich sind. Das heißt, dass allein durch die Rekombination keine Verbesserung der Individuen erzeugt werden kann. Bei dieser Situation hilft die Mutation, schnell neue Individuen zu generieren, was besonders vorteilhaft ist bei einer kleinen Population.

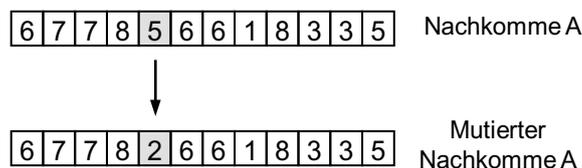


Abbildung 2-28. Beispiel für eine Mutation eines ganzzahligen Nachkommen

Durch die Mutation werden die Nachkommen nach der Rekombination zufällig verändert. Die für ganzzahlige und reelle Variablen verwendeten Mutationsverfahren sind die kontinuierliche und die verteilte Mutation.<sup>133</sup> Für binäre Variablen besteht die Mutation aus einer einfachen Umschaltung des Variablenwertes. Für diesen Schritt müssen zwei Dinge bestimmt werden: die Wahrscheinlichkeit, mit der eine Variable mutiert und die Größe des Mutationsschritts. Die am meisten empfohlene Wahrscheinlichkeit beträgt  $1/n$  ( $n$ : Anzahl der Variablen eines Individuums).<sup>134</sup> Die

<sup>132</sup> Syswerda 1989

<sup>133</sup> Voigt et al. 1995

<sup>134</sup> Mühlenbein/Schlierkamp-Voosen 1993

Festlegung des Mutationsschrittes reeller und ganzzahliger Variablen ist etwas schwieriger. Ein kleiner Wert des Mutationsschritts hat normalerweise einen sehr positiven Einfluss auf die Suche, jedoch kann ein größerer Wert schneller zum Erfolg führen. Es ist deswegen empfehlenswert, mit einem kleinen Mutationsschritt zu arbeiten und diesen nur zu vergrößern, wenn es nötig ist. MÜHLENBEIN und SCHLIERKAMP-VOOSEN<sup>135</sup> haben einen Mutationsoperator entsprechend der folgenden Gleichung vorgeschlagen, wobei bei ganzzahligen Variablen die minimalen Mutationsschritte 1 betragen:

$$\text{Var}_i^{\text{Mut}} = \text{Var}_i + s_i \cdot r_i \cdot a_i \quad (2.5)$$

wobei  $i \in \{1, 2, \dots, n\}$  gleichverteilt,

$s \in \{-1, +1\}$  gleichverteilt,

$r_i = r \cdot \text{Domain}_i$ ,  $r$ : Mutationsbereich,  $\text{Domain}$ : Definitionsbereich der Variable  $i$

$a_i = 2^{-uk}$ ,  $u \in [0, 1]$  gleichverteilt,

$k$ : Mutationspräzision

### Abbruchkriterium

Ein genetischer Algorithmus läuft, bis das vorher definierte Abbruchkriterium erfüllt ist. Die direkten Abbruchkriterien sind die maximale Anzahl von Generationen, die maximale Anzahl der Berechnungen, die maximale Rechnerzeit und die Differenz des besten Zielfunktionswertes zu einem vorgegeben Wert.<sup>136</sup>

Am häufigsten wird die maximale Anzahl der Generationen eingesetzt. Durch dieses Kriterium wird der Abbruch des Suchvorgangs sehr überschaubar und ist garantiert. Das Rechnerzeit-Kriterium lässt sich einsetzen, wenn die Rechnerzeit begrenzt ist, z. B. bei einem Echtzeitsystem.

## **2.8 Stand der Forschung – Produktarchitekturfestlegung**

Die Idee einer Festlegung der Produktarchitektur möglichst früh im Projektverlauf ist zweifellos nicht neu. Sie wurde bereits in mehreren bekannten Entwicklungsprozessen angewandt und auch in einigen Forschungsberichten veröffentlicht.

Die Entwicklungsmethode und die vorherigen relevanten Forschungsarbeiten zur Festlegung der Produktarchitektur werden in diesem Abschnitt erläutert. Zunächst werden die relevanten Aspekte der VDI-Richtlinie 2221 zu diesem Thema aufgezeigt.

---

<sup>135</sup> Schlierkamp-Voosen/Mühlenbein 1996

<sup>136</sup> Pohlheim 2004

Danach werden fünf relevante Methoden von STONE, ALLEN und CARLSON-SKALAK, VAN WIE, HÖLTTÄ ET AL. und ULRICH und EPPINGER vorgestellt.

### 2.8.1 VDI-Richtlinie 2221

Bereits im Jahr 1993 wurde die Produktarchitekturfestlegung während der Konzeptphase in der Produktentwicklungsrichtlinie genannt. Die VDI-Richtlinie 2221, die ein generelles Vorgehen zum Entwickeln und Konstruieren technischer Produkte empfiehlt, hat diese Festlegung als einen Arbeitsabschnitt vorgeschlagen. Hierbei sollen realisierbare Module entworfen werden, wobei die modulare Struktur des Produkts als Arbeitsergebnis von diesem Arbeitsschritt erwartet wird. In der folgenden Abbildung ist das Vorgehen bei der Entwicklung zu sehen.

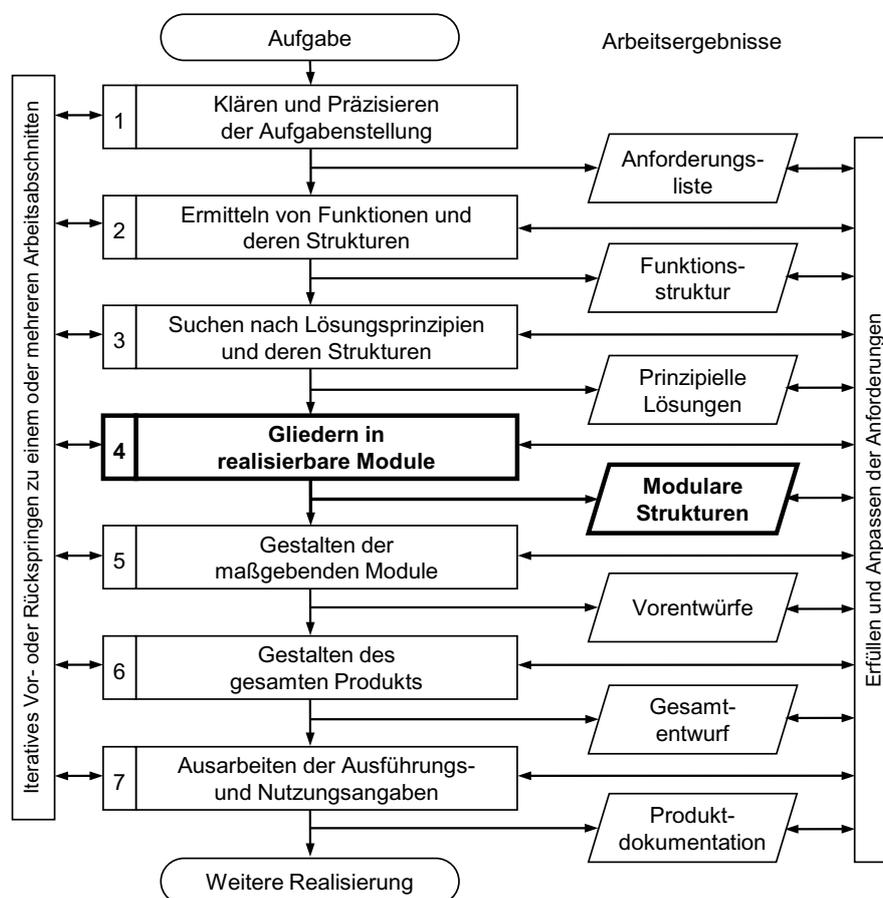


Abbildung 2-29. Generelles Vorgehen beim Entwickeln und Konstruieren<sup>137</sup>

Die Festlegung der Produktarchitektur wird im Arbeitsabschnitt 4 („Gliedern in realisierbare Module“) durchgeführt (siehe Abbildung 2-29). Dieser Schritt wird weiterhin in der VDI-Richtlinie 2223, bei der es um das methodische Entwerfen

<sup>137</sup> VDI-Richtlinie 2221 1993, S. 9

technischer Produkte geht, weiter konkretisiert. In der Richtlinie wird er in zwei untergeordnete Schritte aufgeteilt:

1. in das Erkennen gestaltbeeinflussender Anforderungen und Bedingungen und
2. in das Modularisieren der Prinzipskizze.

Danach ergibt sich die modulare Architektur als Arbeitsergebnis.

Der erste Schritt behandelt die Bedingungen und Anforderungen, die für das Gestalten maßgeblich sind. Dies sind zum Beispiel die funktionalen, kinematischen oder geometrischen Bedingungen und Anforderungen. Mittlerweile sind die allgemeinen Anforderungen bereits in der Anforderungsliste aufgelistet. Nach der Auswahl einer prinzipiellen Lösung muss der Entwickler alle Anforderungen und Randbedingungen hinsichtlich ihrer Gültigkeit überprüfen.

Im zweiten Schritt werden unterschiedliche zusätzliche Gesichtspunkte betrachtet. Sie sind im Folgenden in Form einer Fragenliste beschrieben:

- Wie kann man das Produkt derart in Module gliedern, dass diese beim Entwerfen möglichst unabhängig voneinander bearbeitet werden können?
- Welche Module sind für das Erfüllen der Anforderungen besonders wichtig und damit vorrangig zu bearbeiten?
- Gibt es Unterschiede in der Dringlichkeit der Bearbeitung von Modulen beim Entwerfen?
- Welche Module sind neu zu entwerfen?
- Welche organisatorischen Bedingungen sind in der Konstruktion gegeben?
- Lassen sich Termine durch gezielte Parallelarbeit im Sinne von „Simultaneous Engineering“ erheblich verkürzen?
- Welche Arbeitsschritte lassen sich besser im Team und welche in Einzelarbeit erledigen?

Die für diesen Abschnitt vorgeschlagenen rechnergestützten Systeme in der VDI-Richtlinie 2223 sind allgemeine CAD-Werkzeuge. Es handelt sich um die Produktmodelle (PM), Operationsmethoden (OM), Editoren (ED) und Daten- und Wissensbank (DW).

### 2.8.2 Drei-Heuristiken-Methode nach STONE

Im Jahr 1997 hat STONE<sup>138</sup> eine Methode zur Identifizierung der Module in der Funktionsebene des Produktentstehungsprozesses entwickelt. Die Methode kann sowohl für eine Neukonstruktion als auch eine Weiterkonstruktion angewendet werden, unter der Voraussetzung, dass zunächst eine Funktionsstruktur des Systems erstellt werden muss. Um die Module aus der Funktionsstruktur zu identifizieren, sind drei Heuristiken<sup>139</sup>, nämlich „Dominant Flow“, „Branching Flow“ und „Conversion-Transmission Modules“, ein zentraler Ansatz seiner Arbeit.

Die „Dominant Flow“ Heuristik interessiert sich in erster Linie für den nicht verzweigten und wichtigsten Fluss (Information/Energie/Material) im System aus Sicht des Kunden. Die Subfunktionen, die dieser Fluss darstellt, werden dann zusammen gruppiert. Diese identifizierten Subfunktionen werden als ein Modul definiert. Andere Flüsse, die das Modul verbinden, werden dann „Interaktionen“ genannt (Abbildung 2-30 a).

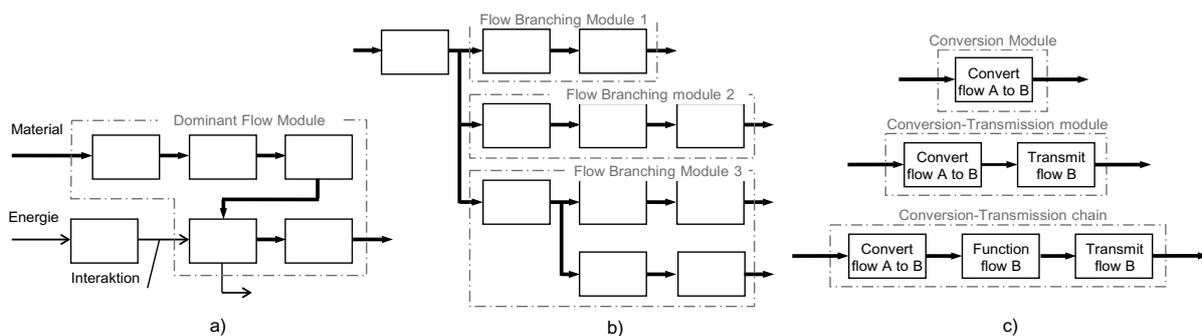


Abbildung 2-30. Drei Heuristiken nach STONE; a) Dominant Flow, b) Branching Flow und c) Conversion-Transmission Modules<sup>140</sup>

Die zweite Heuristik „Branching Flow“ benötigt zuerst die Identifizierung von Flüssen in der Funktionsstruktur, die sich in parallele Funktionen verzweigen. Jeder abgezweigte Fluss formt das Modul der Funktionen, die er durchfließt (Abbildung 2-30 b).

Die letzte Heuristik („Conversion-Transmission Modules“) befasst sich mit den Subfunktionen, die die Flüsse umwandeln oder weiterleiten. Die

<sup>138</sup> Stone 1997

<sup>139</sup> Snowden 2000, „Heuristiken“ (griech.) sind Erfahrungsregeln. Manche Heuristiken sind z. B. als Verfahrensanweisung dokumentiert, andere kennt man im Unternehmen seit Jahren, ohne dass sie irgendwo niedergeschrieben sind. Damit können sie sowohl explizites als auch stilles Wissen des Unternehmens sein. Ähnlich wie im komplexen Straßenverkehr, wo einfache, leicht merkbare Regeln helfen, sind Heuristiken im Geschäftsalltag ideal, um mit der zunehmenden Komplexität umzugehen und Wissen zu verbreiten.

<sup>140</sup> Stone 1997, S. 48, 58, 64

Umwandlungsfunktion wird als ein Modul definiert, wenn diese Umwandlungsfunktion keine Weiterleitfunktionen verbindet. Wenn ja, werden diese Umwandlungsfunktionen zusammen in einem Modul gruppiert. Falls eine andere Funktion zwischen beiden Funktionen existiert und sich nur mit umgewandeltem Fluss darstellen lässt, sind die Funktionen zusammen als Modul zu gruppieren (Abbildung 2-30 c).

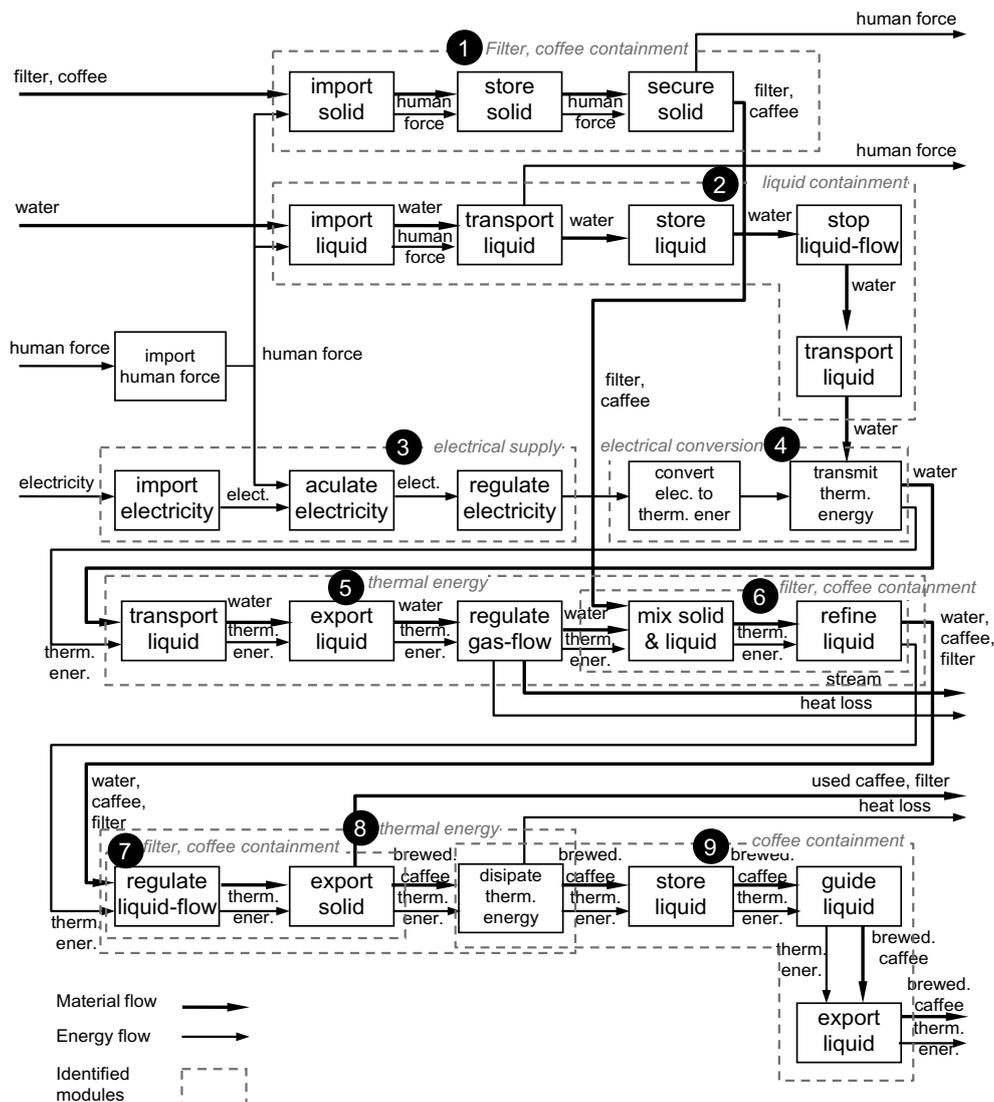


Abbildung 2-31. Funktionsstruktur mit den neun identifizierten Modulen einer Kaffeemaschine<sup>141</sup>

Zahlreiche Haushaltgeräte wurden in der Arbeit von STONE<sup>142</sup> analysiert. Abbildung 2-31 zeigt das Beispiel einer Gruppierung der Teilfunktionen einer Kaffeemaschine mit der Drei-Heuristiken-Methode. Insgesamt werden neun mögliche Module mit der

<sup>141</sup> Vgl. Stone 1997, S. 76

<sup>142</sup> Stone 1997

drei-Heuristiken-Methode identifiziert (5 Dominant Flow Module, 3 Branching Flow Module und 1 Conversion-Transmission Modul). Davon sind nur sechs unabhängige Module, da Modul 5 und 6 sich überlappen und ebenso die Module 7, 8 und 9. STONE beschreibt die Umsetzung aller drei Heuristiken, allerdings gibt er keine Empfehlung, wie die Ergebnisse aller Heuristiken zusammengefasst werden sollen. Die mittels der Heuristiken identifizierten Module können sich überlappen. Letztendlich liegt die Entscheidung, welche Module zu implementieren sind, beim Konstrukteur.

Außerdem hat STONE zur Unterstützung seines Ansatzes eine quantitative Methode entwickelt, durch die eine Bewertung der Kundenbedürfnisse mit den entsprechenden Funktionen verknüpft wird. Die Analyse wird durch die sogenannten Device-Funktion-Matrix und Funktion-Funktion-Matrix verwirklicht. Das Ergebnis der Analyse zeigt, ob das Produkt einem anderen Produkt ähnlich ist und ob die identifizierten Module die Kundenbedürfnisse erfüllen.

### **2.8.3 Produktarchitekturfestlegung nach ALLEN und CARLSON-SKALAK**

Ein Jahr später haben ALLEN und CARLSON-SKALAK<sup>143</sup> eine Arbeit über die Festlegung einer Produktarchitektur während der Aktivitäten der Modellierung von Prinzip und Gestalt veröffentlicht. Die Methode basiert auf einer Modifikation eines bestehenden Produkts mit dem Ziel der gleichzeitigen Optimierung der Produktarchitektur und des Entwicklungsteams. ALLEN und CARLSON-SKALAK betrachten insbesondere, wie die Produktarchitektur und Teamstruktur zusammen definiert werden.<sup>144</sup> Abbildung 2-32 zeigt eine Übersicht der Methode, die mit der Identifikation der Module der vorherigen Produktgeneration anfängt.

Das vorherige Produkt wird zuerst zerlegt und in einem inversen Fischgrätendiagramm dokumentiert. Die Entscheidung darüber, welche Bauelemente zu welchem Modul gehören, liegt bei den Entwicklern. Dann wird die Identifikation der Funktionsstruktur des Produkts durchgeführt unter Berücksichtigung der geeigneten Entwickler und Teamstruktur. Eine Baumstruktur der Funktionen und die zugehörigen Teams werden in diesem Schritt ausgearbeitet. Im letzten Schritt werden die Eigenschaften wie Material, Lebenszyklus und Elementanzahl in die Grundfunktionen im Systemfunktions-Diagramm (Abbildung 2-32) eingefügt, wobei für die Berechnung Matrizen benutzt werden, um zu zeigen, wie modular ein Produkt ist.

---

<sup>143</sup> Allen/Carlson-Skalak 1998

<sup>144</sup> Vgl. Allen/Carlson-Skalak 1998, S. 2

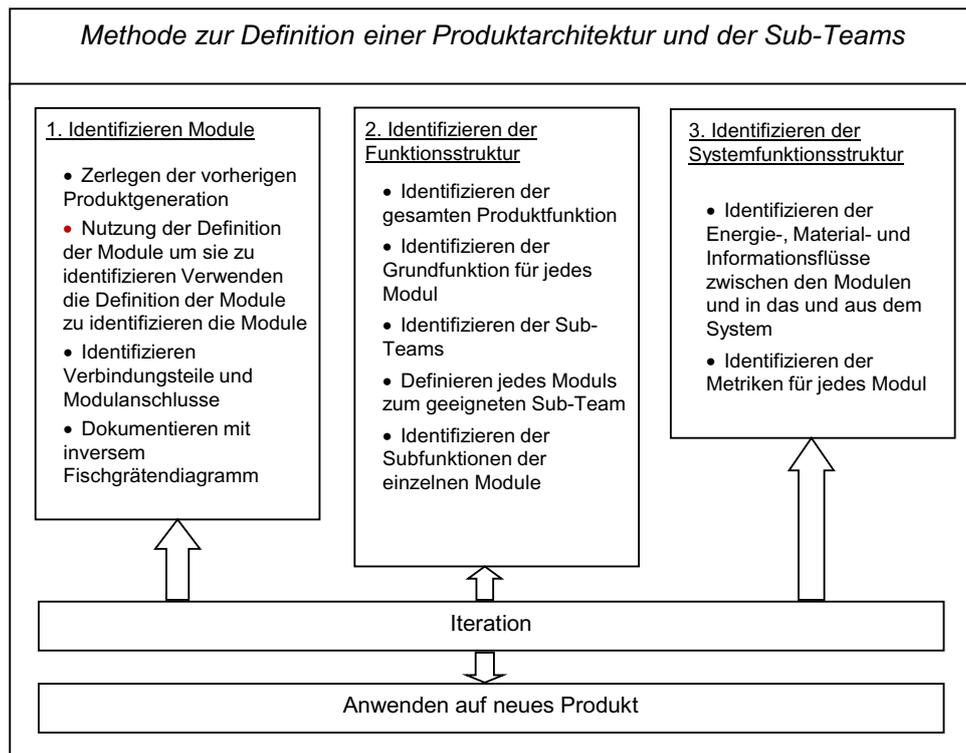


Abbildung 2-32. Methode zur Festlegung einer Produktarchitektur nach ALLEN und CARLSON-SKALAK<sup>145</sup>

Während der Durchführung aller drei Schritte ist es wichtig, dass der Prozess mehrfach durchlaufen wird, um die Informationen aller Schritte zu verknüpfen und das Ergebnis auf seine Richtigkeit hin zu überprüfen.

#### 2.8.4 Produktarchitekturfestlegung nach VAN WIE

VAN WIE<sup>146</sup> hat ein systematisches Vorgehen zur Festlegung einer Produktarchitektur im Jahr 2002 vorgestellt. Diese Methode, die man sowohl für eine Neukonstruktion als auch für eine Weiterkonstruktion anwenden kann, zielt auf die Vermeidung von Iterationen bei der Suche nach einer optimalen Produktarchitektur durch Darstellungen der Produktarchitektur und Handlungsempfehlungen.

Das Vorgehen besteht aus drei Schritten, wie folgende Abbildung zeigt:

<sup>145</sup> Vgl. Allen/Carlson-Skalak 1998, S. 3

<sup>146</sup> Van Wie 2002

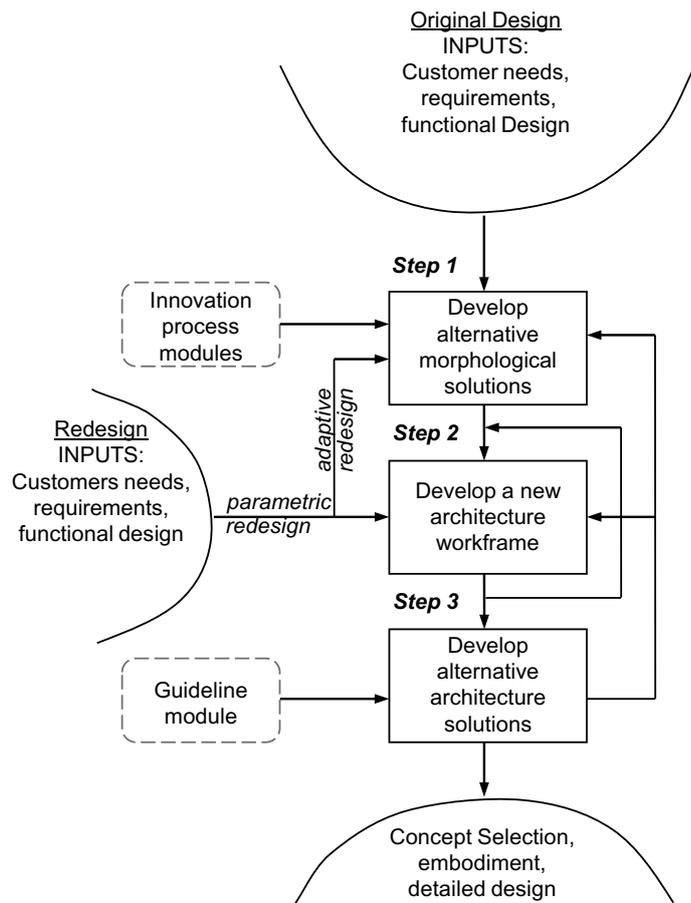


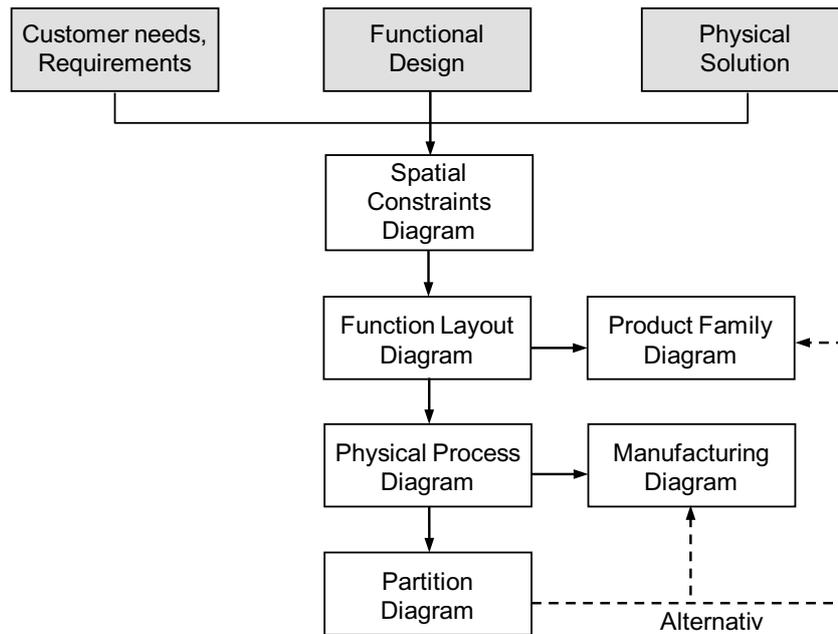
Abbildung 2-33. Vorgehensweise zur Festlegung der Produktarchitektur nach VAN WIE<sup>147</sup>

Die Ausgangsinformationen der Methode betreffen die Kundenanforderungen und Produktfunktionen. Der erste Schritt ist die Generierung alternativer physischer Lösungen, die die Produktfunktionen leisten. Das Ziel dieses Schritts ist eine direkte Umsetzung der Funktionen in eine Geometrie. Jedoch wird er nur begrenzt, um die morphologischen Lösungen zu erstellen<sup>148</sup>. Die Produktmerkmale wie zum Beispiel Bauteilanzahl, Schnittstelle, Fertigungsprozess, Montage usw. werden nicht betrachtet.

Im zweiten Schritt sind die sechs oben genannten Diagramme der Produktarchitektur zu erstellen. Die Reihenfolge zur Erstellung der oben genannten Diagramme wird wie in Abbildung 2-34 vorgeschlagen:

<sup>147</sup> Van Wie 2002, S. 125

<sup>148</sup> Vgl. Van Wie 2002, S. 126

Abbildung 2-34. Reihenfolge der Diagrammerstellung<sup>149</sup>

Die Diagramme sind das Ergebnis des zweiten Schrittes, die als Informationsbasis für den dritten Schritt dienen. Sechs Diagramme, die wesentlich für die Produktarchitekturfestlegung sind, werden auch in seiner Arbeit beschrieben. Das sind:

- Spatial-Constraints-Diagramm,
- Function-Layout-Diagramm,
- Physical-Process-Diagramm,
- Product-Family-Diagramm,
- Manufacturing-Diagramm und
- Partition-Diagramm.

Tabelle 2-5 zeigt die Beispiele der oben genannten Darstellungen als ein Wissensbasis zur Festlegung der Produktarchitektur nach VAN WIE.

<sup>149</sup> Vgl. Van Wie 2002, S. 128

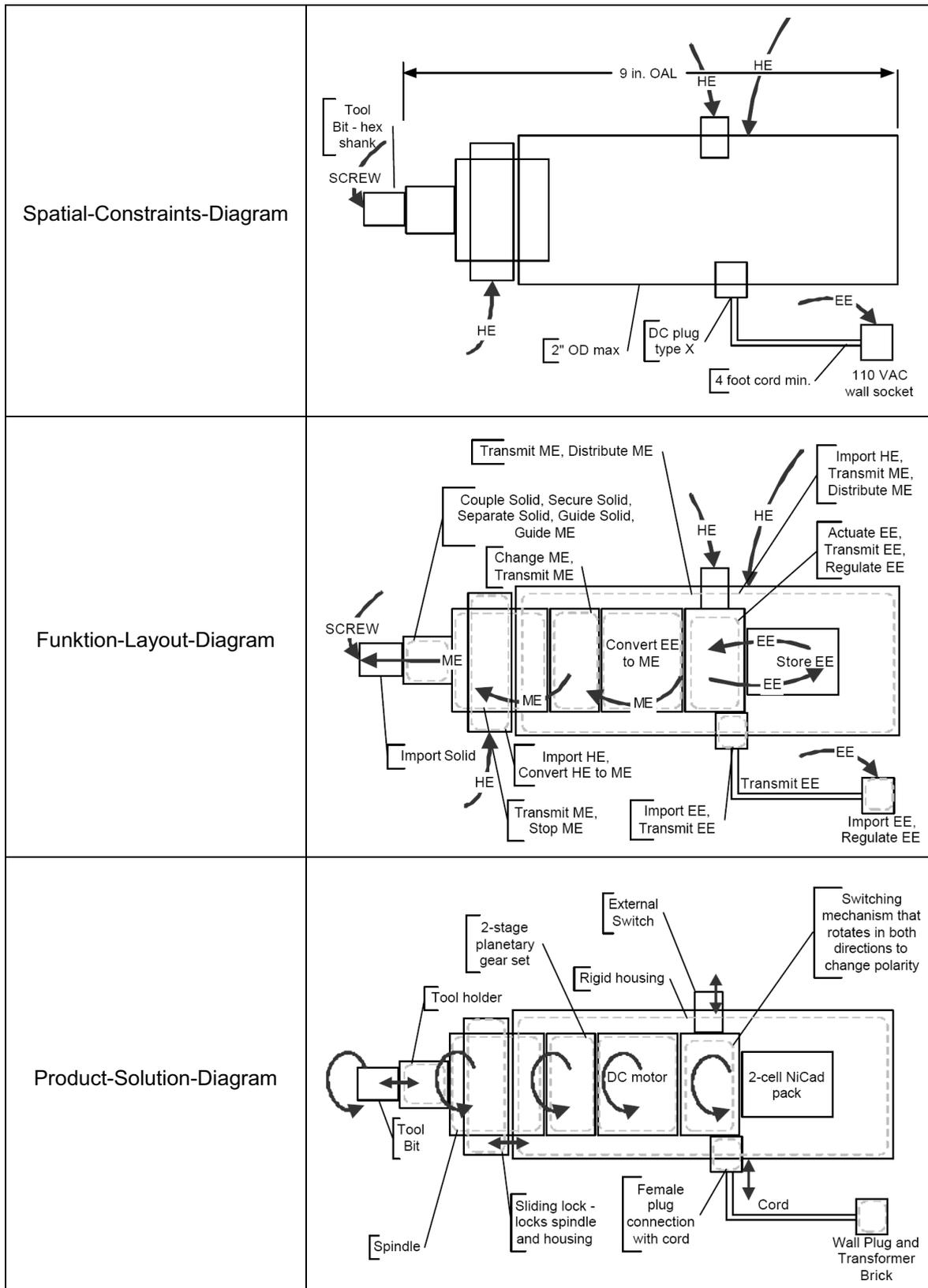


Tabelle 2-5 Beispiele für sechs verschiedene Diagramme eines Akkuschaubers als Ergebnisse des zweiten Schrittes der Architekturfestlegung nach VAN WIE

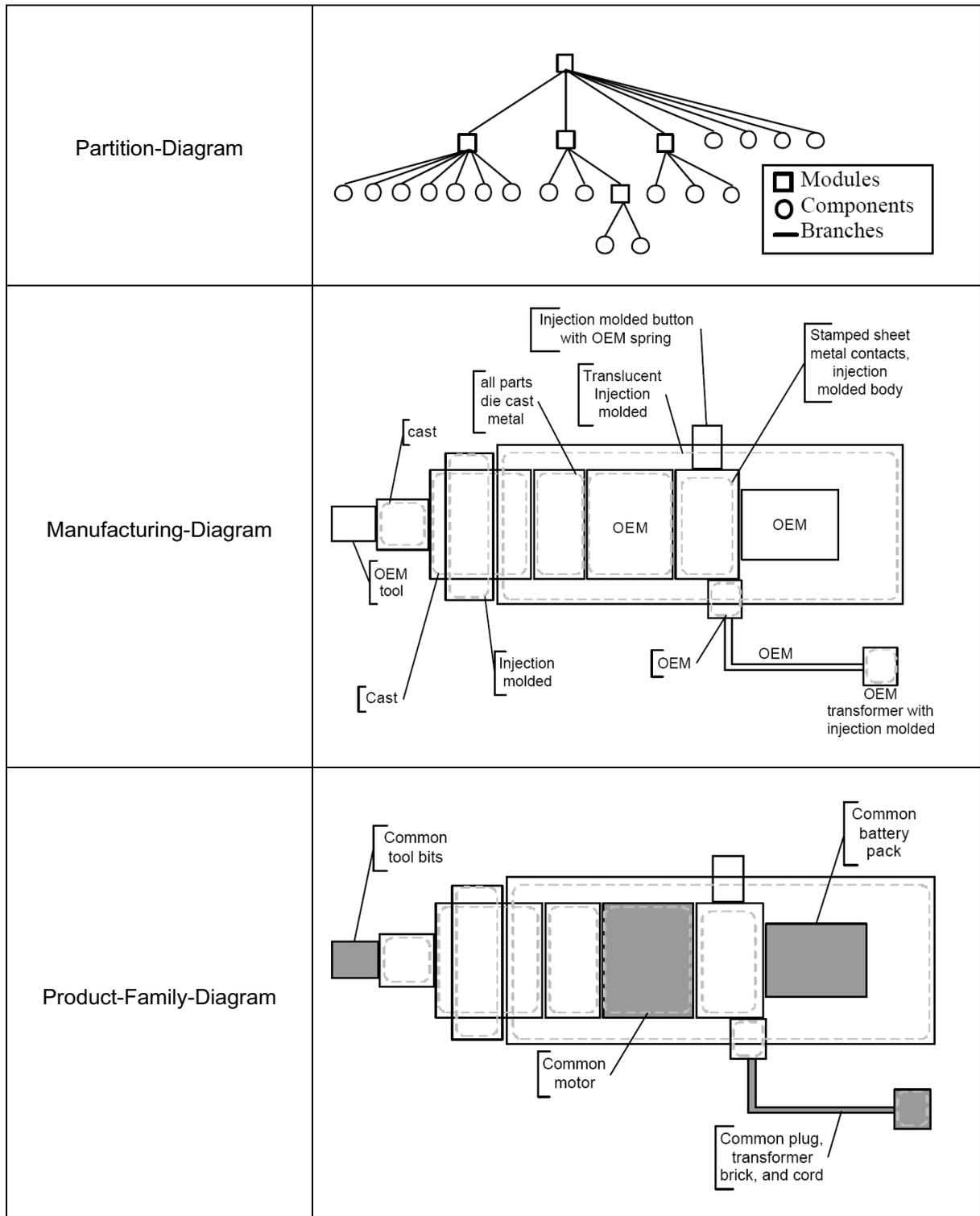


Tabelle 2-5. Beispiele für sechs verschiedene Diagramme eines Akkuschaubers als Ergebnisse des zweiten Schrittes der Architekturfestlegung nach VAN WIE<sup>150</sup> (Fortsetzung)

<sup>150</sup> Vgl. Van Wie 2002, S. 41, 43, 45, 47, 49, 51

Abschließend wird die Produktarchitektur anhand von neun grobe Handlungsempfehlungen („Guideline-Modul“) weiter verfeinert.

In dritten Schritt werden alle sechs Diagramme als Basis für das Guideline-Modul (siehe Abbildung 2-33). Das für die Produktarchitekturfestlegung Guideline-Modul ist eine Liste der Empfehlungen, die von bestehenden Produkten hinsichtlich des Montagekostens analysiert, bewertet werden. In Tabelle 2-6 werden die Empfehlungen in Guideline-Modul nach VAN WIE<sup>151</sup> gezeigt.

Aspects	Recommendation
Product boundary	Smooth the product boundary
User interfaces	Position user interfaces to match user activities.
Multi-configuration	Generate multiple device configurations according to user activities.
Motion	Reduce device motion complexity.
Partition	Partition the device into appropriate modules and components
Excessive loading	Reduce the negative effects of excessive loading
Structural excess reduction	Reduce unnecessary device structure.
Energy and material flowpath	Reduce energy and material in the device
Automation	Increase device automation

Tabelle 2-7. Das Guideline-Modul (zusammengefasst aus VAN WIE<sup>152</sup>)

### 2.8.5 Produktarchitekturfestlegung nach HÖLTTÄ ET AL.

HÖLTTÄ ET AL.<sup>153</sup> modularisieren die Produktarchitektur ebenso wie andere Autoren aus Sicht der Funktionsseite. Die Arbeit von HÖLTTÄ ET AL. fokussiert auf Flüsse statt auf Funktionen der Funktionsstruktur. Das zentrale Werkzeug der Methode ist ein Dendrogramm, welches eine Hierarchie der Produktarchitektur darstellt. Die Funktionen des Produkts werden durch einen Algorithmus, der aus fünf Schritten besteht, in Module gruppiert. Zunächst werden einzelne Produktfunktionen in Blackboxes ( $m^1, m^2, \dots, m^n$ ) umbenannt (1). Dann werden die Eingangs- und Ausgangsinformationen jeder Blackbox eingefügt (2). Anschließend ist zu untersuchen (3) und zu bewerten (4), wie ähnlich und wie unterschiedlich jedes Paar der Blackboxes zueinander ist. Diese Berechnungen werden weiter ausgearbeitet (5).

<sup>151</sup> Vgl. Van Wie 2002, S. 103

<sup>152</sup> Vgl. Van Wie 2002, S. 103 ff.

<sup>153</sup> Hölttä et al. 2003

Die Ergebnisse sind die sogenannten Pseudoabstände, die eine Ähnlichkeit zwischen zwei Produktfunktionen beschreiben. Alle Werte werden in eine Matrix eingetragen und dienen als Basis zur Erstellung eines Dendrogramms. Die Gruppierung der „Blackboxes“ wird als ein Vorschlag für die Produktarchitektur angesehen. Abbildung 2-35 zeigt ein Beispiel für eine Distanz-Matrix und ein Dendrogramm für die Festlegung der Produktarchitektur nach HÖLTTÄ. In der Matrix werden die berechneten Werte der Ähnlichkeiten zwischen den Systemelementen dargestellt. Mittels einer definierten Vorgehensweise wird aus den Distanzwerten ein Dendrogramm wie auf der rechten Seite von Abbildung 2-35 abgeleitet. Dieses Dendrogramm stellt dar, wie die Systemelemente bei unterschiedlichen Modulanzahlen angeordnet werden sollten. Die Anzahl der Schnittpunkte zwischen einer durchgehenden horizontalen Linie und den vertikalen Linien des Dendrogramms ergibt dabei die Anzahl der Module. Im Beispiel wird ein Fall mit drei Modulen betrachtet. Aus dem Dendrogramm kann entnommen werden, dass die drei Module des Beispielsystems folgendermaßen zusammengesetzt sein sollten: Modul 1: ( $m^{2a}$ ,  $m^1$ ,  $m^{4b}$ ), Modul 2: ( $m^{2b}$ ,  $m^3$ ,  $m^{4a}$ ) und Modul 3: ( $m^{2c}$ )

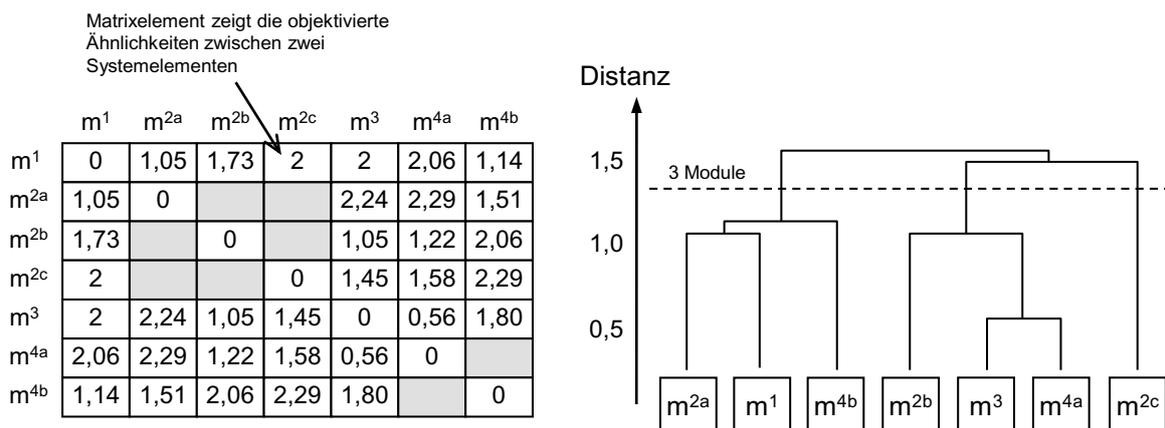


Abbildung 2-35. Beispiele einer Distanz-Matrix und eines Dendrogrammes<sup>154</sup>

### 2.8.6 Produktarchitekturfestlegung nach ULRICH und EPPINGER

ULRICH und EPPINGER<sup>155</sup> stellen in ihrem Buch „Product Design and Development“ auch eine Methode vor, mit der die Architektur eines Produktes definiert werden kann. Diese Methode wird am Ende der Konzeptentwicklungsphase („Concept-Development-Phase“)<sup>156</sup> ausgeführt. Die Vorgehensweise ist in vier Schritte eingeteilt:

<sup>154</sup> Vgl. Hölttä et al. 2003, S. 7

<sup>155</sup> Ulrich/Eppinger 2008

<sup>156</sup> Vgl. Ulrich/Eppinger 2008, S. 14

- 1) Produktschema erstellen,
- 2) Elemente des Schemas gruppieren,
- 3) eine grobe Geometrie erzeugen,
- 4) fundamentale und nebensächliche Interaktionen identifizieren.

Zuerst wird das Schema (oder die Funktionsstruktur) des Produkts entwickelt. Alle Funktionselemente und die Interaktionen, die nach Energie, Material und Information untergliedert werden, werden im Schema eingetragen und zusammen verknüpft.

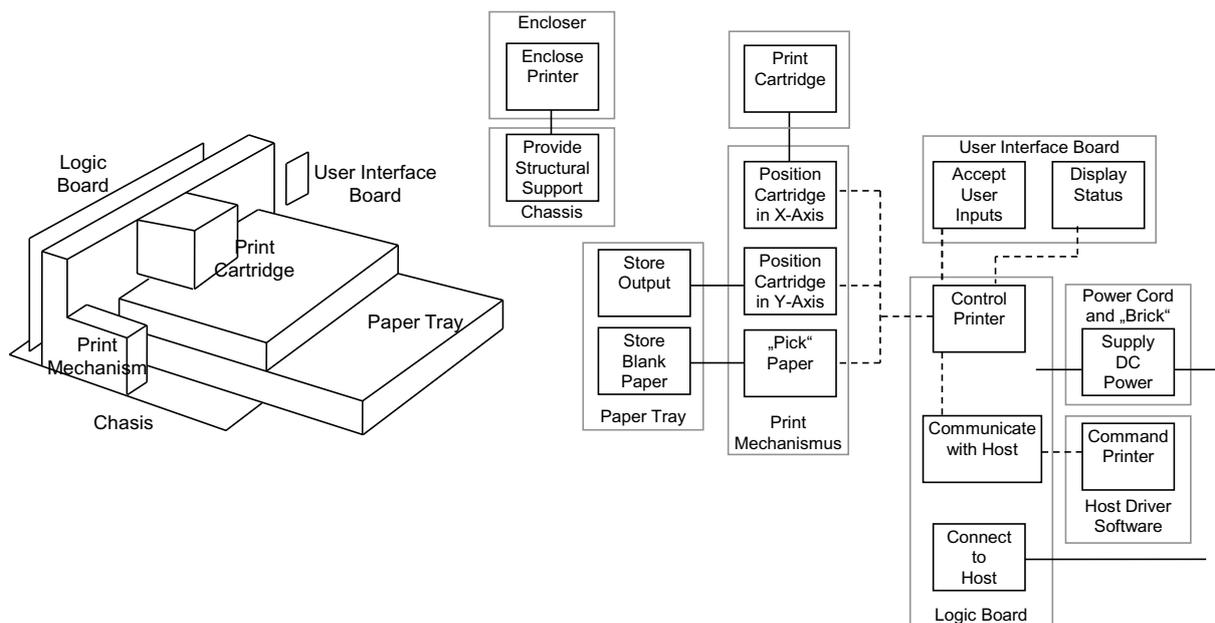


Abbildung 2-36. Beispiel für Chunks und Skizze eines Druckers<sup>157</sup>

Danach werden sie nach ULRICH und EPPINGER<sup>158</sup> – unter Berücksichtigung der folgenden Punkte gruppiert.

- Geometrische Integration
- Funktionsverteilung
- Zulieferer
- Ähnlichkeit des Designs und der Fertigungstechnik
- Lokalisation der Änderungsmöglichkeiten
- Produktvariationen
- Standardisierung
- Beweglichkeit der Schnittstellen

<sup>157</sup> Ulrich/Eppinger 2008, S. 174, 175

<sup>158</sup> Vgl. Ulrich/Eppinger 2008, S. 173 ff.

Der dritte Schritt folgt nach der Gruppierung. Eine Skizze (2D oder 3D) des Produkts wird entsprechend des Ergebnisses des letzten Schritts angefertigt. Sie beschreibt grob, wo alle Funktionselemente platziert werden und ihre ungefähre Gestalt. Letztendlich werden die Interaktionen zwischen den Gruppen identifiziert. Das Ziel dieses Schritts ist die Erleichterung der Kommunikation und des Informationsaustausches zwischen den Entwicklungsteams. Idealerweise müssen alle Teams schon früh im Systemlevel<sup>159</sup> (siehe Abbildung 2-5) verstehen, was für eine Interaktionen zwischen den Gruppen besteht. Das Ergebnis kann man in Form eines Schaltbilds oder in einer Matrix darstellen.

### 2.8.7 Zusammenfassung und Fazit zum Stand der Forschung

Eine Zusammenfassung und die wichtigsten Merkmale der ausgewählten relevanten Literatur werden in Tabelle 2-8 dargestellt.

Quelle	Werkzeuge	Quantitativer Ansatz	Funktions-/ Gestaltinform.- basiert	Für Neu-/Umkonstruktion
VDI-Richtlinie 2221	Checkliste, CAD, PM, OM, Daten- und Wissensbank	Nein	Gestaltinform.	Neu-/Umkonstruktion
STONE	Device-Funktion, Funktion-Funktion-Matrices	Ja, durch Bewertung der Kundenbedürfnisse	Funktionsinform.	Umkonstruktion
ALLEN und CARLSON-SKALAK	Fischgrätendiagramm, Funktionsstruktur, System-Funktions-Diagramm	Ja, Modularität berechnen	Funktionsinform.	Umkonstruktion
VAN WIE	Spatial Constraints, Function Layout, Physical Process, Product Family, Manufacturing, Partition Diagram	Nein	Funktions- und Gestaltinform.	Neu-/Umkonstruktion
HÖLTTÄ ET AL.	Dendrogramm	Ja, Distanz Matrix	Funktionsinform.	Neu-/Umkonstruktion
ULRICH und EPPINGER	Produktschema, grobe Geometrie	Nein	Funktions- und Gestaltinform.	Neu-/Umkonstruktion

Tabelle 2-8. Relevante Literatur für die Festlegung der Produktarchitektur zu frühen Zeitpunkten in Produktentwicklungsprojekten

<sup>159</sup> Ulrich/Eppinger 2008, S. 15

Die im Abschnitt 2.8.1 bis 2.8.6 vorgestellten Forschungsarbeiten haben sich zwar mit der Festlegung der Produktarchitektur früh im Produktentwicklungsprojekt beschäftigt, aber die Methoden und die zugehörigen Hilfswerkzeuge sind sehr unterschiedlich. Das macht den Stand der Forschung unübersichtlich und die dort vorgestellten Methoden sind schlecht durchgängig anwendbar. Im folgenden Diagramm (Abbildung 2-37) wird ein Fazit aus dem Stand der Forschung gezogen:

Die genannten Forschungsarbeiten werden in zwei Gruppen eingeteilt:

- Intuitive Ansätze, sowohl mit Funktionsinformationen, als auch dem Layout eines Systems (rechts unten Abbildung 2-37)
- Quantitative Ansätze auf der Grundlage einer Integrationsanalyse der Funktionsinformationen (links oben Abbildung 2-37)

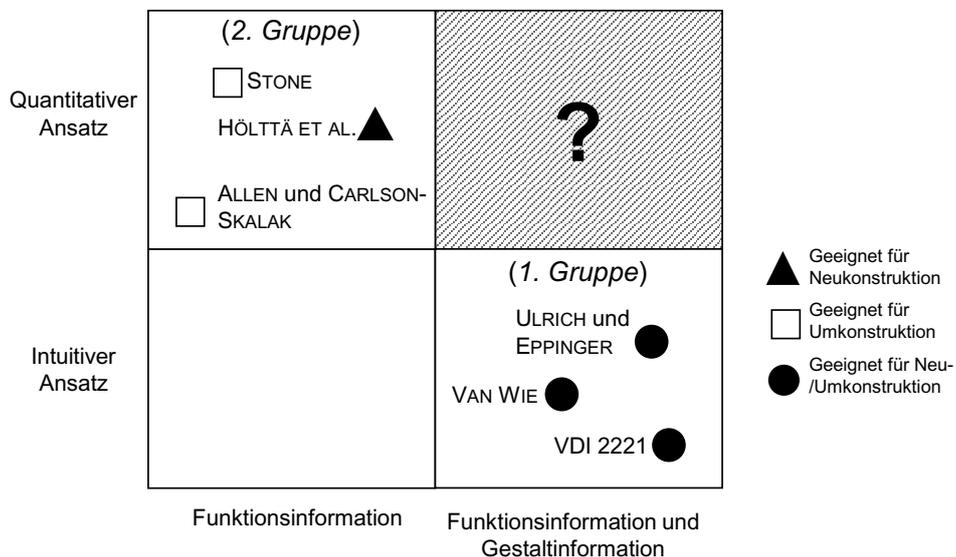


Abbildung 2-37. Klassifizierung der Literatur

Die Ansätze in der ersten Gruppe werden ohne numerische Berechnung angewendet. Die VDI-Richtlinie 2221 schlägt einen Schritt zur Entscheidung über die Produktarchitektur durch Gruppierung der Elemente der prinzipiellen Lösung vor. Verschiedene Aspekte des Produktlebenszyklus lassen sich betrachten, um die Produktarchitektur zu konfigurieren. Ebenso empfehlen ULRICH und EPPINGER eine Methode mit vier Schritten, um zur entscheidenden Produktarchitektur in der sogenannten Systemlevel-Phase<sup>160</sup> zu gelangen. Funktionselemente in der Funktionsstruktur sind in Chunks geclustert, aus denen die Module in der Architektur abgeleitet werden. VAN WIE betrachtet die Funktionsinformation zusammen mit vielen

<sup>160</sup> Vgl. Ulrich/Eppinger 2008, S. 15 ; die Systemlevel-Phase wird in Abbildung 2-2 dargestellt.

Arten von Diagrammen, die das System beschreiben und seine Produktarchitektur bestimmen. Aufgrund ihrer Einfachheit und der Nutzung sowohl der Funktionsinformation als auch der Gestaltinformation sind viele Vorteile daraus zu ziehen, wie zum Beispiel einfache Anwendbarkeit oder weniger Zeitaufwand; trotzdem sollten sie von erfahrenen Entwicklern angewendet werden.

In der zweiten Gruppe basieren die Methoden hauptsächlich auf einer quantitativen Integrationsanalyse der Funktionselemente in der Funktionsstruktur. Diese Gruppe enthält Ansätze von HÖLTTÄ ET AL., STONE und ALLEN und CARLSON-SKALAK. HÖLTTÄ ET AL. analysieren die Integration der Systemfunktionen mit einem Dendrogramm. STONE schlägt drei heuristische Methoden vor. Seine Methode basiert auf der Bewertung der Kundenwünsche zusammen mit einer Integrationsanalyse der Funktionen. In der gleichen Weise beschließen ALLEN und CARLSON-SKALAK die Produktarchitektur durch eine Analyse der Funktionsintegration sowie Produktarchitekturen von Erzeugnissen aus früheren Generationen und durch die Vorgaben der Entwicklungsteams. Anders als die erste Gruppe verwenden diese Methoden die Vorteile einer numerischen Suche nach der optimalen Lösung, die durch einen Vergleich der verschiedenen Architekturen numerisch bestimmt werden kann, auch wenn der Entwickler unerfahren ist.

## 2.9 Problemstellung

Sowohl klassische als auch moderne Produktentstehungsprozessmodelle und -methoden konzentrieren sich sehr wenig auf die Produktarchitektur. Auch in der Praxis erfolgt die Festlegung der Produktarchitektur meist durch eine intuitive Entscheidung. Um bei der Festlegung der Produktarchitektur eine gute Entscheidung treffen zu können, wird ein talentierter und erfahrener Entwickler benötigt. Jedoch ist es so besonders bei komplexen Systemen nicht immer möglich, zu einem optimalen Ergebnis zu kommen.

Die bereits entwickelten Methoden kann man in die zwei vorher genannten Kategorien einteilen. Sie sind bislang unabhängig voneinander und besitzen jeweils ihre eigenen Vorteile. Allerdings kann die Umsetzung der Ansätze in der ersten Gruppe Schwierigkeiten bereiten, insbesondere wenn viele Faktoren zu beachten sind oder wenn das Produkt noch unbekannt ist. Andererseits sind die Ansätze in der zweiten Gruppe nicht auf die Analyse der Gestaltinformation ausgerichtet. Dies kann zu einem Konflikt zwischen einem analysierten Ergebnis der Produktarchitektur und der entwickelten Gestalt führen. Zwei wesentliche Gründe, die die beiden zuvor erläuterten Schwierigkeiten verursachen, sind die folgenden:

- Gestaltinformation und Funktionsinformation sind getrennt oder nicht ausreichend verbunden.

- Die Gestaltinformation wird bei der Festlegung der Produktarchitektur, wenn überhaupt, nicht quantitativ berücksichtigt.

Es besteht also Forschungsbedarf in Bezug auf die Entwicklung einer neuen Methode, die beide Probleme eliminiert und die die Vorteile der beiden Gruppen der vorhandenen Ansätze miteinander verbindet.

Die in der vorliegenden Forschungsarbeit vorgestellte Methode kann im Vorgehen der VDI-Richtlinie 2223 als Ergänzung und konkrete Handlungsanleitung für den Arbeitsschritt 4.2 „Modularisieren der prinzipiellen Lösung“ (siehe Abbildung 2-38) eingeordnet werden.

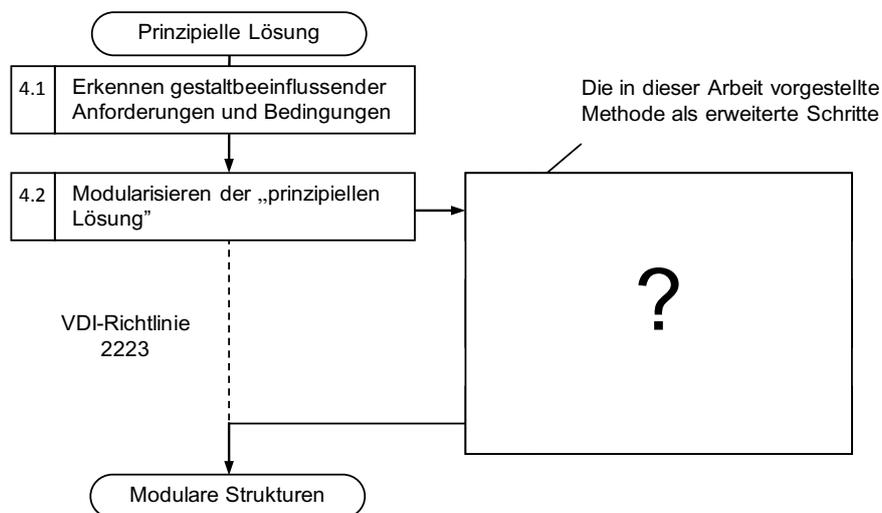


Abbildung 2-38. Die hier vorgestellte Methode als Erweiterung des 4. Schritts „Gliedern in realisierbare Module“ der VDI-Richtlinie 2223<sup>161</sup>

Bei der Suche nach prinzipiellen Lösungen spielt die Festlegung der Produktarchitektur noch keine Rolle, bzw. wird noch nicht bewusst bedacht. Die VDI-Richtlinie 2223 schlägt einen eigenen Schritt (Schritt 4 in Abbildung 2-29) für die Festlegung der Produktarchitektur vor. Dieser Schritt wird in zwei weitere Schritte „Erkennen gestaltbeeinflussender Anforderungen und Bedingungen“ und „Modularisieren der prinzipielle Lösung“ gegliedert, jedoch basieren diese Arbeitsschritte, wie schon im Kapitel 2.8.1 beschrieben, nur auf subjektiven Entscheidungen. In der vorliegenden Arbeit wird der Handlungsschritt „Modularisierung der prinzipiellen Lösung“ aus quantitativer Sicht behandelt. Im nächsten Abschnitt wird die hieraus abgeleitete Zielsetzung der Arbeit konkreter beschrieben.

<sup>161</sup> Vgl. VDI-Richtlinie 2223, S. 18

---

### 3 Zielsetzung der Arbeit

In der vorliegenden Arbeit werden Methoden und benötigte Werkzeuge zur frühzeitigen Festlegung der Produktarchitektur in Produktentwicklungsprojekten diskutiert. Die richtige Produktarchitektur ist ein wesentlicher Erfolgsfaktor eines Produktes und damit auch eines Unternehmens. In der Literatur sind aber nur wenige Methoden und Hilfswerkzeuge zur Unterstützung der Architekturfestlegung in der Entwicklung beschrieben. Insbesondere die wichtigen Zusammenhänge zwischen Funktion und Gestalt können mit den Methoden aus dem Stand der Forschung daher nicht sinnvoll und quantitativ berücksichtigt werden. Bis jetzt fokussieren alle bereits entwickelten Methoden in diesem Gebiet entweder auf einer funktionsbasierten Integrationsanalyse oder einer Systematisierung des intuitiven Vorgehens.

Ein notwendiges Werkzeug für die Produktarchitekturfestlegung ist ein Integrationsanalysewerkzeug. Bei der Integrationsanalyse handelt es sich um die Untersuchung des Grades der Integration der Grundelemente und der Interaktionen, die zwischen den Grundelementen auftreten. Die Ergebnisse der Integrationsanalyse sind die Gruppierungen der Grundelemente, welche das Zielsystem bestmöglich erfüllen und somit hilfreich für die Gestaltung des Produkts in den weiteren Arbeitsschritten sind.

Ziel dieser Forschungsarbeit ist die Entwicklung einer Methode und eines dazugehörigen rechnergestützten Werkzeugs, die sowohl Funktionsinformation als auch Gestaltinformation in quantitativer Art und Weise zur Festlegung der Produktarchitektur nutzen und somit eine transparente und effiziente Definition der Produktarchitektur ermöglichen.

Die Methode muss folgende Bedingungen erfüllen:

- Die Funktionsinformation und die Gestaltinformation müssen in der Methode gemeinsam und nicht isoliert voneinander betrachtet werden.
- Um die Varianten der Produktarchitekturen vergleichen zu können, müssen die Produktarchitekturen in der Methode hinsichtlich des Grades der Zielsystemerfüllung quantitativ bewertet werden können.
- Die Methode muss sowohl für Neukonstruktionen als auch für Variantenkonstruktionen und Anpassungskonstruktionen geeignet sein.
- Eine rechnergestützte Optimierung für die effiziente Festlegung von Produktarchitekturen muss gegeben sein.

---

## 4 Lösungsansatz

Der erste Teil dieses Abschnittes enthält eine Übersicht und ein Lösungsansatz, der die vorgestellten Zielsetzungen der Arbeit erfüllt. Das aus dem Ansatz abgeleitete Vorgehen wird im Kapitel 4.2 erklärt. Anschließend werden im Kapitel 4.3 die Umsetzungsmöglichkeiten und die Vorteile des Lösungsansatzes gegenüber einem herkömmlichen Ansatz dargelegt.

### 4.1 Lösungsansatz

Die in dieser Arbeit vorgestellte Methode zur Festlegung der Produktarchitektur ist relevant für mehrere Aktivitäten der Produktentstehung. Diese Methode eignet sich besonders für die Anwendung im Rahmen der Aktivität „Modellierung von Prinzip und Gestalt“. Weiterhin gibt es Verknüpfungen zu anderen Aktivitäten der Produktentstehung, insbesondere zur Aktivität Validierung, die nach ALBERS<sup>162</sup> die zentrale Aktivität des Produktentstehungsprozesses ist. Bei der Aktivität Validierung wird ein erreichter IST-Zustand (Objektsystem) mit einem geplanten SOLL-Zustand (Zielsystem) verglichen. Durch den kontinuierlichen und systematischen Vergleich beider Zustände können Handlungssysteme im Verlauf von Produktentstehungsprozessen geprüft und gegebenenfalls zielgerichtet angepasst werden.

Idealerweise wird die Methode eingesetzt, nachdem eine Produktidee ausgewählt wurde und eine Prinzipskizze erstellt und ausgewählt wurde. In dieser Prinzipskizze wird hauptsächlich die Gestalt des Produktes beschrieben, wobei die Funktionsinformationen des Produktes zu diesem Zeitpunkt bereits schon bekannt sein sollten, jedoch werden sie in der Regel nicht explizit beschrieben und dargestellt. Die Verknüpfung zwischen Gestaltbeschreibung und Funktion obliegt damit dem Betrachter, muss von ihm konstruiert werden und ist somit immer subjektiv und damit in der Regel für jeden Betrachter unterschiedlich. Dies stellt ein deutliches Risiko für Missverständnisse dar. Der geplante SOLL-Zustand, der nach der Durchführung der Methode vorliegen soll, ist ein Modell des Produktes, in dem mittels C&C-M die Gestalt mit der dazugehörigen Funktionsinformation verknüpft wird und gleichzeitig die optimale Produktarchitektur dargestellt wird. Dadurch wird deutlich, dass das Ziel der vorliegenden Arbeit darin besteht, eine Methode für die

---

<sup>162</sup> Vgl. Albers 2010, S. 5

Bestimmung einer optimalen Produktarchitektur bereit zu stellen sowie die für die effiziente Durchführung der Methode notwendigen Unterstützungswerkzeuge.

Wie bereits in Kapitel 2.9 beschrieben, müssen zwei wesentliche Probleme für die Erreichung der eben beschriebenen Zielsetzung gelöst werden. Dies sind zum einen die mangelnde integrierte Berücksichtigung von Gestaltinformation und Funktionsinformation und zum anderen die nicht-quantitative Berücksichtigung von Gestaltinformation bei der Festlegung von Produktarchitekturen.

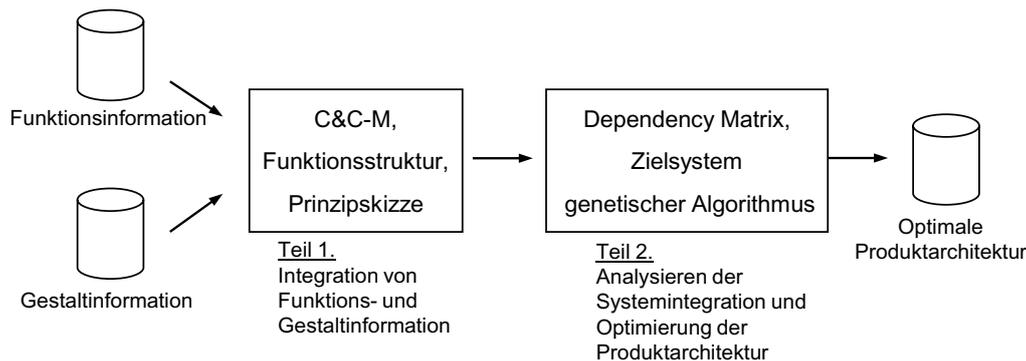


Abbildung 4-1. Schema der Methode<sup>163</sup>

Die Methode zur Lösung dieser Problemstellungen wird entsprechend der Darstellung in Abbildung 4-1 ebenso in zwei Teile unterteilt:

- 1) Integration von Funktionsinformation und Gestaltinformation und
- 2) Analyse der Systemintegration und Optimierung der Produktarchitektur.

Im ersten Teil werden die Modellelemente (Leitstützstrukturen und Wirkflächenpaare, siehe detailliertere Beschreibungen im Kapitel 2.5) und die Hypothese zur Modellierung des C&C-M<sup>164</sup> angewendet. Im zweiten Teil wird nach einer optimalen Lösung auf Basis der im ersten Teil generierten C&C-M Dependency-Matrix (CDM) gesucht. Ein Bewertungsvorgehen und ein genetischer Algorithmus werden in diesem Teil genutzt.

## 4.2 Methodisches Vorgehen

Das Lösungskonzept in Kapitel 4.1 befasst sich hauptsächlich mit der Strategie zur Lösung der Probleme, ohne eine detaillierte Vorgehensbeschreibung vorzugeben. In diesem Abschnitt wird der Weg für die Festlegung der Produktarchitektur Schritt für Schritt dargestellt.

<sup>163</sup> Die Definition „Prinzipskizze“ in Abbildung wird in Kapitel 5.1.1 beschrieben.

<sup>164</sup> Albers 2010, S. 6

Zwei Arten von Information, die üblicherweise sowohl in der Theorie als auch in der Praxis früh in einem Entwicklungsprojekt der Produktentwicklung zur Verfügung stehen, sind die *Anforderungen/Randbedingungen* (Teile des Zielsystems) und eine *Prinzipiskizze* (im Objektsystem, siehe Abschnitt 5.1.1). In vielen allgemeinen Entwicklungsmethoden wie zum Beispiel der VDI 2221, bei PAHL ET AL.<sup>165</sup> oder ULRICH und EPPINGER wird vorgeschlagen, zu Beginn eines Projektes die Anforderungsliste zu erstellen. Danach werden Produktideen generiert und eine davon für das weitere Vorgehen ausgewählt. Anschließend werden Prinzipiskizzen für die Umsetzung der Idee entwickelt, von denen eine, die am besten bewertete, ausgewählt und weiter ausgearbeitet wird.

Die Information aus dem Zielsystem und aus der Prinzipiskizze sind die wichtigsten Eingangsgrößen für die hier vorgestellte Methode. Das Vorgehen wird, wie in Abbildung 4-2 dargestellt, in drei Hauptschritte unterteilt: nämlich die Erstellung, die Bewertung und die Optimierung der C&C-M Dependency-Matrix.

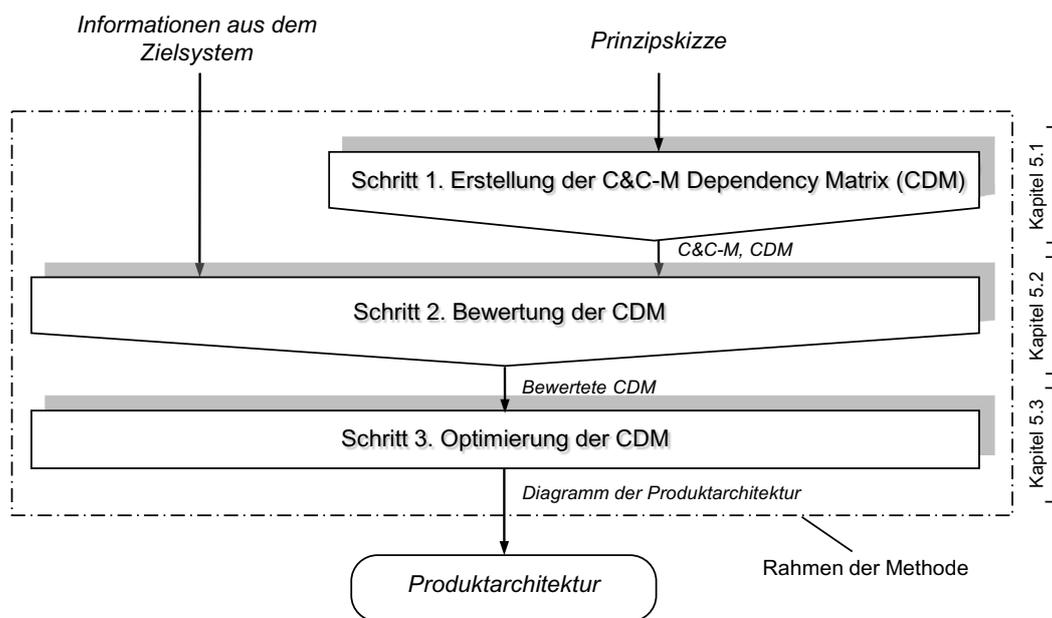


Abbildung 4-2. Methodisches Vorgehen der Produktarchitekturfestlegung

Im ersten Arbeitsschritt wird eine Funktionsstruktur oder eine Funktionsbeschreibung für eine Prinzipiskizze erstellt. Diese Funktionsbeschreibung und die Gestaltinformation (von der Prinzipiskizze) werden weiter durch die Elementmodelle des C&C-M miteinander verbunden. Das Ergebnis lässt sich in der C&C-M Dependency-Matrix darstellen. Die Matrix wird weiter im zweiten Schritt bewertet. Die

<sup>165</sup> Pahl et al. 2003, S. 204

Beziehungen der Modellelemente werden hinsichtlich Funktion und Kompatibilität quantitativ beurteilt. Schließlich wird die bewertete Matrix analysiert, um die beste Anordnung der Produktarchitektur zu finden. Die Ergebnismatrix lässt sich wiederum in der C&C-M Darstellung am Ende der Methode zeigen.

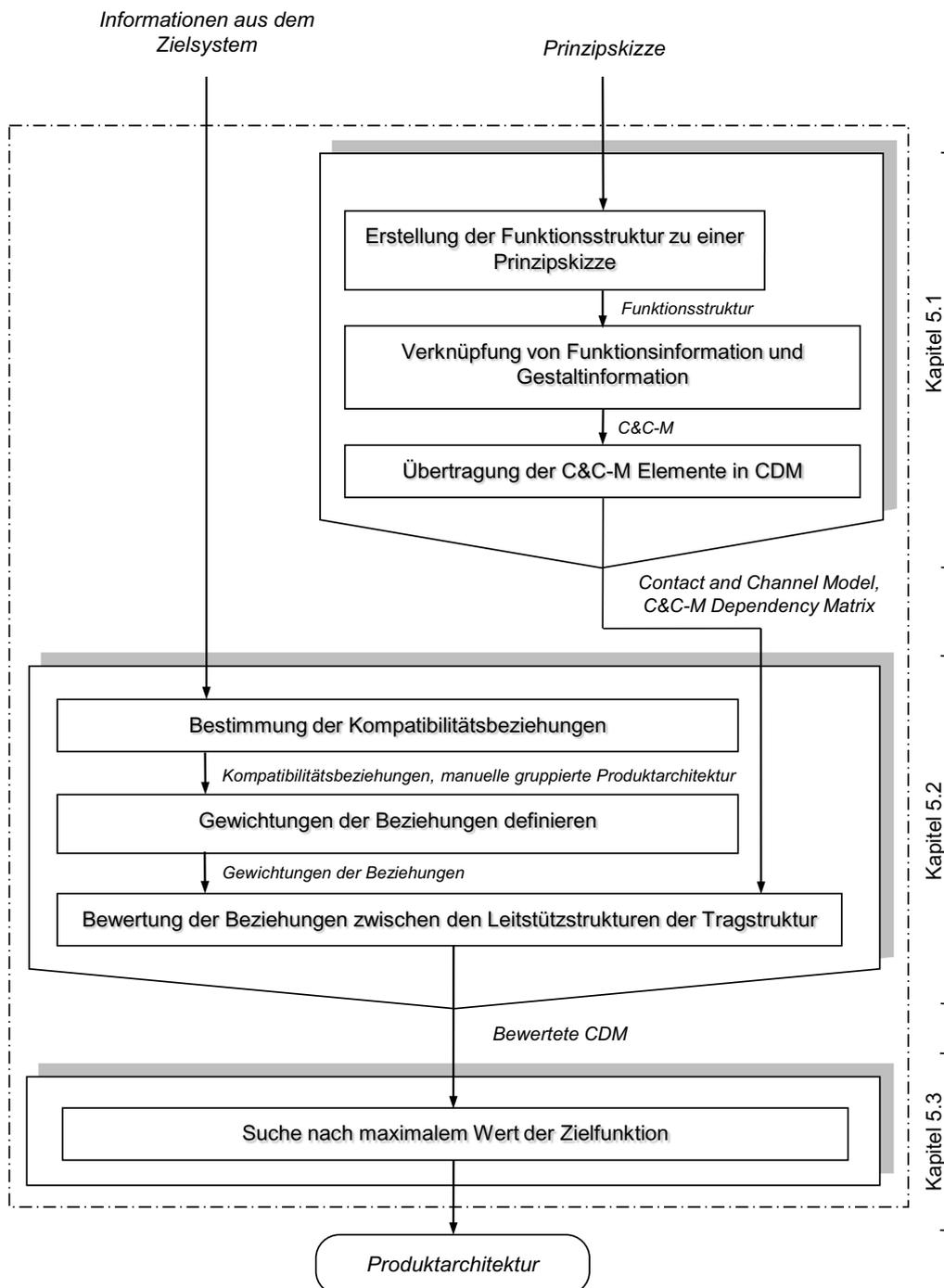


Abbildung 4-3. Vollständiges Ablaufdiagramm für die in dieser Arbeit entwickelte Methode

Ein vollständiges für den in dieser Arbeit entwickelten Ansatz wird in Abbildung 4-3 gezeigt, wobei die konkreten Informationen und die nötigen Beschreibungen aller drei Schritte in den Kapiteln 5.1, 5.2 und 5.3 schrittweise erklärt werden.

### 4.3 Methodeneinsatz für Umkonstruktion

Nicht nur für die Neu- sondern auch bei der Umkonstruktion lässt sich der Ansatz einsetzen. Die Anwendung herkömmlicher Umkonstruktionen (Anpassungskonstruktion und Variantenkonstruktion), bei denen die Produktarchitektur keine besondere Beachtung findet, führt in der Entwicklung komplexer Systeme zu Weiterentwicklungen, die meistens entweder ähnlich oder integraler sind als das vorhandene Produkt. Nach PAHL ET AL.<sup>166</sup> heißt die Anpassungskonstruktion eine Konstruktion bei bekannten und bewährten Lösungsprinzipien und passt die Gestaltung an veränderte Randbedingungen an. Bei Variantenkonstruktionen werden im Zuge der Auftragsabwicklung die Größe und/oder die Anordnung von Teilen und Baugruppen innerhalb von Grenzen vorausgedachter System variiert.

Die derzeitige Umkonstruktion fokussiert auf die Erfüllung der Produktfunktionen. Dagegen werden andere Anforderungen wie „Design für X“ (zum Beispiel fertigungsgerechte, montagegerechte Konstruktion) oder ein modularer Aufbau der Produktarchitektur, der viele weitere Vorteile verschaffen könnte, häufig nur als sekundäre Ziele berücksichtigt oder manchmal gar nicht beachtet.

Bei Umkonstruktionen stellen Entwickler eine bestehende Produktarchitektur oft nicht in Frage und gehen davon aus, dass diese nicht Gegenstand der Umkonstruktion ist sondern einfach beibehalten werden kann. Es wird nicht überprüft, ob eine bestehende Produktarchitektur bei Berücksichtigung der angepassten Ziele und Randbedingungen noch optimal ist oder angepasst werden sollte. Dies führt dazu, dass bestehende Produktarchitekturen bei Umkonstruktionen einfach übernommen werden. Möglicherweise ist eine bestehende Produktarchitektur nicht die beste Lösung, auch wenn sie geänderte Ziele und Randbedingungen erfüllt.

Solche Probleme haben keinen großen Einfluss auf einfache Systeme, die aus einer geringen Anzahl von Komponenten bestehen. In diesem Fall kann der Entwickler möglicherweise eine optimale Produktarchitektur durch seine Erfahrung konfigurieren. Jedoch ist die Umkonstruktion eines komplexen Systems etwas anderes. Sie hat viel mehr herausfordernde Funktionsanforderungen, viele Komponenten und schwierige Randbedingungen müssen berücksichtigt werden. In dieser Situation wird die erste Priorität auf die Funktionserfüllung unter den gegebenen Randbedingungen gelegt, wobei die Optimierung der Produktarchitektur oft mit sehr viel geringerer Priorität verfolgt oder übersehen wird. Unter diesen komplexen Bedingungen ist es nicht leicht für den Entwickler, über die Produktarchitektur nur durch Erfahrung zu entscheiden. Eine transparente und

---

<sup>166</sup> Pahl et al. 2003, S. 4

effiziente Unterstützung für die Entscheidung ist von wesentlicher Bedeutung. Der Mangel an Unterstützungswerkzeugen kann dazu führen, dass keine optimale Produktarchitektur erzielt wird, obwohl dies möglich wäre.

Die in dieser Arbeit entwickelte Methode beschreibt ein Vorgehen für die effiziente Festlegung einer optimalen Produktarchitektur (mit bestehenden Komponenten). Sowohl für Umkonstruktionen als auch für die Festlegung von Produktarchitekturen existieren bereits verschiedene Entwicklungsansätze. Die Methoden für beide Anwendungsbereiche werden meist separat ohne gegenseitige Berücksichtigung eingesetzt. Existierende Ansätze zur Produktarchitekturfestlegung werden erst spät im Entwicklungsprozess eingesetzt wenn die Gestaltdefinition bereits weit fortgeschritten ist. Nach der Produktarchitekturfestlegung muss die Gestalt an die neu definierte Produktarchitektur angepasst werden und einige Komponenten sind neu zu gestalten. Ein solcher Ablauf erfordert ein iteratives Vorgehen und verursacht somit unnötigen Zeit- und Konstruktionsaufwand.

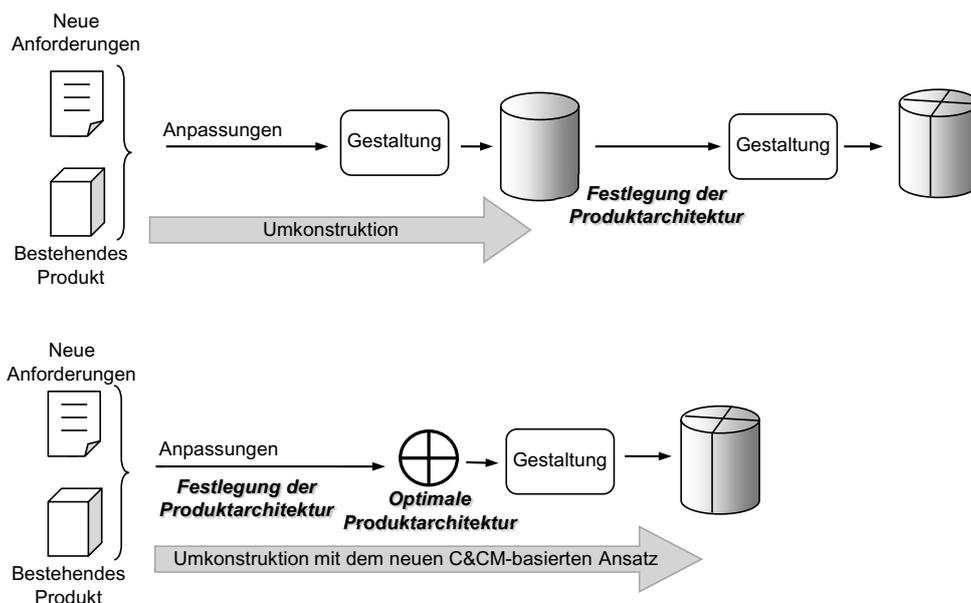


Abbildung 4-4. Vergleich zwischen herkömmlichem Vorgehen und dem neuen C&CM-basierten Ansatz zur Festlegung der Produktarchitektur bei Umkonstruktionen

In dieser vorliegenden Forschungsarbeit wird die Untersuchung einer Optimierung der Produktarchitektur früh im Produktentwicklungsprojekt verschoben. Der Ansatz besteht aus der Kombination von Methode und Werkzeug. Das ist schematisch in Abbildung 4-4 dargestellt. Die Kombination der beiden Methoden unterstützt den Entwickler dabei, eine Iteration bei Weiterentwicklungen zu vermeiden.<sup>167</sup>

<sup>167</sup> Vgl. Albers et al. 2008b, S. 3

---

## 5 Methodisches Vorgehen

In diesem Kapitel werden in drei Abschnitten die drei wesentlichen Schritte der Methode beschrieben. Zunächst wird die Erstellung des Contact and Channel Models (C&C-M) gezeigt. Danach werden die Vorgänge zum Bewerten der C&C-M Dependency-Matrix (CDM) beschrieben. In diesem Abschnitt wird das Zielsystem für die Architekturfestlegung erklärt, wobei die hierfür relevanten Ziele und Randbedingungen erörtert werden. Im Folgenden werden die Beziehungswerte zwischen allen Leitstützstrukturen also die Tragstruktur präsentiert. Anschließend wird ausführlich auf die CDM-Optimierung eingegangen, die durch einen genetischen Algorithmus erfolgt, und es wird ein Beispiel für die Bewertung der CDM dargestellt. Dazu werden alle drei Handlungsschritte durchgängigen in einem akademischen Beispiel veranschaulicht.

### 5.1 Erstellung des C&C-M und der C&C-M Dependency-Matrix

Im ersten Arbeitsschritt der in dieser Arbeit vorgestellten Methode zur Produktarchitekturfestlegung „Erstellung der C&C-M Dependency-Matrix“ wird eine Funktionsstruktur<sup>168</sup> oder eine Funktionsbeschreibung für eine Prinzipskizze erstellt. Hier geht man von der Skizzenebene auf die Funktionsebene. Dieses Wechseln der Betrachtungsebenen wird in Entwicklungsprozessen oft nicht bewusst und gezielt durchgeführt. Meist wird eine prinzipielle Lösung, z. B. in Form einer Prinzipskizze, weiter konkretisiert, und zwar z. B. durch die weitere Gestaltung in CAD-Systemen.

In diesem Schritt wird darauf abgezielt, ein C&C-M auf Basis der Prinzipskizze zu erstellen. Die Modellierung des C&C-M kann grundsätzlich wie folgt erfolgen: Der Ersteller analysiert das System und identifiziert alle möglichen WFP im System anhand einer Prinzipskizze. Die Positionen alle WFP werden dann in der Prinzipskizze markiert, z. B. mit den Symbolen in Tabelle 2-2. Die WFP werden weiter mit den LSS anhand der Modellierungshypothese (siehe Kapitel 2.5) in der Prinzipskizze verbunden. Um das C&C-M zu vervollständigen werden nach dieser Verknüpfung der Modellelemente Funktionsinformationen dazu explizit eingefügt, zum Beispiel in Form einer Tabelle mit einer sprachlichen Funktionsbeschreibung und einer eindeutigen Zuordnung zu den WFP, grafisch oder über eindeutige Benennung der WFP.

---

<sup>168</sup> Siehe Definition Funktion in Abschnitt 2.4.1

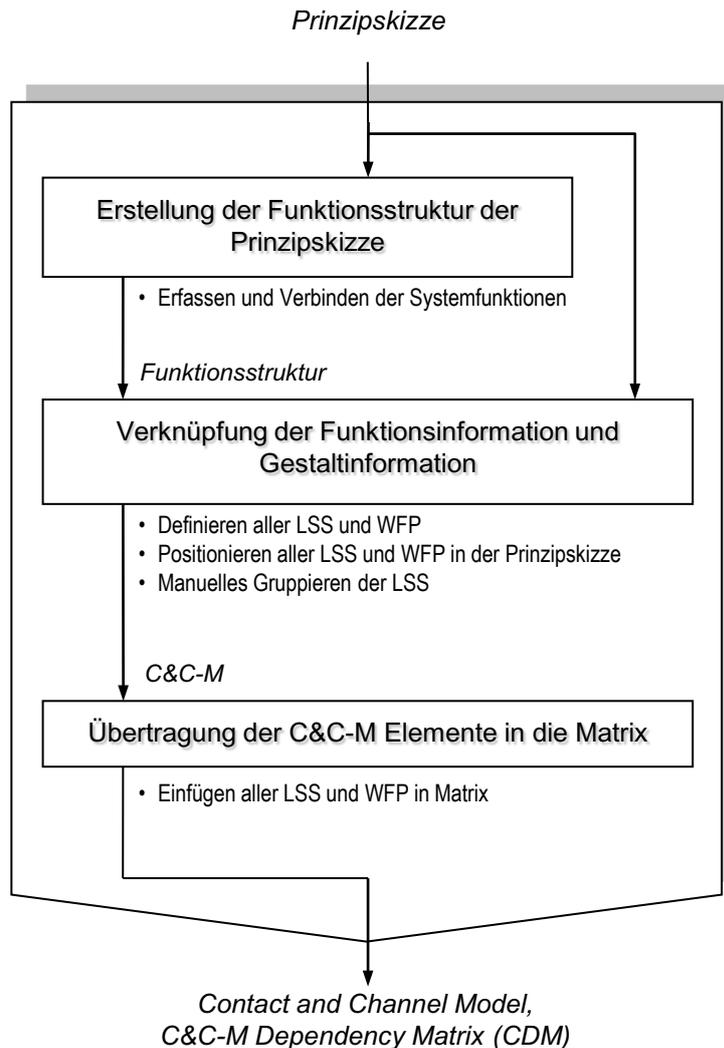


Abbildung 5-1. Vorgehen bei der Erstellung einer C&C-M Dependency-Matrix

Hierbei ist anzumerken, dass mit dieser Vorgehensweise die Funktionen oder Funktionsinformation des Systems nur intuitiv über das Systemverständnis des Modellerstellers in die Darstellung in der Prinzipiskizze einfließen, ohne sie zunächst vor der C&C-M Modellierung separat explizit zu beschreiben. Eine solche Vorgehensweise der Modellierung ist jedoch aus folgenden Gründen nicht immer problemlos möglich:

- Die Systemmodellierung mit C&C-M zeichnet sich durch große Flexibilität aus. Auf allen Abstraktionsniveaus kann ein System mit C&C-M durch WFP und LSS dargestellt werden. Es fällt manchmal schwer, zu entscheiden, mit welchem Abstraktionsgrad ein System mittels C&C-M modelliert werden sollte.
- Außerdem ist es aufgrund des C&C-M Darstellungsschemas nicht einfach möglich, ein komplexes System, z. B. einen hoch-integrierten Mechanismus, in einer 3-D-Skizze unmittelbar zu modellieren.

Bei der Wahl des Ausgangspunktes für die Modellerstellung für die C&C-M Dependency-Matrix spielen natürlich auch die individuellen Vorlieben der beteiligten Personen eine große Rolle. Je nachdem was die beteiligten Personen bevorzugen kann mit einer Prinzipskizze, einer Funktionsstruktur oder einer integrierten Abbildung von Funktion und Gestalt mittels C&C-M begonnen werden. Dies führt lediglich zu einem unterschiedlichen Startpunkt bei dem in Abbildung 5-1 dargestellten Vorgehen, führt aber zum gleichen Ergebnis.

### **5.1.1 Erstellung der Funktionsstruktur zu einer Prinzipskizze**

In der Funktionsstruktur des 2. Arbeitsschritts der VDI-Richtlinie 2221 werden die Grundfunktionen mit den Nebenfunktionen vermischt.<sup>169</sup> Obwohl diese Vorgehensweise den Vorteil einer vollständigen Funktionsdarstellung besitzt, kann sie zu einer sehr komplexen Funktionsstruktur führen, die möglicherweise sehr unübersichtlich wird. Außerdem lässt sich nach dieser Vermischung aufgrund der zu erwartenden Vielzahl von Interaktionen die Integrationsanalyse wahrscheinlich nicht effektiv durchführen.

Eine Möglichkeit, um ein komplexes Problem zu vereinfachen, ist die Problemdekomposition.<sup>170</sup> In der in dieser Arbeit entwickelten Methode wird eine Dekomposition in zwei Klassen von Funktionen angestrebt, die Grundfunktionen und die Nebenfunktionen. In dieser Arbeit werden zunächst die Grundfunktionen betrachtet. Dabei ergibt sich eine neue, vereinfachte Funktionsstruktur. Diese neue Funktionsstruktur ist für die Analyse besser geeignet.

Bei diesem Schritt liegt eine Schwierigkeit darin, dass entschieden werden muss, welche Funktionen zu den Haupt- und welche zu den Nebenfunktionen zählen. Diese Unterscheidung ist nicht immer leicht und eindeutig zu treffen. Bei einer Änderung der betrachteten Systemgrenzen können Nebenfunktionen zu Grundfunktionen werden und umgekehrt.<sup>171</sup>

Diese Entscheidung muss vom Konstrukteur getroffen werden. Hilfsfunktionen werden in dieser Arbeit bei der Festlegung der Produktarchitektur nicht berücksichtigt da sich diese oft erst aus der identifizierten Produktarchitektur ergeben. Funktionen die in allen zulässigen Produktarchitekturen vorhanden sein müssen, sind als Grundfunktionen zu deklarieren. Die Nebenfunktion wie beispielsweise Hilfsfunktionen werden bei der Produktarchitektur nicht berücksichtigt, da diese vor Festlegung der Produktarchitektur meist nicht sinnvoll definiert werden können.

---

<sup>169</sup> Vgl. VDI-Richtlinie 2221, S. 17 f.

<sup>170</sup> Vgl. Chen et al. 2005 S. 184

<sup>171</sup> Vgl. Pahl et al. 2003, S. 43

Für die Erstellung der Funktionsstruktur in diesem Schritt wird die Prinzipskizze als Eingangsinformation benötigt. Die hierfür angefertigte Prinzipskizze muss keine ausführlichen Informationen über Geometrie, Masse oder Werkstoffe beinhalten, jedoch sollten in ihn alle Informationen dargestellt werden, die notwendig sind um das Prinzip verständlich zu erklären. In diesem Fall sollten z. B. keine skizzierten Bauteile überdeckt werden. In dieser vorliegenden Arbeit wird der Begriff „Prinzipskizze“ wie folgend definiert.

Eine Prinzipskizze ist ein grafisches Modell, das ein System mit geringem Detaillierungsgrad und erhöhtem Abstraktionsgrad abbildet. Sie enthält nur die für das pragmatische Merkmal (Zweck) dieses Modells notwendige Detaillierung und den dafür notwendigen Konkretisierungsgrad der Gestalt. Prinzipskizzen können auch durch symbolische und/oder kurze sprachliche Beschreibungen ergänzt werden. Ziel der Prinzipskizze ist es mit geringer Detaillierungstiefe die Gestalt einer Lösung abzubilden, die eine bestimmte Funktion erfüllen soll. Die Prinzipskizze (ohne C&C-M) enthält meist keine Funktionsinformation. Die Verknüpfung zwischen dieser Gestaltbeschreibung und der Funktion obliegt dem Betrachter, muss von ihm konstruiert werden und ist somit immer subjektiv und damit in der Regel für jeden Betrachter unterschiedlich.

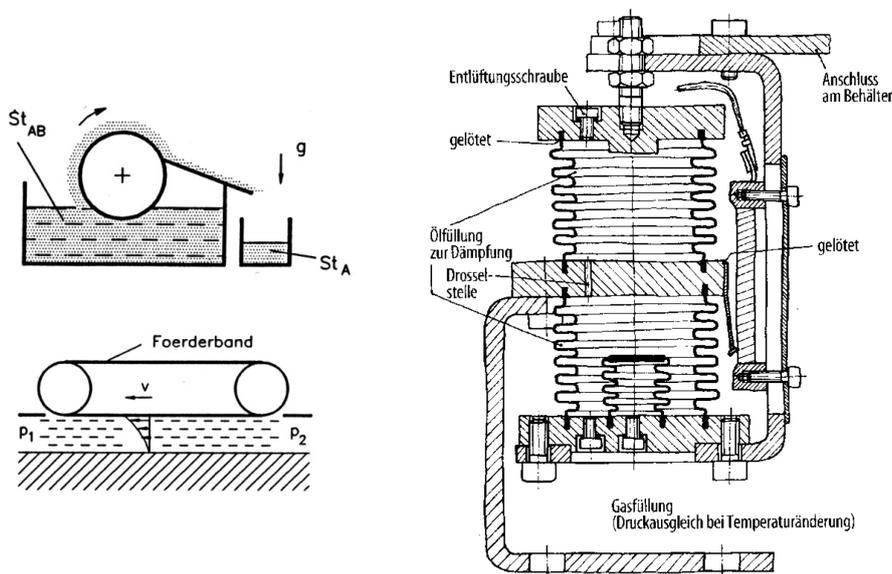


Abbildung 5-2. Beispiele für „Prinzipskizzen“ nach KOLLER<sup>172</sup> (links) und „Prinzipielle Lösung“ nach PAHL ET AL.<sup>173</sup> (rechts)

<sup>172</sup> Vgl. Koller/Kastrup 1994, S.

<sup>173</sup> Vgl. Pahl et al. 2003, S. 240

Eine „Prinzipskizze“ nach der oben beschriebenen Definition ist eine mögliche Abbildung einer „Prinzipiellen Lösung“ nach dem Verständnis von PAHL/BEITZ<sup>174</sup>. Allerdings haben andere Autoren den Begriff anders definiert. KOLLER und KASTRUP<sup>175</sup> benutzen die Prinzipskizze als eine abstrakte grafische Darstellung der Umsetzung/prinzipiellen Lösung für eine Elementarfunktion. Aus der Prinzipskizze ist der Ort der Erfüllung der Elementarfunktion nicht ersichtlich (siehe Beispiele in Abbildung 5-2). ULRICH und EPPINGER verwenden die Prinzipskizze (siehe Beispiel in Abbildung 2-36) als eine einfache Repräsentationen der groben Gestalt mittels mehrerer Quader um die Räumlichkeit der Systemkomponenten darzustellen. Um Missverständnisse aufgrund der unterschiedlichen Bedeutungen des Begriffs der „Prinzipskizze“ zu vermeiden, wird in den verbleibenden Abschnitten dieser Arbeit der Begriff entsprechend der oben angegebenen Definition verwendet. Beispiele für unterschiedliche Prinzipskizzen mit unterschiedlichen Abstraktionsgraden von einem Kreuzgelenk sind in Abbildung 5-3 dargestellt.

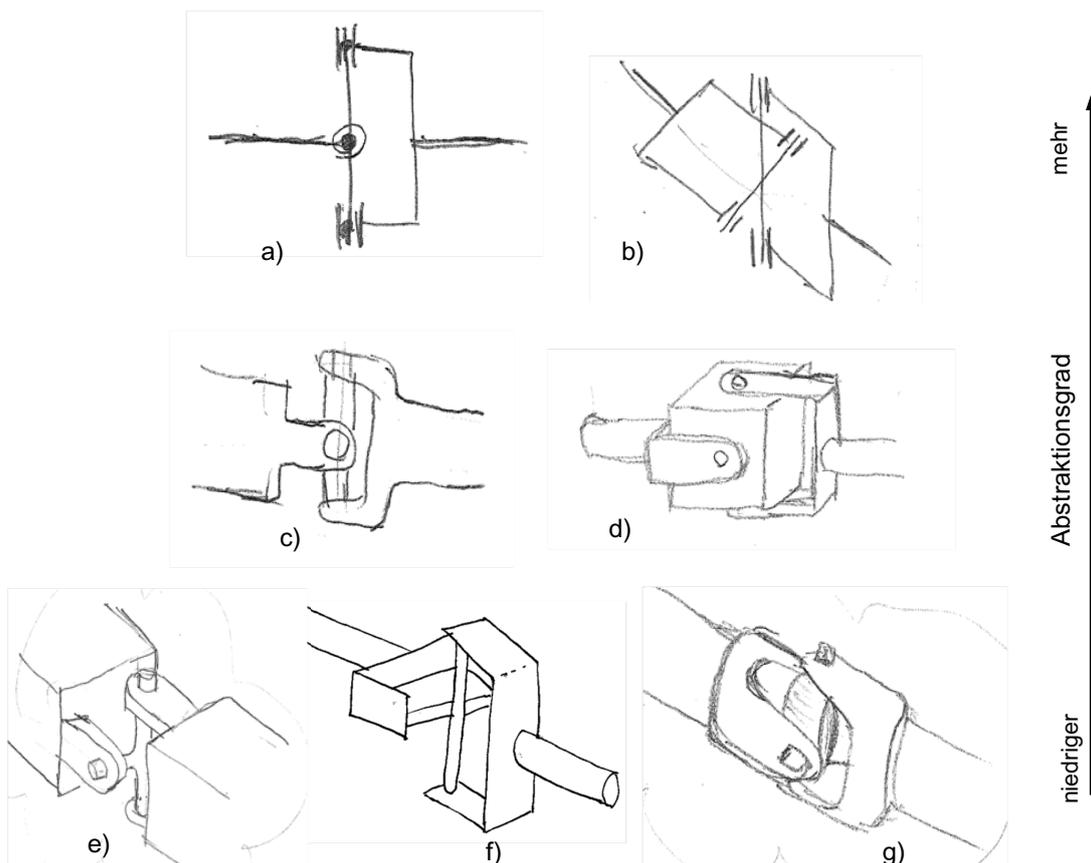


Abbildung 5-3. Unterschiedliche Prinzipskizzen für ein Kreuzgelenks

<sup>174</sup> Vgl. Pahl et al. 2003, S. 240

<sup>175</sup> Vgl. Koller/Kastrup 1994

**Erfassen und verbinden der Systemfunktionen:**

Das Abstraktionsniveau der Betrachtung für die Festlegung der Produktarchitektur wird über die Prinzipskizze definiert (Detail im Abschnitt 2.4) Die sehr detailliertere Funktionsformulierung in der Funktionsstruktur ist für das Systemverständnis des Entwicklers sehr hilfreich, jedoch ist sie in vorliegender Methode nicht notwendig, weil diese Funktionsstruktur nicht für die explizite Systembeschreibung erzeugt wird. Sie dient als ein Zwischenschritt, um ein C&C-M für die Produktarchitekturfestlegung zu erstellen.

Alle notwendigen Funktionen, die auf den Energie-, Stoff- und Informationsfluss einwirken, vom Eingang bis zum Ausgang des Systems, werden erfasst. Das kann in Form von einer Funktionsstruktur geschehen.

Alle vorher definierten Funktionen werden entsprechen der bestehenden Wechselwirkungen in der Darstellung der Funktionsstruktur miteinander verbunden. Es ergibt sich die Funktionsstruktur des Systems, die sich aus dem Betrachtungsniveau der Prinzipskizze definiert.

Beispiel zum Kapitel 5.1.1: Die Erstellung einer Funktionsstruktur für ein Kreuzgelenk wird hier als Beispiel dargestellt. Beim gewählten Betrachtungsniveau müssen zwei Grundfunktionen erfüllt werden. Es ist unmöglich, von dieser Struktur eine Produktarchitektur abzuleiten. Der Detaillierungsgrad der Funktionsstruktur ist hierfür zu gering.

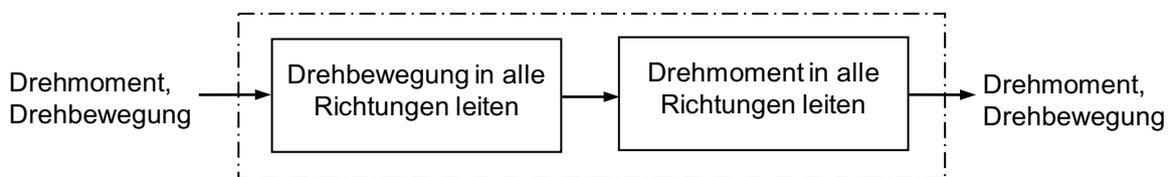


Abbildung 5-4. Funktionsstruktur des Kreuzgelenks

Aus diesem Grund wird eine neue Funktionsstruktur, die mehr Details beinhaltet, erstellt. Abbildung 5-5 zeigt eine ausgewählte Prinzipskizze mit der das Abstraktionsniveau der Funktionsstruktur für das weitere Vorgehen definiert wird.

Dieser Schritt beinhaltet die Erfassung und Verknüpfung der Funktionen aus allen Energie-, Stoff- und Informationsflusses, wobei in diesem Beispiel nur der Energiefluss dargestellt wird, da dieser aus funktionaler Sicht der in diesem Beispiel einzig relevante ist. Weitere Flussarten, wie z. B. der Informationsfluss beim Drehwinkel, spielen hier keine Rolle für die Produktarchitekturfestlegung.

Die Funktionen und ihre Verknüpfungen bilden eine Funktionsstruktur (Abbildung 5-6). Ihr Abstraktionsniveau wird vom Entwickler über den gewählten

Detailierungsgrad der Prinzipskizze definiert. Das heißt, falls das betrachtete System umfangreicher ist, kann es sein, dass dieses Gelenksystem als eine Teilfunktion des Gesamtsystems zu betrachten ist (siehe Kapitel 6.1).

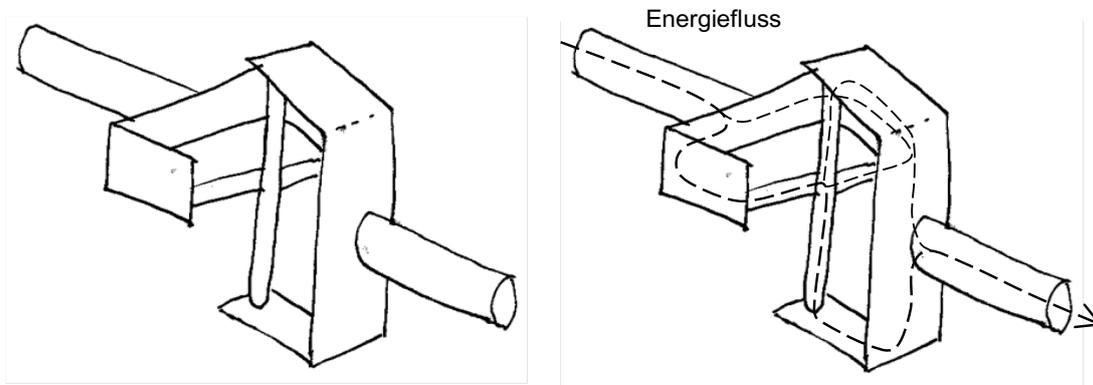


Abbildung 5-5. Ausgewählte Prinzipskizze und der Energiefluss

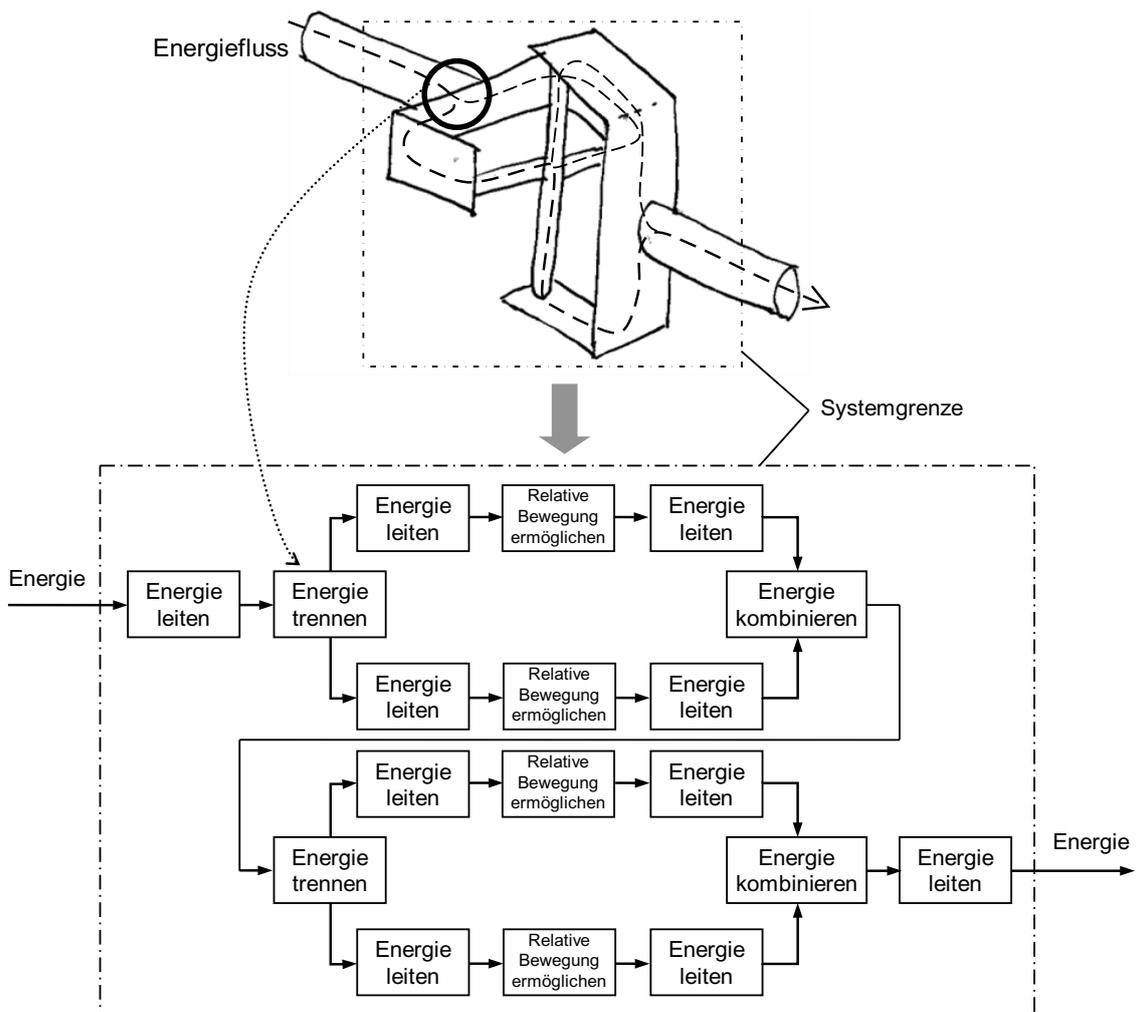


Abbildung 5-6. Beispiel für eine Funktionsstruktur eines Kreuzgelenks

### 5.1.2 Verknüpfung von Funktionsinformation und Gestaltinformation

#### ***Definieren aller LSS und WFP:***

Die LSS, die WFP und die Funktionen werden in diesem Schritt so nummeriert, dass eine eindeutige Zuordnung von C&C-M Modellelementen zu den Funktionen möglich ist. Erst dadurch entsteht eine sinnvolle C&C-M Modellierung die die Entwickler unterstützen kann. Das System wird hier von der Funktionsdomäne in die „Contact-and-Channel“-Domäne abgebildet. Dieser Schritt kann auch parallel zum folgenden Schritt durchgeführt werden. Ab dem nächsten Schritt werden hauptsächlich die „Contacts“ (WFP) und „Channels“ (LSS), die mit den Elementen von der Prinzipskizze verbunden werden, betrachtet.

#### ***Positionieren aller LSS und WFP in der Prinzipskizze:***

In die Prinzipskizze werden nun alle nummerierten LSS und WFP aufgenommen und einer konkreten oder generalisierten Fläche oder Volumen zugeordnet. Hiermit wird einer Funktion explizit und eindeutig die Gestalt zugeordnet, die die Funktion erfüllt. Üblicherweise werden hierfür Symbole verwendet wie sie in Tabelle 2-2 abgebildet sind. Andere Formen der Darstellung sind möglich und können an den jeweiligen Bedarf und Vorlieben der Anwender angepasst werden.

#### ***Manuelles Gruppieren der LSS :***

In diesem Schritt werden die LSS, die in einem Bauteil gruppiert werden, identifiziert und zusammengestellt. Dies gilt normalerweise für Standardbauteile, die viele Funktionen besitzen, oder für LSS, für die sich aus dem Zielsystem ergibt, dass sie zusammen gefertigt werden müssen. Hierbei handelt es sich also um eine manuelle Gruppierung und nicht um eine automatische Gruppierung als Ergebnis eines Optimierungsschritts. Hierbei muss darauf geachtet werden, dass nur die wichtigen, unlösbaren Gruppierungen bereits im Vorfeld der Optimierung vorgenommen werden. Dies vereinfacht nicht nur die Erstellung der Matrix in nächsten Schritt, sondern auch die Optimierung der Matrix.

Beispiel zum Kapitel 5.1.2: Nun werden einige LSS aus dem vorherigen Beispiel in Abbildung 5-6 manuell gruppiert. Die LSS, die durch eine einzige physische Komponente implementierbar sind, werden zu einer LSS zusammengefasst. Basierend auf der Bauteilherstellung wird in diesem Beispiel angenommen, dass die LSS für die Funktionen „Energie leiten“ und „Energie trennen“ aufgrund einer Vorgabe im Zielsystem unbedingt in dem U-förmigen Bauteil platziert werden sollen; ebenso sollen „Energie kombinieren“ und „Energie leiten“ in dem skizzierten Strang verwirklicht werden (siehe Abbildung 5-7). Daher reduziert sich die gesamte Anzahl der LSS nach der manuellen Gruppierung durch den Entwickler auf 10 (siehe Abbildung 5-8).

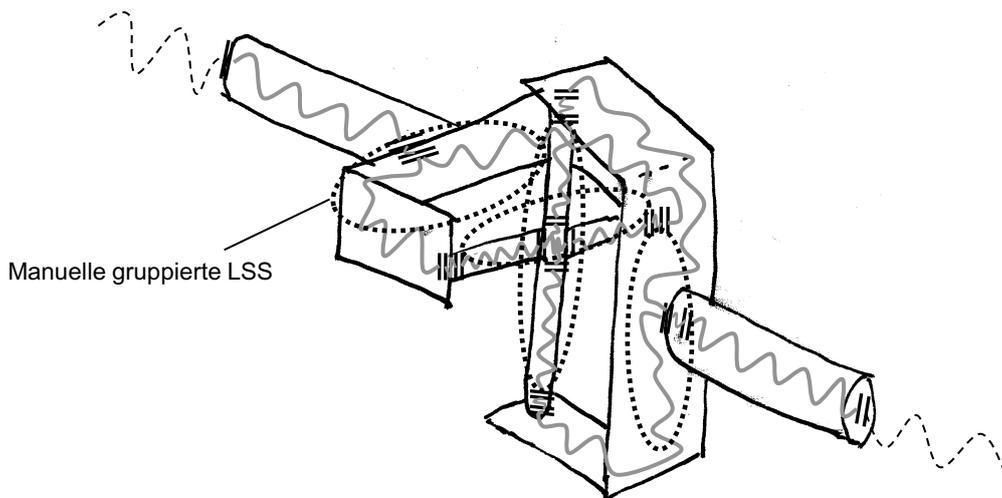


Abbildung 5-7. Integration der LSS und WFP in die Prinzipskizze

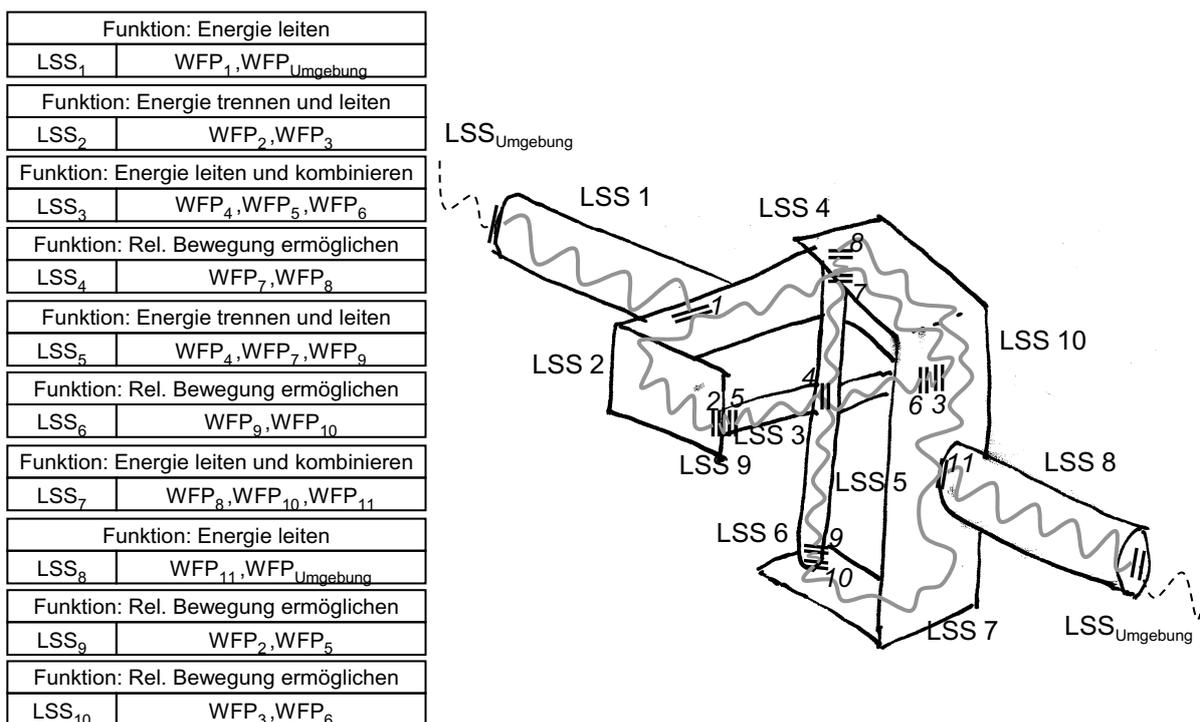


Abbildung 5-8. Das C&C-M des Kreuzgelenks

### 5.1.3 Übertragung der C&C-M Elemente in CDM

#### **Einfügen aller LSS und WFP in die Matrix:**

An einem komplexeren System sind oft viele LSS und WFP beteiligt. Daraus ergibt sich, dass solche Darstellungen leicht unübersichtlich werden und sind zusätzlich für die mathematische Analyse ungeeignet. Aus diesem Grund benötigt man noch eine andere Darstellungsform, die die Grundelemente und Verbindungen zeigt und sich

auch für die Integrationsanalyse eignet. Eine Matrixdarstellung ist in diesem Fall sehr geeignet und wird als Hauptdarstellung in dieser Methode verwendet.

In diesem Schritt werden daher alle LSS und WFP des C&C-M in eine Matrix übertragen. Die LSS werden sowohl als Spaltenparameter als auch als Zeilenparameter aufgelistet. Als Verknüpfungen zwischen den LSS werden die WFP in die entsprechenden Matrixfelder eingetragen (vergl. Abbildung 5-9). Zum Zeitpunkt der Matrixerstellung ist noch nicht endgültig klar welche LSS zusammen in einem Bauteil sein werden. Dies hängt vom Ergebnis der Analyse ab. Es ist möglich, dass nach der Analyse mehrere LSS zu einem Bauteil integriert werden wenn es die dazugehörigen Funktionen zulassen, z.B. ohne Relativbewegung. Da Bauteile, und damit die Tragstruktur, beim Erstellen der Matrix nicht feststehen, muss hier mittels LSS modelliert werden.

Beispiel zum Kapitel 5.1.3: Alle LSS werden in der Matrixkopfzeile und -kopfspalte aufgelistet. Die WFP werden dann als Dateneingabe in die Matrix eingefügt. Alle WFP sind nun eingetragen, jedoch sind sie noch nicht bewertet und nur z. B. mit „X“ gekennzeichnet, um zu zeigen, dass das Wirkflächenpaar existiert, nicht aber welche Bedeutung dieser funktionalen Beziehung zugeordnet wird.

	LSS 1	LSS 2	LSS 3	LSS 4	LSS 5	LSS 6	LSS 7	LSS 8	LSS 9	LSS 10
LSS 1										
LSS 2	X								X	X
LSS 3					X				X	X
LSS 4					X		X			
LSS 5			X	X		X				
LSS 6					X		X			
LSS 7				X		X		X		
LSS 8							X			
LSS 9		X	X							
LSS 10		X	X							

„X“ = Ein Wirkflächenpaar zwischen den Leitstützstrukturen LSS2 und LSS 10

Abbildung 5-9. Unbewertete CDM des Kreuzgelenks

## 5.2 Bewertung der C&C-M Dependency-Matrix

Eine C&C-M Dependency-Matrix (CDM) mit den Bewertungen der Beziehungen ist das Hauptanalysewerkzeug in dieser Methode. Alle Beziehungen der LSS in der CDM werden in Bezug auf die Wichtigkeit der funktionalen Beziehungen und der Kompatibilitätsbeziehungen bewertet.

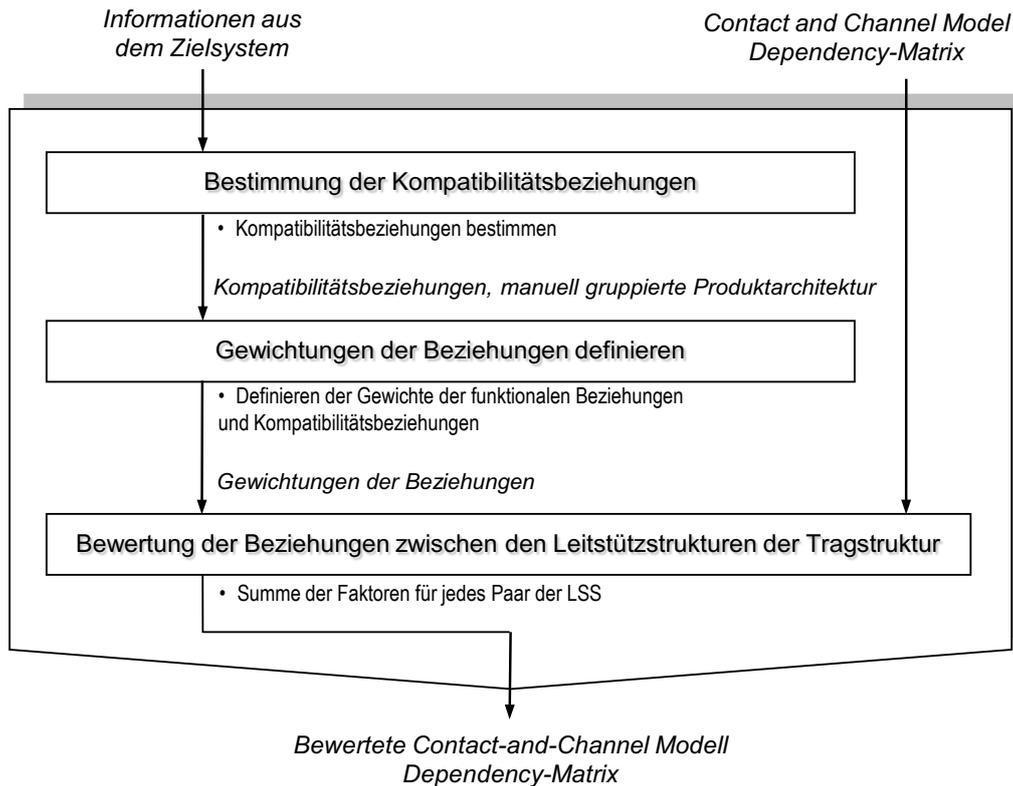


Abbildung 5-10. Vorgehen bei der Bewertung der Beziehungen der Leitstützstrukturen

### 5.2.1 Bestimmung der Kompatibilitätsbeziehungen

#### ***Kompatibilitätsbeziehungen bestimmen:***

Nach ALBERS<sup>176</sup> werden beim Systemansatz des iPeMs alle relevanten Ziele, deren Randbedingungen, Abhängigkeiten und Wechselwirkungen im Zielsystem beschrieben. Die Anforderungsliste, in der die Produktmerkmale und Produkteigenschaften gesammelt werden, ist eine Möglichkeit für die Dokumentation von Teilen des Zielsystems bei einem Entwicklungsprojekt. Diese ist allerdings nicht in der Lage, alle Aspekte des Zielsystems abzubilden.

Ein erstes Zielsystem muss zu Beginn eines Entwicklungsprojektes vorhanden sein oder erstellt werden und im Verlauf eines Projektes kontinuierlich erweitert und aktualisiert werden. Wird kein Zielsystem explizit modelliert ist es zumindest üblich eine Anforderungsliste (oder Lasten- und Pflichtenhefte) für die Steuerung von Entwicklungsprojekten zu nutzen.

<sup>176</sup> Vgl. Albers 2010, S. 5

In der VDI-Richtlinie 2221<sup>177</sup> beispielweise besteht der erste Schritt im Erstellen der Anforderungsliste, wobei alle Anforderungen der vom Kunden oder von der Produktplanung formulierten Aufgabe zu klären und zu präzisieren sind. Diese Liste ist für alle folgenden Arbeitsabschnitte eine begleitende und stets auf dem aktuellen Stand zu haltende Informationsgrundlage. PAHL ET AL.<sup>178</sup> stellen eine Leitlinie zur Ergänzung der Anforderungen vor. In dieser Leitlinie sollten folgende Hauptmerkmale in der Anforderungsliste enthalten sein; Geometrie, Kinematik, Kräfte, Energie, Stoff, Information, Sicherheit, Ergonomie, Fertigung, Kontrolle, Montage, Transport, Gebrauch, Instandhaltung, Recycling, Kosten und Termin.

Zur Analyse und Synthese des Systems mit Hilfe des C&C-M werden hauptsächlich die funktionalen Beziehungen zwischen den C&C-M Grundelementen untersucht. Wie bereits im Kapitel 2.3 dargestellt, sollten andere Beziehungsarten zwischen LSS, die nicht direkt aus einer zur Nutzung Funktion entsteht, ebenfalls bei der Festlegung der Produktarchitektur berücksichtigt werden können. Aus diesem Grund ist es notwendig zur Festlegung der Produktarchitektur anhand des C&C-M noch eine andere nicht-funktionale Beziehungsart zwischen C&C-M Grundelementen, die Kompatibilitätsbeziehung, zu definieren.

Die Kompatibilitätsbeziehung ist eine nicht-funktionale Beziehung zwischen zwei Leitstützstrukturen (LSS), also zwei abstrakten Konstrukten zur Leitung von Material, Energie und Information. Kompatibilitätsbeziehungen werden aus dem Zielsystem abgeleitet. Mit Hilfe der Kompatibilitätsbeziehungen kann ein Entwickler angeben, welche Paare von LSS aufgrund ihrer Eigenschaften und der Zielvorgaben aus dem Zielsystem zusammen in einem Modul angeordnet sein sollten. Hierbei werden jeweils immer nur zwei LSS betrachtet, es werden aber keine konkreten Module auf Basis der Kompatibilität alleine definiert.

Die Kompatibilitätsbeziehung ergibt sich aus den Zielvorgaben für die Produktarchitekturfestlegung im Zielsystem. Sie gibt an, wie kompatibel zwei LSS miteinander sind, die in einem Modul gruppiert werden sollen.

Die Kompatibilitätsbeziehung wird z. B. aus den oben genannten Produktmerkmalen in der Anforderungsliste als Teil des Zielsystems bestimmt. Aufgrund der vielseitigen Verknüpfungen der Produktarchitektur mit verschiedenen Domänen im Unternehmen

---

<sup>177</sup> Vgl. VDI-Richtlinie 2221, S. 9 f.

<sup>178</sup> Vgl. Pahl et al. 2003, S. 193 f.

(vgl. Abbildung 1-2) wird die Produktarchitektur durch viele Produktmerkmale im Zielsystem beeinflusst.

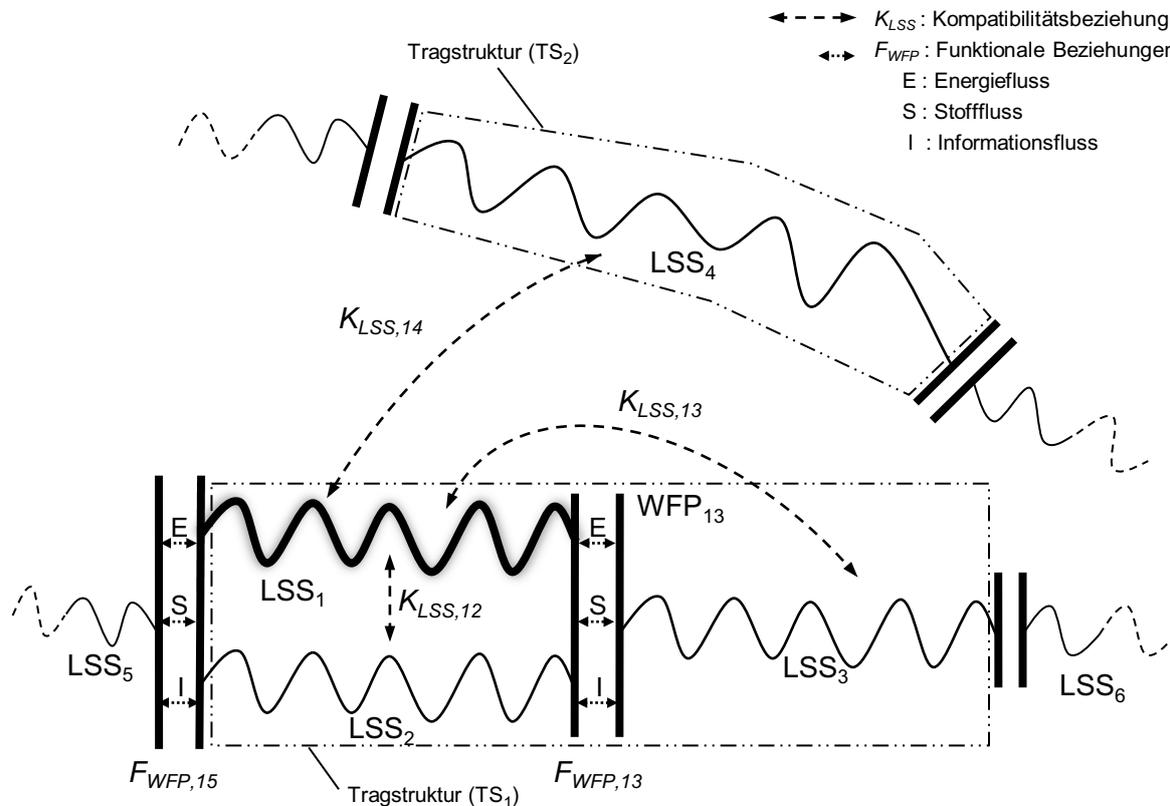


Abbildung 5-11. Funktionale Beziehungen und Kompatibilitätsbeziehung zwischen  $LSS_1$  und anderen LSS in einem System<sup>179</sup>

Diese Kompatibilitätsbeziehungen ( $K_{LSS,12}$  und  $K_{LSS,13}$  in Abbildung 5-11) zwischen zwei LSS können sowohl innerhalb einer Tragstruktur, als auch ( $K_{LSS,14}$  in Abbildung 5-11) zwischen zwei unterschiedlichen Tragstrukturen betrachtet werden.

Eine Tragstruktur kann aus einer einzigen LSS, mehreren LSS oder vielen LSS bestehen, die nicht alle gleichzeitig existieren müssen. Eine Tragstruktur, die aus einer Vielzahl von LSS besteht, wie sie z. B. durch Topologieoptimierung unter Berücksichtigung dynamischer Belastungen entsteht, eignet sich nicht immer für die Fertigung in einem Einzelbauteil. Hierfür kann es verschiedene Gründe geben, z. B. Herstellbarkeit, Vereinfachung des Herstellungsprozesses oder Zugänglichkeit der Tragstruktur. Um so eine komplexe Tragstruktur in mehreren Gruppen von LSS aufzuteilen, sind die Kompatibilitätsbeziehungen zwischen den LSS zu berücksichtigen (siehe Anwendungsbeispiel im Abschnitt 6.3).

<sup>179</sup> Um die Abbildung anschaulich darzustellen werden nur die Beziehungen zwischen der  $LSS_1$  und anderen LSS in dieser Abbildung dargestellt. Andere existierende Beziehungen werden in dieser Abbildung nicht angezeigt.

Ein anderes Beispiel für eine Kompatibilitätsbeziehung soll anhand ein Systems zu Datenübertragung über das Hausstromnetz (Homeplug AV / Powerline Communication) veranschaulicht werden. Hier muss die Kompatibilität zwischen einer LSS zur Übertragung von Strom und einer LSS zur Übertragung von Information über ein Hausstromnetz sicher gestellt werden. Diese beiden LSS können in einem Modul oder in einem Bauteil integriert werden solange sie kompatibel zueinander sind. Problematisch wird es beispielsweise wenn ein Trenntrafo in die Stromleitung integriert wird, Dann ist die LSS für die Energieübertragung nicht mehr kompatibel mit der LSS für die Informationsübertragung und beide LSS können nicht mehr integriert werden.

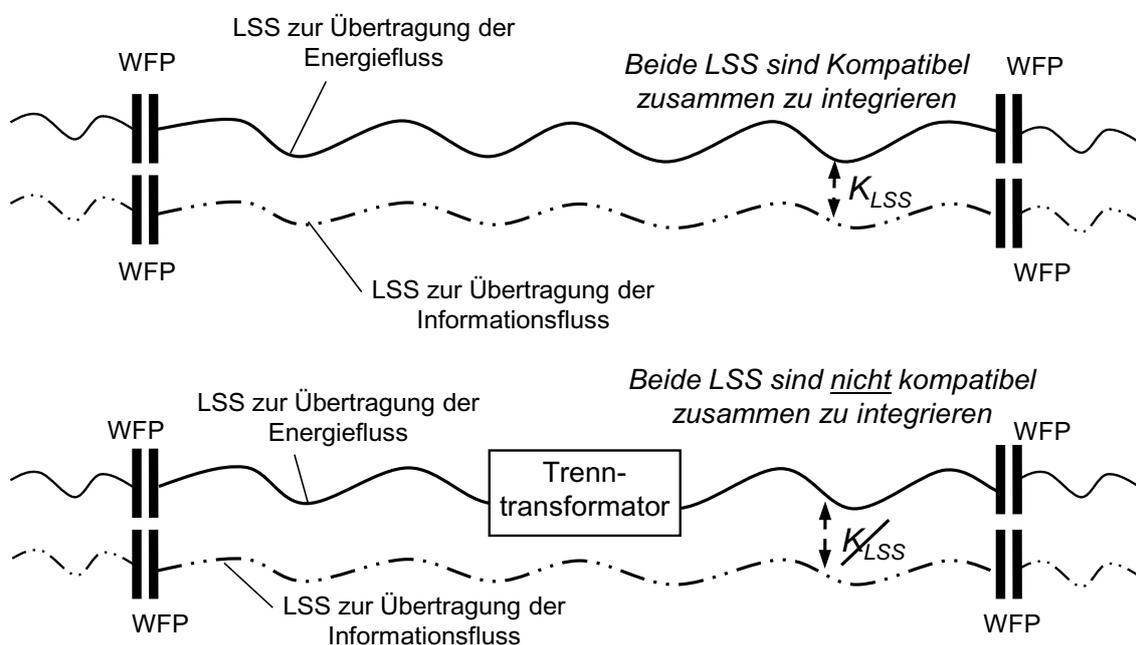


Abbildung 5-12. Beispiel der Kompatibilitätsbeziehung zwischen LSS zur Übertragung von Energiefluss und Informationsfluss

Zwei Leitstützstrukturen können beispielsweise auch eine Kompatibilitätsbeziehung haben, wenn sie vorhersehbar aus identischen Materialien bestehen, einen ähnlichen Produktionsprozess und eine ähnliche Lebensdauer aufweisen sollen.

Die genaue Anzahl der Kompatibilitätsbeziehungen und der Module kann nicht pauschal genau angegeben werden, sie hängt vom jeweiligen System ab. Das Ergebnis dieses Schritts kann in Form einer Auslistung der Kompatibilitätsbeziehungen und der davon betroffenen LSS visualisiert werden. Die Gewichtungen für die Kompatibilitätsbeziehungen im Vergleich zu den funktionalen Beziehungen werden im nächsten Schritt definiert.

Beispiel zum Kapitel 5.2.1: In diesem Kapitel wird das Beispiel aus Kapitel 5.1.3 fortgeführt. Jetzt werden Produktmerkmale im Zielsystem betrachtet und ausgewählt, um die Kompatibilitätsbeziehungen zu bestimmen. Zwei angenommene Ziele, von

denen die Kompatibilitätsbeziehungen abgeleitet werden, sind „einfach abzumontieren“ und „möglichst wenige Einzelteile“. Das Gelenk soll einfach zu montieren sein (Ziel A) und aus möglichst wenigen Einzelteilen bestehen (Ziel B). Daraus entstehen die nicht-funktionale Beziehungen zwischen den LSS. Danach werden die Gewichtungen der Kompatibilität mit funktionalen Beziehungen verglichen.

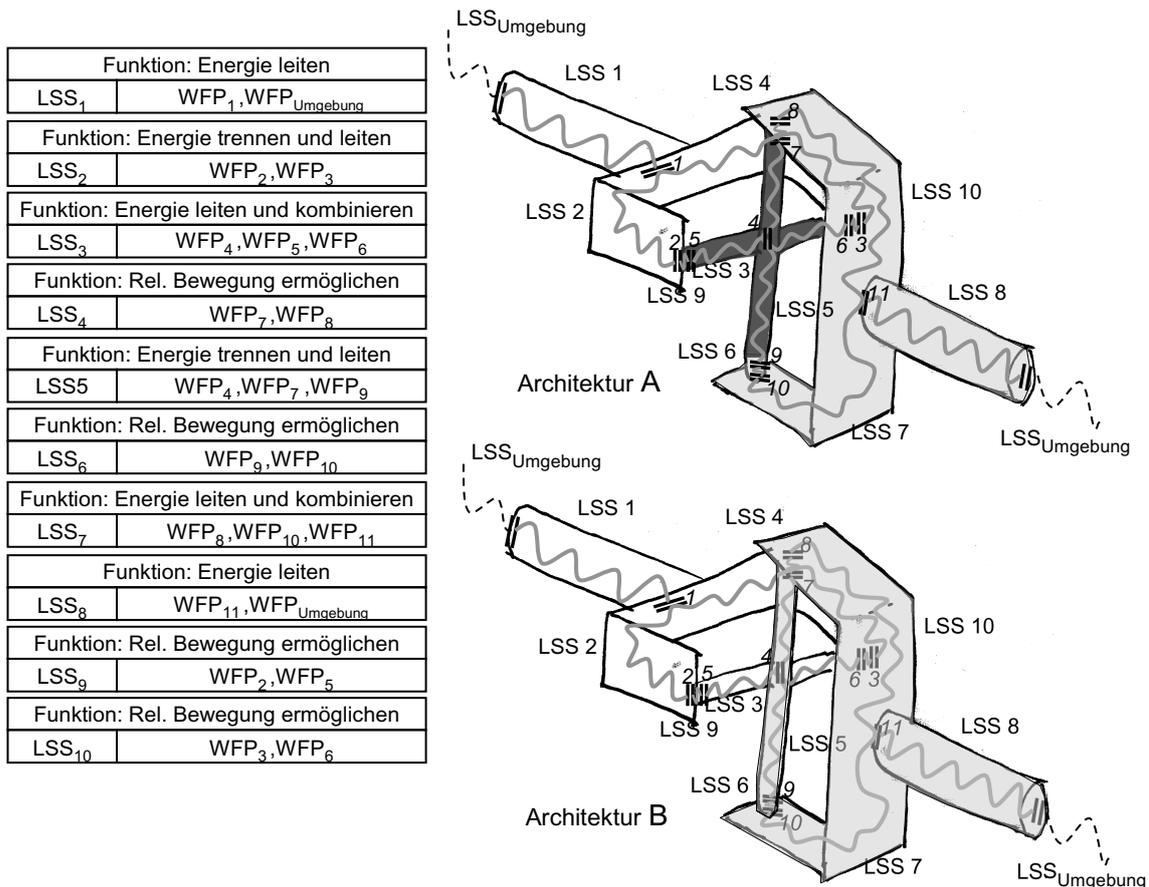


Abbildung 5-13. Beispiel der zwei manuellen Gruppierungen der LSS

### 5.2.2 Gewichtungen der Beziehungen definieren

#### **Definieren der Gewichtungen der funktionalen Beziehungen und der Kompatibilitätsbeziehungen:**

Zwei Arten von Beziehungen zwischen Leitstützstrukturen, die in der hier vorliegenden Methode betrachtet werden, sind die funktionalen Beziehungen und die Kompatibilitätsbeziehungen. Die funktionalen Beziehungen sind direkte Beziehungen zwischen Leitstützstrukturen, die durch die Grundfunktionen verursacht werden. Diese Beziehungen sind für das System notwendig und können nicht vermieden werden. Diese Beziehungen kann man in drei Typen unterteilen, je nachdem, ob sie vom Energie-, Stoff- oder Informationsfluss abgeleitet werden. Nur durch die Wirkflächenpaare werden die Leitstützstrukturen verknüpft (siehe Abbildung 5-11).

Die Kompatibilitätsbeziehungen leiten sich nicht von den Grundfunktionen ab; sie lassen sich aus der Zielvorgabe der Produktarchitekturfestlegung im Zielsystem ermitteln. Diese Beziehung entsteht durch die Eigenschaften der Leitstützstrukturen.

Jedes Ziel und jede funktionale Wirkung wird nun miteinander verglichen. Der Entwickler entscheidet in diesem Schritt, wie wichtig die funktionalen Beziehungen im Vergleich zu den Kompatibilitätsbeziehungen sind.

Jeder Beziehung lässt sich entsprechen der ihr zugewiesenen Wichtigkeit ein Faktor zu ordnen. Ein beispielhaftes Bewertungsschema wird in Tabelle 5-1 gezeigt.

Wichtigkeiten	Faktor
nicht wichtig	0
wenig wichtig	1
wichtig	2
sehr wichtig	3

Tabelle 5-1. Bewertungsschema der Beziehungen

Die Gewichtung jeder Beziehung lässt sich durch die Gleichung (5.1) berechnen, wobei die Summe aller Gewichtungen in diesem Fall gleich 1 gesetzt wird.

$$\text{Gewichtung der Beziehung} = \frac{\text{Wichtigkeitsfaktor der Beziehung}}{\text{Summe der Faktoren aller Beziehungen}} \quad (5.1)$$

Diese Verteilung der Gewichtung (durch die Entscheidung der Wichtigkeit) muss sorgfältig bestimmt werden, weil sie einen großen Einfluss auf das Ergebnis der Architektursuche hat. Der Einfluss der Gewichtung auf das Suchergebnis wird im Anwendungsbeispiel in Kapitel 6.3 gezeigt.

Beispiel zum Kapitel 5.2.2: Es wird in diesem Beispiel angenommen, dass der Entwickler die funktionalen Beziehungen als *sehr wichtig* (entspricht Faktor 3 in Tabelle 5-1) einschätzt. Die weiteren zwei Kompatibilitätsbeziehungen, die aus dem Zielsystem abgeleitet werden, sind *wichtige* Beziehungen (entspricht Faktor 2 in Tabelle 5-1). Alle Beziehungen der Leitstützstrukturen werden wie in der Gleichung (5.1) bewertet. Der maximale Wert jeder  $B_{LSS}$  wird in diesem Ansatz gleich 1 gesetzt. Die Konfigurationen der Architektur bezüglich der Ziele A und B werden vom Entwickler entschieden, wie in folgender Tabelle gezeigt.

	Gewichtung der Beziehung ( $F_{WFP_{ij}}$ und $K_{LSS_{ij}}$ )	LSS in Modul 1	LSS In Modul 2	LSS in Modul 3
Funktionale Interaktionen (Faktor 3)	$F_{WFP_{ij}} = \frac{3}{3+2+2} \approx 0,4$			
Einfach abzumontieren (Ziel A) (Faktor 2)	$K_{LSS_{ijA}} = \frac{2}{3+2+2} \approx 0,3$	LSS3, LSS5	LSS4, LSS6, LSS7, LSS8	LSS1, LSS2, LSS9, LSS10
Wenige Einzelteile (Ziel B) (Faktor 2)	$K_{LSS_{ijB}} = \frac{2}{3+2+2} \approx 0,3$	LSS4, LSS5, LSS6, LSS7, LSS8	LSS1, LSS2, LSS3, LSS9, LSS10	-

Tabelle 5-2. Die Anordnungen der LSS für Ziel A und Ziel B

### 5.2.3 Bewertung der Beziehungen der Leitstützstrukturen der Tragstruktur

#### Summe der Faktoren für jedes Paar der LSS:

Zwei wesentliche Berechnungen für den Ansatz sind die Bewertungen der „Beziehungen zwischen LSS“ ( $B_{LSS}$ ) und die Ermittlung des Zielfunktionswertes (Z). Für die Bewertung jeder Interaktion zwischen LSS werden zwei Parametersätze benötigt. Das sind die „funktionale Beziehung der Wirkflächenpaare“ ( $F_{WFP}$ ) und die „Kompatibilitätsbeziehung der Leitstützstruktur“ ( $K_{LSS}$ ).

Die Bewertungen kann man in folgender Gleichung (5.2) darstellen. Die  $B_{LSS_{ij}}$  ist eine Summe der beiden Parameter „ $F_{WFP_{ij}}$ “ und „ $K_{LSS_{ij}}$ “. Die  $F_{WFP}$  betrifft drei funktionale Interaktionen (Energie, Stoff und Information).<sup>180</sup> Die  $K_{LSS}$  stellt die Kompatibilität dar, also wie geeignet zwei Leitstützstrukturen aus verschiedenen Aspekten sind, um in einem Modul gruppiert zu werden.

$$B_{LSS_{ij}} = \sum_{k=1}^3 F_{WFP_{ij,k}} + \sum_{k=1}^m K_{LSS_{ij,k}} \tag{5.2}$$

- wobei  $B_{LSS_{ij}}$  die Beziehungszahl zwischen  $LSS_i$  und  $LSS_j$  ist und
- $F_{WFP_{ij}}$  die Gewichtung der funktionalen Beziehung (Energie, Stoff, Information) zwischen  $LSS_i$  und  $LSS_j$
- $K_{LSS_{ij}}$  die Gewichtung der Kompatibilitätsbeziehung zwischen  $LSS_i$  und  $LSS_j$  jedes Ziels
- $m$  die Anzahl der Kompatibilitätsbeziehung zwischen beiden LSS.

<sup>180</sup> Vgl. Pahl et al. 2003, S. 43

Beispiel zum Kapitel 5.2.3: In diesem Kapitel wird das Beispiel aus Kapitel 5.2.2 fortgeführt. Zum Beispiel lässt sich  $B_{LSS}$  zwischen LSS 3 und LSS 9 (siehe Markierung „X“ in Abbildung 5-14 links) wie folgt berechnen:  $F_{WFP_{39}} = 0,4$ ,  $K_{LSS_{39A}} = 0$ , da LSS 3 und LSS 9 keine Kompatibilitätsbeziehung haben (siehe Tabelle 5-2),  $K_{LSS_{39B}} = 0,3$ , weil sie für Ziel B eine Kompatibilitätsbeziehung haben (siehe Tabelle 5-2). Das heißt,

$$B_{LSS_{39}} = F_{WFP_{39}} + K_{LSS_{39A}} + K_{LSS_{39B}} = 0,4 + 0 + 0,3 = 0,7.$$

Oder die Beziehungszahl zwischen LSS 1 und LSS 9 lässt sich gleicherweise nach

$$B_{LSS_{19}} = F_{WFP_{19}} + K_{LSS_{19A}} + K_{LSS_{19B}} = 0 + 0,3 + 0,3 = 0,6$$

berechnen. Alle  $B_{LSS_{ij}}$  werden analog zu der oben gezeigten Gleichung berechnet. Das Ergebnis der Bewertung der gesamten Matrix lässt sich wie in Abbildung 5-14 darstellen. Man sieht, dass viele nicht angekreuzte Felder jetzt nach der Bewertung aufgrund der zusätzlichen Berücksichtigung der Kompatibilitätsbeziehungen nicht mehr leer sind.

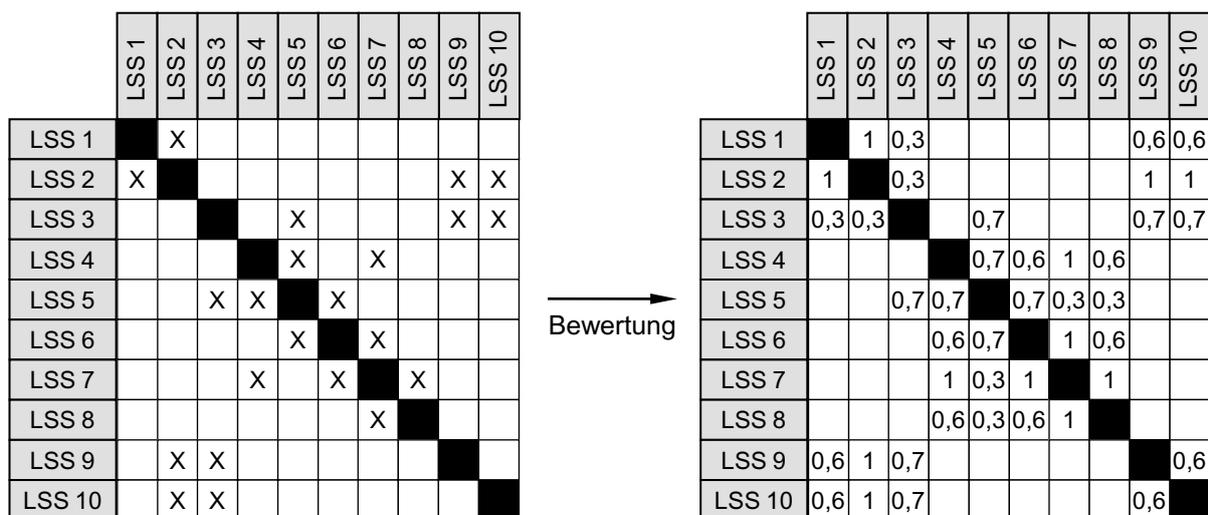


Abbildung 5-14. Links: die unbewertete CDM, rechts: die bewertete CDM des Beispielsystems

Bis zu diesem Zeitpunkt wird die CDM komplett bewertet und zur Optimierung herangezogen. Die Optimierung der Matrix dieses Beispiels wird im Kapitel 5.3.2 weiter konkretisiert. Nachdem alle  $B_{LSS_{ij}}$  bewertet wurden, ist die CDM bereit für den nächsten Schritt.

### 5.3 Optimierung der C&C-M Dependency-Matrix

In diesem Schritt wird die Zielfunktion für die CDM auf Basis des Modularitätsgrades formuliert. Der Wert der Zielfunktion (siehe Gleichung 5.3 und 5.4) zeigt, wie gut die

Produktarchitektur die Ziele für die Produktarchitekturfestlegung erfüllt. In der hier erarbeiteten Methode ist es erwünscht, den Zielfunktionswert zu maximieren. Aufgrund der Diskontinuität und Parametervielzahl der Gleichung ist es kaum möglich, das Maximum der Zielfunktion analytisch zu bestimmen. Deswegen kommt ein Suchalgorithmus zum Einsatz.

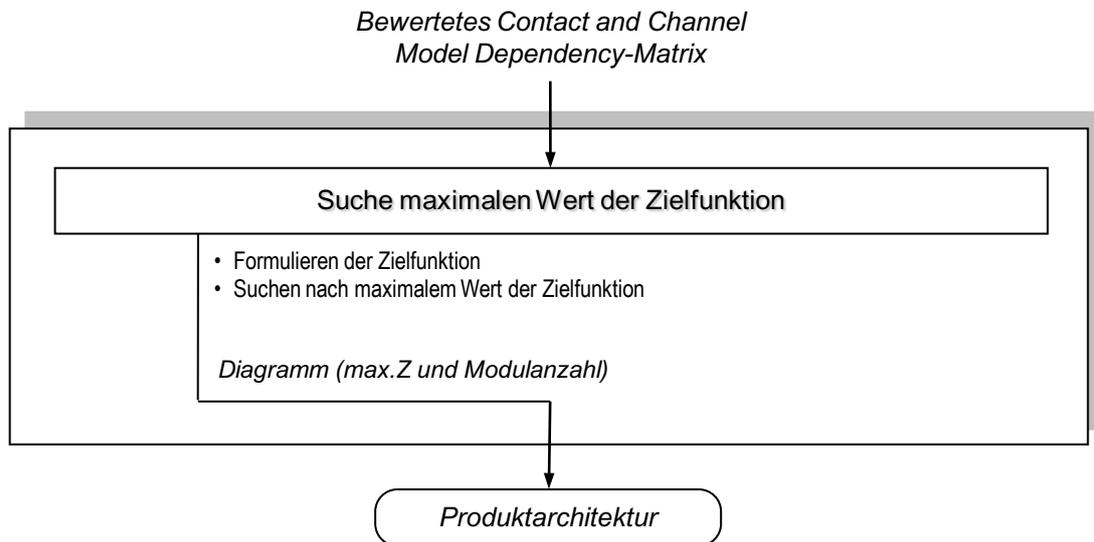


Abbildung 5-15. Vorgehen bei der Optimierung der C&C-M Dependency-Matrix

Die aktuell am besten geeigneten Suchalgorithmen für ein solches komplexes Clusteringproblem sind genetische Algorithmen, die in diesem Schritt angewendet werden. Die Analyseergebnisse ergeben sich in Form von Gruppen der LSS oder der Gliederung der LSS in Module. Jede Gruppe der LSS entspricht einem vorgeschlagenen Modul für das Produkt, was wichtig für die Auslegung der Gestalt im nächsten Schritt ist.

### 5.3.1 Suche nach maximalem Wert der Zielfunktion

#### **Formulieren der Zielfunktion:**

Um mehrere unterschiedliche, durch die Anordnung der LSS in CDM dargestellten Produktarchitekturen vergleichen zu können, braucht man eine Bewertungsfunktion. Diese Funktion sollte bestimmen, wie gut die Produktarchitektur die Ziele und Randbedingungen erfüllen kann.

Die Formulierung der Zielfunktion ( $Z$ ) lässt sich aus dem Bewertungsprinzip des Modularitätsgrades herleiten.<sup>181</sup> Je mehr die Anzahl der Beziehungen innerhalb der Module liegt, desto höher wird die Zielfunktion bewertet – und umgekehrt. Diese

<sup>181</sup> Vgl. Guo/Gershenson 2004, S. 264

Funktion  $Z$  hängt von der Anzahl der Module ( $N_m$ ), den Bewertungen der Beziehungen zwischen den LSS ( $B_{LSS}$ ) und der Anordnung der LSS in der CDM ab.

$$\text{Zielfunktion } (Z) = f \{ N_m, B_{LSS}, \text{Anordnung der LSS} \} \quad (5.3)$$

Ähnlich wie die Gleichung nach GUO und GERSHENSON lässt die Zielfunktion sich dann wie folgt formulieren.

$$Z = \underbrace{\frac{\sum_{k=1}^{N_m} \sum_{i=n_k}^{m_k} \sum_{j=n_k}^{m_k} B_{LSSij}}{(m_k - n_k + 1)^2}}_{\text{Innerhalb der Module}} - 2 \times \underbrace{\frac{\sum_{k=1}^{N_m-1} \sum_{p=k+1}^{N_m} \sum_{i=n_k}^{m_k} \sum_{j=n_p}^{m_p} B_{LSSij}}{(m_p - n_p + 1)(m_k - n_k + 1)}}_{\text{Außerhalb der Module}} \quad (5.4)$$

wobei  $n_k, n_p$  der Index der ersten Leitstützstruktur im k-ten und p-ten Modul,  
 $m_k, m_p$  der Index der letzten Leitstützstruktur im k-ten und p-ten Modul,  
 $N_m$  die gesamte Anzahl der Module im Produkt,  
 $B_{LSSij}$  die Bewertung der Beziehungen jedes Paares der Leitstützstrukturen  
 und  
 $i, j$  Zeile und Spalte in der Matrix sind.

Eine Anordnung der LSS in der CDM ergibt einen Wert der Zielfunktion.

Die Anzahl der verschiedenen Möglichkeiten zur Anordnung der LSS in der CDM ist hoch. Theoretisch kann die Produktarchitektur mit einer Modulanzahl von einem Modul (vollständig integral) bis maximal  $N_{LSS}$  Module (vollständig modular) gestaltet werden. Man kann die gesamte Anordnungsanzahl einfach nach folgender Gleichung berechnen:

$$\text{Gesamte Anzahl der Anordnungen} = (N_{LSS})^{N_{LSS}} \quad (5.5)$$

wobei  $N_{LSS}$  die gesamte Anzahl der LSS ist.

Der Zielfunktionswert ( $Z$ ) ist einfach aus der Matrix zu berechnen. Jedoch sind der maximale Wert ( $Z_{max}$ ) und die entsprechende Anordnung nur sehr schwer analytisch zu berechnen. Die Berechnung für alle Möglichkeiten braucht sehr viel Zeit. Bei z. B. einer 10x10-CDM müssen insgesamt  $10^{10}$  Berechnungen zum Evaluieren der Zielfunktion durchgeführt werden. Ein Suchalgorithmus kommt daher in diesem Fall zum Einsatz.

Die evolutionären Algorithmen sind geeignet für diese Art von Zielfunktionen (Gleichung 5.4), da es sich hier um nichtlineare oder diskontinuierliche Funktionen handelt.<sup>182</sup>

**Suchen nach maximalem Wert der Zielfunktion:**

Wie in Kapitel 2.7 beschrieben, wird die Kodierung durch die Grundregel von GOLDBERG<sup>183</sup> bestimmt. „Das Alphabet der Kodierung sollte so klein wie möglich gewählt werden, wie es einem natürlichen Ausdruck der Lösung entgegenkommt.“ Das Alphabet des genetischen Algorithmus bei dieser Optimierung ist für eine ganzzahlige Kodierung geeignet. Durch Anwendung der binären Kodierung, die normalerweise im GA benutzt wird, in diese vorliegende Arbeit könnten viele Individuen, die sich nicht im Suchraum befinden, erzeugt werden. Dies verursacht allerdings unnötige Rechenzeit. Das Individuum der Lösung lässt sich wie folgt darstellen:

LSS	1	2	3	4	...	$N_{LSS-1}$	$N_{LSS}$
Gehört zur Modulnummer (Individuen)	$M_1$	$M_2$	$M_3$	$M_4$	...	$M_{N_{LSS-1}}$	$M_{N_{LSS}}$

,wobei Modulnummer  $\in \{1, 2, \dots, N_{LSS}\}$ .

Alle Individuen der ersten Generation werden zufällig generiert, wie in Kapitel 2.7 beschrieben. Tabelle 5-3 zeigt ein Beispiel wie eine möglich erste Population aussehen kann.

LSS	1	2	3	4	...	$N_{LSS-1}$	$N_{LSS}$	Zielfunktionswert
Individuum 1	1	4	5	3	...	6	6	0,124
Individuum 2	3	5	2	1	...	3	1	-0,512
Individuum 3	1	7	6	3	...	9	2	-0,230
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
Individuum n	2	3	1	1	...	3	8	-1,187

Tabelle 5-3. Beispiel der Initialisierung der ersten Generation

<sup>182</sup> Vgl. Pohlheim 2000

<sup>183</sup> Goldberg 1989

Alle Individuen werden dann evaluiert. Danach werden Selektion, Rekombination und Mutation durchgeführt.

Den maximalen Zielfunktionswert kennt man vor der Suche nicht. Aus diesem Grund besteht das Abbruchkriterium in der Vorgabe einer Anzahl der Generationen die in der Optimierung untersucht werden sollen. Je mehr Generationen und Populationen der Algorithmus benutzt, desto größer ist die Wahrscheinlichkeit, die optimale Lösung zu finden. Gleichzeitig erhöht sich aber mit der Populationsgröße und mit der Anzahl der Generationen auch die Rechenzeit. Die Lösung des Problems ist das beste Individuum der letzten Generation, das den maximalen Wert liefert.

Wenn selbst nach Einsatz der bekannten Verfahren die Lösung des Problems nicht zufriedenstellend ist, muss über das weitere Einbringen von problemspezifischem Wissen nachgedacht werden. Dies beinhaltet Maßnahmen wie<sup>184</sup>:

- die spezielle Initialisierung der Individuen der Anfangspopulation:
  - Nutzung eines Ergebnisses einer vorangegangenen Optimierung (z. B. auch durch Anwendung anderer Optimierungsverfahren)
  - Einbringen von Wissen durch Experten
  - Nutzung bereits bekannter Lösungen
  - Einbindung problemspezifischer Informationen, die aus der Zielfunktion ableitbar sind
- das angepasste Einschränken der Wertebereiche der Variablen und
- die Entwicklung spezieller Operatoren.

POHLHEIM hat vorgeschlagen, dass die ersten beiden Punkte immer angewendet werden sollten, wenn dieses Wissen vorliegt.<sup>185</sup> Dadurch kommt es mindestens zu einer deutlichen Verkleinerung des Suchbereiches und damit zu einer Beschleunigung des Auffindens zufriedenstellender Lösungen. Die Entwicklung spezieller Operatoren ist in den meisten Fällen schwierig und nicht erforderlich gewesen.

Nach Ablauf der Optimierung durch den genetischen Algorithmus kann der Maximalwert der Zielfunktion  $Z_{max}$  und die dazugehörige Anordnung als Endergebnis ausgegeben werden. Dabei werden aber die Einflüsse einzelner Parameter nicht analysiert. Dazu gehört auch der für diese Arbeit besonders relevante Zusammenhang zwischen der Modulanzahl ( $N_m$ ) und dem maximalen Zielfunktionswert  $Z_{max}$ .

---

<sup>184</sup> Pohlheim 2000, S. 62, 193

<sup>185</sup> Pohlheim 2000, S. 193

Um die optimale Modulanzahl für die Produktarchitektur besser zu verstehen, wurde der Einfluss von der Modulanzahl auf  $Z_{max}$  bzw.  $Z_{min}$  weiter untersucht.  $N_m$  in der Zielfunktion wird als konstant (von 1 bis  $N_{LSS}$ ) für jede Suche nach  $Z_{max}$  variiert.  $Z_{max}|_{N_m}$  und  $Z_{min}|_{N_m}$  werden mit dem genetischen Algorithmus für jede Modulanzahl gesucht. Der Zusammenhang zwischen  $Z_{max}$ ,  $Z_{min}$  und  $N_m$  wird beispielhaft in Abbildung 5-16 gezeigt, wobei man die Zielfunktionswerte für die integrale Produktarchitektur ( $Z_{integral}$ ) und die völlig modulare Produktarchitektur ( $Z_{modular}$ ) einfach ohne Suchalgorithmus mit den Gleichungen (5.6) und (5.7) berechnen kann:

$$Z_{integral} = Z|_{N_m=1} = \frac{\sum_{i=1}^{N_{LSS}} \sum_{j=1}^{N_{LSS}} B_{LSSij}}{N_{LSS}^2} \quad (5.6)$$

$$Z_{modular} = Z|_{N_m=N_{LSS}} = -(N_{LSS} \times Z_{integral}) \quad (5.7)$$

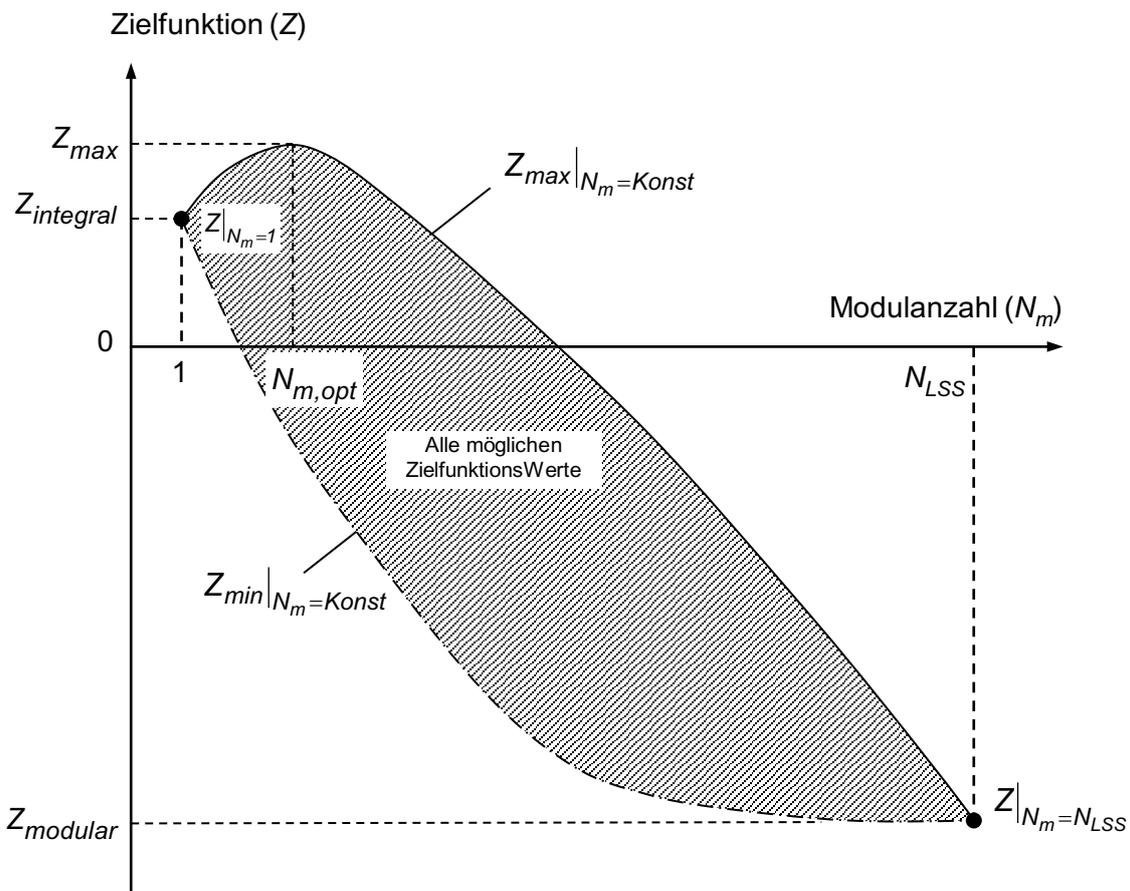


Abbildung 5-16. Zusammenhang zwischen den Zielfunktionswerten ( $Z$ ) und der Modulanzahl ( $N_m$ )

Wie in der Abbildung 5-16 gezeigt, besitzen die  $Z_{max|N_m}$  und  $Z_{min|N_m}$  gleiche Werte, falls  $N_m = 1$  und  $N_m = N_{LSS}$  ist. Wenn die integrale Produktarchitektur nicht die beste Produktarchitektur ( $Z_{integral}$  ungleich  $Z_{max}$ ) ist, erhöht sich die  $Z_{max|N_m}$  weiter bis zum  $Z_{max}$ . Danach verringert sich  $Z_{max|N_m}$  fast linear bis zu  $Z_{modular}$ .  $Z_{min|N_m}$  fällt schon bei  $N_m = 1$  sehr schnell linear bis  $Z_{min|N_m}$  ungefähr gleich wie  $Z_{modular}$ . Danach bleibt der Zielfunktionswert fast konstant bis  $N_m = N_{LSS}$ .

Der schattierte Bereich im Diagramm zeigt alle möglichen Werte der Zielfunktion, wobei die gesamte Anzahl durch die Gleichung (5.5) berechnet werden kann.

Nachdem eine  $Z_{max}$  ermittelt wurde, kann die Produktarchitektur einfach durch eine Umwandlung von der Matrixdarstellung in eine C&C-M Darstellung modelliert werden. Mit dem Ergebnis kann die optimale Produktarchitektur erstellt und in C&C-M Darstellung zusammen mit der Prinzipskizze visualisiert werden (Beispiel in Abbildung 5-21 links). Die folgenden Punkte sollen aus der Darstellung ersichtlich sein:

- Anzahl der Module im Produkt
- Funktionen oder LSS in jedem Modul
- Grenze jedes Moduls und die Wirkflächenpaare des Moduls
- Trennmöglichkeit und die Verbindungspositionen

Beispiel zum Kapitel 5.3.1: Das Ziel ist das Finden des maximalen Zielfunktionswerts durch den genetischen Algorithmus. Eine Erklärung zur Berechnung der Zielfunktion wird hier ebenfalls vorgestellt. Die CDM vom Kapitel 5.2.3 wird wie in Abbildung 5-18 (rechts) so konfiguriert, dass sich zwei Module ergeben.

	LSS 1	LSS 2	LSS 3	LSS 4	LSS 5	LSS 6	LSS 7	LSS 8	LSS 9	LSS 10
LSS 1	■	1	0,3						0,6	0,6
LSS 2	1	■	0,3						1	1
LSS 3	0,3	0,3	■		0,7				0,7	0,7
LSS 4				■	0,7	0,6	1	0,6		
LSS 5			0,7	0,7	■	0,7	0,3	0,3		
LSS 6				0,6	0,7	■	1	0,6		
LSS 7				1	0,3	1	■	1		
LSS 8				0,6	0,3	0,6	1	■		
LSS 9	0,6	1	0,7						■	0,6
LSS 10	0,6	1	0,7						0,6	■

Abbildung 5-17. Das Ergebnis der CDM vom Abschnitt 5.2.3

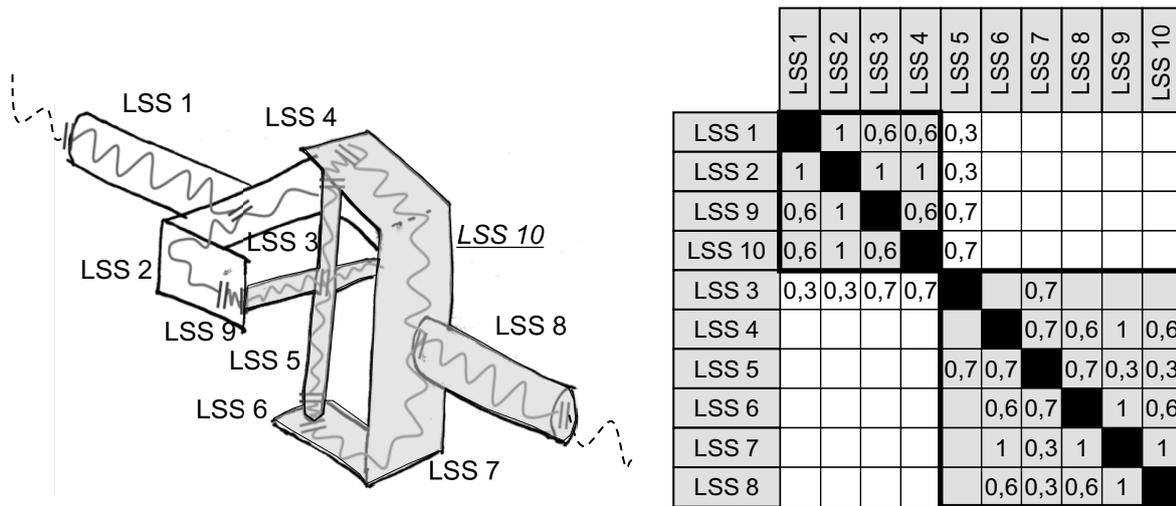


Abbildung 5-18. Beispiel einer Anordnung des Kreuzgelenks mit 2 Modulen und die entsprechende CDM

Das Modul 1 enthält LSS 1, 2, 9, 10 und Modul 2 besteht aus LSS 3, 4, 5, 6, 7, 8. Die Zielfunktion (Z) für diese Produktarchitektur wird wie folgt berechnet:

$$Z = \frac{\sum_{k=1}^{N_m} \sum_{i=n_k}^{m_k} \sum_{j=n_k}^{m_k} B_{LSSij}}{(m_k - n_k + 1)^2} - 2 \times \frac{\sum_{k=1}^{N_m-1} \sum_{p=k+1}^{N_m} \sum_{i=n_k}^{m_k} \sum_{j=n_p}^{m_p} B_{LSSij}}{(m_p - n_p + 1)(m_k - n_k + 1)}$$

$$Z = \frac{\sum_{i=1}^4 \sum_{j=1}^4 B_{LSSij} + \sum_{i=5}^{10} \sum_{j=5}^{10} B_{LSSij}}{2} - 2 \times \frac{\sum_{i=5}^{10} \sum_{j=1}^5 B_{LSSij}}{2}$$

$$Z = \frac{\binom{9,6}{16} + \binom{15}{36}}{2} - 2 \times \frac{\binom{2}{24}}{2} = 0,425$$

Die CDM kann theoretisch aus einem Modul oder bis zu maximal 10 Modulen bestehen. Das heißt, dass jede LSS zu Modul 1 bis Modul 10 gehören kann. Alle  $10^{10}$  Lösungsmöglichkeiten zu evaluieren, erfordert zu viel Rechenzeit. Es ist sehr unwahrscheinlich, dass eine Produktarchitektur völlig modular ist. Erfahrungsmäßig beträgt sehr selten eine optimale Modulanzahl der optimalen Produktarchitektur mehr als die Hälfte der Anzahl der LSS. In diesem Beispiel wird der maximale Modulanzahl mit 5 angegeben. Trotzdem ergeben sich immer noch  $5^{10}$  oder 9.765.625 Lösungsmöglichkeiten. Es wurde der genetische Algorithmus benutzt, um die optimale Lösung zu suchen, damit die lange Rechenzeit, die zur Evaluierung aller Möglichkeiten nötig wäre, vermieden werden kann.

Das Ergebnis ist in Abbildung 5-19 dargestellt. Am Anfang erhöhen sich die gefundenen höchsten Zielfunktionswerte ziemlich schnell. Nach etwa 200 Generationen (insgesamt 4.000 Individuen oder 5% aller Lösungsmöglichkeiten) findet der genetische Algorithmus schon den maximalen Zielfunktionswert. Es hat insgesamt etwa 30 Sekunden für alle 1000 Generationen auf einem normalen Rechner (PC) gedauert. Dieses Ergebnis wurde mit weiteren Durchläufen ausgehend von unterschiedlichen Startpopulationen bestätigt.

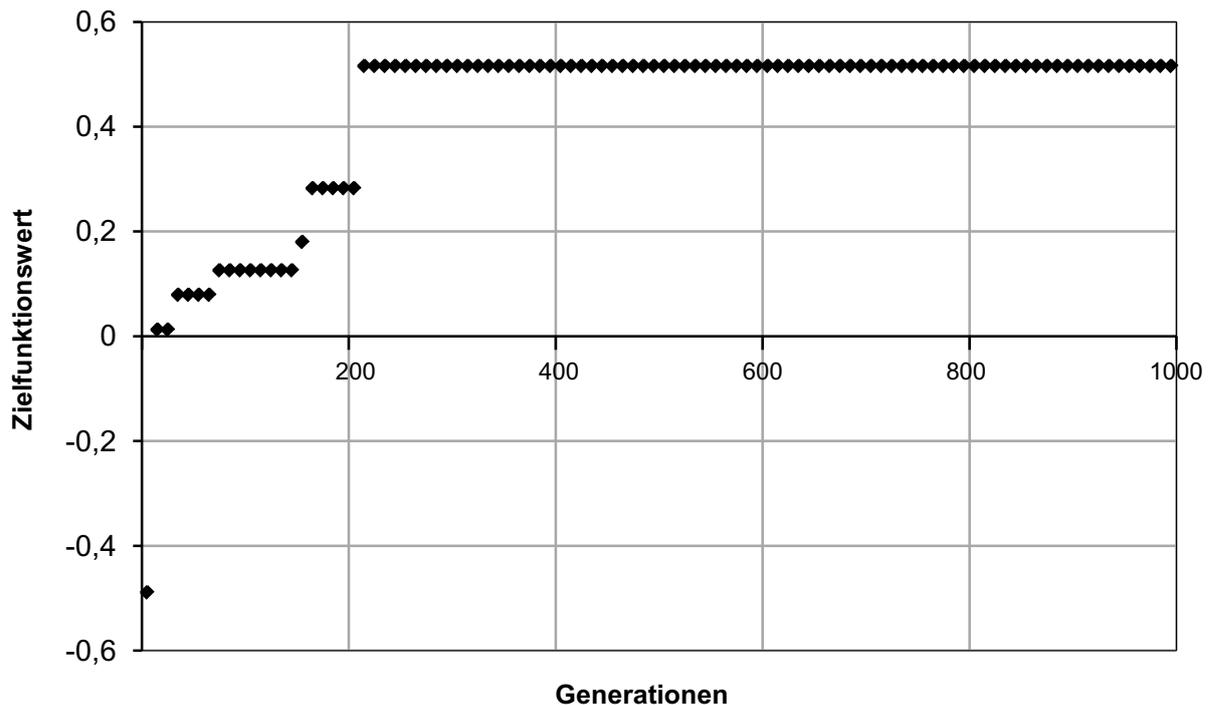


Abbildung 5-19. Verlauf des besten Zielfunktionswerts bei der Suche mit genetischem Algorithmus (mit 20 Populationen, 1000 Generationen)

Der genetische Algorithmus ergibt, dass der maximale Zielfunktionswert  $0,516$  beträgt. Von daher wird die Konfiguration wie in der Abbildung 5-21 als optimale Produktarchitektur für die weiter Gestaltdefinition vorgeschlagen. Die Produktarchitektur für den Fall A liefert nur den Wert  $0,183$ .

Der Zusammenhang zwischen den Zielfunktionswerten und der Modulanzahl ist in Abbildung 5-20 dargestellt. Es ist zu erkennen, dass die meisten Werte der Zielfunktion im Minusbereich liegen. Nur wenn das Produkt weniger als vier Module hat, können die Zielfunktionswerte positiv sein. Die optimale Modulanzahl ist zwei.

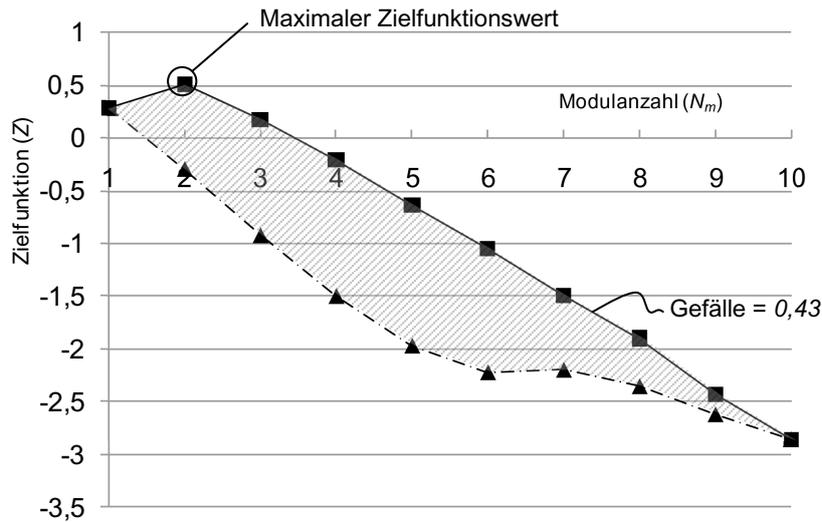


Abbildung 5-20. Zusammenhang zwischen den Zielfunktionswerten und Modulanzahl des Kreuzgelenksystems

Bei Betrachtung des Gefälles der oberen Linie in Abbildung 5-20 kann man sagen, dass je mehr Module in der Produktarchitektur verwendet werden, umso mehr reduziert sich der Zielfunktionswerte auf rund 0,43.

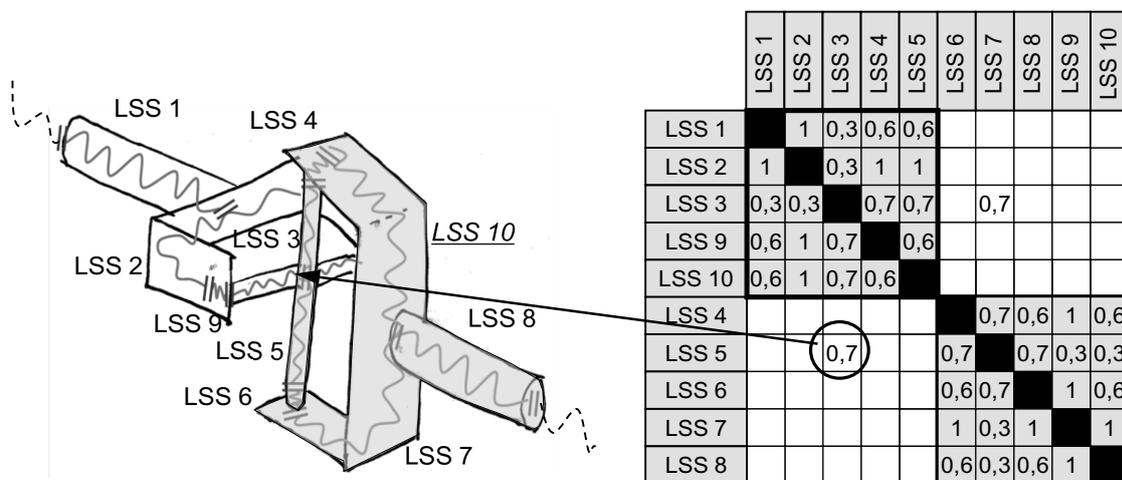


Abbildung 5-21. Die optimale Produktarchitektur

Das Ergebnis der Anwendung der in dieser Arbeit entwickelten Methode zur Festlegung der Produktarchitektur mit C&C-M ist ein Vorschlag für eine Produktarchitektur als Basis für die weitere Gestaltdefinition die die Vorgaben aus dem Zielsystem hinsichtlich der Produktarchitektur optimal erfüllt. In dem Fallbeispiel für das Kreuzgelenk ergibt sich folgende Produktarchitektur:

- Zwei Module in der Produktarchitektur
- Modul 1 besteht aus LSS 1, 2, 3, 9, 10 und Modul 2 aus LSS 4, 5, 6, 7, 8
- Trennungsmöglichkeit für WFP zwischen LSS 3 und LSS 5

### 5.3.2 Unterstützungswerkzeug

Um die Erstellung und die Optimierung der CDM einfacher durchführen zu können, wurde eine Software programmiert. Eingangsvariablen, die durch den Benutzer definiert werden müssen, sind die Größe der CDM und danach alle Beziehungsbewertungen sowohl für die funktionalen Beziehungen als auch für die Kompatibilitätsbeziehungen. Die Gruppierung der LSS in dieser Software erfolgt manuell oder automatisch. Die Hauptvorteile gegenüber der Erstellung in einer Tabelle (zum Beispiel in Excel) sind eine einfachere Eingabe und die Möglichkeit zur Berechnung des Wertes der Zielfunktion. Die Optimierung der CDM ist in dieser Software ebenfalls integriert.

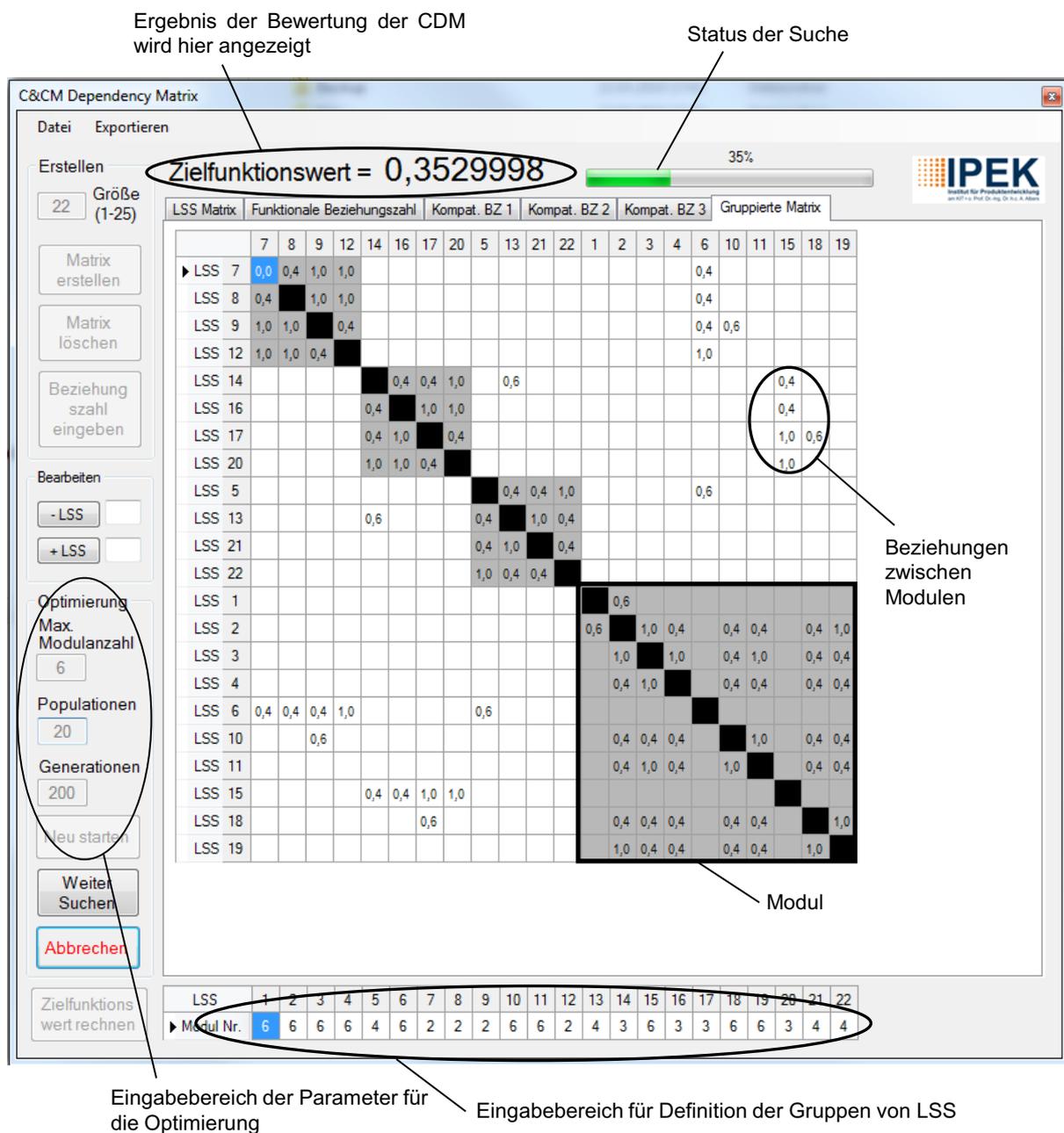


Abbildung 5-22. Die Software zur Unterstützung der Erstellung, Bewertung und Optimierung der CDM

Ein sogenannter einfacher genetischer Algorithmus wird in dieser Software angewendet. Die Anzahl der Populationen und Generationen kann vom Benutzer frei gewählt werden, um die Optimierung durchzuführen. Abbildung 5-22 zeigt die Benutzeroberfläche dieser Software.

Das Ergebnis der Optimierung wird in Form der Module und zugehörigen LSS abgebildet. In der Matrix wird deutlich gezeigt, welche Beziehungen sich innerhalb eines Moduls befinden bzw. welche Beziehungen die Module verbinden.

Die Optimierung der CDM mit einem genetischen Algorithmus kann man auf verschiedenen Wegen durchführen. Man kann diesen Optimierungsprozess selber programmieren wie bei der Software in Abbildung 5-22 oder eine Software für den genetischen Algorithmus benutzen.

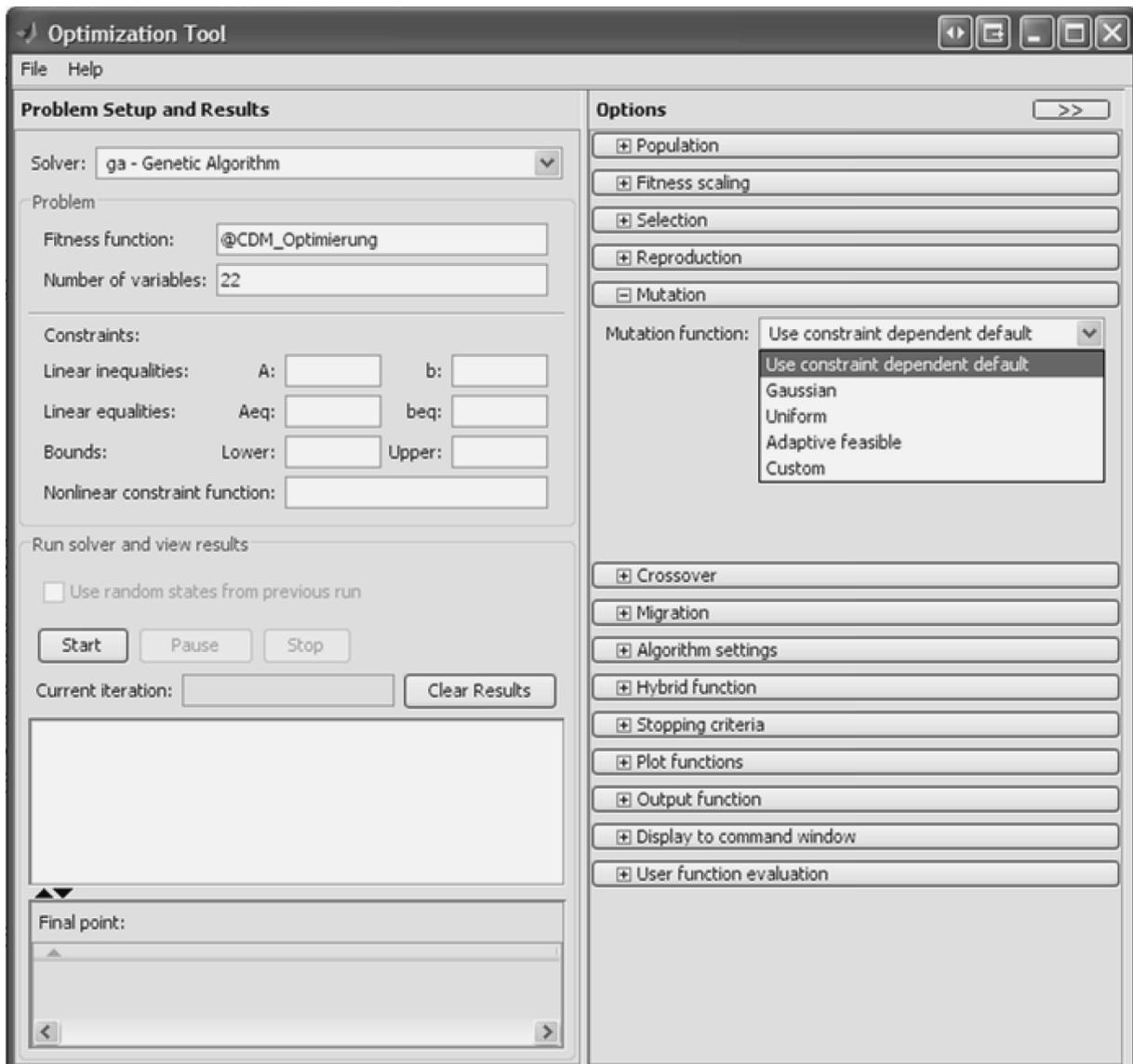


Abbildung 5-23. Die MATLAB Optimization Toolbox zur Optimierung der CDM

Eine andere Möglichkeit, die relativ einfach einzusetzen ist, ist die Verwendung der MATLAB Optimization Toolbox. Bei dieser Toolbox muss man die Zielfunktion und die bewertete CDM in einem M-File programmieren. Dieses M-File und andere notwendige Dateien für die Optimierung unter MATLAB kann bequem und mit wenig Aufwand durch die in Abbildung 5-22 dargestellte Software erzeugt werden. In der Optimization Toolbox von MATLAB kann aus einer Vielzahl an Varianten von genetischen Algorithmen ausgewählt werden, zum Beispiel Gaussian, Uniform, Adaptive feasible oder Custom (individuelle Programmierung) bei der Mutation.

---

## 6 Anwendungsbeispiele der Methode

### 6.1 Produktarchitekturfestlegung eines Roboterunterarms

Die Entwicklung eines Roboterunterarms für einen humanoiden Roboter (ARMAR III) des Sonderforschungsbereiches 588 (SFB 588) „Humanoide Roboter – Lernende und kooperierende multimodale Roboter“ der Deutschen Forschungsgemeinschaft (DFG) wird als Anwendungsbeispiel für die, in dieser Arbeit entwickelten, Methode vorgestellt.

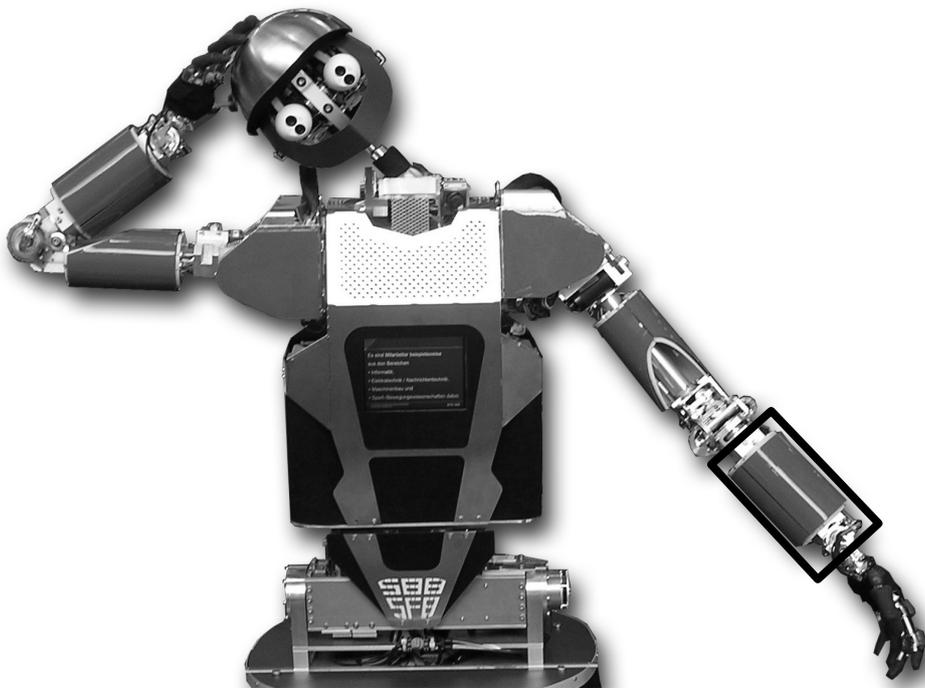


Abbildung 6-1. Humanoider Roboter ARMAR III<sup>186</sup>

Die Funktionsstruktur, die in der folgenden Abbildung dargestellt ist, wurde vor Anwendung der Methode erstellt und wird für die Modularisierung als bekannt vorausgesetzt.

Die Struktur hat vier Hauptfunktionen<sup>187</sup> (siehe Abbildung 6-2): Eine Seite des Unterarms wird mit dem Ellenbogengelenk des Roboters verbunden und die andere Seite mit der Roboterhand. Informationen und Energie müssen zwischen dem

---

<sup>186</sup> Albers et al. 2006

<sup>187</sup> Albers et al. 2009b

Unterarm und der Hand sowie dem Unterarm und dem Ellenbogengelenk ausgetauscht werden. Die Gesamtfunktion dieses Roboterunterarms ist die Realisierung einer menschenähnlichen Bewegung des Arms von der Schulter bis zum Handgelenk. Dieser Roboter wird in mehreren Instituten erprobt, und es muss damit gerechnet werden, dass der Roboterunterarm mit Objekten in seinem Umfeld oder mit sich selbst kollidiert.

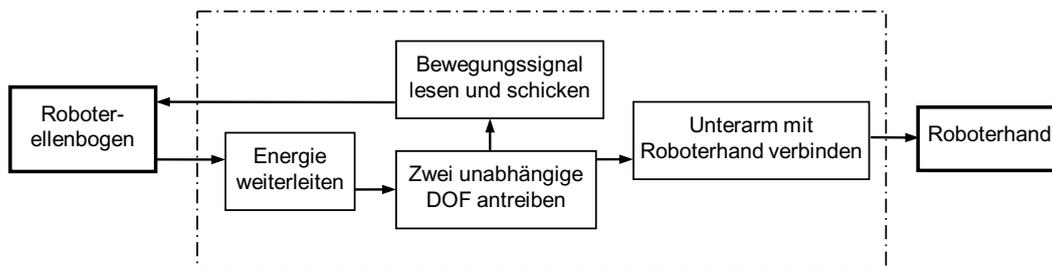


Abbildung 6-2. Die Funktionsstruktur des Roboterunterarms

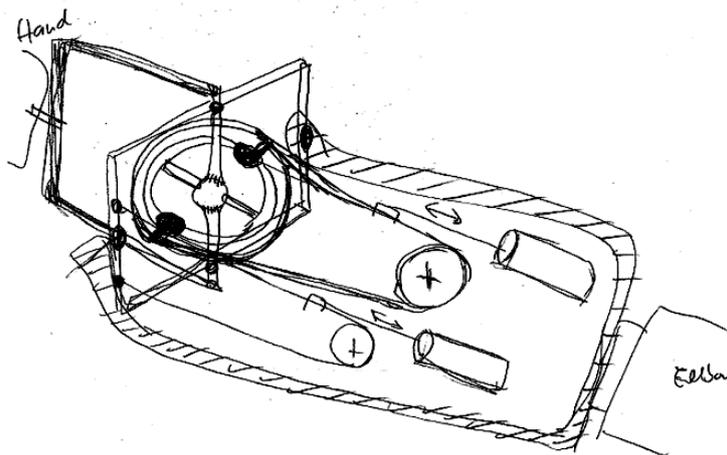


Abbildung 6-3. Ausgewählte Prinzipskizze<sup>188</sup>

Die Produktarchitektur wird unter Berücksichtigung sowohl der Funktionen als auch von Aspekten hinsichtlich Wartung und Austauschbarkeit festgelegt. Nach der Generierung verschiedener Lösungsideen, wurde eine Prinzipskizze für die ausgewählte Lösung wie in der Abbildung 6-3 erstellt.

Es ist nicht immer einfach zu entscheiden mit welchem Abstraktionsniveau ein C&C-M erstellt werden sollte. Daher ist eine direkte Ableitung des C&C-M alleine auf Basis der Prinzipskizze schwierig. Es muss beispielweise entschieden werden, ob es Sinn macht jedes skizzierte Bauelement als eine LSS zu modellieren oder eine

<sup>188</sup> Albers et al. 2009b

abstraktere oder konkretere Modellierung zu wählen. Für die Wahl des Abstraktionsniveaus sollte bei der Modellerstellung daher auch immer die Funktionsbeschreibung mit berücksichtigt werden.

Ähnlich wie in Abbildung 5-6 bleiben alle Details dieser Prinzipskizze auf einem Abstraktionsniveau ähnlich der dritte Hierarchiestufe in Abbildung 2-12. Alle Grundfunktionen, die für das Verständnis der Prinzipskizze notwendig sind, werden in die Funktionsstruktur aufgenommen. Hilfsfunktionen, die für die Festlegung der Produktarchitektur nicht notwendig sind, werden in diesem Handlungsschritt vernachlässigt. Die Funktionsformulierung der Energieflüsse wird hier in diesem Fall mit einem „Substantiv-Verb-Paar“ formuliert und bei Bedarf mit Adjektiven ergänzt, z. B. „Energie übertragen“ oder „elektrische in mechanische Energie umwandeln“ (siehe Abbildung 6-4). Bei diesem Betrachtungsniveau muss der Roboterunterarm insgesamt 22 Grundfunktionen mit 28 Interaktionen erfüllen.

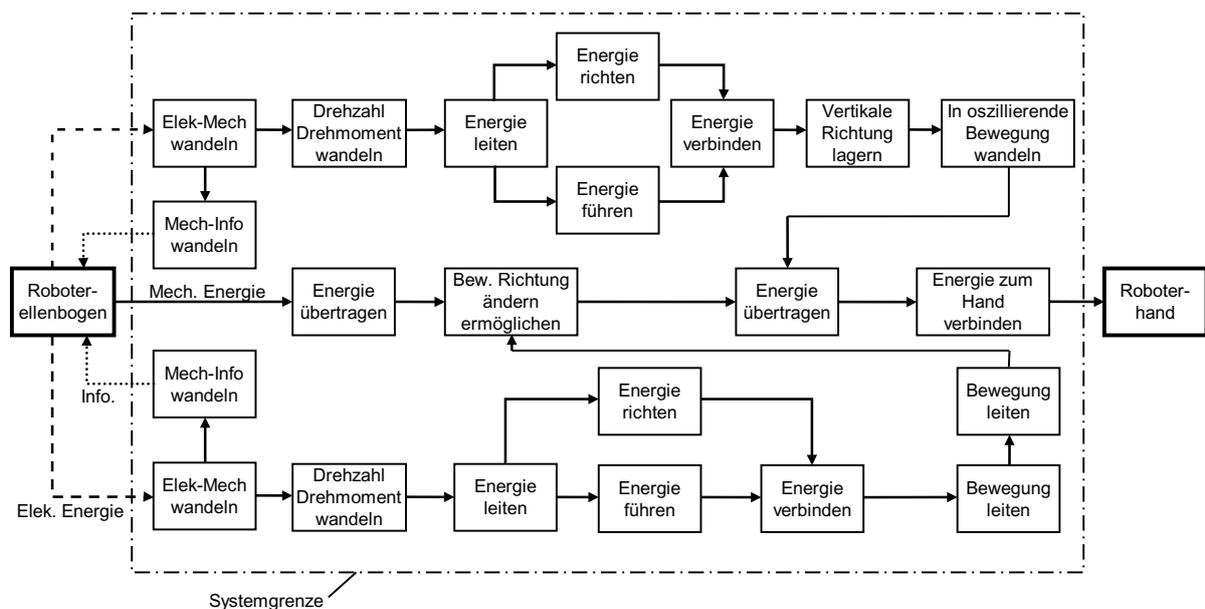


Abbildung 6-4. Modifizierte Funktionsstruktur des Roboterunterarms

Zur Erstellung der Funktionsbeschreibung werden alle Grundfunktionen und deren Interaktionen zuerst in der Nomenklatur des C&C-M definiert. Durch diese schrittweise Erstellung der Funktionsbeschreibung bekommt man eine klarere Vorstellung, auf welchem Abstraktionsniveau die Funktionen in C&C-M integriert werden müssen. Die Funktionsbeschreibung und die Prinzipskizze lassen sich durch das C&C-M miteinander verknüpfen, wobei alle Positionen der LSS und WFP in der Prinzipskizze definiert werden (vgl. Abbildung 6-5). In diesem Schritt der Erstellung der C&C-M werden die LSS, die in einem Bauteil gruppiert werden sollten, identifiziert und gruppiert.

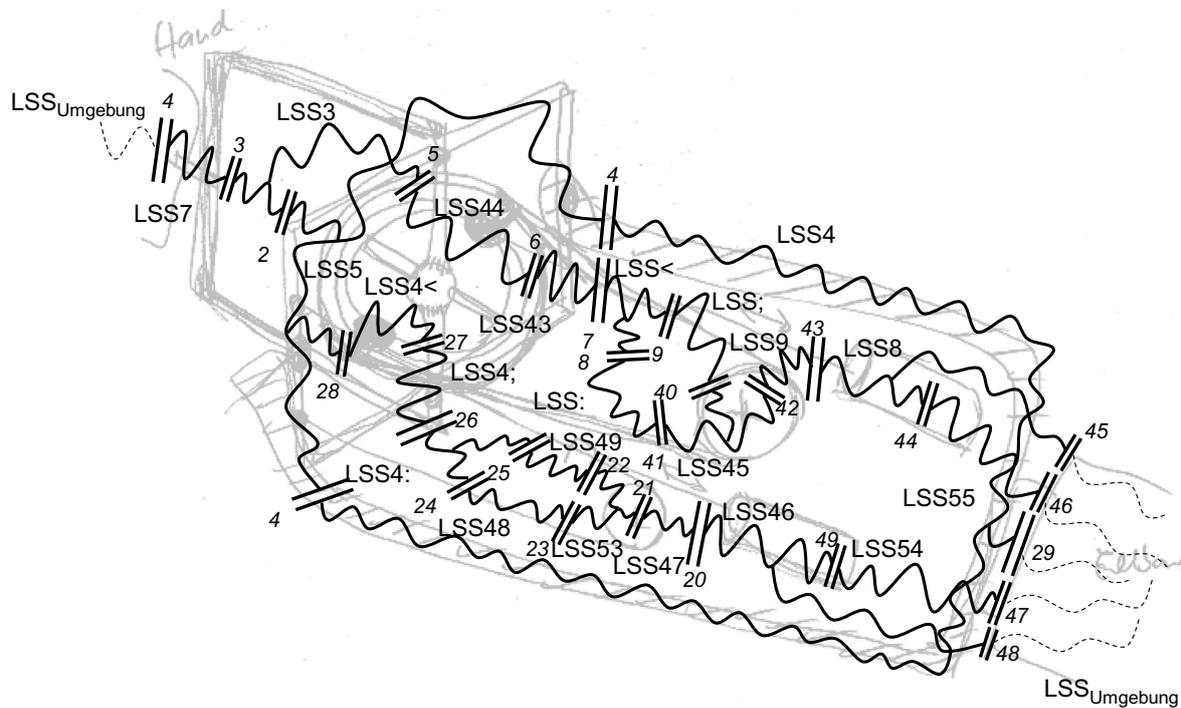


Abbildung 6-5. C&amp;C-M des Roboterunterarms

Leitstützstrukturen	Grundfunktionen	Wirkflächenpaaren
LSS 1	Energie übertragen	WFP 1, 29
LSS 2	Änderung der Bewegungsrichtung ermöglichen	WFP 1, 2
LSS 3	Energie übertragen	WFP 2, 3, 5
LSS 4	Energie zur/von Hand übertragen	WFP 3, 4
LSS 5	Energie Elek-Mech wandeln	WFP 13, 14, 15
LSS 6	Drehzahl, Drehmoment wandeln	WFP 12, 13
LSS 7	Energie führen	WFP 8, 11
LSS 8	Energie richten	WFP 9, 10
LSS 9	Energie verbinden	WFP 7, 8, 9
LSS 10	Vertikale Richtung lagern	WFP 6, 7
LSS 11	In oszillierende Bewegung wandeln	WFP 5, 6
LSS 12	Energie leiten	WFP 10, 11, 12
LSS 13	Energie Elek-Mech wandeln	WFP 19, 20
LSS 14	Drehzahl, Drehmoment wandeln	WFP 20, 21
LSS 15	Energie führen	WFP 23, 24
LSS 16	Energie richten	WFP 22, 25
LSS 17	Energie verbinden	WFP 24, 25, 26
LSS 18	Bewegung leiten	WFP 26, 27
LSS 19	Bewegung leiten	WFP 27, 28
LSS 20	Energie leiten	WFP 21, 22, 23
LSS 21	Mech-Info wandeln	WFP 17, 19
LSS 22	Mech-Info wandeln	WFP 14, 16

Tabelle 6-1 Liste der LSS, WFP und Funktionen des C&amp;C-M

Dies gilt beispielsweise für Standardbauteile oder Zukaufteile, die viele Funktionen besitzen. Weiterhin ist es sinnvoll LSS zu gruppieren die aus Sicht des Entwicklers

zusammen gefertigt werden sollten. In diesem Beispiel findet der Entwickler keine LSS, die manuell gruppiert werden muss.

Daraus ergibt sich, dass dieses Roboterunterarmsystem aus 22 LSS und 29 WFP besteht. Die im System vorhandenen LSS und WFP können beliebig nummeriert werden. Die Funktionsstruktur mit den dazugehörigen LSS und WFP wird in Tabelle 6-1 abgebildet.

Im nächsten Schritt werden die Wichtigkeiten der funktionalen Beziehungen und Kompatibilitätsbeziehungen definiert. In diesem Beispiel werden die funktionalen Beziehungen etwas wichtiger als die Kompatibilitätsbeziehungen der Wartung/Austauschbarkeit gewichtet – nämlich „sehr wichtig“ gegen „wichtig“. Das heißt, dass die Gewichtung der funktionalen Beziehungen  $3/5$  (oder  $0,6$ ) beträgt und  $2/5$  (oder  $0,4$ ) für die Kompatibilitätsbeziehungen der Wartung/Austauschbarkeit. Diese Gewichtung wird vom Entwickler vorgenommen, wird im Zielsystem dokumentiert, basiert auf Erfahrung und ist das Ergebnis mehrere Diskussionen im Entwicklungsteam.

Die C&C-M Dependency-Matrix (CDM) als Basis für die rechnerunterstützte Architekturfestlegung kann daraufhin erstellt werden. Alle Beziehungen zwischen den LSS ( $B_{LSS}$ ) werden bewertet und in einer CDM wie in Abbildung 6-6 kompakt und übersichtlich dargestellt.

	LSS 1	LSS 2	LSS 3	LSS 4	LSS 5	LSS 6	LSS 7	LSS 8	LSS 9	LSS 10	LSS 11	LSS 12	LSS 13	LSS 14	LSS 15	LSS 16	LSS 17	LSS 18	LSS 19	LSS 20	LSS 21	LSS 22	
LSS 1	1	0,6																					
LSS 2	0,6	1	0,4						0,4	0,4								0,4	1				
LSS 3	1		1						0,4	1								0,4	0,4				
LSS 4	0,4	1		1					0,4	0,4								0,4	0,4				
LSS 5					1	0,6						0,4									0,4	1	
LSS 6				0,6	1	0,4	0,4	0,4			1												
LSS 7				0,4		1	0,4	1			1												
LSS 8				0,4	0,4	1	1	1	0,6		0,4												
LSS 9				0,4	1	1	1	1	0,6		0,4												
LSS 10	0,4	0,4	0,4					0,6	1	1								0,4	0,4				
LSS 11	0,4	1	0,4						1	1	1							0,4	0,4				
LSS 12					1	1	1	0,4			1												
LSS 13				0,4								1	0,6								1	0,4	
LSS 14												0,6	1	0,4	0,4	0,4				1			
LSS 15												0,4	1	0,4	1				1				
LSS 16												0,4	0,4	1	1				1				
LSS 17												0,4	1	1	1	0,6			0,4				
LSS 18	0,4	0,4	0,4					0,4	0,4								0,6	1	1				
LSS 19	1	0,4	0,4					0,4	0,4								1	1					
LSS 20												1	1	1	0,4					1			
LSS 21				0,4								1									1	0,4	
LSS 22				1								0,4									0,4	1	

Abbildung 6-6. CDM vor der Optimierung

Zu Beginn der Optimierung beträgt der Zielfunktionswert ( $Z$ )  $0,135$  bei einer vollständig integralen Produktarchitektur. Aus der CDM ist noch keine sinnvolle

Moduleinteilung erkennbar und daher wird die Produktarchitektur als integral betrachtet (ein einziges Modul). Die manuelle Berechnung der Zielfunktion und die Suche nach der optimalen Konfiguration sind in diesem Fall sehr schwierig, weil die Modulanzahl und die möglichen Anordnungen aller LSS unbekannt sind. Eine optimale Produktarchitektur kann mittels eines genetischen Algorithmus identifiziert, wobei die Zielfunktion in Form der Gleichung (5.4) formuliert wird:

$$Z = \frac{\sum_{k=1}^{N_m} \frac{\sum_{i=n_k}^{m_k} \sum_{j=n_k}^{m_k} B_{LSSij}}{(m_k - n_k + 1)^2}}{N_m} - 2 \times \frac{\sum_{k=1}^{N_m-1} \sum_{p=k+1}^{N_m} \frac{\sum_{i=n_k}^{m_k} \sum_{j=n_p}^{m_p} B_{LSSij}}{(m_p - n_p + 1)(m_k - n_k + 1)}}{N_m}$$

Wie schon im Kapitel 2.7 beschrieben, wird die Integerkodierung im einfachen genetischen Algorithmus eingesetzt. Prinzipbedingt besteht bei genetischen Algorithmen die Gefahr, dass nur ein lokales Optimum und nicht das globale Optimum gefunden wird. Es existieren jedoch Strategien und Vorgehensweisen mit denen dieses Risiko verringert werden kann und die bei der Durchführung der in dieser Arbeit entwickelten Methode berücksichtigt werden sollten. Hierzu zählen zum Beispiel die Variation des Startdesigns zu größeren Populationen sowie der Prozeßparameter des genetischen Algorithmus z. B. eine Erhöhung der Anzahl der Generationen.

Unter Berücksichtigung dieser Randbedingungen wurde für das in diesem Kapitel vorgestellte Anwendungsbeispiel ein maximaler Wert der Zielfunktion von 0,446 gefunden (siehe Verlauf in Abbildung 6-7).

Um die Verteilung der Zielfunktionswerte besser erkennen zu können, lässt man den Algorithmus  $Z_{max}$  und  $Z_{min}$  für jedes gleichbleibende  $N_m$  suchen. Das Ergebnis wird in Abbildung 6-8 gezeigt.

Die Werteverteilung der Zielfunktionen ist ähnlich wie beim Beispiel des Kreuzgelenks. Die meisten Werte sind negativ. Nur wenige Konfigurationen der LSS können zu positiven Zielfunktionswerten führen. Die optimale Modulanzahl (bei globaler Lösung) beträgt vier Module. Der Zielfunktionswert einer Drei-Modul-Architektur ist hier allerdings nur geringfügig kleiner als  $Z_{max}$  bei vier Modulen.

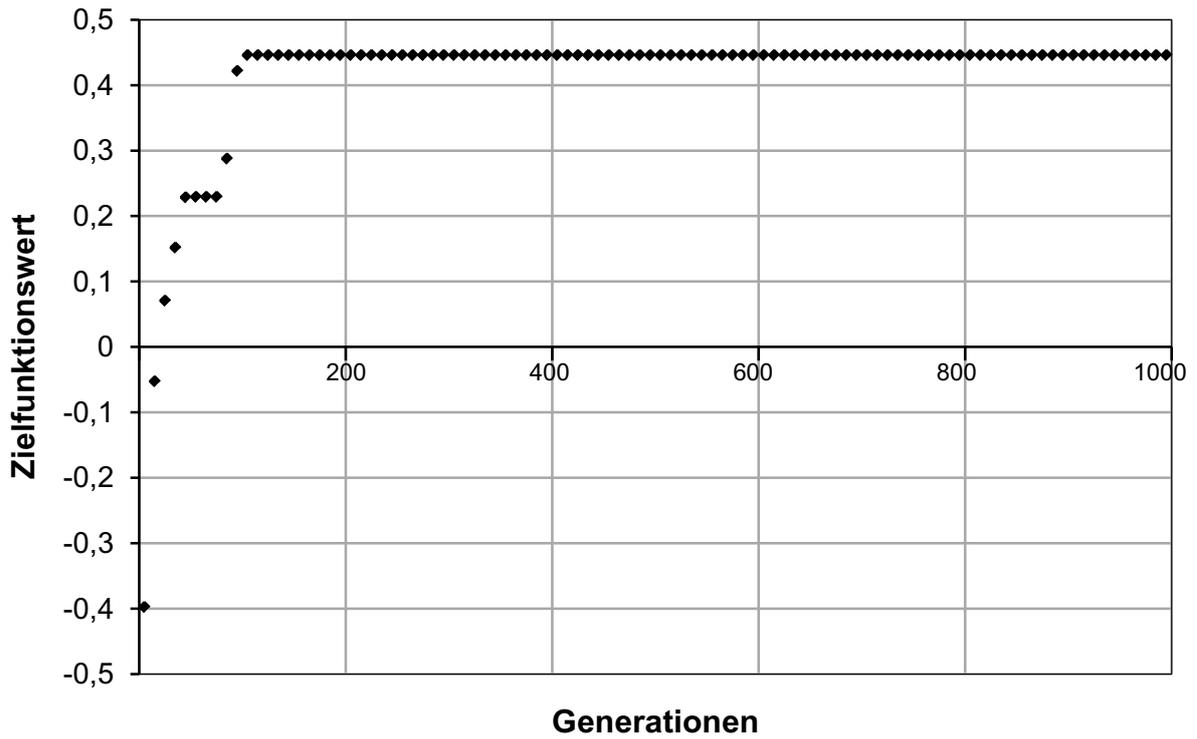


Abbildung 6-7. Verlauf des besten Zielfunktionswerts bei der Suche mit dem genetischen Algorithmus (mit 200 Populationen, 1000 Generationen)

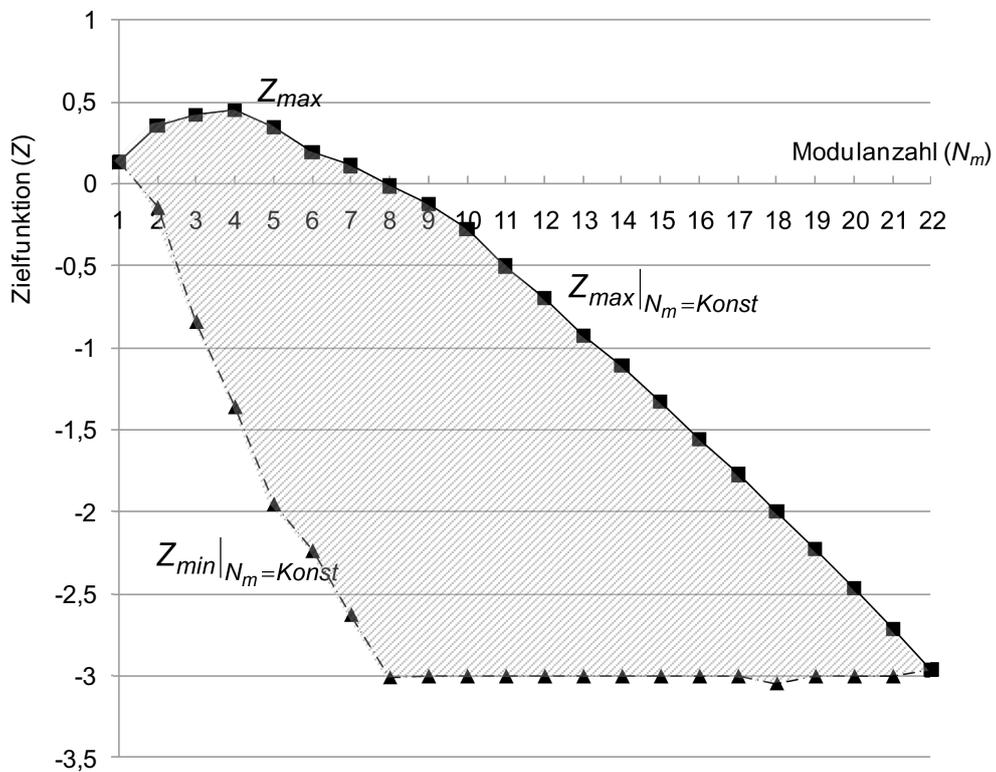


Abbildung 6-8. Zusammenhang zwischen den Zielfunktionswerten und der Modulanzahl des Roboterunterarms

Abbildung 6-9 zeigt die Anordnungen der LSS, sodass der Roboterunterarm aus vier Modulen besteht und Z einen maximalen Wert annimmt.

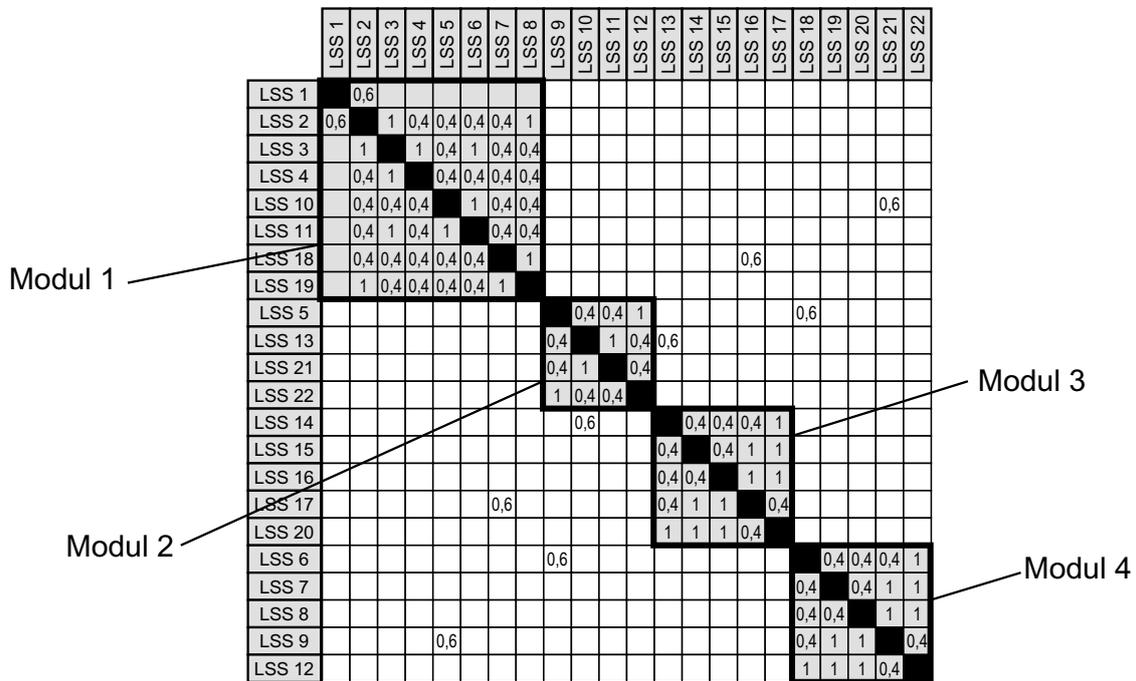


Abbildung 6-9. CDM nach der Optimierung

Ohne die Anwendung der in dieser Arbeit entwickelten Methode wäre es wahrscheinlich, dass die Produktarchitektur des Roboterunterarms bei einer intuitiven Vorgehensweise eine integrale Architektur besäße. Abbildung 6-10 zeigt das Ergebnis der Produktarchitekturfestlegung in einem C&C-M.

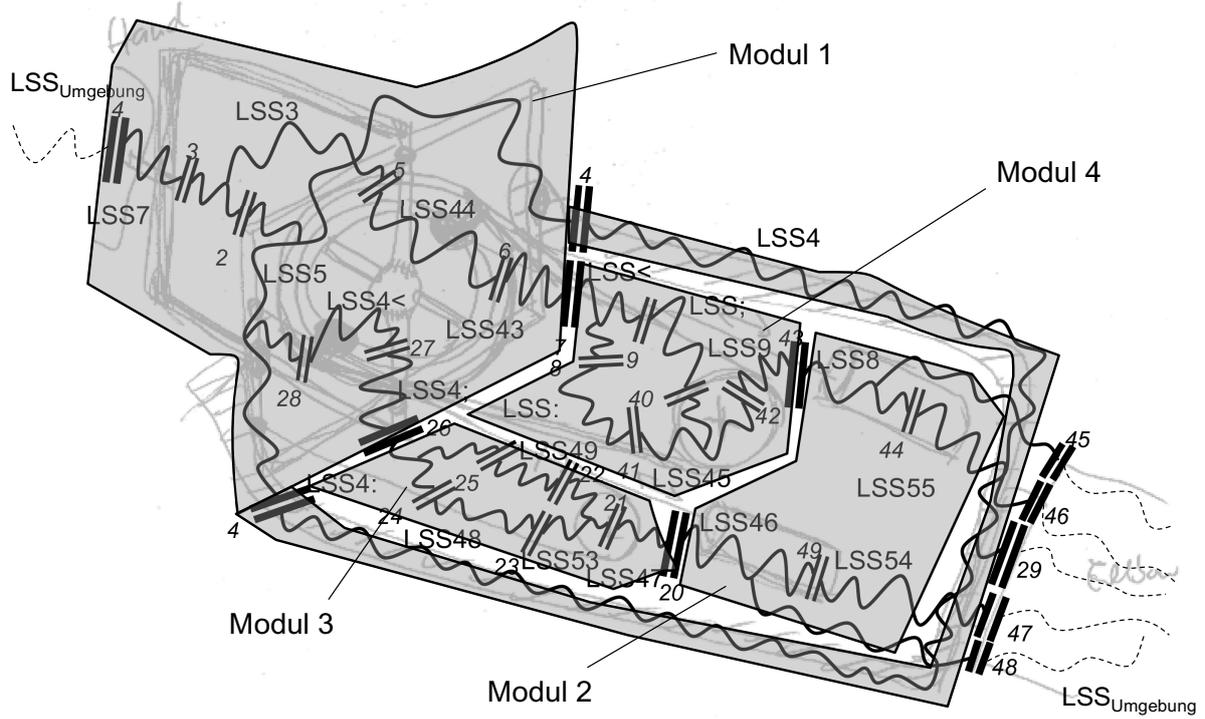


Abbildung 6-10. Ergebnis des Ansatzes

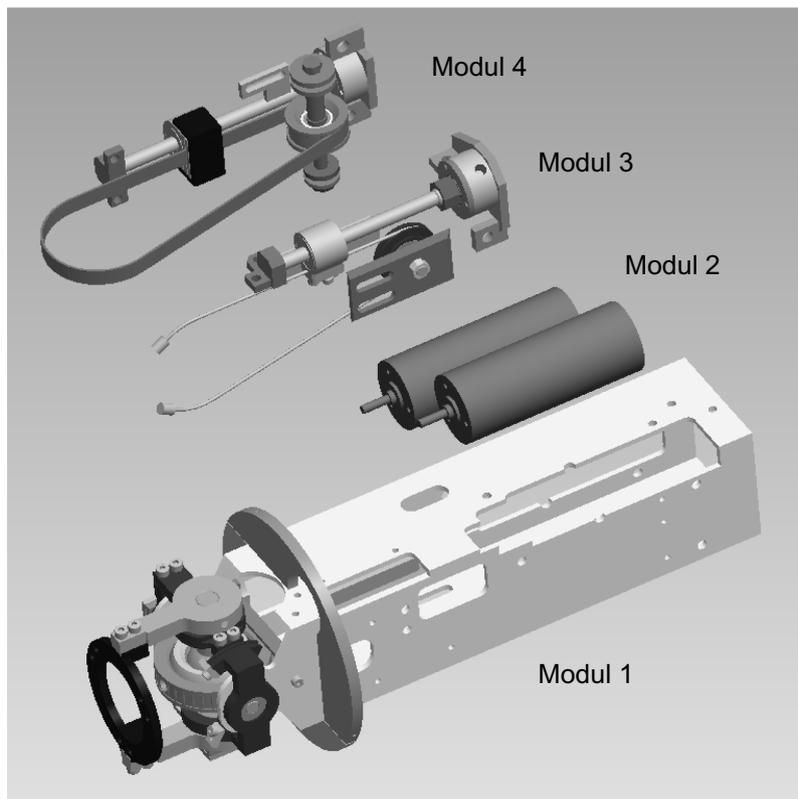


Abbildung 6-11. CAD-Modell des Roboterunterarms

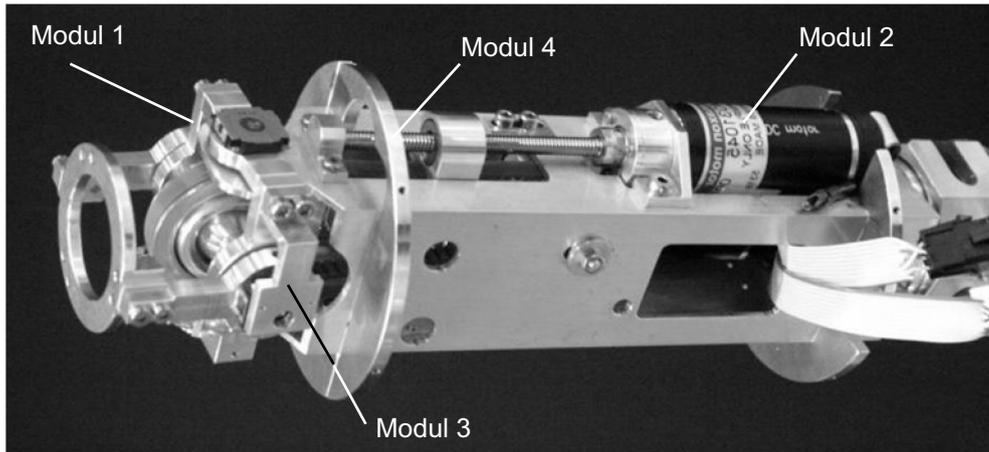


Abbildung 6-12. Der fertig aufgebaute Roboterunterarm

Das Analyseergebnis zeigt einige interessante Punkte, die wertvoll für die weitere Festlegung der Gestalt sind. Der Entwickler sollte eine Trennungsmöglichkeit zwischen den Modulen, die Wirkflächenpaare Nummer 1, 7, 13, 20 und 26 realisieren (siehe Abbildung 6-10). Die Hilfsfunktionen zur Verbindung der genannten WFP sollten eine Trennung ermöglichen. Die interne Architektur aller Module kann integral sein ermöglicht damit eine leichte und kompakte Gestaltung der Baugruppen. Nach der Produktarchitekturfestlegung wurde die Gestalt mit der vorgeschlagenen Architektur weiter konkretisiert und umgesetzt. Abbildung 6-12 zeigt den aufgebauten Roboterunterarm in seiner endgültigen Form.

## 6.2 Produktarchitekturfestlegung einer Bohrmaschine

Dieses Beispiel stellt die Festlegung der Produktarchitektur einer Bohrmaschine vor. Dazu wird angenommen, dass bereits ein Zielsystem für die Bohrmaschine erstellt wurde und dass das Ergebnis der Analyse der Funktionsstrukturen sowie die Prinzipskizze bereits das weitere Vorgehen zur Verfügung stehen. Dieses Beispiel fokussiert auf den Schritt der Produktarchitekturfestlegung, in dem die optimale Produktarchitektur für ein gegebenes Zielsystem identifiziert werden soll. Die Abbildung 6-13 zeigt die Prinzipskizze der Bohrmaschine. In diesem Beispiel wird angenommen bzw. ist aufgrund der Erfahrungen der Entwickler erwünscht, dass die Architektur in zwei Module und die verbleibenden Systemelemente unterteilt wird – nämlich in das Informationsverarbeitungsmodul, das Schaltermodul und die Plattform.

Mit Hilfe der in dieser Arbeit entwickelten Methode soll nun geprüft werden, ob diese Einteilung optimal für das Gesamtsystem ist.

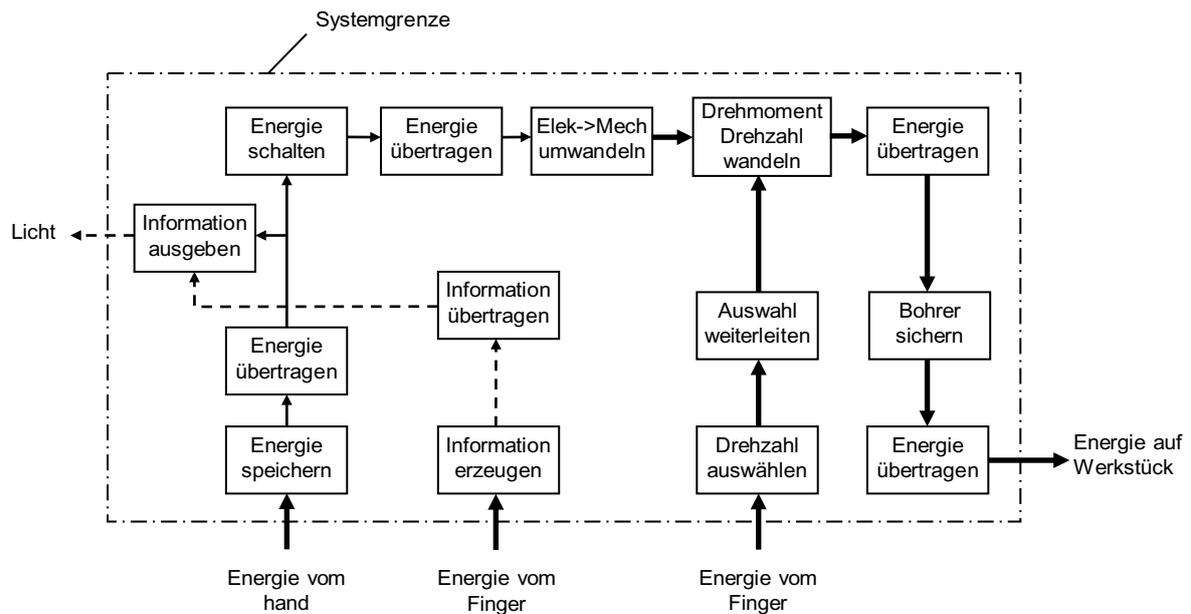


Abbildung 6-13. Funktionsstruktur der Bohrmaschine

Zu Beginn wird ein C&C-M erstellt. Alle Funktionen und Funktionsinteraktionen werden nun mit C&C-M Elementen modelliert und in einer Tabelle aufgelistet. Die Sortierung ist zu diesem Zeitpunkt vernachlässigbar und kann zunächst beliebig erfolgen.

Danach werden die Funktions- und Gestaltinformationen aus der Prinzipskizze in einer Darstellung (Abbildung 6-14) kombiniert. Die Positionen der LSS und WFP lassen sich aus der Skizze definieren. Es entsteht die kombinierte Information, die in der C&C-M Darstellung gezeigt wird. Abbildung 6-14 zeigt die C&C-M Darstellung der Bohrmaschine.

Im nächsten Schritt werden die LSS, die aufgrund der Vorgaben aus dem Zielsystem unbedingt in einem Bauteil gruppiert werden müssen, manuell gruppiert (Detail im Kapitel 5.1.1). In diesem Beispiel befinden sich die LSS 3 und 6 in einer LSS-Gruppe, da im Zielsystem vorgegeben ist, dass sie zusammen als ein Zukaufteil eingebunden werden sollen.

Es gilt, unterschiedliche Aspekte der Kompatibilitätsbeziehungen zur Produktarchitekturfestlegung zu betrachten. Diese Aspekte können aus dem Zielsystem entnommen werden oder, falls solche Vorgaben bisher noch nicht Teil des Zielsystems sind, müssen die Kompatibilitätsbeziehungen jetzt festgelegt und dem Zielsystem zugeführt werden.

LSS <sub>1</sub>	Funktion: Energie speichern WFP <sub>1</sub> , WFP <sub>Umgebung</sub>
LSS <sub>2</sub>	Funktion: Energie übertragen WFP <sub>2</sub> , WFP <sub>3</sub>
LSS <sub>3</sub>	Funktion: Information ausgeben WFP <sub>4</sub> , WFP <sub>5</sub> , WFP <sub>Umgebung</sub>
LSS <sub>4</sub>	Funktion: Information erzeugen WFP <sub>6</sub> , WFP <sub>Umgebung</sub>
LSS <sub>5</sub>	Funktion: Information übertragen WFP <sub>4</sub> , WFP <sub>6</sub>
LSS <sub>6</sub>	Funktion: Energie schalten WFP <sub>3</sub> , WFP <sub>5</sub> , WFP <sub>7</sub>
LSS <sub>7</sub>	Funktion: Energie übertragen WFP <sub>7</sub> , WFP <sub>8</sub>
LSS <sub>8</sub>	Funktion: Elek.->Mech umwandeln WFP <sub>8</sub> , WFP <sub>9</sub>
LSS <sub>9</sub>	Funktion: Drehmoment, Drehzahl wandeln WFP <sub>9</sub> , WFP <sub>10</sub> , WFP <sub>11</sub>
LSS <sub>10</sub>	Funktion: Auswahl weiterleiten WFP <sub>10</sub> , WFP <sub>12</sub>
LSS <sub>11</sub>	Funktion: Drehzahl auswählen WFP <sub>12</sub> , WFP <sub>Umgebung</sub>
LSS <sub>12</sub>	Funktion: Energie übertragen WFP <sub>11</sub> , WFP <sub>13</sub>
LSS <sub>13</sub>	Funktion: Bohrer sichern WFP <sub>13</sub> , WFP <sub>14</sub>
LSS <sub>14</sub>	Funktion: Energie übertragen WFP <sub>14</sub> , WFP <sub>Umgebung</sub>

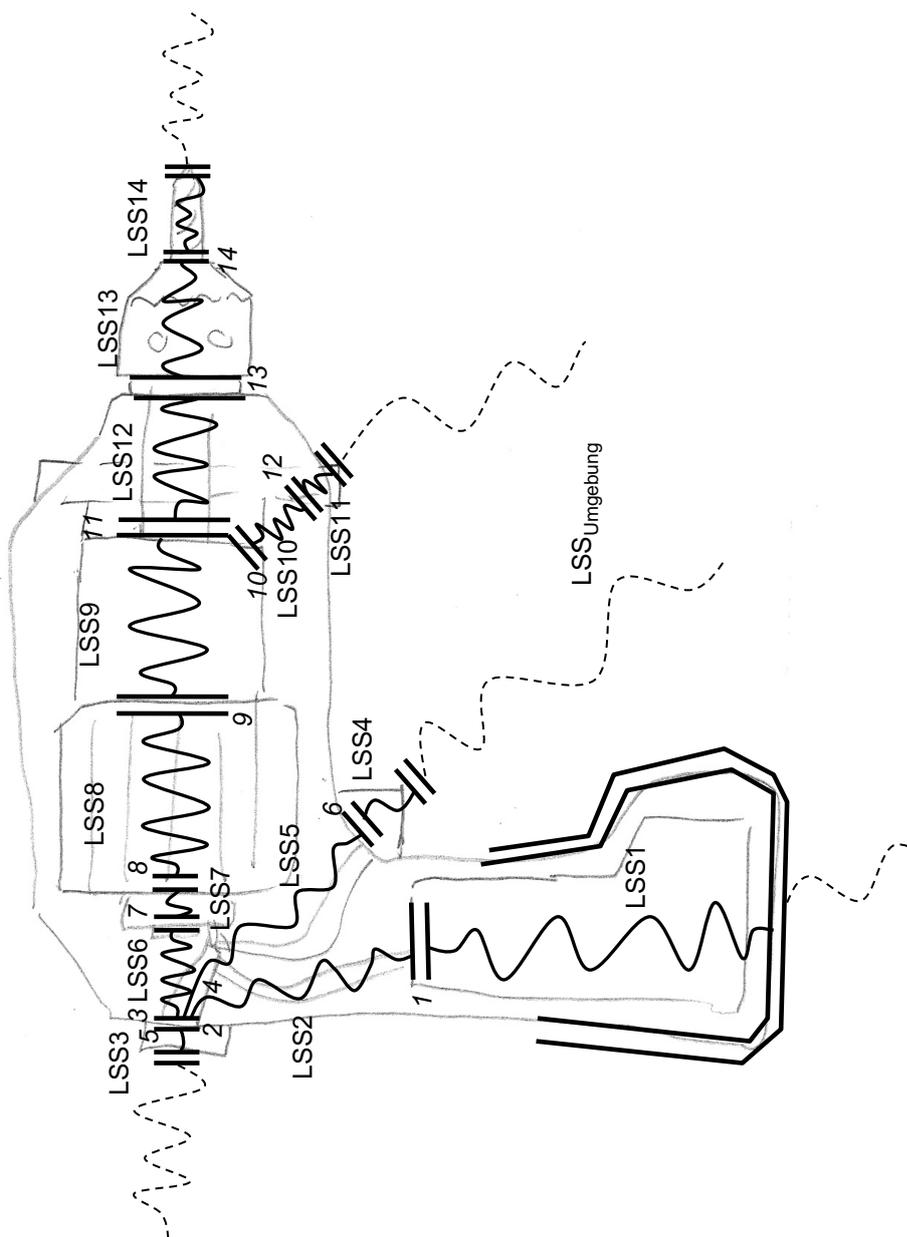


Abbildung 6-14. C&C-M Darstellung einer Bohrmaschine

In diesem Beispiel wird angenommen bzw. ist aus Erfahrung erwünscht, die Architektur in drei Module zu unterteilen – nämlich das Informationsverarbeitungsmodul, das Schaltermodul und das Plattformmodul. Somit sind wird von den Entwicklern die Unterteilung in zwei Module und die verbleibenden Systemelemente (vgl. folgende Tabelle) vorgeschlagen. Jetzt müssen die Entwickler festlegen, wie wichtig die einzelnen Beziehungen zwischen den Systemelementen sind. Angenommen, es wird entschieden, dass die funktionalen Beziehungen mit einem Gewichtungsfaktor von 2 (*wichtig*) eingehen und die LSS-Kompatibilitätsbeziehung mit einem Gewichtungsfaktor von 1 (*weniger wichtig*), dann ergibt sich, dass die Gewichtung der funktionalen Beziehung 2/3 oder ca. 0,67 ist und die der Kompatibilitätsbeziehung 1/3 oder ca. 0,33.

	Gruppe 1	Gruppe 2	Gruppe 3
Kompatibilitätsbeziehungen	LSS (3,6), 4, 5	LSS 9, 10, 11	LSS 1, 2, 7, 8, 12, 13, 14

Tabelle 6-2. Gruppierung der LSS aufgrund der Kompatibilitätsbeziehungen

Anschließend werden alle Beziehungen zwischen LSS ( $B_{LSS}$ ) in die C&C-M Dependency-Matrix eingetragen. Nach der Erstellung der Matrix sieht die neue Systemdarstellung wie in Abbildung 6-15 aus.

	LSS 1	LSS 2	LSS 3,6	LSS 4	LSS 5	LSS 7	LSS 8	LSS 9	LSS 10	LSS 11	LSS 12	LSS 13	LSS 14
LSS 1	1	0,67											
LSS 2	0,67	1	0,67										
LSS 3,6		0,67	1	0,33	1	0,67							
LSS 4			0,33	1									
LSS 5			1	1	1								
LSS 7		0,67				1	0,67						
LSS 8						0,67	1	0,67					
LSS 9							0,67	1	0,33	0,67			
LSS 10								1	1				
LSS 11								0,33	1				
LSS 12								0,67			1	0,67	
LSS 13											0,67	1	0,67
LSS 14											0,67	1	1

Abbildung 6-15. Die bewertete CDM der Bohrmaschine

Jetzt sind die LSS einfach entsprechend ihrer Nummerierung sortiert, und es ist noch keine sinnvolle Unterteilung in Module (Gruppen von LSS) erkennbar. Durch die Gleichung (5.7)

$$Z_{integral} = Z|_{N_m=1} = \frac{\sum_{i=1}^{N_{LSS}} \sum_{j=1}^{N_{LSS}} B_{LSSij}}{N_{LSS}^2} = \frac{\sum_{i=1}^{13} \sum_{j=1}^{13} B_{LSSij}}{13^2}$$

ergibt sich ein Zielfunktionswert von  $\frac{20,4}{13^2}$  oder 0,121 für die integrale Architektur.

Nach der Optimierung der CDM durch den genetischen Algorithmus folgt als Ergebnis, dass dieses Produkt in drei Module aufgeteilt werden sollte. Die zugehörigen LSS jedes Moduls können einfach aus der CDM in Abbildung 6-16 abgelesen werden.

	LSS 1	LSS 2	LSS 7	LSS 8	LSS 12	LSS 13	LSS 14	LSS 3,6	LSS 4	LSS 5	LSS 9	LSS 10	LSS 11
LSS 1	0,67												
LSS 2	0,67							0,67					
LSS 7			0,67					0,67					
LSS 8			0,67								0,67		
LSS 12					0,67						0,67		
LSS 13					0,67	0,67							
LSS 14					0,67								
LSS 3,6	0,67	0,67						0,33	1				
LSS 4								0,33	1				
LSS 5								1	1				
LSS 9			0,67	0,67								1	0,33
LSS 10												1	1
LSS 11											0,33	1	

Abbildung 6-16. Das Ergebnis der optimierten CDM

Der maximale Zielfunktionswert beträgt dann 0,2966 wie folgt berechnet.

$$Z = \frac{\sum_{k=1}^{N_m} \sum_{i=n_k}^{m_k} \sum_{j=n_k}^{m_k} B_{LSSij}}{N_m} - 2 \times \frac{\sum_{k=1}^{N_m-1} \sum_{p=k+1}^{N_m} \sum_{i=n_k}^{m_k} \sum_{j=n_p}^{m_p} B_{LSSij}}{N_m}$$

$$= \frac{\sum_{i=1}^7 \sum_{j=1}^7 B_{LSSij}}{(8-1+1)^2} + \frac{\sum_{i=8}^{10} \sum_{j=8}^{10} B_{LSSij}}{(10-8+1)^2} + \frac{\sum_{i=11}^{13} \sum_{j=11}^{13} B_{LSSij}}{(13-11+1)^2} - 2 \times \frac{\sum_{i=1}^7 \sum_{j=8}^{10} B_{LSSij}}{(10-8+1)(7-1+1)} + \frac{\sum_{i=1}^7 \sum_{j=11}^{13} B_{LSSij}}{(13-11+1)(7-1+1)}$$

$$= \frac{5,36}{7^2} + \frac{4,66}{3^2} + \frac{4,66}{3^2} - 2 \times \frac{1,34}{3 \times 7} + \frac{1,34}{3 \times 7} = 0,2966$$

In der folgenden Abbildung wird die C&C-M Darstellung statt der Matrixdarstellung verwendet. Diese ist etwas leichter verständlich als die Matrixdarstellung.

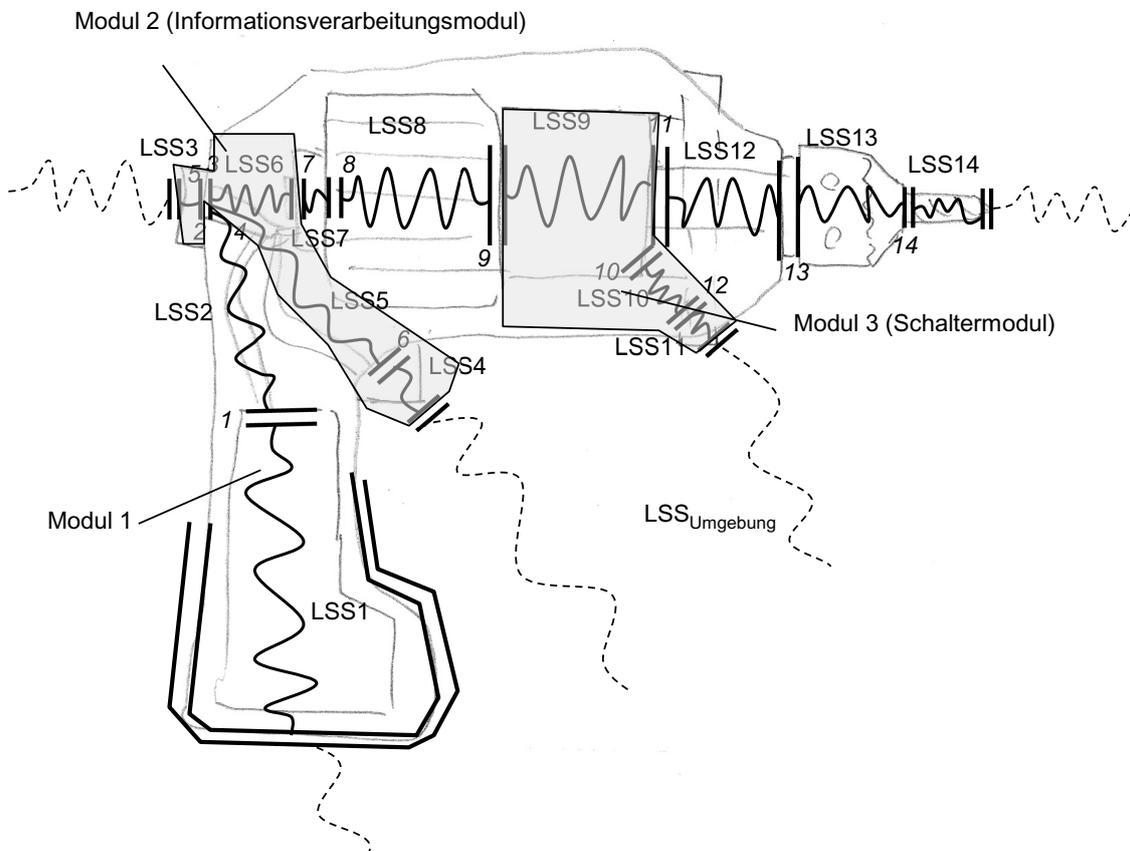


Abbildung 6-17. C&C-M Darstellung der vorgeschlagenen Produktstruktur für weitere Aktivitäten

Durch die Anwendung der in dieser Arbeit entwickelten Methode konnte eine vorgeschlagene Produktarchitektur sowie eine Funktionsstruktur analysiert werden. Es konnte überprüft werden, ob der ursprüngliche Vorschlag optimal für das Gesamtsystem ist.

### 6.3 Produktarchitekturfestlegung für einen humanoiden Roboterthorax

Das Beispiel in diesem Abschnitt unterscheidet sich von den ersten beiden Beispielen dadurch, dass die Prinzipskizze nicht durch eine Handskizze vorgegeben wird, sondern durch die rechnergestützte Methode der Topologieoptimierung. In diesem Beispiel wird die Methode zur Unterstützung der Architekturfestlegung bei der Aktivität der Modellierung von Prinzip und Gestalt des Thorax des humanoiden Roboters ARMAR III des SFB 588 (siehe Abbildung 6-18) angewendet.

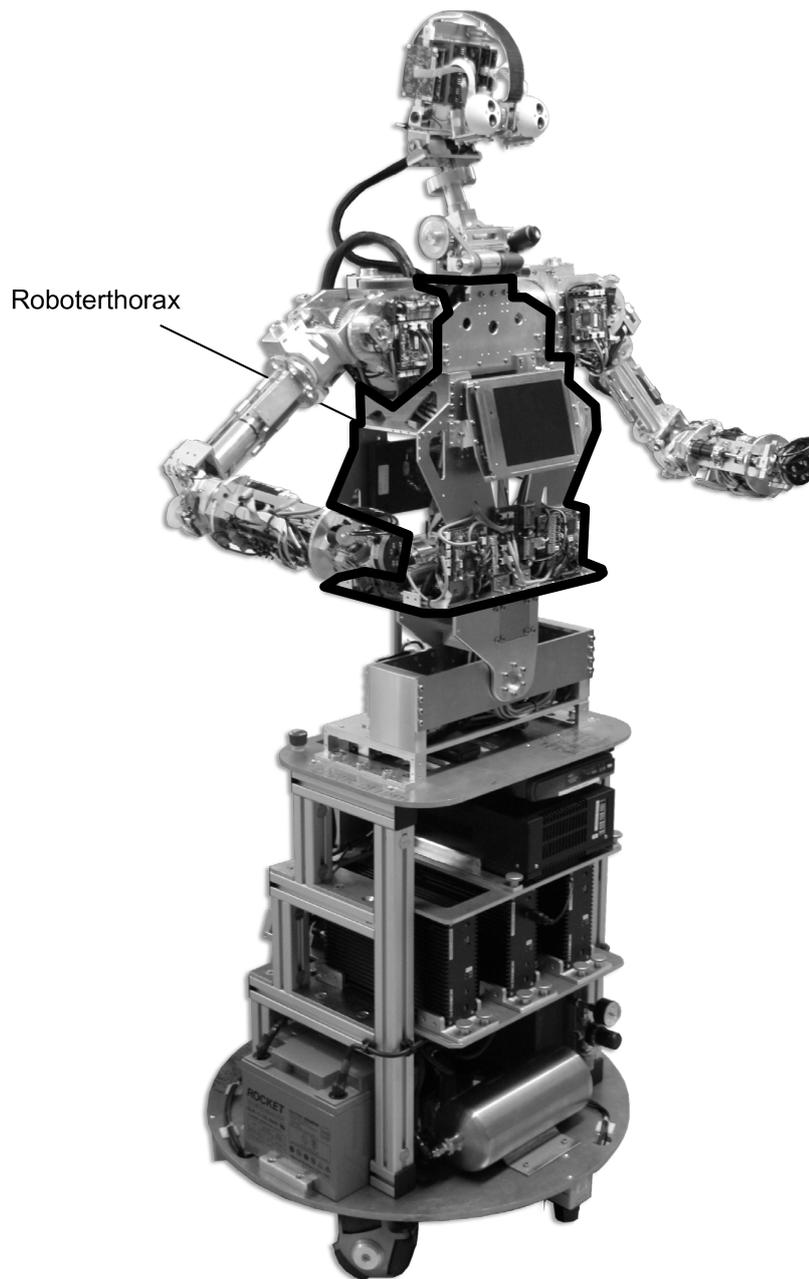


Abbildung 6-18. Der Roboterthorax als zentrale Tragstruktur des Roboters ARMAR III (SFB 588)

Verschiedene Komponenten wie Arme, Hals oder Torsogelenk des humanoiden Roboters ARMAR III werden durch den Thorax verbunden, wobei verschiedene elektrische und mechanische Komponenten wie z. B. UCOM<sup>189</sup>-Mikrokontroller oder die Antriebseinheiten des Ellenbogengelenks ebenfalls in den Thorax integriert werden müssen. Aufgrund zu erwartender Änderungen an der Elektronik,

---

<sup>189</sup> Regenstein et al. 2007

Reparaturen oder Wartung muss es möglich sein, den Thorax oder Teile des Thorax leicht zu demontieren. Aus diesem Grund soll das Zielsystem zur Produktarchitekturfestlegung nicht nur die Funktionsintegration, sondern auch den Wartungs-/Zugänglichkeitsaspekt berücksichtigen.



Abbildung 6-19. Ergebnis der Topologieoptimierung des Thorax<sup>190</sup>

Das Ergebnis einer Topologieoptimierung<sup>191</sup> ersetzt die Skizze bei diesem Anwendungsbeispiel für die Methode. Die Struktur wird für verschiedene vordefinierte Lastfälle optimiert, wobei sie das Ergebnis der Topologieoptimierung keinen Architekturvorschlag ergibt, sondern nur eine einzige integrale Tragstruktur<sup>192</sup> (vgl. das Beispiel in Abbildung 6-19). Üblicherweise lässt sich das Ergebnis der topologisch optimierten Struktur bei den weiteren Aktivitäten einfach intuitiv konstruktiv umsetzen und kann intuitiv in mehrere Teile zerlegen werden, falls es nicht oder nur mit großem Aufwand als ein Bauteil gefertigt werden kann. Bei einer intuitiven Unterteilung ist es nicht sicher, dass die Unterteilung, die die Produktarchitektur definiert, optimal ist. Mit der in dieser Arbeit entwickelten Methode wird daher die Architektur des strukturoptimierten Thorax wie in Kapitel 5.1, 5.2 und 5.3 analysiert und optimiert.

---

<sup>190</sup> Albers et al. 2007

<sup>191</sup> Albers et al. 2007

<sup>192</sup> Siehe Definition Tragstruktur in Abschnitt 2.5

Mithilfe der Struktur in Abbildung 6-19 ist es möglich, ein C&C-M des topologisch optimierten Thorax zu erstellen und mit der Darstellung der Struktur zu kombinieren. Allerdings kann es beim direkten Skizzieren der LSS und WFP auf solche 3-D-Darstellungen möglicherweise schwer fallen, zu erkennen, welche Bauteile zu welchen LSS gehören. In diesem Fall ist eine Funktionsstruktur eine gute weitere Darstellungsmöglichkeit, um das modellierte System einfacher zu verstehen. Zum Beispiel kann man in Abbildung 6-23 die Position der LSS16 sehr schwer erkennen, jedoch sieht man in der Funktionsstruktur in Abbildung 6-22, dass die Funktion der Leitstützstruktur 16 „Energie zwischen Hals, Armen und Torsogelenk leiten“ ist. Damit kann der Betrachter leichter verstehen, dass die LSS 16 in Abbildung 6-23 sich unten im Thorax befindet und dass diese LSS 16 den Thorax mit dem Torsogelenk verbindet.

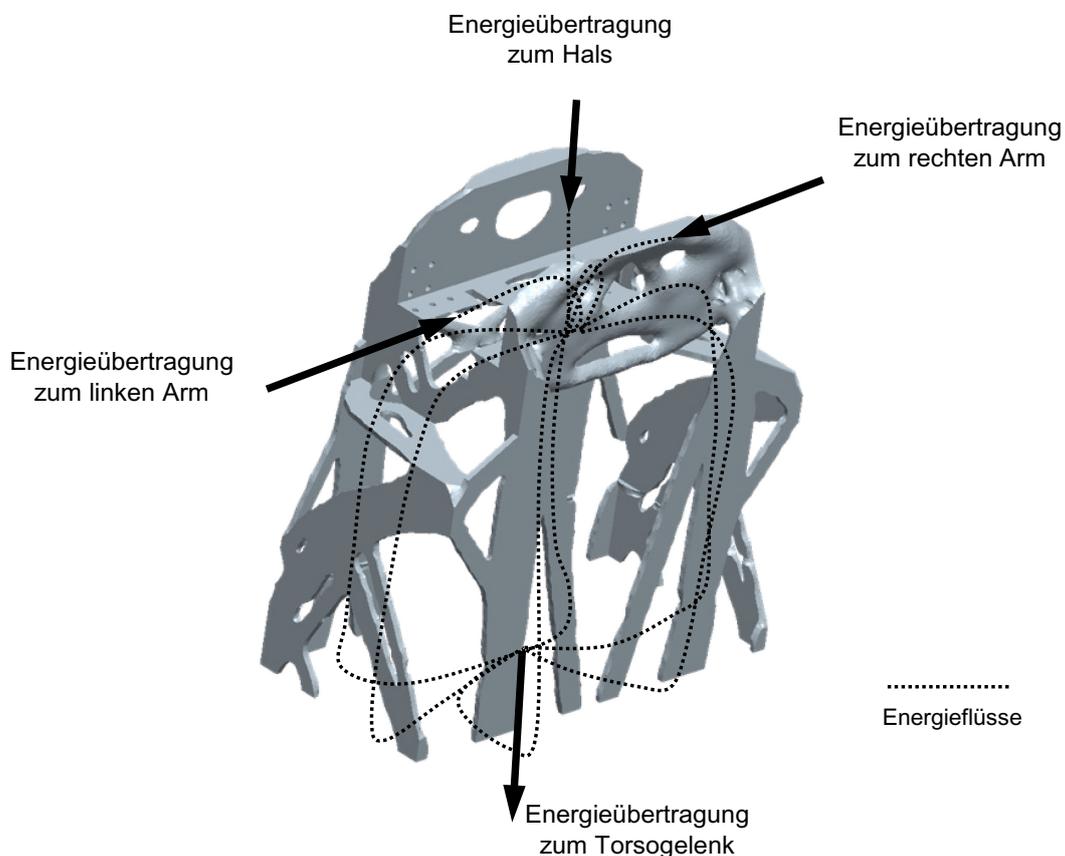


Abbildung 6-20. Energieflüsse in der Tragstruktur des Robotertorax

Die Energieflüsse zwischen den Armen, dem Hals und dem Torsogelenk werden in Abbildung 6-20 dargestellt. Die Energieflüsse werden zum einen zusammen geführt und werden dann wieder in sechs Richtungen verteilt – nämlich nach vorne links, vorne rechts, hinter links, hinter rechts und links und rechts. Danach werden sie noch einmal zusammengeführt und zum Torsogelenk weitergeleitet.

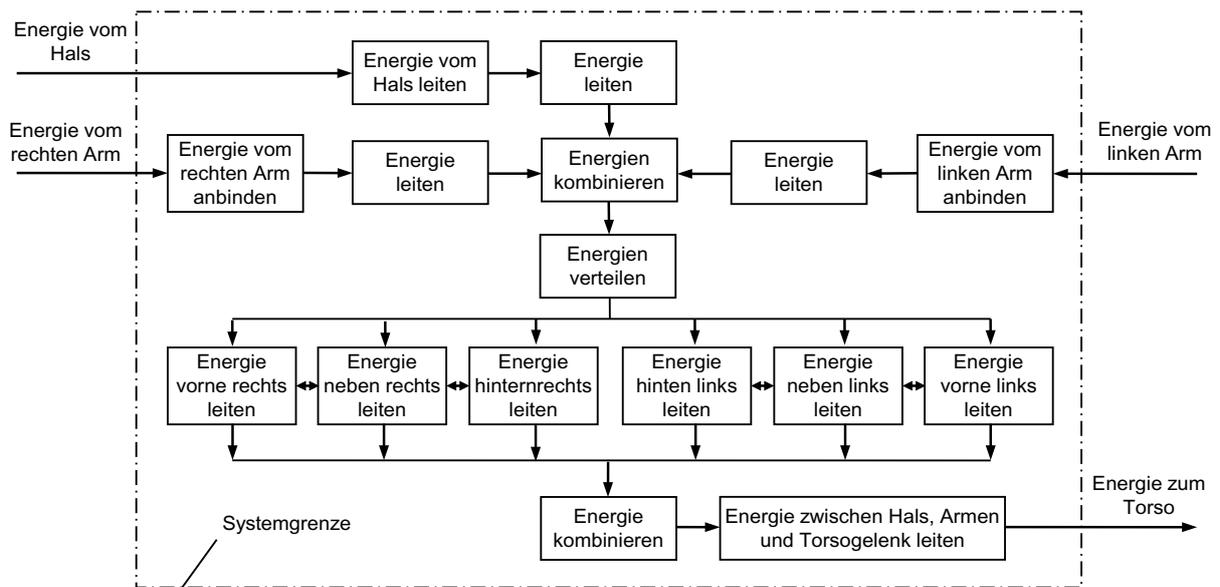


Abbildung 6-21. Funktionsstruktur des Thorax

Alle Funktionen auf Energieflüsse beruhen lassen sich sammeln und verknüpfen, wie in Abbildung 6-21 dargestellt. Es gibt insgesamt 16 Funktionen in dieser Funktionsstruktur. Nach der Erstellung der Funktionsstruktur werden sie mit der Geometrieinformation kombiniert und daraus ergibt sich ein C&C-M.

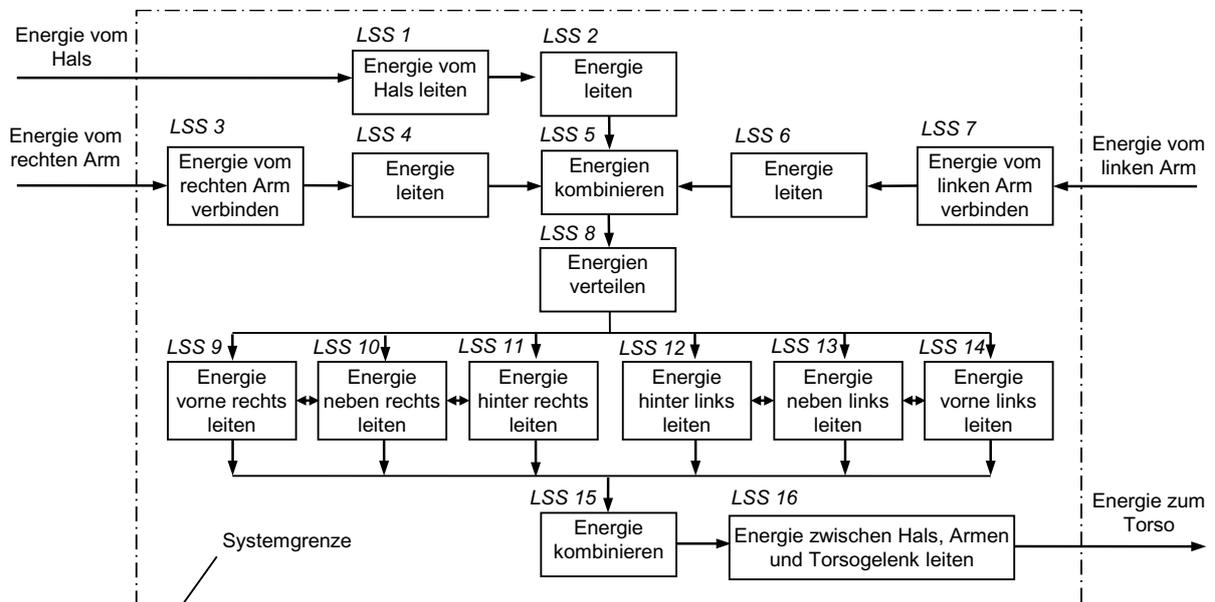


Abbildung 6-22. Funktionsstruktur mit den LSS

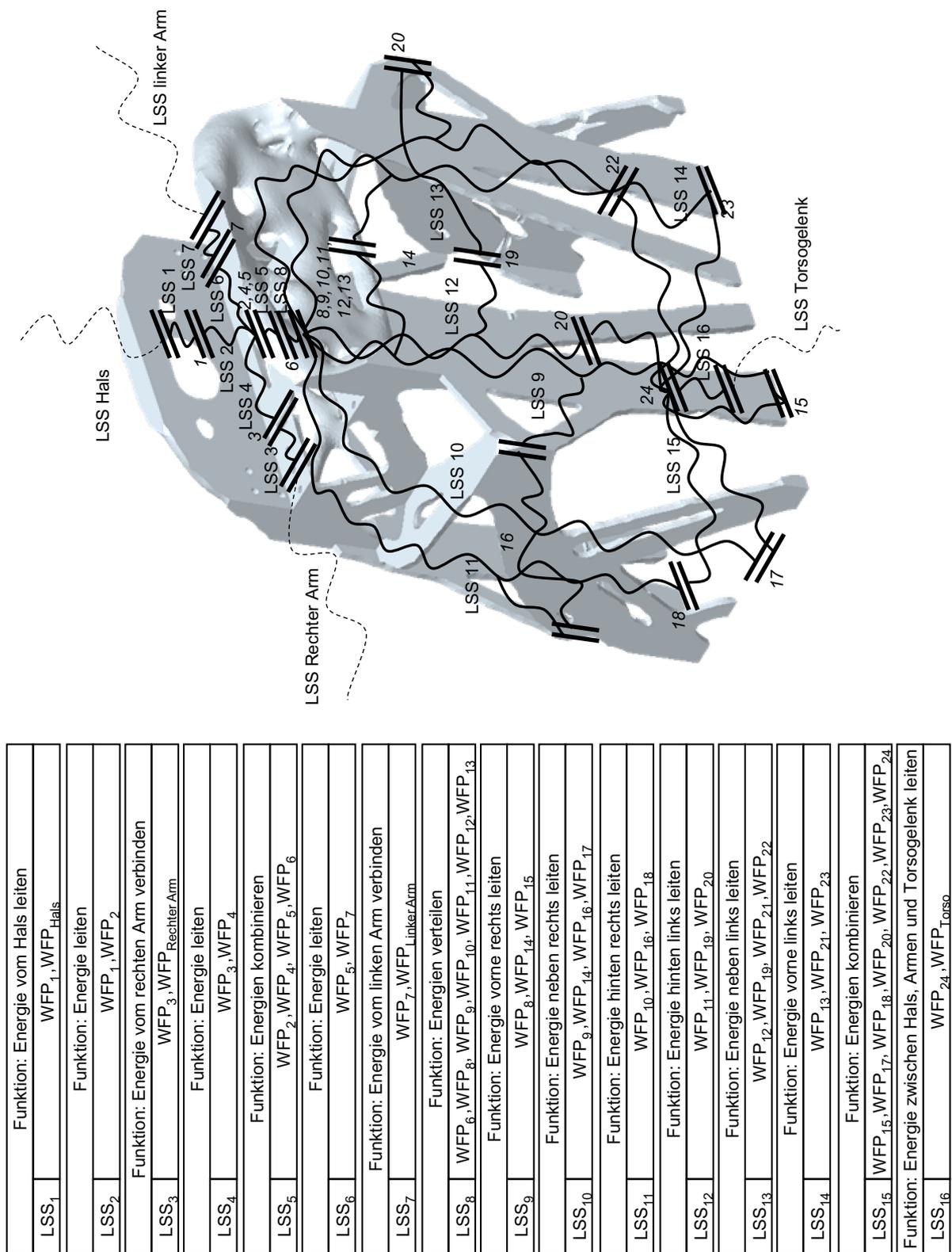


Abbildung 6-23. Das C&C-M des Roboterthorax

Alle 16 LSS und 24 WFP sind in Abbildung 6-22 und Abbildung 6-23 dargestellt. Die Darstellung im 3D-Raum ist nicht ganz so anschaulich, weil das Modell räumlich dargestellt wird und dabei einige LSS aufeinander zu liegen kommen. Die LSS und WFP werden nun in die CDM übertragen. Abbildung 6-24 zeigt das Ergebnis dieser Transformation. Diese CDM ist nur ein Modell der funktionalen Beziehungen zu sehen, noch ohne Berücksichtigung der Kompatibilitätsbeziehungen ( $K_{LSS}$ ).

	LSS 1	LSS 2	LSS 3	LSS 4	LSS 5	LSS 6	LSS 7	LSS 8	LSS 9	LSS 10	LSS 11	LSS 12	LSS 13	LSS 14	LSS 15	LSS 16
LSS 1	X															
LSS 2		X														
LSS 3			X													
LSS 4				X												
LSS 5					X											
LSS 6						X										
LSS 7							X									
LSS 8								X								
LSS 9									X							
LSS 10										X						
LSS 11											X					
LSS 12												X				
LSS 13													X			
LSS 14														X		
LSS 15															X	
LSS 16																X

Abbildung 6-24. CDM des Roboterthorax

Aufgrund des Wartungsaspekts sollten drei LSS – nämlich 9, 11 und 12 – zusammen in einem Modul gruppiert werden. Diese Gruppe der LSS wird dann als eine Öffnungsmöglichkeit für die Thorax vorgesehen. Die meisten elektronischen und mechanischen Komponenten sollen innerhalb des Thorax angeordnet werden. Deshalb kann es schwierig werden, wenn diese ohne eine vorgesehene Öffnungsmöglichkeit des Thorax reparieren oder zu entnehmen. Es ist für den Entwickler ohne weitere Hilfsmittel nicht einfach zu entscheiden, ob es die sinnvoll ist, ein Modul der vorgeschlagenen Architektur weiter zu unterteilen. Hier soll nun systematisch die Frage beantwortet werden, ob es andere gute mögliche Lösungen oder eine optimale Lösung gibt.

Durch die Bewertung und Optimierung der CDM lässt sich diese Frage beantworten.

Wenn die funktionalen Beziehungen als genauso eingeschätzt werden wie die Kompatibilitätsbeziehungen (Gewichtungsfaktor gleich 0,5 für beide Beziehungsarten) dann schlägt der Algorithmus vor, dass es sich nicht lohnt, noch ein weiteres Modul für den Wartungsaspekt zu definieren, da dieses Modul zahlreiche Interaktionen außerhalb des Moduls verursachen und damit die Produktarchitektur des Gesamtsystems verschlechtern würde. Bei einer schwächeren Berücksichtigung der Funktion falls die Kompatibilitätsbeziehung des Zielsystems wichtiger ist als die funktionalen Interaktionen, wie zum Beispiel 0,7 zu

0,3 (vgl. Abbildung 6-26), schlägt der Algorithmus vor, dass die Architektur aus zwei Modulen bestehen sollte. Ein Modul umfasst dann die LSS 9, 11 und 12, das zweite die verbleibenden LSS.

Dieses Beispiel zeigt, wie wichtig die Verteilung der Faktoren bzw. die Gewichtung der unterschiedlichen Interaktionsarten ist. Die optimale Produktarchitektur hängt nicht nur von der Anordnung der LSS und der Modulanzahl ab, sondern auch davon, wie die Beziehungsfaktoren gewichtet werden.

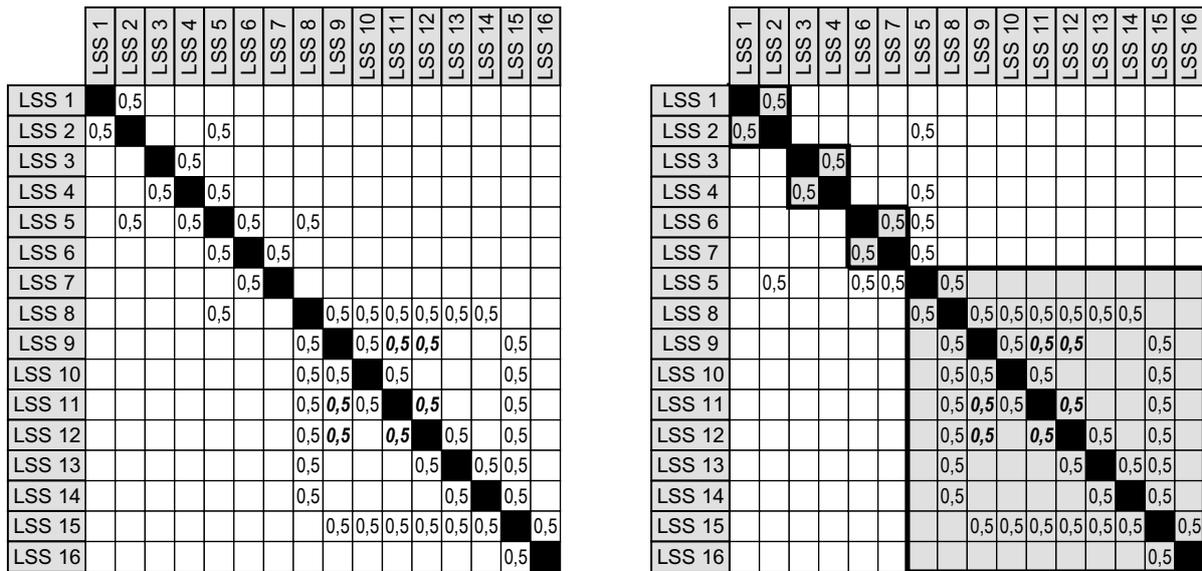


Abbildung 6-25. Mit  $F_{WFP} = K_{LSS} = 0,5$  Links: integrale Architektur ( $Z = 0,105$ ), Rechts: optimale Architektur ( $Z_{max} = 0,203$ )

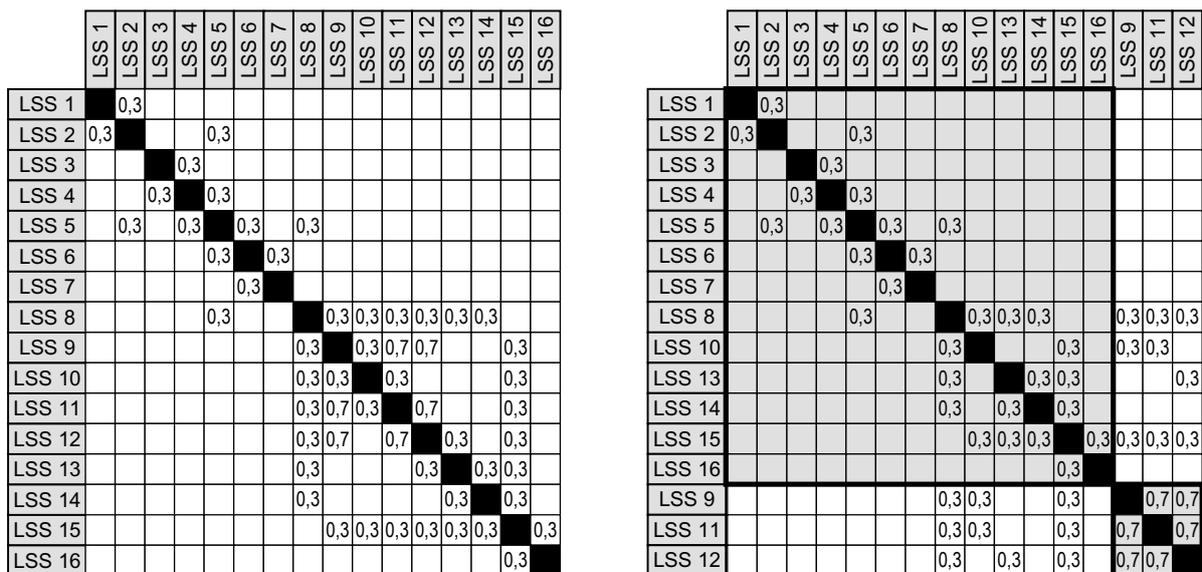


Abbildung 6-26. Mit  $F_{WFP} = 0,3$  und  $K_{LSS} = 0,7$  Links: integrale Architektur ( $Z = 0,073$ ), Rechts: optimale Architektur ( $Z_{max} = 0,191$ )

Durch die Anwendung der in dieser Arbeit entwickelten Methode wurde festgestellt, dass die Architektur des Thorax für den Roboter ARMAR III in vier Module geteilt werden sollte, wenn die Wichtigkeiten der funktionalen Interaktionen und der Wartungsaspekte aus dem Zielsystem als gleichwertig angesehen werden. Falls der Wartungsaspekt als wichtiger angesehen wird, ist eine Architektur aus nur zwei Modulen (vgl. Abbildung 6-27) vorteilhafter.

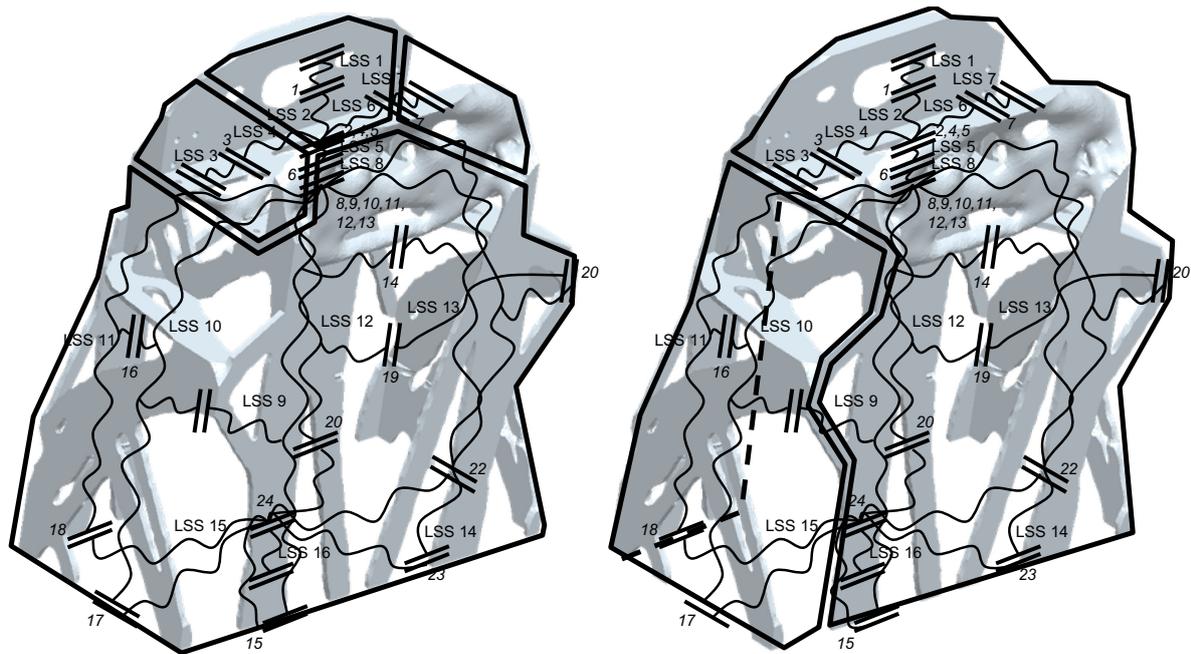


Abbildung 6-27. optimale Architekturen des Thorax, fürs a)  $F_{WFP} = K_{LSS} = 0,5$  (links) und b)  $F_{WFP} = 0,3, K_{LSS} = 0,7$  (rechts)

---

## 7 Zusammenfassung und Ausblick

Die Produktarchitektur ist ein wesentlicher Erfolgsfaktor für die Produkte eines Unternehmens und damit für das ganze Unternehmen. Dennoch wird bei der Festlegung von Produktarchitekturen oft nicht ausreichend sorgfältig analysiert und nicht systematisch, nach quantitativen Kriterien und rechnerunterstützt vorgegangen. Manchmal wird sogar die Produktarchitektur einer vorherigen Version eines Produkts oder eines Konkurrenzprodukts einfach übernommen. Besonders bei Produktentwicklungsprojekten von komplexen Systemen, in denen die Bestimmung der funktionalen Anforderungen im Zielsystem schwierig ist und mit großem Aufwand verfolgt werden muss, wird die Produktarchitektur oft vernachlässigt.

Die Idee, die Produktarchitektur möglichst früh im Projektverlauf festzulegen, ist nicht neu. Sie wurde bereits in mehreren bekannten Entwicklungsprozessen angewandt und auch in einigen Forschungsberichten veröffentlicht. Die bisherigen relevanten Forschungen lassen sich in zwei Gruppen einteilen:

- Intuitive Ansätze, die sowohl Funktionsinformation als auch das Layout eines Systems berücksichtigen (ULRICH und EPPINGER<sup>193</sup>, VAN WIE<sup>194</sup> und VDI-Richtlinie 2221<sup>195</sup>)
- Quantitative Ansätze auf der Grundlage von Integrationsanalysen von Funktionsinformation (STONE<sup>196</sup>, HÖLTTÄ ET AL.<sup>197</sup> und ALLEN und CARLSON-SKALAK<sup>198</sup>)

Bislang sind diese beiden Alternativen unabhängig voneinander und besitzen jeweils ihre eigenen Vorteile und Nachteile. Allerdings kann die Umsetzung der Ansätze in der ersten Gruppe Schwierigkeiten bereiten, insbesondere wenn viele Faktoren zu beachten sind oder wenn es sich um Neukonstruktionen handelt und das Produkt noch unbekannt ist. Andererseits sind die Ansätze in der zweiten Gruppe nicht gut für die Analyse der Gestaltinformation geeignet. Dies kann zu einem Konflikt zwischen einem analysierten Ergebnis und der Gestaltung führen.

---

<sup>193</sup> Ulrich/Eppinger 2008

<sup>194</sup> Van Wie 2002

<sup>195</sup> VDI-Richtlinie 2221 1993

<sup>196</sup> Stone 1997

<sup>197</sup> Hölttä et al. 2005

<sup>198</sup> Allen/Carlson-Skalak 1998

Zwei wesentliche Gründe, die die beiden zuvor erläuterten Schwierigkeiten verursachen, sind folgende:

- Gestaltinformation und Funktionsinformation sind getrennt oder nicht ausreichend verbunden.
- Die Festlegung der Produktarchitektur ist nicht-quantitativ bestimmt, was die Gestaltinformation betrifft.

Ziel der vorliegenden Arbeit ist die Entwicklung einer Methode mit dazugehörigem Hilfswerkzeug, die sowohl Funktionsinformation als auch Gestaltinformation quantitativer Art zur Festlegung der Produktarchitektur früh im Produktentwicklungsprojekt nutzen.

Das Konzept zur Lösung der Problemstellung wird in zwei Teile unterteilt. Der erste Teil ist das Verbinden von Gestalt- und Funktionsinformation und im zweiten Teil wird die Systemintegration analysiert. Im ersten Teil werden die Modellelemente (Leitstützstruktur und Wirkflächenpaar) des Contact and Channel Models (C&C-M) nach ALBERS<sup>199</sup> und eine Modellierungshypothese angewendet. Die Modellierung mittels C&C-M kann das Systemverständnis durch Kennzeichnung der Orte der Funktionserfüllung in der Darstellung einer prinzipiellen Lösung unterstützen. Der zweite Teil konzentriert sich auf die Suche nach einer optimalen Lösung auf Basis des im ersten Teil generierten C&C-M. Eine Matrixdarstellung, ein Bewertungsvorgehen und ein genetischer Algorithmus werden hierfür genutzt. Die Produktarchitekturfestlegung eines Roboterunterarms, einer Bohrmaschine und eines humanoiden Roboterthorax werden als Beispiele vorgestellt.

Die frühzeitige Definition der Produktarchitektur im Entwicklungsprojekt besitzt die vorher beschriebenen Vorteile, jedoch basiert diese Entscheidung auf Informationen, die noch veränderlich sind. Während der Iterationen, z. B. durch die Veränderung der Randbedingungen in späteren Aktivitäten oder die Modifikation der Prinzipskizze, könnte sich die optimale Produktarchitektur ändern und daher ist es wichtig, dass die Produktarchitektur schnell aktualisiert werden kann, und das funktioniert am besten über eine Rechnerunterstützung für die Methode. Die Aktualisierung lässt sich in der hier vorgeschlagenen Methode durch die Rechnerunterstützung mit gespeicherter CDM und einem Optimierungsalgorithmus ohne großen Aufwand durchführen.

Weitere Forschungsarbeiten sollten sich mit der Entwicklung einer einfach und intuitiv zu bedienenden Software, einem effizienten Suchalgorithmus und mit der Verbesserung der Methode im Umgang mit Restriktionen beschäftigen. Die

---

<sup>199</sup> Albers et al. 2008a

Empfindlichkeit der Methode hinsichtlich Änderungen der Gewichtungsfaktoren bei mehreren gleichzeitig zu berücksichtigenden Zielen für die Festlegung der Produktarchitektur sollte weiter untersucht werden. Außerdem ist die Implementierung der Methode für dreidimensionale Modelle, die für die Einbindung der Methode in moderne 3D-CAD-Systeme notwendig ist, ein weiteres anspruchsvolles Ziel.

---

## 8 Literaturverzeichnis

- Albers 2010** Albers, Albers: Five Hypotheses about Engineering Processes and their Consequences. In: *International Symposium series on Tools and Methods of Competitive Engineering – TMCE 2010* (Ancona 2010)
- Albers et al. 2009a** Albers, Albert; Sauter, Christian; Maier, Thomas; Geier, Martin und Otnad, Jens: Academic Engineering Design Education In *A Realistic Environment. ASME 2009 - International Design Engineering Technical Conferences & Computers and Information in Engineering Conference, 2009*
- Albers et al. 2009b** Albers, Albert ; Sedchaicharn, Korkiat ; Sauter, Christian ; Burger Wolfgang: A Method for Defining a Product Architecture Early in Product Development Using the Contact and Channel Model. In: *17<sup>th</sup> International Conference on Engineering Design – ICED'09* (Stanford 2009)
- Albers et al. 2008a** Albers, Albert ; Burkardt, Norbert ; Sauter, Christian ; Sedchaicharn, Korkiat: A Modularization Method in the Early Phase of Product Development. In: *International Design Conference – Design 2008* (Dubrovnik 2008).
- Albers et al. 2008b** Albers, Albert ; Sedchaicharn, Korkiat ; Sauter, Christian ; Burger Wolfgang: An Approach for the Modularization of a Product Architecture in Redesign Processes of Complex Systems. In: *ASME International Design Engineering Technical Conferences & Computers and Information in Engineering Conference – DETC/CIE 2008* (New York 2008).
- Albers et al. 2007** Albers, Albert ; Otnad, Jens ; Weiler, Helge ; Häussler Pascal: Methods for Lightweight Design of Mechanical Components in Humanoid Robots. In: *International Conference on Humanoid Robots – Humanoid 2007, IEEE-RAS* (Pennsylvania 2007).
- Albers / Meboldt 2007** Albers, A.; Meboldt, M.: iPeMM- Integrated Product Development Process Management Model, Based on Systems Engineering and Systematic Problem Solving. In: *Proceedings of the 16th International Conference on Engineering Design - ICED 07*, Paris, France, 2007.
- Albers et al. 2006** Albers, Albert ; Brudniok, Sven ; Otnad, Jens ; Sauter, Christian ; Sedchaicharn, Korkiat: Upper Body of a new Humanoid Robot – the Design of ARMAR III. In: *International Conference on Humanoid Robots – Humanoids 2006, IEEE-RAS* (Genua 2006).
- Albers / Meboldt 2006** Albers, Albert ; Meboldt, Mirko: A new Approach in Product Development, Based on Systems Engineering and Systematic Problem Solving. In: *Applied Engineering Design Science – AEDS 2006* (Pilsen 2006). : S.5 – 12
- Albers et al. 2005** Albers, Albert ; Burkardt, Norbert ; Ohmer, Manfred: Approaches for the Synthesis of Technical Systems within the Contact and Channel Model C&CM. In: *International Conference on Engineering Design – ICED 05* (Melbourne 2005).
- Albers / Matthiesen 2002** Albers, Albert ; Matthiesen, Sven: Konstruktionsmethodisches Grundmodell zum Zusammenhang von Gestalt und Funktion technischer Systeme - Das Elementmodell "Wirkflächenpaare & Leitstützstrukturen" zur Analyse und Synthese technischer Systeme. In: *Konstruktion - Zeitschrift für Produktentwicklung und Ingenieur-Werkstoffe/* Band 54 (2002), Nr. 7/8 – 2002, S.55 – 60

- Allen / Carlson-Skalak 1998** Allen, Kevin R. ; Carlson-Skalak, Susan: Defining Product Architecture During Conceptual Design. In: *ASME Design Engineering Technical Conferences – DETC98* (Atlanta 1998).
- Anderson 2004** Anderson, David: *Design for Manufacturability & Concurrent Engineering How to Design for Low Cost, Design in High Quality, Design for Lean Manufacture, and Design Quickly for Fast Production*. California : CIM Press, 2004
- Baker 1987** Baker, James E.: Reducing Bias and Inefficiency in the Selection Algorithm. In: *International Conference on Genetic Algorithms – ICGA* (New Jersey 1987) S.14 – 21
- Birkhofer 2000** Birkhofer, Herbert ; Berger, B.: *Produktentwicklung I*, Wintersemester 2000/2001; Darmstadt, 2000
- Braess / Seiffert 2007** Braess, Hans-Hermann.; Seiffert, Ulrich: *Vieweg Handbuch Kraftfahrzeugtechnik ATZ-MTZ Fachbuch*. 5. Aufl. Wiesbaden : Friedr. Vieweg & Sohn, 2007
- Browning 2001** Browning, Tyson R.: Applying the Design Structure Matrix to System Decomposition and Integration Problems - A Review and New Directions. In: *IEEE Transactions on Engineering Management* (2001). S.292 – 306
- Buttelmann / Lohmann** Buttelmann, Maik ; Lohmann , Boris: Optimierung mit Genetischen Algorithmen und eine Anwendung zur Modellreduktion. In: *Automatisierungstechnik* (2004), S.1521 –163
- Chen et al. 2005** Chen, Li ; Ding, Zhendong ; Li, Simon: Tree-Based Dependency Analysis in Decomposition and Re-decomposition of Complex Design Problems. In: *Journal of Mechanical Design*/ Vol. 127 (2005), S.12 – 23
- Child et al. 1991** Child, Peter ; Diederichs, Raimund ; Sanders, Falk-Hayo ; Wisniowski, Stefan: The Management of Complexity. In: *The McKinsey Quarterly*, Nr. 4, S.52 – 68
- Deuflhard / Hohmann 2002** Deuflhard, Peter; Hohmann, Andreas: *Numerische Mathematik I – Eine algorithmisch orientierte Einführung*. 3. Aufl. Berlin : De Gruyter, 2002
- Dieter 2000** Dieter, George Ellwood: *Engineering Design: a Materials and Processing Approach*. 3. Aufl. McGraw-Hill, 2000
- Dong / Whitney 2001** Dong, Qi; Whitney, Daniel E.: Design a Requirement Driven Product Development Process. In: *ASME International Design Engineering Technical Conferences & International Conference on Design Theory and Methodology – DETC/DTM 2001* (Pittsburgh 2001)
- Ehrlenspiel 1995** Ehrlenspiel, K.: *Integrierte Produktentwicklung*. 3. Aufl., Hanser, München, 2003.
- Eppinger 1997** Eppinger, Steven D.: A Planning Method for Integration of Large-scale Engineering Systems. In: *International Conference on Engineering Design – ICED'97* (Tampere 1997)
- Enkler 2010** Enkler, Hans-Georg: *Rechnergestützter Entwurf von Bauteilen mit stark streuenden Leitstützstrukturen am Beispiel hochbelastbarer urgeformter mikromechanischer Systeme*. Karlsruhe Institut für Technologie, Fakultät für Maschinenbau, Diss., 2010
- Erixon 1998** Erixon, Gunnar: *Modular Function Deployment – A Method for Product Modularization*. Stockholm, KTH, Institute of Technology, Diss., 1998
- Feldmann 1999** Feldmann, Martin: *Natural analoge Verfahren – Metaheuristiken zur Reihenfolgeplanung*. Wiesbaden : Deutscher Universitäts Verlag, 1999

- Flemming et al. 1996** Flemming, Manfred ; Ziegmann, Gerhard ; Roth, Siegfried: *Faserverbundbauweisen – Halbzeuge und Bauweise*. Berlin : Springer, 1996
- Gerdes et al. 2004** Gerdes, Ingrid ; Klawonn, Frank ; Kruse, Rudolf: *Evolutionäre Algorithmen Genetische Algorithmen Strategien und Optimierungsverfahren Beispielanwendungen*. 1. Aufl. Wiesbaden : Friedr. Vieweg & Sohn, 2004
- Gershenson et al. 2004** Gershenson, John K. ; Prasad G. Jagannath. ; Zhang Y.: Product modularity: measures and design methods. In: *Journal of Engineering Design* (2004), Nr.1, S.33 – 51
- Gierse 1996** Gierse, Franz Josef: *Beigleittext zur Vorlesung Methodik des Konstruierens*. Universität Gesamthochschule Siegen, 1996
- Goldberg 1989** Goldberg, David E.: *Genetic Algorithms in Search, Optimization, and Machine Learning*. Bonn : Addison-Wesley Longman, 1989
- Goldberg / Deb 1991** Goldberg, David E. ; Deb, Kalyanmoy: A Comparative Analysis of Selection Schemes Used in Genetic Algorithms. In: *Foundation of Genetic Algorithms*. 1991
- Göpfert 1999** Göpfert, Jan: *Modulare Produktentwicklung – Zur gemeinsamen Gestaltung von Technik und Organisation*. Wiesbaden : Gabler, 1999
- Guo / Gershenson 2004** Guo, Fang ; Gershenson, John K.: A Comparison of Modular Product Design Methods Based on Improvement and Iteration. In: *ASME International Design Engineering Technical Conferences & Computers and Information in Engineering Conference – DETC/CIE 2004* (Utah 2004), S.261 – 269
- Hansen 1968** Hansen, Friedrich: *Konstruktionssystematik*. Berlin : VEB, 1968
- Holland 1975** Holland, John: *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control and Artificial Intelligence*. Ann Arbor : University of Michigan Press, 1975
- Höltkä et al. 2005** Höltkä, Katja ; Suh, Eun Suk ; De Weck, Olivier: Tradeoff Between Modularity and Performance for Engineered Systems and Products. In: *International Conference on Engineering Design – ICED'05* (Melbourne 2005)
- Höltkä et al. 2003** Höltkä, Katja ; Tang, Victor ; Seering, Warren P.: Modularizing Product Architecture Using Dendrograms. In: *International Conference on Engineering Design – ICED'03* (Stockholm 2003)
- Huang 2000** Huang, Chun-Che: Overview of Modular Product Development. In: *The National Science Council ROC(A)* Vol. 24, Bd.3 (2000), S.149 – 165
- Huthwaite 2007** Huthwaite, Bart: *The Lean Design Solution – A Practical Guide to Streamlining Product Design and Development*. Michigan, 2007
- Huwlyer / Weyland 2004** Huwlyer, Cédric; Weyland, Mathias: *Evolutionäre Optimierung*, 2. Fassung, 2004
- Kahn et al. 2003** Kahn K. B.; Franzak, F.; Griffin, A.: Editorial: *Identification and Consideration of Emerging Research Questions*. In: *Journal of Product Innovation Management*, Vol.20, No. 3, Page. 192 – 201, 2003.
- Kamrani 2001** Kamrani, Ali K. ; Sa'ed Salhieh: *Product Design for Modularity* : Massachusetts : Kluwer Academic, 2001

- Keller et al. 2005** Keller, Rene ; Eckert, Claudia ; Clarkson, John P.: Multiple Views to Support Engineering Change Management for Complex Products. In: *International Conference on Coordinated & Multiple Views in Exploratory Visualization – CMV'05* (London 2005)
- Koller / Kastrup 1994** Koller, Rudolf ; Kastrup, Norbert: *Prinziplösungen zur Konstruktion technischer Produkte*. Springer, 1994
- Koller 1998** Koller, Rudolf: *Konstruktionslehre für den Maschinenbau: Grundlagen zur Neu- und Weiterentwicklung technischer Produkte mit Beispielen*. Berlin, Heidelberg, New York, 1998
- Kusiak / Huang 1996** Kusiak, Andrew ; Huang, Chen-Che: Development of Modular Products. In: *IEEE Transactions on Components, Packaging, and Manufacturing Technology*. 1996, S.523 – 538
- Li et al. 2003** Yu, Tian-Li ; Yassine, Ali A. ; Goldberg, David E.: A Genetic Algorithm for Developing Modular Product Architectures. In: *ASME International Design Engineering Technical Conferences & Computers and Information in Engineering Conference – DETC'03* (Illinois 2003)
- Lindemann et al. 2008** Lindemann, Udo ; Maurer, Maik ; Braun, Thomas: *Structural Complexity Management: An Approach for the Field of Product Design* : Springer, 2008
- Lindstedt / Burenus 2003** Lindstedt, Per ; Burenus, Jan: *The Value Model – How to Master Product Development and Create Unrivalled Customer Value*. Sweden, 2003
- Matthiesen 2002** Matthiesen, Sven: *Ein Beitrag zur Basisdefinition des Elementmodells "Wirkflächenpaare & Leitstützstrukturen" zum Zusammenhang von Funktion und Gestalt technischer Systeme*". Karlsruhe, Universität, Fakultät für Maschinenbau, Diss., 2002
- Mayer 2007** Mayer, Axel: *Modularisierung der Logistik – Ein Gestaltungsmodell zum Management von Komplexität in der industriellen Logistik*. Berlin, Technische Universität, Diss., 2002
- McCord / Eppinger 1993** McCord, Kent R. ; Eppinger, Steven D.: Managing the Integration Problem in Concurrent Engineering. In: *MIT Sloan School of Management Working Paper* (Massachusetts 1993)
- Meboldt 2008** Meboldt, Mirko: *Ganzheitliche Modellbildung der Produktentstehung -von mentalen Modellen bis zum Workflow für den Umgang mit Komplexität- als Beitrag zum integrierten Produktentstehungsmodell (IPeM)*. Karlsruhe, Universität, Fakultät für Maschinenbau, Diss., 2008
- Meister 2005** Meister, Andreas: *Numerik linearer Gleichungssysteme – Eine Einführung in Moderne Verfahren*. 2. Aufl. Wiesbaden : Vieweg, 2005
- Mühlenbein / Schlierkamp-Voosen 1993** Mühlenbein, Heinz ; Schlierkamp-Voosen, Dirk: Predictive Models for the Breeder Genetic Algorithm – I. Continuous Parameter Optimization Evolutionary Computation. S.25 – 49
- Oerding 2009** Oerding, Jochen: *Ein Beitrag zum Modellverständnis der Produktentstehung – Strukturierung von Zielsystemen mittels C&CM*. Karlsruhe, Universität, Fakultät für Maschinenbau, Diss., 2009
- Otto / Holttta 2004** Otto, Kevin ; Hoeltta, Katja: A Multi-Criteria Framework for Screening Preliminary Product Platform Concepts. In: *ASME International Design Engineering Technical Conferences and Computers and Information in Engineering Conference – DETC'04* (Utah 2004)

- Pahl et al. 2003** Pahl, Gerhard ; Beitz, Wolfgang ; Feldhusen, Jörg ; Grote, Karl-Heinrich: Konstruktionslehre: Grundlagen erfolgreicher Produktentwicklung Methoden und Anwendung. 5. Aufl. Berlin : Springer, 2003
- Pahl / Beitz 1993** Pahl, Gerhard ; Beitz, Wolfgang: *Konstruktionslehre*. Berlin, Heidelberg, New York, 1993
- Pimmler / Eppinger 1994** Pimmler, Thomas U. ; Eppinger Steven D.: Integration Analysis of Product Decompositions. In: *ASME Design Theory and Methodology Conference – DTM*. (Minneapolis 1994)
- Pohlheim 2000** Pohlheim, Hartmut: *Evolutionäre Algorithmen – Verfahren, Operatoren und Hinweise für die Praxis*. Berlin : Springer, 2000
- Rapp 1999** Rapp, Thomas: *Produktstrukturierung*. St. Gallen, Universität, Wirtschaftsrechts- und Sozialwissenschaften, Diss., 1999
- Rechenberg 1973** Rechenberg, Ingo: *Evolutionstrategie*. Friedrich Frommann Verlag, Stuttgart-Bad Cannstatt; 1973
- Regenstein et al. 2007** Regenstein, Kristian ; Kerscher, Thilo ; Asfour, Tamim ; Zöllner, Marius ; Dillmann , Rüdiger: Universal Controller Module (UCoM) – Component of a Modular Concept in Robotic Systems. In: *IEEE International Symposium on Industrial Electronics*. (Vigo 2007)
- Rodenacker 1991** Rodenacker, W. G.: *Methodisches Konstruieren*; Konstruktionsbücher Band 27; Belin, Heidelberg, New York; 1991
- Rogers et al. 2006** Rogers, James L. ; Korte, John J. ; Bilardo, Vincent J. Jr.: Development of a Genetic Algorithm to Automate Clustering of a Dependency Structure Matrix. In: *NASA Center for Aerospace Information (CASI)*. (Virginia 2006)
- Romesburg 1984** Romesburg, H. Charles: *Cluster Analysis for Researchers*. North Carolina : Wadsworth Inc, 1984
- Schlierkamp-Voosen / Mühlenbein 1996** Schlierkamp-Voosen, Dirk ; Mühlenbein, Heinz: Adaptation of Population Sizes by Competing Subpopulations. In: *International Conference on Evolutionary Computation – ICEC 1996*. (Nagoya 1996) S.330 – 335
- Schuh 2005** Schuh, Günther: *Produktkomplexität managen, Strategien Methoden Tools*, 2. Aufl. München : Hanser, 2005
- Schuh / Schwenk 2001** Schuh, Günther ; Schwenk, Urs: *Produktkomplexität managen*. Hanser, 2001
- Sharman / Yassine 2003** Sharman, David M. ; Yassine, Ali A.: Characterizing Complex Product Architectures. In: *Systems Engineering* Vol. 7 (2004) Nr. 1, S.35 – 60
- Sharman 2002** Sharman, Davia M.: *Valuing Architecture for Strategic Purposes – Comments on Applying the Dependency Structure Matrix With Real Options Theory*. Massachusetts Institute of Technology, 2002
- Snowden 2000** Snowden D.: *The ASHEN-Model an enabler of action*. Knowledge Management, Band 3, 7/2000, S. 14 – 17, 2000
- Sosa et al. 2000** Sosa, Manuel ; Eppinger, Steven D. ; Rowles, Craig M.: Designing Modular and Integrative Systems. In: *ASME International Design Engineering Technical Conferences & Computers and Information in Engineering Conference – DETC'00* (Maryland 2000)

- Sosale et al. 1997** Sosale, Swaroop ; Hashemain, Mehdi ; Gu, Peihua: Product Modularization for Reuse and Recycling. In: *Concurrent Product Design and Environmentally Conscious Manufacturing – ASME 1997 (Texas 1997)* S.195 – 206
- Strawbridge et al. 2002** Strawbridge, Zeke; McAdams, Daniel A.; Stone, Robert B.: A Computational Approach to Conceptual Design. In: *ASME International Design Engineering Technical Conferences – DETC'02 (Montreal 2002)*
- Stone 1997** Stone Robert B.: *Towards a Theory of Modular Design*. Texas, Universität, Diss., 1997
- Syswerda 1989** Syswerda, Gilbert: Uniform Crossover in Genetic Algorithms. In: *International Conference on Genetic Algorithms – ICGA (Virginia 1989)* S.2 – 9
- Ulrich / Eppinger 2008** Ulrich, Karl T. ; Eppinger Steven D.: *Product Design and Development*. 4. Auf. McGrawHill, 2008
- Ulrich 1995** Ulrich, Karl: The role of product architecture in the manufacturing firm. In: *Research Policy* 24 (1995), S. 419-440
- Van Wie et al. 2001** Van Wie, Mike J. ; Greer, James L. ; Campbell, Matthew I. ; Stone, Robert B. ; Wood, Kristin L.: Interfaces and Product Architecture. In: *ASME International Design Engineering Technical Conferences & Computers and Information in Engineering Conference – DETC'01 (Pennsylvania 2001)*
- Van Wie 2002** Van Wie, Michael James: *Designing Product Architecture – A Systematic Method*. Texas, Universität, Diss., 2002
- VDI Richtlinie 2206 2004** Verein Deutscher Ingenieure – Gesellschaft Entwicklung Konstruktion Vertrieb: Entwicklungsmethodik für mechatronische Systeme. *VDI-Richtlinie*, Juni 2004
- VDI Richtlinie 2221 1993** Verein Deutscher Ingenieure – Gesellschaft Entwicklung Konstruktion Vertrieb: Methodik zum Entwickeln und Konstruieren technischer Systeme und Produkte. *VDI-Richtlinie*, Mai 1993
- VDI Richtlinie 2223 2004** Verein Deutscher Ingenieure – Gesellschaft Entwicklung Konstruktion Vertrieb: Methodisches Entwerfen technischer Produkte, *VDI-Richtlinie*, Januar 2004
- VDI Richtlinie 2803 1996** Verein Deutscher Ingenieure – Gesellschaft Entwicklung Konstruktion Vertrieb: Funktionenanalyse Grundlagen und Methode. *VDI-Richtlinie*, Oktober 1996
- Voigt et al. 1995** Voigt, Hans-Michael ; Mühlenbein, Heiz ; Cvetkovic, Dragan: Fuzzy Recombination for Continuous Breeder Genetic Algorithm. In: *International Conference on Genetic Algorithms – ICGA (Pittsburgh 1995)* S.104 – 111
- Whitfield et al. 2001** Whitfield, Robert I. ; Smith, Joanne S. ; Duffy, Alex B.: Identify Component Modules. In: *Artificial Intelligence in Design '02*. (2002), S. 571 – 592
- Yassine / Wissman 2007** Yassine, Ali A. ; Wissman, Luke A.: The Implications of Product Architecture on the Firm. In: *System Engineering* 10 (2007) , Nr. 2, S. 118 – 137

---

## 9 Stichwortverzeichnis

- Abbruchkriterium 42, 46, 92
- Abstraktionsgrad 75
- Aktivität 7, 65
- Aktivitäten der Problemlösung 6
- Aktivitäten der Produktentstehung 6
- Anfangspopulation 92
- Anpassungsfunktion 29
- ARMAR III 101, 115
- Baukastensystematik 28
- Bauweise 16
- Bewertungsschema 86
- Beziehungszahl 87
- Branching Flow 49
- C&C-M 29, 103, 111, 120
- C&C-M Dependency-Matrix 71, 80
- CDM 5, 105
- Chromosom 38
- Contact and Channel Modell 5, 29
- Conversion-Transmission modules 49
- Das integrierte Produktentstehungsmodell 5
- Dendrogramm 57
- Dependency-Matrix 33
- Design für X 69
- Design-Structure-Matrix 33
- Detailierungsgrad 13, 27
- Detailierungsgraden 28
- Dominant Flow 49
- Einfacher genetischer Algorithmus 42
- Elementarfunktion 28
- Entwicklungsprozess 11
- Evolutionären Algorithmen 38
- Evolutionsstrategie 41
- Festlegung der Produktarchitektur 3, 24, 46
- Fitness 38
- Frühe Phase 13
- Function-Layout-Diagramm 54
- Funktion 25
- Funktionale Abhängigkeit 19
- Funktionale Beziehung 80, 82, 85, 87, 105
- Funktionale Gliederung 28
- Funktionsformulierung 26, 28
- Funktionsgliederung 28
- Funktionsinformation 62, 66
- Funktionsstruktur 15, 61
- Gen 38
- Generation 38
- Genetische Algorithmen 38
- Genetischer Algorithmus 92
- Gesamtfunktion 28
- Gesamtstruktur 31
- Gestaltinformation 62, 66
- Gewichtung der Beziehung 86
- Grundfunktion 29
- Guideline Module 57
- Handlungssystem 6, 65
- Hilfsfunktion 29, 73
- Humanoide Roboter 101
- Humanoiden Roboterthorax 115
- Hypothese 31
- Integerkodierung 106
- Integrale Produktarchitektur 16
- Intuitive ansätze 61

- iPeM 5, 13
- Kodierung und Initialisierung 43
- Kompatibilitätsbeziehung 82, 83, 86, 105, 113
- Komponentenstruktur 15
- Konzeptphase 11
- Leitstützstruktur 78
- Leitstützstrukturen 31
- Manufacturing-Diagramm 54
- Modul 17
- Modulanzahl 92, 93
- Modulare Produktarchitektur 16
- Modular-Function-Deployment 20
- Modularität 19, 23
- Modularitätsgrad 22, 88
- Mutation 45
- Nebenfunktion 73
- Objektsystem 6, 65
- Partition-Diagramm 54
- Phase 11
- Physical-Process-Diagramm 54
- Physische Abhängigkeit 19
- Population 38
- Prinzipskizze 61, 65, 67, 74, 102
- Product-Family-Diagramm 54
- Produktarchitektur 15, 18, 25
- Produktarchitekturfestlegung 65
- Produktarchitekturtypen 23
- Pseudoabstände 58
- Quantitative ansätze 61
- Referenzmodell 9
- Rekombination 44
- Reproduktion 38
- Reststruktur 31
- Selektion 44
- Sonderfunktion 29
- Spatial-Constraints-Diagramm 54
- System-Level-Design-Phase 11
- Teilfunktion 28
- Topologieoptimierung 115, 117
- Tragstruktur 30, 83
- Umkonstruktion 69
- Validierung 9
- VDI Richtlinien 2221 47
- Vorentwicklungsphase 14
- Wirkflächen 30
- Wirkflächenpaar 78
- Wirkflächenpaaren 31
- ZHO-Systeme 5
- Zielfunktion 89
- Zielfunktionswert 90
- Zielsystem 5, 65

# Lebenslauf

## Persönliche Daten

Name: Korkiat Sedchaicharn  
Geburtsdatum: 24. Oktober 1976  
Geburtsort: Bangkok, Thailand  
Staatsangehörigkeit: thailändisch  
Familienstand: ledig

## Studium

05/88 – 03/93 Gymnasium, Patumkongka Schule in Bangkok, Thailand  
06/93 – 03/97 Maschinenbaustudium, King Mongkut's Institute of Technology Thonburi (KMITT) in Bangkok, Thailand – Abschluss: Bachelor Ingenieur  
04/96 – 05/96 Praktikum CAD/CAM bei Auckland Institute of Technology in Auckland, Neuseeland  
03/01 – 10/04 Studium „Ergänzung Maschinenbau“, Technische Universität Clausthal, Clausthal-Zellerfeld – Abschluss: Diplom Ingenieur  
10/03 – 03/04 Diplomarbeit bei Robert BOSCH GmbH, Feuerbach  
06/10 Promotionsprüfung zum Doktor-Ingenieur der Fakultät für Maschinenbau

## Berufstätigkeit

05/97 – 10/97 Wissenschaftlicher Mitarbeiter und Hilfsassistent bei King Mongkut's Institute of Technology Thonburi (KMITT) in Bangkok, Thailand  
11/97 – 12/97 Forscher bei Institute of Field Robotics (FIBO) in Bangkok, Thailand  
01/98 – 11/99 Produktionsplanung Ingenieur bei Siam United Steel (1995) Co., Ltd in Rayong, Thailand