

Knowledge-based Engineering in der Technischen Logistik

Diplomarbeit an der TU-Graz
Institut für Technische Logistik



von

Stefan Steinkellner

Graz, Dezember 2012

EIDESSTATTLICHE ERKLÄRUNG

Ich erkläre an Eides statt, dass ich die vorliegende Arbeit selbstständig verfasst, andere als die angegebenen Quellen/Hilfsmittel nicht benutzt und die den benutzten Quellen wörtlich und inhaltlich entnommenen Stellen als solche kenntlich gemacht habe.

Graz, am

.....

(Unterschrift)

Vorwort

Diese Diplomarbeit wurde zur Erlangung des akademischen Grades „Diplomingenieur Wirtschaftsingenieurwesen-Maschinenbau“ verfasst. Sie ist im Zuge meines Studiums an der TU Graz am Institut für Technische Logistik entstanden.

Für die sehr gute Unterstützung und Betreuung möchte ich mich bei Herrn Ass.Prof. Dipl.-Ing. Dr.techn. Christian Landschützer bedanken.

Ein herzliches Dankeschön gebührt meinen Eltern, die mich während der Studienzeit unterstützt und mir diese Ausbildung ermöglicht haben.

Bedanken möchte ich mich noch bei meiner Schwester Renate, für das Korrekturlesen dieser Arbeit.

Ein weiteres Dankeschön gilt an all jene, die mir bei dieser Arbeit mit Rat und Tat Beiseite standen.

Kurzfassung

Deutsch

In dieser Diplomarbeit wird das Thema Knowledge-based Engineering (KBE) behandelt, KBE automatisiert Konstruktionsschritte, aufgrund einer Regelverarbeitung in einem CAD-System. Dabei werden neben der Thematik KBE, die tangierenden Themen Wissensmanagement und Produktentwicklung behandelt. Für die Erstellung von KBE-Anwendungen existieren Methodiken, die im Rahmen dieser Arbeit betrachtet werden.

Zur systematischen Vorgehensweise in der Produktentwicklung werden die Verfahren Axiomatic Design und die Design Structure Matrix untersucht. Ein Anwendungsbeispiel zeigt die Repräsentation von Wissen, mithilfe einer Ontologie, über den Fördertyp Rollenförderer. Im Zuge des Anwendungsbeispiels veranschaulicht die Erstellung einer Design Structure Matrix die Abbildung von Parameterabhängigkeiten des Rollenförderers.

Der abschließende Teil der Arbeit führt die Erkenntnisse die aus der Durchführung des Anwendungsbeispiels gewonnen werden konnten, an. Außerdem erfolgen in diesem Abschnitt der Arbeit eine Bewertung der verwendeten Literatur sowie ein Vergleich der KBE-Methodiken.

Englisch

This master thesis treats the topic Knowledge-based Engineering (KBE). KBE automates design steps, based on rule processing in a CAD-system. Besides KBE the related topics knowledge management and product development will be discussed. There are existing methodologies for the preparation of KBE-applications, which will be treated in this thesis.

For the systematic approach in product development the procedures Axiomatic Design and the Design Structure Matrix will be explained. One application example indicates the representation of knowledge, by means of ontology about the conveyor type roller conveyor. Using this example the creation of Design Structure Matrix and the parameter dependency of the roller conveyor will be shown.

The final part demonstrates the insights which could be reached by executing this example. Furthermore a rating of the used literature and a comparison of the KBE-methodologies has been taken.

Inhaltsverzeichnis

1	Einleitung.....	3
2	Grundlagen.....	4
2.1	KBE (Knowledge-based Engineering).....	4
2.1.1	Entwicklung und aktueller Stand von KBE.....	4
2.1.2	Anwendungsgründe von KBE-Systemen.....	5
2.1.3	Vorteile und Nachteile von KBE-Systemen	5
2.1.4	Einsparungspotential von KBE-Systemen.....	6
2.2	Wissensmanagement	8
2.2.1	Einführung Wissensmanagement	8
2.2.2	Wissensrepräsentation.....	10
2.3	Produktentwicklung	12
2.3.1	Einführung Produktentwicklung.....	12
2.3.2	Methodisches Vorgehen bei der Produktentwicklung	13
2.3.3	PLM (Product Lifecycle Management)	15
3	Systematische Analyse zur Verwendung von KBE.....	16
3.1	Knowledge-based Engineering Methodiken	16
3.1.1	MOKA-Methodik	16
3.1.1.1	MOKA-Hauptaspekte	16
3.1.1.2	MOKA-KBE-Lebenszyklus	17
3.1.1.3	Informelles und Formelles Modell	21
3.1.1.4	MML (MOKA-Modelling Language)	22
3.1.2	KOMPRESSA-Methodik	29
3.1.2.1	KOMPRESSA-Merkmale.....	29
3.1.2.2	KOMPRESSA-Aktivitätsgruppen	29
3.1.3	DEE-Methodik.....	32
3.1.3.1	DEE-Komponenten und Prozessablauf.....	32
3.1.3.2	DEE-Agenten System	34
3.1.4	KNOMAD-Methodik.....	36
3.1.4.1	KNOMAD-Merkmale	36
3.1.4.2	KNOMAD-Phasen.....	36
3.2	MDE (Model Driven Engineering)	40
3.2.1	Modellbasierte Softwareentwicklung	40
3.2.2	Modellgetriebene Softwareentwicklung	40
3.2.2.1	Modellierung	41
3.2.2.2	Ziele der modellgetriebenen Softwareentwicklung	42
3.3	Knowledge-based Systems – Expertensysteme	43
3.3.1	CommonKADS-Methodik.....	44
3.3.1.1	CommonKADS-Modellreihenfolge	45
3.3.1.2	CommonKADS-Prozesspersonen	46
3.4	Matrixbasierte Methoden	49
3.4.1	DSM (Design Structure Matrix)	49
3.4.1.1	DSM Typen.....	50
3.4.1.2	DSM Analysealgorithmen	50
3.4.2	DMM (Domain Mapping Matrix).....	53
3.4.3	MDM (Multiple Domain Matrix)	54
3.5	Axiomatic Design	55

3.6	Wissensrepräsentation mit Softwaretools	59
3.6.1	Compendium	59
3.6.2	CmapTools	60
3.6.3	VUE (Visual Understanding Environment)	61
3.6.4	iMapping	62
3.6.5	CAM (Cambridge Advanced Modeller)	63
3.6.6	Set Visualiser	64
4	Eigene Ansätze – Ontologie und DSM am Beispiel eines Rollenförderers....	66
4.1	Definition Stetigförderer	66
4.2	Berechnungsgrundlagen Stetigförderer	68
4.3	Definition Rollenförderer.....	70
4.4	Erstellung einer Rollenfördererstrecke	71
4.5	Erstellung einer Ontologie in der xKBE-App	73
4.5.1	Einpflegen der Förderstrecke in die xKBE-App	73
4.5.2	Einbetten von Normen und Theorie in die xKBE-App	74
4.5.3	Funktionsumfang der xKBE App	76
4.5.4	Möglichkeiten und Nutzen der xKBE App	78
4.6	Design Structure Matrix für Rollenförderer.....	79
4.6.1	Erstellung einer DSM für Parameter eines Rollenförderers.....	79
4.6.2	Analysefunktionen des CAM	81
4.6.3	Parameterdarstellung im Set Visualiser.....	81
5	Rechercheergebnisse der Anwendung von KBE-Methodiken – Erkenntnisse.. des Anwendungsbeispiels	83
5.1	Anwendung der KBE-Methodiken in der Praxis.....	83
5.2	Erkenntnisse aus dem Anwendungsbeispiel	84
6	Bewertung	86
6.1	Bewertung der Literatur	86
6.2	Bewertung der KBE Methodiken	87
6.3	Bewertung der Wissensrepräsentationstools	88
7	Zusammenfassung	90
8	Verzeichnisse.....	91
8.1	Literaturverzeichnis	91
8.2	Abbildungsverzeichnis.....	95
8.3	Tabellenverzeichnis	96
9	Anhang	97
9.1	Technische Datenblätter	98
9.2	Parameterauflistung.....	104
9.3	Umstrukturierte DSMs	108

1 Einleitung

Die letzten Jahrzehnte waren durch starke Automatisierungen und Optimierungen in den Bereichen Produktion und Herstellung geprägt. Das Einsatzfeld der Produktentwicklung und Konstruktion bietet jedoch enormes Potential für innovative Prozessoptimierungen, Automatisierungen und Standardisierungen. In vielen Konstruktionsabteilungen werden nach wie vor identische oder ähnliche und oft sehr zeitintensive Konstruktionsprozesse für verschiedene Neu-, Anpassungs- und Weiterentwicklungen wiederholt. Aufgrund der immer kürzer werdenden Entwicklungszeiten stehen die Entwickler und Konstrukteure unter einem zunehmenden Zeitdruck, der die Einführung von Standardisierungs- und Automatisierungsprozesse aus zeitlichen Gründen verhindert. Die Vorrangigkeit des Tagesgeschäftes führt zu einer Unterordnung der Dringlichkeit zur Einleitung von Standardisierungs- und Automatisierungsprozesse. Jedoch bietet diese Problematik die Chance, für viele Unternehmen den Teufelskreis aus steigendem Zeit- und Kostendruck zu unterbrechen und den Konstrukteuren mehr Zeit für kreative und innovative Aufgaben zur Verfügung zu stellen. Eine enorme Effektivitätssteigerung in den Konstruktionsprozessen kann durch die „Wissensbasierte Konstruktion“ erzielt werden. Unter „Wissensbasierter Konstruktion“ (englische Bezeichnung: Knowledge-based Engineering, KBE), versteht man die Verwendung von Wissen, mit dem Konstruktionschritte automatisiert oder teilautomatisiert, aufgrund der Anwendung einer Regelverarbeitung in einer CAD-Umgebung durchgeführt werden. Durch die „Wissensbasierte Konstruktion“ wird der Konstrukteur von zeitintensiven Routinetätigkeiten befreit, was ihm einen größeren Spielraum für die Herausforderung zur Suche von neuen, kreativen und innovativen Lösungen verschafft. (vgl. [HSA12])

Gliederung

In dieser Arbeit erfolgt im ersten Kapitel Grundlagen eine allgemeine Erläuterung der Thematik KBE, sowie den damit verwandten Themen Wissensmanagement und Produktentwicklung. Das zweite Kapitel systematische Analyse zur Verwendung von KBE ist in drei Teile aufgegliedert. Der erste Teil beinhaltet die Darlegung von existierenden KBE-Methodiken. Im zweiten Teil wird ein Verfahren zur Hilfestellung in der Produktentwicklung, das den Namen Axiomatic Design trägt beschrieben. Des Weiteren wird für die Produktentwicklung das Werkzeug Design Structure Matrix angeführt. Im letzten Teil dieses Kapitels werden verschiedene Softwaretools zur Anwendung von Wissensmanagement und zur Erstellung einer Design Structure Matrix vorgestellt. Das vierte Kapitel beinhaltet die Dokumentation der Durchführung eines Anwendungsbeispiels. Dabei wurde eine Ontologie über den Fördertype Rollenförderer erstellt, sowie eine parameterbasierte Design Structure Matrix diesen Fördertyps. Das folgende Kapitel enthält die Rechercheergebnisse einer Untersuchung über eine Praxisanwendung der KBE-Methodiken. Des Weiteren beinhaltet es die gewonnenen Erkenntnisse des Anwendungsbeispiels. Das sechste Kapitel enthält eine Bewertung der im dritten Kapitel verwendeten Literatur, sowie eine Gegenüberstellung der KBE-Methodiken und den angeführten Softwaretools. Das letzte Kapitel beinhaltet eine Zusammenfassung der hier vorliegenden Arbeit.

2 Grundlagen

Zunächst werden die grundlegenden Themen wie, Knowledge-based Engineering, Wissensmanagement und Produktentwicklung näher betrachtet, um einen Überblick einer innovativen Produktentwicklung zu verschaffen.

2.1 KBE (Knowledge-based Engineering)

2.1.1 Entwicklung und aktueller Stand von KBE

In den 1980er Jahren wurden bereits in der 2D-CAD-Technik simple Konstruktionsaufgaben durch einfaches kopieren und wiederverwenden vereinfacht. Der nächste Schritt in der Konstruktionsautomatisierung war die berechnungsgetriebene Feature-Technologie, dabei wurden Wellengeneratoren und Schraubenberechnungstools eingesetzt, die hinterlegte Dimensionierungsberechnungen enthielten. Als in den 1990er Jahren die parametrische Konstruktion aufkam, war es möglich die ursprünglichen „statischen“ Parameterauslegungen einfach zu adaptieren und die hinterlegten Teilebibliotheken wurden durch Parametertabellen angesteuert, die auf einem parameterautomatisierten Berechnungsprozess basieren. Dadurch war eine Anpassungskonstruktion durch die Ansteuerung von Parametern möglich, diese Vorgehensweise wird auch heute noch in vielen Konstruktionsbüros so praktiziert. Bei den gegenwärtigen KBE-Systemen muss man zwischen zwei unterschiedlichen KBE-Ansätzen unterscheiden. Zum einen gibt es die full-KBE-Lösung, die mit einer Spezialsoftware umgesetzt wird, welche vor allem in der Flugzeug- und Fahrzeugindustrie eingesetzt wird. Zum anderen gibt es die KBE-fähigen augmented CAD-Systeme. Hierunter fallen parametrische CAD-Baugruppen, die durch übergeordnete Berechnungstools, wie MS EXCEL, MathCad, MATLAB oder selbstständige VB-Programme generiert werden. Allerdings muss von den Spezialsoftwareprodukten der CAD-Hersteller abgegrenzt werden, wie z.B., der Hersteller CATIA mit Knowledge Ware oder NX mit Knowledge Fusion, die eine Reihe der eigenständigen KBE-Lösungen darstellen. Der Nachteil dieser augmented KBE-Lösungen besteht in der „statischen“ Wissensformulierung, weshalb eine Kommunikation mit Wissensmanagementsystemen nur begrenzt umzusetzen ist. Das erfasste Wissen steht nur innerhalb des CAD/KBE-Systems zur Verfügung. Jedoch haben diese Systeme einen geringeren Implementierungsaufwand. Für die Erstellung einer KBE-Lösung einer Produktpalette soll jedoch der Aspekt betrachtet werden, das erfasste Wissen außerhalb des CAD-Systems zur Verfügung zu stellen. Für solche Herangehensweise existieren einige KBE-Methodiken die im Abschnitt 3.1 genauer beschrieben werden. Da die Erstellung einer KBE-Anwendung sehr softwarelastig ist, wurde noch das Model Driven Engineering, welches im Punkt 3.2 näher betrachtet wird, angeführt. Dabei handelt es sich um eine automatische Programmcodegenerierung aus einem zuvor festgelegten Modell. (vgl. [LJ12])

2.1.2 Anwendungsgründe von KBE-Systemen

Einer der wichtigsten Aspekte in heutigen Zeiten für Unternehmen wettbewerbsfähig zu bleiben, ist der richtige Umgang mit der Ressource Wissen, das spezifische Produktwissen eines Unternehmens spielt dabei die tragende Rolle. Bei der Handhabung des Unternehmens-Know-how können sich Fehler einschleichen, die durch eine Anwendung eines KBE-Systems vermieden werden können, die folgende Auflistung beinhaltet einige dieser Fehlerquellen: (vgl. [ALC12])

- **Unsicherheit:** Die Dokumentationen und Erfahrungen aus bereits abgewickelten Projekten, sowie die Konstruktionsregeln oder Datenbanken werden nicht sorgfältig verwaltet oder existieren gar nicht.
- **Fluktuation:** Das Fachwissen von qualifizierten Mitarbeitern geht mit dem Austritt des Mitarbeiters aus dem Unternehmen verloren.
- **Aktualität:** Ein dezentrales Datenbanksystem überschneidet sich unumgänglich und kann nicht in allen Bereichen auf den aktuellsten Stand gehalten werden.
- **Benutzerfreundlichkeit:** Die bisherigen Systeme sind oft schwer zu bedienen oder deren erfasstes Wissen wurde nicht ausreichend dokumentiert, was zum Resultat führte, das es schwer verständlich war.
- **Unordnung:** Die Wissenserfassung ist essentiell, jedoch zu viel Wissen führt zur Verwirrungen und kann deshalb nicht wiederverwendet werden.

2.1.3 Vorteile und Nachteile von KBE-Systemen

Die folgende Anführung beinhaltet den unternehmerischen Nutzen, der durch die Anwendung eines KBE-Systems laut (vgl. [ALC12]) erfolgt:

- **Schnelligkeit:** Eine wertvolle Zeitersparnis durch die Automatisierung von wiederholender und monotoner Routinearbeiten, die andernfalls produktive Arbeitszeit in Anspruch nehmen.
- **Fehlerfreiheit:** Ermöglicht komplexe Berechnungen fehlerfrei durchzuführen, sowie ein schnelles nachschlagen in diversen Datenbanken.
- **Kernkompetenzen:** Der Ingenieur benötigt kein spezifisches Wissen anderer Fachdisziplinen für die Ausführung seiner Aufgaben.
- **Einfacher Konstruktionsprozess:** Das Einschreiten des Ingenieurs ist nur mehr bei wenigen Konstruktionsentscheidungen von Nöten, daraus erfolgt eine Vereinfachung des Entwicklungsprozesses.

Die Autoren (vgl. [MMA08]) erwähnen weitere Vorteile von KBE-Systemen sowie einige Nachteile von KBE-Anwendungen.

Vorteile:

- Reduktion der Produkteinführungs- und Vorlaufzeiten
- Ermöglicht die Wiederverwendung von kritischem Wissen
- Ermöglicht die gemeinsame Nutzung von Konstruktionen in Echtzeit unter der Verwendung von WWW, HTML und Hyperlinking Protokollen
- Reduktion der Produktionskosten
- Automatisiert wiederholende Arbeiten
- Auch für die Verwendung von Analyse- und Simulationsprozessen geeignet
- Kosten- und Zeitersparnis bis zu 90%

- Ergänzt das traditionelle CAD

Nachteile:

- Kostenintensive Implementierung
- Benötigt anfänglich viel Zeit für die Erfassung des Konstruktionswissens
- KBE-Anwendungen sind nur gerechtfertigt, wenn ähnliche Probleme mehrmals vorkommen

Die KBE-Systeme können nicht für jeweilige Problemstellung im Konstruktionsbereich angewendet werden, die folgende Auflistung enthält laut (vgl. [VBD11]) einige Beispiele, wann es nicht sinnvoll ist, KBE einzusetzen.

- Die Konstruktionsaufgabe ist sehr einfach zu modellieren und benötigt wenige Ressourcen, anstatt für diesen Fall eine anspruchsvolle KBE-Anwendung zu entwickeln.
- Das Unternehmen zeigt keine Bereitschaft oder verfügt über keine finanziellen Mittel um ein KBE-System einzuführen. Heutige Unternehmen neigen dazu, Commercial-Of-The-Shelf (Kommerzielle Produkte aus dem Regal) Lösungen einzuführen und scheuen den Weg eine betriebseigene Software zu entwickeln, welche jedoch für den Einsatz einer KBE-Entwicklung benötigt wird.
- Der Konstruktionsprozess besteht vorwiegend aus kreativen Prozessen und die Produkte sind starken Änderungen unterworfen.
- Das Wissen für die angestrebte Anwendung ist nicht vorhanden.
- Der Konstruktionsprozess kann nicht eindeutig definiert werden, es besteht keine Möglichkeit, die einzelnen Schritte im Konstruktionsprozess abzugrenzen und zu bestimmen.
- Es herrschte eine ständige Technologieänderung im Konstruktionsprozess.

2.1.4 Einsparungspotential von KBE-Systemen

Einer der Hauptvorteile bei der Anwendung von KBE-Systemen ist die Automatisierung von nicht-kreativen und sich wiederholender Konstruktionsaufgaben. Die Automatisierung ermöglicht nicht nur enorme Zeit- und Kostenersparnisse sondern befreit den Konstrukteur auch von monotonen Arbeiten und erlaubt ihm, dadurch sich auf kreative und innovative Konstruktionstätigkeiten zu konzentrieren. Der Konstrukteur benötigt des Weiteren 20% seiner Arbeitszeit mit der Suche nach Informationen, durch die Wissenswiederverwendung von KBE-Systemen erlaubt dieser Ansatz eine weitere Arbeitszeiteinsparung. Die Abbildung 2-1 veranschaulicht die Umverteilung der repetitiven Arbeiten zu den kreativen und innovativen Konstruktionstätigkeiten. (vgl. [VBD11])

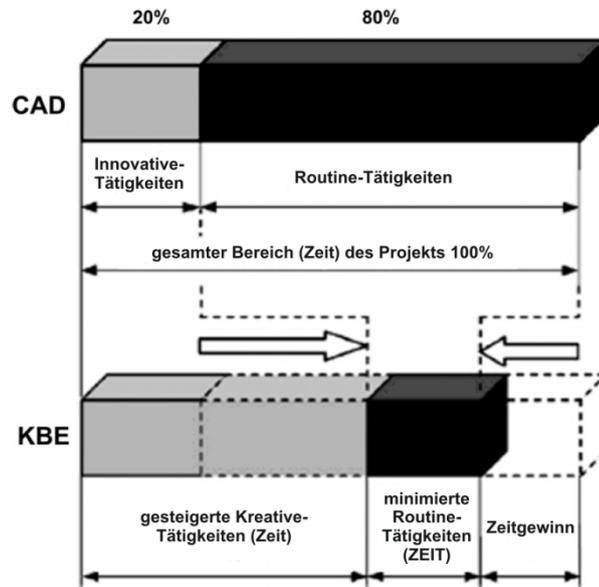


Abbildung 2-1: Erreichbare Zeiteinsparung durch KBE –Einsatz (angelehnt an [VBD11])

Eine quantitative Anführung für die Zeiteinsparung einer KBE-Anwendung liefert der Artikel [EM07] . Dabei wurde im speziellen Anwendungsfall ein KBE-System im Fachbereich Flugzeugbau für eine Flügelkonstruktion, die aus Metallfaserlaminaten besteht, eingesetzt. Für diesen Anwendungsfall benötigt die Entwicklungszeit einer herkömmlichen CAD-Konstruktion sechzehn Mannwochen. Die Abbildung 2-2 veranschaulicht die Gegenüberstellung einer herkömmlichen CAD-Konstruktion zu einer KBE-Lösung. Dabei zeigt die Grafik, dass bei sechs Konstruktionszyklen die KBE-Konstruktion mit der herkömmlichen CAD-Konstruktion in ihrem zeitlichen Aufwand gleichgestellt ist. Das anfängliche Zeitdefizit der KBE-Anwendung gegenüber der traditionellen Konstruktion, ergibt sich aus der Erfassung des Wissens und der Erstellung des Codes. Das Diagramm zeigt, dass bei der Durchführung von zehn herkömmlichen Konstruktionszyklen in derselben Zeit hundert KBE-Konstruktionszyklen ausgeführt werden können. Dieses Diagramm gilt jedoch nur für diesen konkreten Anwendungsfall und kann nicht für jede KBE-Lösung verallgemeinert werden, da das Verhältnis der beiden Konstruktionszyklen sehr stark von anderen Faktoren abhängt, wie z.B. der Komplexität der Konstruktion. (vgl. [EM07])

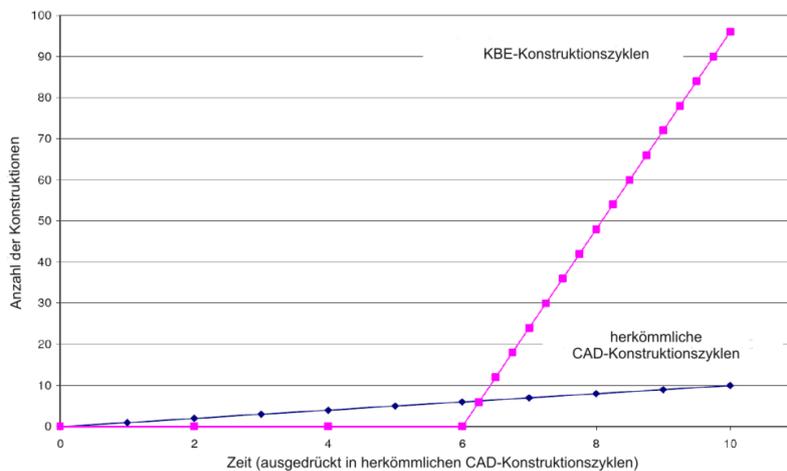


Abbildung 2-2: KBE-System Entwicklungszeit versus herkömmliche Konstruktionszeit (angelehnt an [EM07])

2.2 Wissensmanagement

Da im vorigen Abschnitt erwähnt wurde, dass es zielführend ist ein KBE-System mit einem Wissensmanagement zu verbinden, wodurch das erfasste Wissen auch außerhalb der CAD-Umgebung wiederverwendet werden kann, erfolgt in diesem Punkt eine kurze Beschreibung über Wissensmanagement und Wissensrepräsentation.

2.2.1 Einführung Wissensmanagement

Durch die Verkürzung der Innovationszyklen, die zunehmenden Globalisierung der Unternehmen und der Situation durch alterndes Personal, ruft in vielen Branchen eine effizientere Handhabung mit der Wettbewerbsressource „Wissen“ hervor. Vor allem besteht die Herausforderung darin, das Fachwissen in der Disziplin der Ingenieurwissenschaften aktuell zu halten. Daher ist eine Eingliederung des Wissensmanagements in den Prozess der Wertschöpfung eines Unternehmens unvermeidbar. Die heutigen Informationsmedien erlauben sofortige Recherchen in globalen Informationsspeichern, dabei entsteht jedoch die Schwierigkeit der Qualitätsbewertung und der Verlässlichkeit dieser Informationen. Zur Abhilfe einer Informationsüberflutung der Belegschaft werden zielorientierte Wissensmanagementsysteme eingesetzt. Ein Kernpunkt bei der Einführung eines Wissensmanagementsystems ist die Bereitschaft der Mitarbeiter solche Systeme anzuwenden, zu pflegen und den Mehrwert des Wissensmanagement für das Unternehmen und das Personal offen zu legen. (vgl. [VDI09])

Begriffe im Wissensmanagement

Zunächst wird der Begriff Wissen etwas näher erläutert, der anhand der Wissenstreppe von North (Abbildung 2-3) dargestellt werden kann. Auf der niedrigsten Ebene sind die Daten angesiedelt, über deren Anwendungszweck noch keine Auskunft gegeben werden kann. Durch eine Analyse und einer Interpretation der Daten, die sie in einem Kontext erscheinen lässt, verwandeln sich die Daten in Informationen. Der Unterschied zwischen Wissen und Informationen liegt darin, dass die Informationen miteinander und innerhalb eines Problemlösungszusammenhangs verknüpft sind und sich durch eine ausgeprägtere Handlungsorientierung kennzeichnen. Wissen im Hinblick auf ein spezielles Einsatzgebiet äußert sich als Können, welches zusammen mit Wollen zum Handeln wird. Korrektes Handeln deutet auf Kompetenz hin, die als Basis für den Wettbewerbsvorteil dient. (vgl. [LEE05] , S. 18)

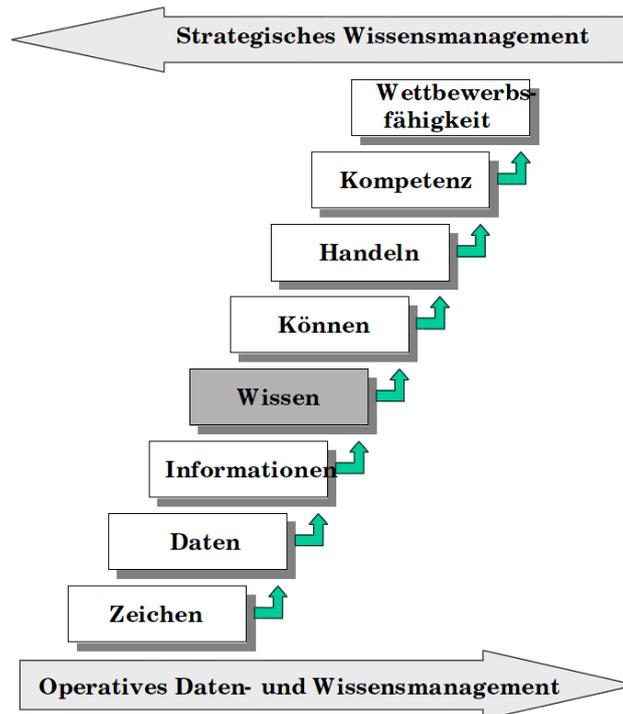


Abbildung 2-3: Wissenstreppe von North ([LEE05] , S.11)

Wissen kann weiter in zwei grundlegende Arten unterteilt werden. Zum einen in implizites Wissen, die aus den subjektiven Erfahrungen eines Menschen erfolgen und sich nicht formalisieren lassen, wie z.B. die Fähigkeit zu schwimmen. Zum anderen in explizites Wissen, welches formal festgehalten werden kann und in digitalen Medien gespeichert werden kann, wie z.B. Berechnungsgrundlagen. (vgl. [STA12] , S. 57)

Eine genaue Definition der Begriffe Daten, Informationen, Wissen, implizites und explizites Wissen ist in der VDI-Richtlinie 5610 [VDI09] enthalten, diese Definitionen werden nachfolgend angeführt.

„*Daten* sind objektive Fakten; sie sind ohne Zusammenhang und weitere Hintergründe nicht deutbar und als „Rohmaterial“ zu verstehen.“ [VDI09]

„*Informationen* sind strukturierte Daten mit Relevanz und Zweck, die in einen Kontext gebracht, kategorisiert, kalkuliert und korrigiert werden können.“ [VDI09]

„*Wissen* ist vernetzte Information, die in die Lage versetzt, Vergleiche anzustellen, Verknüpfungen herzustellen und Entscheidungen zu treffen.“ [VDI09]

„*Implizites Wissen* ist an Personen gebunden, schwer kommunizierbar und kaum formalisierbar.“ [VDI09]

„*Explizites Wissen* ist auf verschiedene Ebenen formalisierbar (z.B. Sprache, Schrift), damit kommunizierbar und in verschiedenen Medien speicherbar.“ [VDI09]

Grundbausteine des Wissensmanagements

Das Wissensmanagement beinhaltet Mittel, die das Ziel einer wirkungsvollen Wissensnutzung anstreben. Es enthält organisatorische, personelle und technische Mittel. Die Abbildung 2-4 veranschaulicht die operativen Elemente des Wissensmanagements. In der Wissensidentifikation werden unternehmensinterne und -externe Gesichtspunkte herangezogen. Dabei wird bestimmt, welches Wissen schon vorhanden ist und die damit erkennbaren Wissenslücken werden durch eine gezielte Identifikation von Quellen erschlossen. Bei dem Element Wissenserwerb wird externes Wissen herangeschafft, wobei in der Wissensentwicklung neues Wissen im Unternehmen erarbeitet wird. Der Bereich Wissensverteilung beschäftigt sich mit der Bereitstellung des Wissens im Unternehmen. In dem nachfolgenden Element Wissensnutzung kann das Wissen für unternehmerisches Handeln und Entscheiden verwendet werden. Der abschließende Bereich Wissensbewahrung hat die Aufgabe das Wissen im Unternehmen zu behalten und einen Wissensverlust zu vermeiden. (vgl. [MUL07] , S.18)

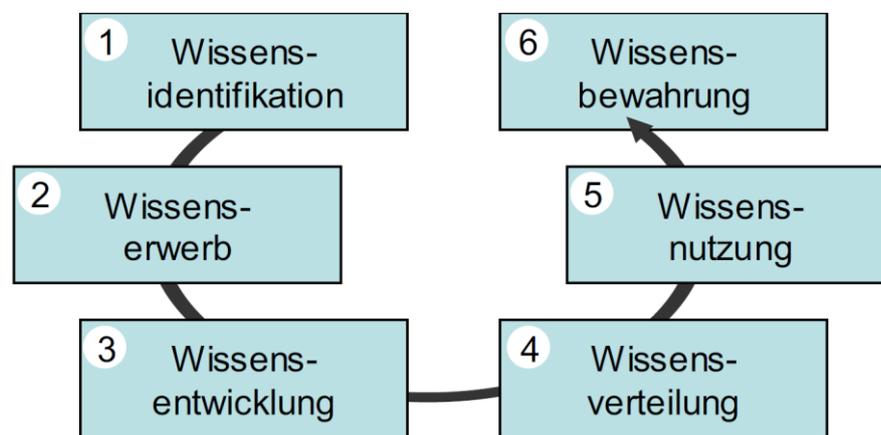


Abbildung 2-4: Operative Wissensmanagementelemente ([MUL07] , S.18)

Ein bekanntes Instrument für Wissensmanagement sind die Expertensysteme, dabei wird das über die Jahre angesammelte Wissen hochspezialisierter Experten in einem EDV-System abgebildet. Mit den jeweiligen Informationen können dem Systemanwender Lösungsmöglichkeiten für ein spezielles Problem angeboten werden. Auf die Expertensysteme wird im Abschnitt 3.3 näher eingegangen. (vgl. [STA12] , S.58)

2.2.2 Wissensrepräsentation

Das Anliegen der Wissensrepräsentation ist die formale Abbildung eines bestimmten Wissensbereiches. Der Ersteller einer Wissensrepräsentation erschafft eine fachliche-thematische Ansicht eines Wissensgebietes. Dabei wird in dem Fachbereich Informatik die zuvor erwähnten Begriffe Daten, Informationen und Wissen für die Erstellung einer Wissensrepräsentation benötigt. Zur Darstellung solcher Wissensdomänen werden semantische Netze und Ontologien angewendet. (vgl. [FAA04] , S.7 ff)

Semantisches Netz

Als semantisches Netz kann man sich ein Modell vorstellen, welches aus einer definierten Menge von begrifflichen Entitäten und aus Beziehungen zwischen den Entitäten besteht. Für das Grundverständnis der Semantik ist die Betrachtung der Zeichenlehre auch als Semiotik bezeichnet, erforderlich. Die Semiotik befasst sich mit der Repräsentationsdarstellung von Informationen in Zeichen und welche Beziehungen zwischen den Informationssender und Informationsempfänger existieren. Hierbei gilt zu unterscheiden, zwischen den Zeichen, der Interpretation der Zeichen und der Wirklichkeit, für welche die Zeichen stehen. Zur Veranschaulichung dieser Beziehungen kann das semiotische Dreieck herangezogen werden, welches in der Abbildung 2-5 dargestellt wird. Der Unterschied zwischen der Semantik und der Syntax besteht darin, dass die Semantik die Bedeutung eines Begriffes zur Wirklichkeit in den Fokus stellt, die Syntax betrachtet hingegen den Zusammenhang der Zeichen zwischen den Zeichnungen, in einer formalen und struktureller Hinsicht. Die Pragmatik definiert den Zusammenhang zwischen der Bezeichnung und dem Bezeichneten in Bezug des Begriffes auf die Wirklichkeit und der Bedeutung für den Benutzer auf die jeweilige Situation. Die Pragmatik stellt die Semantik eines Begriffes in einen stärkeren Kontext mit dem Interpretierenden dar. Die Pragmatik besitzt dadurch eine höhere Subjektivität als die Semantik. (vgl. [GAA10] , S.54 f)

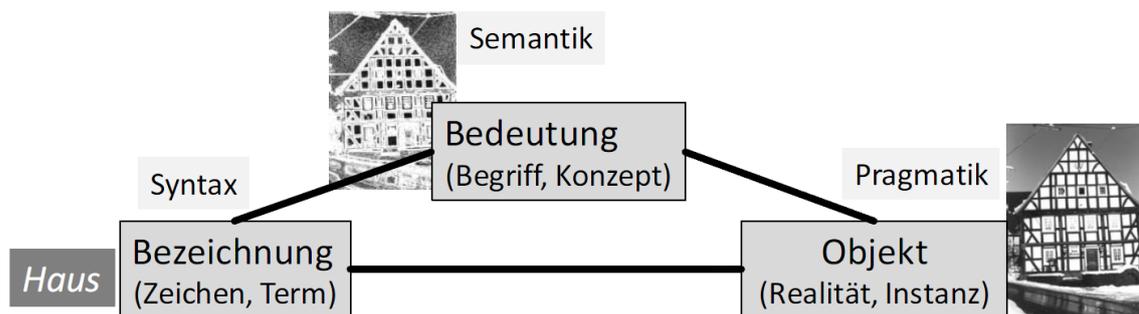


Abbildung 2-5: Semiotisches Dreieck ([GAA10] , S.55)

Ontologie

Unterschiedliche Modelle können eingesetzt werden um die Semantik von Begriffen abzubilden. Diese Modelle unterscheiden sich durch ihre semantische Reichhaltigkeit, zur leichteren Vorstellung kann man sich hierbei eine semantische Treppe vorstellen. Auf der untersten semantischen Stufe befindet sich das Glossar, mit welchen man Begriffe zu einem Thema sammelt. Auf der zweiten Stufe befinden sich die Taxonomien, hier erfolgt zusätzlich zu den Begriffen eine hierarchische Strukturierung. Danach folgt das Modell Thesauri, es enthält weitere Zusatzinformationen zu den Begriffen wie z.B. Synonyme und Deskriptoren. Als nächstes folgen die Topic Maps in denen die Begriffe einem Thema zugewiesen werden und in diesem Thema nach unterschiedlichen Kriterien strukturiert werden. Auf der letzten Stufe der semantischen Treppe und damit am semantisch reichhaltigsten befinden sich die Ontologien. Die Definition einer Ontologie nach dem Autor ([GRU95] , S.908) lautet: „An ontology is a formal, explicit specification of a shared conceptualization.“ Die Konzeptualisierung beschreibt hier die Abbildung der wesentlichen Einheiten eines Anwendungsbereiches der Wirklichkeit als Konzepte. Darin sind explizite Bestimmungen dieser Konzepte und

deren Relationen erforderlich. Ein weiteres Merkmal einer Ontologie ist ihre Formalisierung. Darunter versteht man die formale Dokumentation einer Ontologie, die auch numerisch verarbeitet werden kann. Eine Ontologie illustriert Wissen eines gemeinsamen Verständnisses der Wirklichkeit, jedoch nicht auf dem Verständnis eines einzelnen Menschen. Die Abbildung 2-6 veranschaulicht ein Beispiel einer Ontologie von Fahrzeugen und deren Kraftstoffarten. (vgl. [GAA10] , S.55 f)

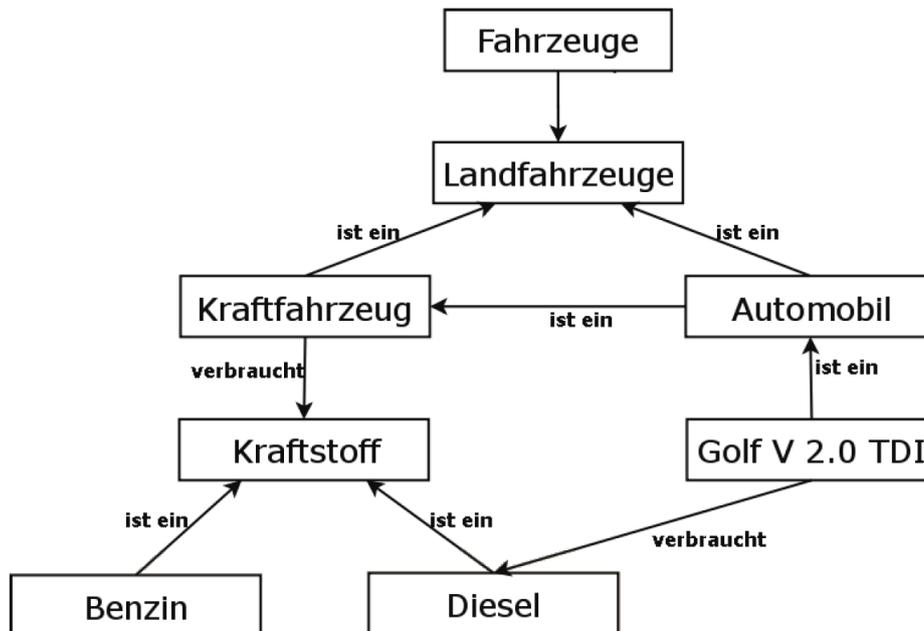


Abbildung 2-6: Beispiel einer Ontologie ([STE09] , S.29)

Im Konstruktionsbereich werden Ontologien für die abstrakte Beschreibung der abhängigen Größen im Konstruktionsprozess und deren Zusammenhänge zueinander verwendet. Die Ontologie erschafft eine Systemumgebung für eine Informationserfassung, die eine Strukturierung und eine Zurverfügungstellung von Informationen bietet. (vgl. [LEE05] , S.36)

In dem Abschnitt 3.6 werden einige Softwaretools zum Thema Wissensrepräsentation betrachtet, des Weiteren wurde eine Ontologie des Fördertyps Rollenförderer erstellt, im Punkt 4.5 erfolgt eine genauere Beschreibung darüber.

2.3 Produktentwicklung

Knowledge-based Engineering wird in der Entwicklung von Produkten eingesetzt, daher wird in diesem Punkt auf die grundlegende Thematik Produktentwicklung sowie auf die methodische Vorgehensweisen bei der Produktentwicklung eingegangen. Des Weiteren wird das Thema Product Lifecycle Management kurz beschrieben.

2.3.1 Einführung Produktentwicklung

Die Produktentwicklung wird durch die planerischen, gestalterischen und produktionsorganisatorischen Elemente bei der Produktentstehung repräsentiert.

Dabei werden alle produktentscheidenden Einflussfaktoren beachtet, damit eine vollständige Berücksichtigung im Bezug der Produktoptimierung ermöglicht werden kann. (vgl. [SCH06] , S.12)

Laut der VDI-Richtlinie 2221 ist die Produktentwicklung folgend definiert:

„Die Produktentwicklung (Entwickeln und Konstruieren) umfasst die Gesamtheit aller Tätigkeiten, die ausgehend von einer Aufgabenstellung für die Entwicklung eines technischen Produktes die für die Herstellung, die Nutzung und das Recycling notwendigen Informationen erarbeitet und dokumentiert.“ [VDI93]

Wie in der Abbildung 2-7 dargestellt, beinhaltet der Produktlebenszyklus die Produktentwicklung, in welcher die Phasen Produktplanung, Produktkonstruktion und Produkterprobung enthalten sind. Die Produktplanung enthält alle Aufgaben, die für die Bestimmung des Gestaltungsrahmens eines zu fabrizierenden Produktes und zur organisatorischen Abwicklung der Produktentwicklung benötigt werden. In der Produktkonstruktion erfolgen die Erstellung eines Lösungskonzeptes aufgrund der geforderten Produktfunktionen, sowie eine konstruktive Verwirklichung des Konzepts. Nach der Erprobung des Produkts durch die Anfertigung eines Prototypen und anschließenden Tests, kann das Produkt zur Herstellung freigegeben werden und die Phase der Produktentwicklung ist damit abgeschlossen. (vgl. [LEE05] , S.8)

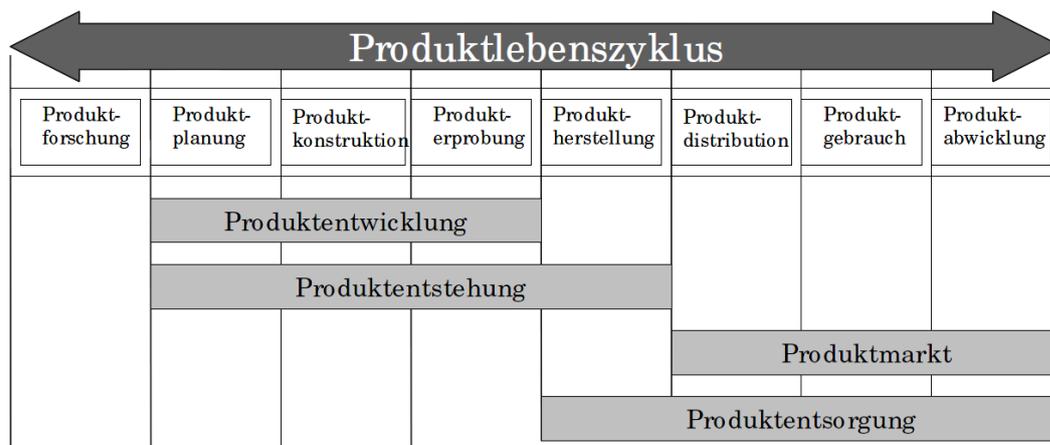


Abbildung 2-7: Einstufung der Produktentwicklung im Produktlebenszyklus ([LEE05] , S7)

2.3.2 Methodisches Vorgehen bei der Produktentwicklung

Da bei der Produktentwicklung die Aufmerksamkeit bezüglich der Kosten und Qualität eines technischen Systems auf alle nachfolgenden Produktlebensphasen gerichtet werden muss, werden systematische und methodische Herangehensweisen benötigt. Die Anforderungen eines Systems bilden den Ausgangspunkt eines methodischen Vorgehens, daraus werden die Funktionen bestimmt und nachträglich sukzessive präzisiert. Die Lösungssuche konzentriert sich dabei auf die funktionsorientierte Perspektive und die Problemabstraktion. Der folgende Abschnitt führt einige Methoden der Produktentwicklung an. (vgl. [GAA10] , S.15)

Vorgehensmodelle zur Produktentwicklung

Die VDI-Richtlinie 2221 bietet ein weitverbreitetes Vorgehensmodell für die Produktentwicklung. Die Inhalte der einzelnen Vorgehensschritte bauen aufeinander auf, jedoch können sie sich zeitlich überschneiden. In der Planungsphase ist das Klären und Präzisieren der Anforderungen und Ziele enthalten. Danach erfolgt die Erstellung des Systemkonzepts. Dies beinhaltet primär die Definition des Strukturumfangs und die prinzipielle Lösungssuche. Die Prinzipiellösungen werden darauffolgend spezifiziert, dies erfolgt durch eine Aufteilung des Systems in Module, welche in weiterer Folge detailliert und gestaltet werden. Die Abbildung 2-8 zeigt die Vorgehensweise nach der VDI-Richtlinie 2221. (vgl. [GAA10] , S.17)

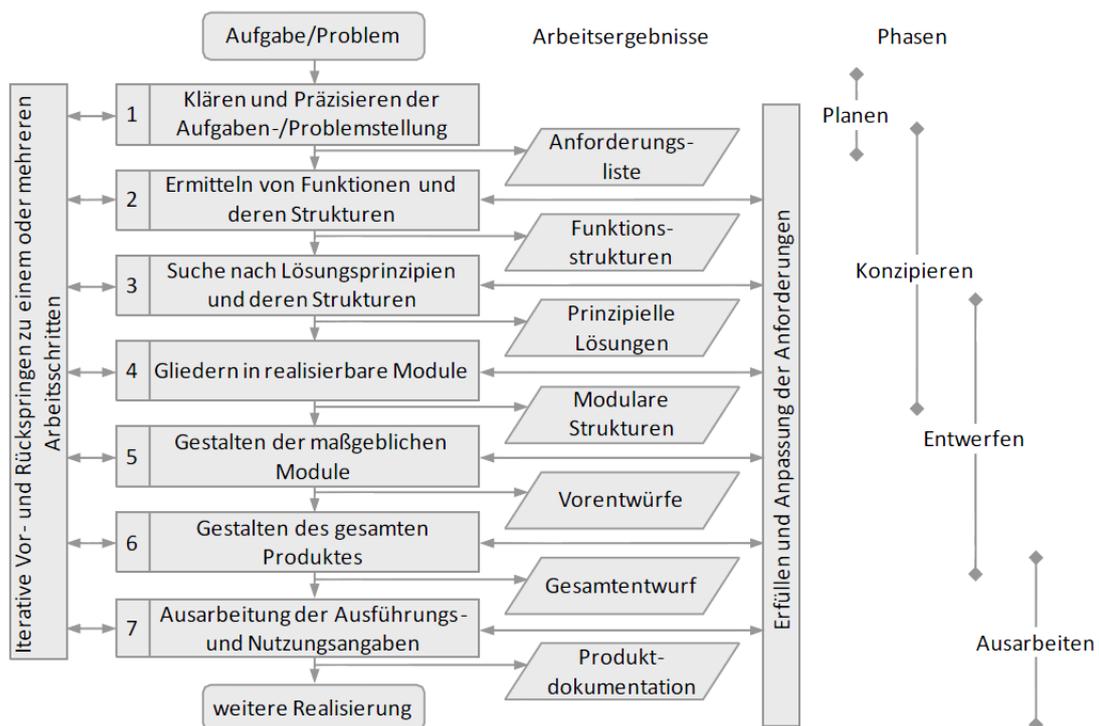


Abbildung 2-8: Vorgehensmodell nach VDI 2221 ([GAA10] , S.17)

Eine weitere Methode bei der Produktentwicklung technischer Systeme ist das Axiomatic Design von Suh, dabei wird der Entwurfsprozess in unterschiedliche Domänen gegliedert, die Zusammenhänge dieser Domänen basieren auf einer mathematischen Verknüpfung und Zuteilung. Die genaue Beschreibung des Axiomatic Design erfolgt im Punkt 3.5. (vgl. [GAA10] , S.18)

Als weiteres Hilfsmittel in der Produktentwicklung und zur Beherrschung komplexer Systeme kann die Design Structure Matrix herangezogen werden. Dabei werden Komponenten oder Parameter eines Produktes in Zeilen und Spalten innerhalb einer symmetrischen Matrix angeordnet und die Abhängigkeiten zwischen den Elementen oder Parameter werden in den Zellen eingetragen. Dadurch lässt sich die Komplexität eines Systems leichter nachvollziehen und verschafft eine Übersicht der Abhängigkeitsbeziehungen zwischen den Parametern oder den Komponenten. Die Design Structure Matrix wird in dem Abschnitt 3.4 näher erläutert. (vgl. [RS12] , S.685)

2.3.3 PLM (Product Lifecycle Management)

Das Product Lifecycle Management abgekürzt PLM, entwickelte sich durch die ständige Erweiterung des Engineering Data Managements (EDM) und Produktdatenmanagements (PDM), welche auf die Phasen Produktentwicklung und Produktentstehung des Produktlebenszyklus fokussiert waren. Der Ansatz PLM umfasst hingegen alle Phasen des Produktlebenszyklus, wie in Abbildung 2-9 dargestellt. Vor allem durch die wachsende Komplexität der Engineering-Prozesse und der verknüpften, digitalen Modelle einschließlich der EDV-Systeme brachten den erweiterten Ansatz PLM hervor. Er konzentriert sich auf die weit verteilten Engineering-Prozesse, dabei werden Prozesse innerhalb des Unternehmens sowie Geschäftsprozesse mit Kunden und Zulieferer mit einbezogen. PLM beinhaltet dabei die notwendige IT-Infrastruktur, das beteiligte Personal und die dynamischen Organisationsaufgaben. Dabei erfolgen das gesamte Stamm- und Entwicklungsdatenmanagement und eine Qualitätssteigerung der Produktentstehungsprozesse. PLM ermöglicht nicht nur eine Einbettung der technischen Daten und Prozesse, sondern bietet auch die Möglichkeit zur Vernetzung mit betriebswirtschaftlichen Systemen, wie ERP-Systeme (Enterprise Resource Planning). (vgl. [SCH06] , S.44)

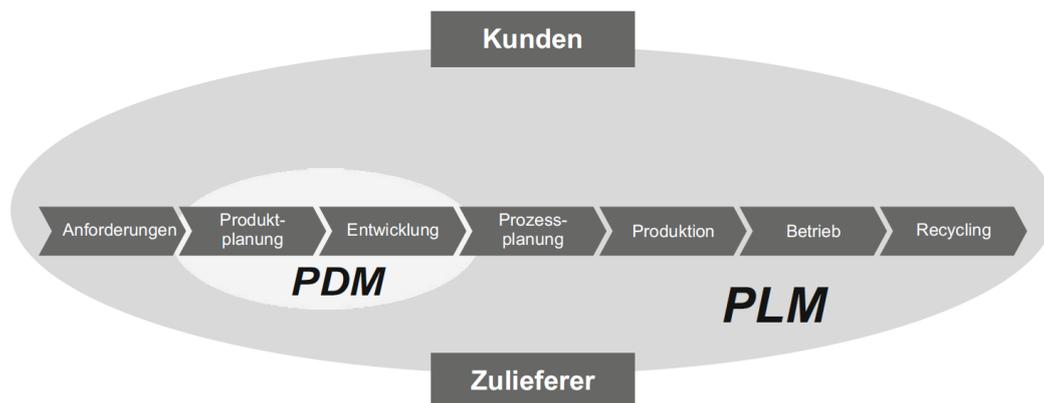


Abbildung 2-9: Erweiterte Integrationstiefe durch PLM ([ES09] , S.37)

3 Systematische Analyse zur Verwendung von KBE

In diesem Kapitel erfolgt die systematische Analyse des Themas KBE und den beiden angrenzenden Thematiken Wissensmanagement und Produktentwicklung. Zum systematischen Ablauf wurden zunächst die zu behandelten Themen spezifiziert. In Bezug auf KBE wurden die KBE-Methodiken und das MDE bestimmt. Im Bereich Wissensmanagement wurden die Expertensysteme sowie Softwaretools zur Wissensrepräsentation festgelegt und zur methodischen Vorgehensweise in der Produktentwicklung wurden die beiden Verfahren Axiomatic Design und Design Structure Matrix festgesetzt. Für die jeweiligen Thematiken wurde zuerst Literatur beschafft. Dabei wurde in Literaturdatenbanken, Bibliothekskatalogen, wissenschaftliche Publikationslisten von Universitäten recherchiert, sowie eine offene Recherche im Internet, indem unterschiedliche Schlagworte verwendet wurden. Nachdem ein erster Überblick über die Thematik verschafft wurde, wurde die Literatur selektiert und darauffolgend beurteilt. Anschließend erfolgte die Ausarbeitung und Dokumentation der betreffenden Themen.

3.1 Knowledge-based Engineering Methodiken

Wie im Kapitel Grundlagen erwähnt, erfolgt unter diesem Punkt die Beschreibung der einzelnen KBE-Methodiken, die den Namen MOKA, KOMPRESSA, DEE und KNOMAD tragen.

3.1.1 MOKA-Methodik

Die Abkürzung MOKA steht für Methodology Oriented to Knowledge-based Engineering Applications. MOKA war ein europäisches Forschungsprojekt, welches am 1. Jänner 1998 begann und sich über einen Zeitraum von 30 Monaten erstreckte. Es sollte als Gerüst für die Strukturierung und Repräsentation von Ingenieurwissen dienen. Dabei wurde das Einsatzfeld einer Knowledge-based Engineering Anwendung vorwiegend auf die Automobil- und Flugzeugindustrie festgelegt. (vgl. [MOK12a])

3.1.1.1 MOKA-Hauptaspekte

Das MOKA-Projekt umfasst die folgenden Hauptaspekte:

- Die Reduktion der Einführungszeit und die damit verbundenen Entwicklungskosten einer KBE-Anwendung um 20-25%.
- Die Verwendung eines konsistenten Weges zur Entwicklung und Wartung einer KBE-Anwendung.
- Die Entwicklung einer Methode, die die Grundlage für den internationalen Standard bilden soll.
- Die Bereitstellung eines Softwaretools zur Unterstützung der Methodik.

Die detaillierte Beschreibung dieser Punkte erfolgt darunter. (vgl. [MOK12b])

Die Reduktion der Einführungszeit und die damit verbundenen Entwicklungskosten einer KBE-Anwendung um 20-25%.

Die primären Kostenkomponenten zur Entwicklung einer KBE-Anwendung, sind die Aufgaben zur Erhebung und Formalisierung des Wissens innerhalb eines bestimmten Fachbereiches. Das mit dieser Aufgabe beauftragte Personal besitzt oft nicht das notwendige Know-how für die Erfüllung dieser Anforderungen. Das führte dazu, dass viele KBE-Programme von Drittunternehmen erstellt wurden. Deswegen hat sich das Projekt MOKA auf die zwei folgenden Punkte konzentriert: (vgl. [MOK12b])

- Die Entwicklung eines strukturierten Gerüsts, welches den Anforderungslevel für die Erfassung und Formalisierung eines Fachbereiches reduziert.
- Die Veranschaulichung des MOKA Softwaretoolpotentials bei der Erstellung eines KBE-Codes. (vgl. [MOK12b])

Die Verwendung eines konsistenten Weges zur Entwicklung und Wartung einer KBE-Anwendung.

Es sollte einfacher werden, bestehende Modelle zu modifizieren oder bereits erarbeitete Lösungen für neue Projekte zu verwenden. MOKA sollte die Modularisierung der Arbeit erleichtern und einheitlichere Ergebnisse von verschiedenen Entwicklergruppen liefern. (vgl. [MOK12b])

Die Entwicklung einer Methode, die die Grundlage für den internationalen Standard bilden soll.

Das Ziel dieses Projektes war es, industriellen Anwendern eine formalisierte Methodik zur Analyse und zur Modellierung von Produkten zu liefern. Dabei sollen die Konstruktionsprozesse mit ihrem hinterlegten Wissen die Kenntnisse eines Unternehmens repräsentieren. Die Methodik soll in der gesamten Industrie gefördert werden und soll als Normungsgremium dienen. (vgl. [MOK12b])

Die Bereitstellung eines Softwaretools zur Unterstützung der Methodik.

Die MOKA-Methodik liefert einen systematischen Denkansatz für die Entwicklung von KBE-Anwendungen. Die Entwicklung einer KBE Anwendung ist meist eine komplexe Aufgabe, in Verbindungen stehen damit Managementbereiche, Regeln und Beschränkungen. Daher ist es wesentlich, dass das Softwaretool die Methodik benutzerfreundlich unterstützt. (vgl. [MOK12b])

Die Anforderungen an das Softwaretool beinhaltet:

- Unterstützung bei der Anwendung der Methodik
- Überprüfung der Modellkonsistenz
- Erleichterung der Iterationen bei der Anwendung des Entwicklungszyklus
- Verbesserung der Softwarequalität

Das Softwaretool ermöglicht den Entwicklern, die Methodik auf die effizienteste Art zu nutzen und verkürzt dabei die Entwicklungszeit. Die Phase der Wissenswartung wird ebenso verbessert. (vgl. [MOK12b])

3.1.1.2 MOKA-KBE-Lebenszyklus

Eines der wichtigsten Ergebnisse des MOKA Projekts war die Definition des sogenannten KBE-Lebenszyklus, der einen detaillierten Leitfaden für die Entwick-

lung einer KBE-Anwendung liefert. Angefangen bei der Identifikation der Möglichkeit einer KBE-Entwicklung bis zum Einsatz der KBE-Anwendung im Unternehmen. Die vier Hauptphasen des KBE-Lebenszyklus dargestellt in Abbildung 3-1 lauten: (vgl. [LAR11] , S.242 f)

- Problem- oder Möglichkeitenidentifikation
- Wissenserfassung und -formalisierung
- Bündelung
- Einsatz

Der Autor [LAR11] hat die Phasen des KBE-Lebenszyklus auf vier Phasen zusammengefasst. In anderen Literaturquellen ist die Phase „Problem- oder Möglichkeitenidentifikation“ auf zwei Phasen aufgeteilt, die „Identifikation“ und „Rechtfertigung“ lauten. Die Phase „Wissenserfassung und -formalisierung“ ist eine eigene Phase für sich und wird nur als „Erfassung“ und „Formalisierung“ bezeichnet.

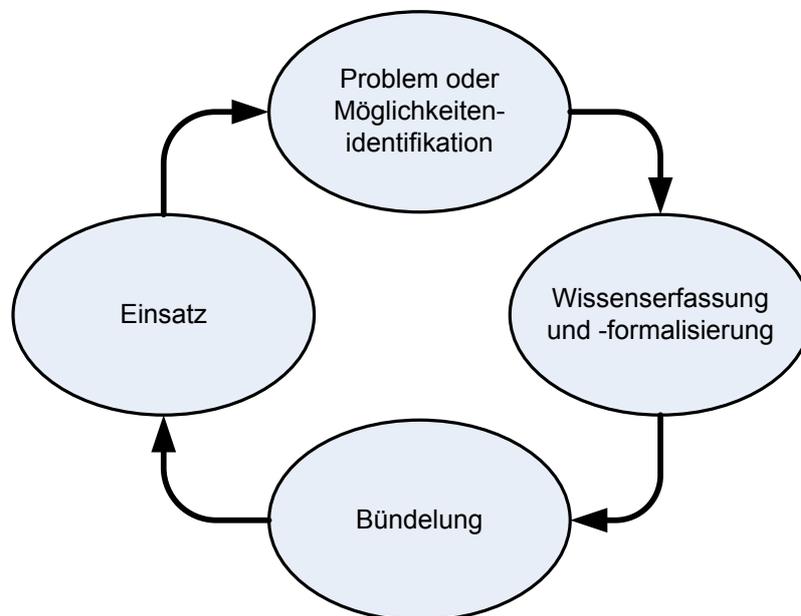


Abbildung 3-1: Hauptphasen des KBE-Lebenszyklus (angelehnt an [LAR11] , S.243)

Problem- oder Möglichkeitenidentifikation

In dieser Vorphase werden die Möglichkeiten zur Entwicklung einer neuen oder zum Modifizieren einer bereits bestehenden KBE-Anwendung abgeklärt. Infolgedessen werden alle notwendigen Ressourcen erhoben, um einen Projektplan zu erstellen. Diese beinhalten die möglichen Wissensquellen, die passenden Wissenserwerbungs-Tools und Techniken, die geeigneten Analyse-Tools und Wissensrepräsentationstechniken. Wenn die erwarteten Vorteile und die technische Realisierbarkeit der neuen bzw. modifizierten KBE-Anwendung gerechtfertigt sind, gibt das Management die Genehmigung um die nächste Phase des KBE-Lebenszyklus durchzuführen. (vgl. [LAR11] , S 243)

Wissenserfassung und -formalisierung

Diese Phase wird in zwei Bereiche untergliedert, die Erfassung und die Formalisierung. Zuerst wird das relevante Wissen von den Experten und anderen Quellen erhoben, danach erfolgt eine informelle Strukturierung des Wissens um es

schematisch darstellen zu können. Damit das erhobene Wissen zurückverfolgt und überprüft werden kann, benötigen die Domänenexperten die informelle Darstellung (d.h. zu gewährleisten, dass während der Erhebungsphase keine Missverständnisse und Unvollständigkeiten aufgetreten sind). Die Wissenserfassungsaktivität beinhaltet die Entfernung der Unklarheiten des gesprochenen Wortes, zur Vorbereitung für die Formalisierung (d.h. die Umsetzung in Regeln). Die Formalisierungsaktivität analysiert das informelle strukturierte Wissen, welches von der Erfassungsaktivität erhoben wurde und repräsentiert es in ein konsistentes und neutrales Format, damit es auf seine Richtigkeit und auf die Fähigkeit zur Wiederverwendung beurteilt werden kann. Diese Tätigkeit könnte dazu führen, andere Iterationen der Erfassung auszulösen, bis alle Widersprüche bzw. Inkonsistenzen beseitigt sind. Die Integration der Erfassungs- und Formalisierungsaktivität vermindert die Lücke des rohen Wissens, welches von den Experten erhoben wurde, für die Formalität der typischen Programmiersprache. In dieser Phase der KBE-Plattform wird noch kein Codierungsvorgang durchgeführt. Der MOKA-Ansatz konzentriert sich hauptsächlich auf diese Phase des KBE-Lebenszyklus. Die beiden Aktivitäten, Erfassung und Formalisierung bestehen aus zwei Modellierungsansätzen, dem informellen Modell und dem formellen Modell, welche im Punkt 3.1.1.3 noch genauer beschrieben werden. Des Weiteren findet eine speziell für die MOKA-Methodik entwickelte Modellierungssprache in dieser Phase ihre Anwendung, eine genauere Erläuterung der Modellierungssprache erfolgt im Punkt 3.1.1.4. (vgl. [LAR11] , S.243 f)

Bündelung

In dieser Phase wird das geprüfte und formelle Wissen, das in der vorhergehenden Phase strukturiert wurde, in die ausgewählte KBE-Umgebung codiert. Im KBE-Lebenszyklus ist dies die eigentliche Programmierungsphase. Die Qualifikation der KBE-Entwickler sowie die Fähigkeiten des ausgewählten KBE-Systems mit seiner Programmiersprache sind ganz entscheidende Faktoren für die KBE-Anwendung. Die Codierung sollte unter der Aufsicht und Genehmigung der Experten und dem Endanwender stehen, welche zur Hilfestellung in der Testung und Validierung dienen sollen, um die Effektivität und Robustheit der entwickelten Module zu gewährleisten. Die Bündelungsphase ist der tatsächliche Übergang von dem neutralen und KBE systemunabhängigen Wissensrepräsentation auf das spezifische Format und die Sprache des gewählten KBE-Systems. Diese Wissenstransformation erfolgt i.d.R. von Hand mit offensichtlichen Grenzen in der Zeiteffizienz und der Transformationsgenauigkeit. Die Erzeugung eines speziellen Übersetzers für die automatische Code-Generierung, auf der Basis einer neutralen Wissensrepräsentation stellt eine anspruchsvolle Aufgabe dar. MOKA zeigt die Machbarkeit der automatischen Generierung aus der Modellierungssprache des formellen Modells. (vgl. [LAR11] , S.244)

Einsatz

In der letzten Phase wird die Anwendung an die potentiellen Benutzer innerhalb einer Organisation verteilt, die davon profitieren könnten. Das KBE-Tool wird installiert, Dokumentation bereitgestellt und die Endanwender werden geschult. In dieser Phase des KBE-Lebenszyklus ist die Unterstützung der IT wesentlich, um eine einwandfreie Infrastruktur für die Speicherung der KBE-Anwendung und für die Bereitstellung an die Benutzer zu liefern. Dies könnte aus einem

Netzwerk von Computern bestehen, die mit einem zentralen Depot verbunden sind und worin die KBE-Anwendung gespeichert wird. Eine adäquate Management Methode für die unterschiedlichen Versionen und Updates der KBE-Anwendung ist allerdings erforderlich (z.B. die Implementierung eines Revision Control System) damit die Entwicklung und Austestung der Anwendung während der gleichzeitigen Bearbeitung von dem Endanwender sichergestellt werden kann. Nachdem die KBE-Anwendung bereitgestellt wurde, um die relevanten Ingenieurtätigkeiten zu unterstützen, könnten neue Möglichkeiten zur Verbesserung oder Erweiterung der Anwendung offensichtlich werden. In diesem Fall kann ein ganz neuer Zyklus von Aktivitäten starten. Eine weitere Spezifizierung und Detaillierung der vier Hauptphasen zeigt die Abbildung 3-2. (vgl. [LAR11] , S.244 f)

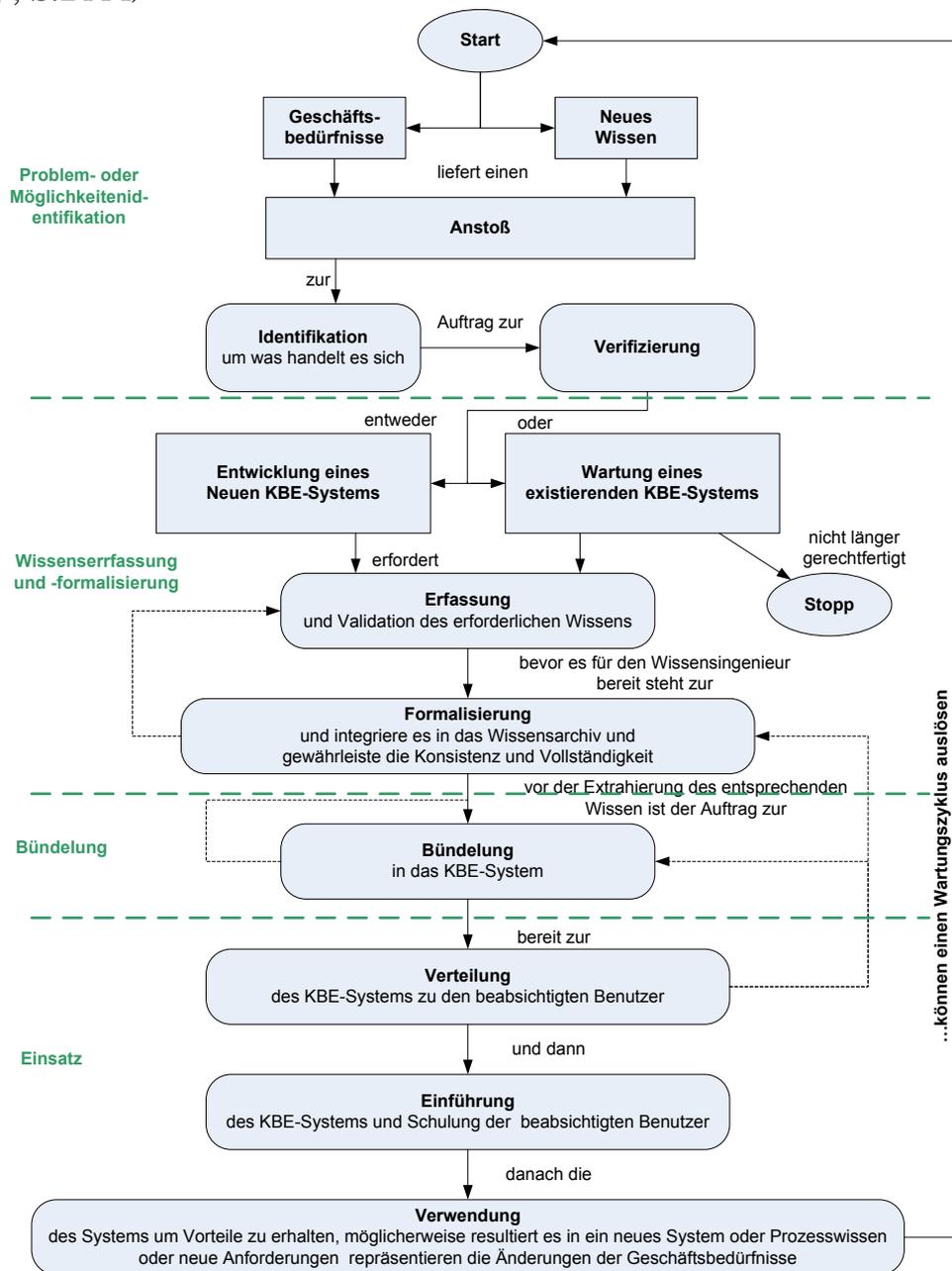


Abbildung 3-2: Spezifizierung der vier Hauptphasen (angelehnt an [LAR11] , S.246)

3.1.1.3 Informelles und Formelles Modell

Informelles Modell

Das informelle Modell wird wie zuvor erwähnt in der Phase Wissenserfassung und –formalisierung erstellt. Es unterstützt die Sammlung und Strukturierung von komplexem Wissen. Dabei bietet es ein nützliches Kommunikationsgerüst zwischen den Wissensingenieuren und den Domänenexperten. Alle Elemente des informellen Modells können zur ursprünglichen Wissensquelle verknüpft werden. Das informelle Modell besteht aus den fünf ICARE Vorlagen, welche wie folgt lauten und in der Abbildung 3-3 dargestellt werden. (vgl. [MOK12c])

- Illustrations (Darstellungen)
repräsentieren das generelle Wissen
z.B.: Übersicht, Beschreibung oder Kommentar
- Constraints (Beschränkungen)
dienen zur Modellierung der Abhängigkeiten zwischen den Entitäten
- Activities (Aktivitäten)
beschreiben die verschiedenen Problemlösungsschritte im Konstruktionsprozess
- Rules (Regeln)
erlauben die Modellierung von Steuerungswissen
- Entities (Entitäten)
beschreiben die Produkt Objekt-Klassen (vgl. [MOK12c])

Die ICARE Vorlagen in dem informellen MOKA Modell können untereinander in Verbindung gesetzt werden und bieten dadurch ein reichhaltiges Repräsentationsnetzwerk für das strukturierte Wissen. In der Abbildung 3-3 kann man die verschiedenen Grafikarten sehen die zur Hilfe herangezogen werden um die Beziehungen der ICARE Vorlagen zu visualisieren. (vgl. [MOK12c])

Das informelle Modell umfasst laut dem Autor (vgl. [MIL08] , S.50) zwei miteinander verbundene Modelle:

- Das informelle Produktmodell:
Die Struktur des Produktes wird so konstruiert, dass die Beschränkungen und Funktionen mit jedem Bauteil verbunden sind.
- Das informelle Prozessmodell:
In jeder Phase sind die Aktivitäten inklusive der Regeln, die vom Konstrukteur ausgeführt werden, beteiligt. (vgl. [MIL08] , S.50)

Formelles Modell

Das formelle Modell hat viel strengere Anforderungen zu erfüllen als das informelle Modell. Formelles Wissen muss in einer Art und Weise digitalisiert werden, dass auf die beabsichtigte Bedeutung des Wissens Rückschlüsse gezogen werden können. All die Konsistenzanforderungen, welche in einer Domäne und bei der Modellierung angewendet werden, müssen durch das formelle Modell erfüllt werden. Dies hat zwei wichtige Konsequenzen zur Folge: (vgl. [MOK12d])

- Das formelle Modell muss in einer einheitlichen und computerverständlichen Form repräsentiert werden.
- Welche die große Vielfalt des Wissens und ihre Aspekte in einem differenzierten Weg darstellen. (vgl. [MOK12d])

Für die Erstellung des formellen Modells wird das informelle Modell als Eingabebasis zur Wissensrepräsentation verwendet. MOKA benützt eine eigene Sprache für die Entstehung des formellen Modelles, die den Namen MOKA Modelling Language trägt und mit MML abgekürzt wird. Die MML ist eine graphisch objektorientierte Repräsentation des Ingenieurwissens und steht eine Ebene über den Anwendungscode. Eine genauere Beschreibung und Erklärung der MML erfolgt im Punkt 3.1.1.4. Das formelle Modell beinhaltet zwei unterschiedliche Modelle: (vgl. [MIL08] , S.50)

- **Das formelle Produktmodell**
Wird für die Spezifizierung der verschiedenen Eigenschaften der Wissens-elemente innerhalb des informellen Modells verwendet. Dabei werden die fünf Ansichten wie, Funktion, Verlauf, Struktur, Repräsentation und Technologie eingesetzt, die ihren Ursprung in der MML haben.
- **Das Design-Prozessmodell**
Dies ist der Ablaufprozess des KBE-Systems und beinhaltet die Beschränkung der Anwendung, des Betriebssystems, die gewählte Programmiersprache und die Spezialistentypen, die in dem Prozess involviert sind. (vgl. [MIL08] , S.50)

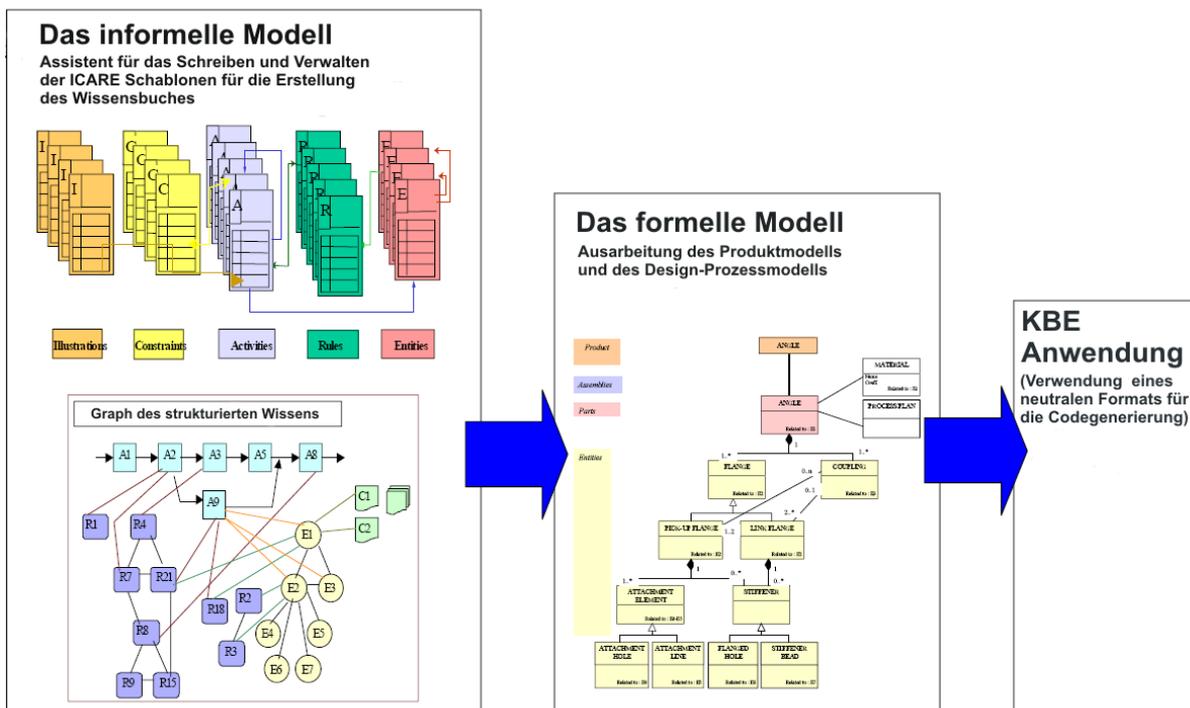


Abbildung 3-3: Die Wissensverarbeitungskette von dem informellen zum formellen Modell (angelehnt an [MOK12d])

3.1.1.4 MML (MOKA-Modelling Language)

Damit ein formelles Modell erstellt werden kann, benötigt der Benutzer eine formelle Modellierungssprache. MOKA hat sich für die UML Unified Modelling Language entschieden, da sie in der Industrie die führende Rolle der objektorientierten Modellierungssprache einnimmt. Ein weiterer Vorteil der UML besteht darin, dass die Softwareentwickler mit dieser Sprache sehr vertraut sind.

MOKA gewährleistet, dass dadurch bewährte Praktiken zum Einsatz kommen. UML ist eine universelle Sprache. Um die Entwicklung von KBE-Anwendungen besser unterstützen zu können, wurde eine eigene Sprache entworfen. Das Ergebnis war die MOKA Modelling Language MML, eine graphisch objektorientierte Modellierungssprache, die auf der UML basiert. Damit kann das konstruktive Ingenieurwissen von KBE-Anwendungen auf einer Benutzerebene repräsentiert werden. MML erbt all die Modellierungsfähigkeiten der UML, dadurch kann der Benutzer bei auftretenden Fehlern auf die Standards der UML zurückgreifen. MML liefert ein Gerüst zur Erstellung von Modellen. (vgl. [MOK12e])

Die MML bietet einen Ratgeber innerhalb der KBE-Entwicklung. Sie ist weder in der Definition noch in der Anwendung starr. MML nimmt bewährte Praktiken von vordefinierten Klassen, Assoziationen und Attributen ein. Diese Elemente ermöglichen dem MML-Benutzer einen strukturierten und logischen Weg für die Modellierung. Ziel ist es, ein Gerüst und einen Leitfaden für den Modellierer zu bieten, der die Verwaltung zur Erstellung und Erfassung von Modellierungspraktiken ermöglicht und nicht vorschreibt, was modelliert werden kann und was nicht. Es ist eine Erweiterung der UML und dem Benutzer steht es frei, ob er auf der UML-Ebene arbeitet, falls es erforderlich ist. Der Benutzer hat die Alternative die Kern MML-Definitionen zu erweitern, damit maßgeschneiderte Lösungen angefertigt werden können. [MOK12f]

Die Kernelemente der MML sind die Produktmodellierung und die Prozessmodellierung. Die Beschreibung der beiden Modellierungen erfolgt darunter. (vgl. [MOK12e])

MML Produktmodellierung

Das Produktmodell besteht aus einer Summe von einzelnen Produktelementen den Produktmodell diagrammen, den Produktmodellansichten, den Produktmodellklassen, den Produktmodellklassenattributen und den Produktmodellbeschränkungen. (vgl. [MOK12e])

Die *Produktmodell diagramme* liefern eine graphische Repräsentation der Produktmodellelemente. Sie stellen Klassen, Klassen-Assoziationen und Klassenattribute dar. (vgl. [MOK12e])

Die *Produktmodellansichten*, dargestellt in der Abbildung 3-4 bestehen aus den einzelnen Ansichten Struktur, Funktion, Verlauf, Technologie und Repräsentation. Benutzer werden jedoch aufgefordert, diese Ansichten nur als Anleitung zu verwenden um ihr eigenes Produktmodell zu erstellen. (vgl. [MOK12e])

- Strukturansicht
- Funktionsansicht
- Verlaufsansicht
- Technologieansicht
- Repräsentationsansicht

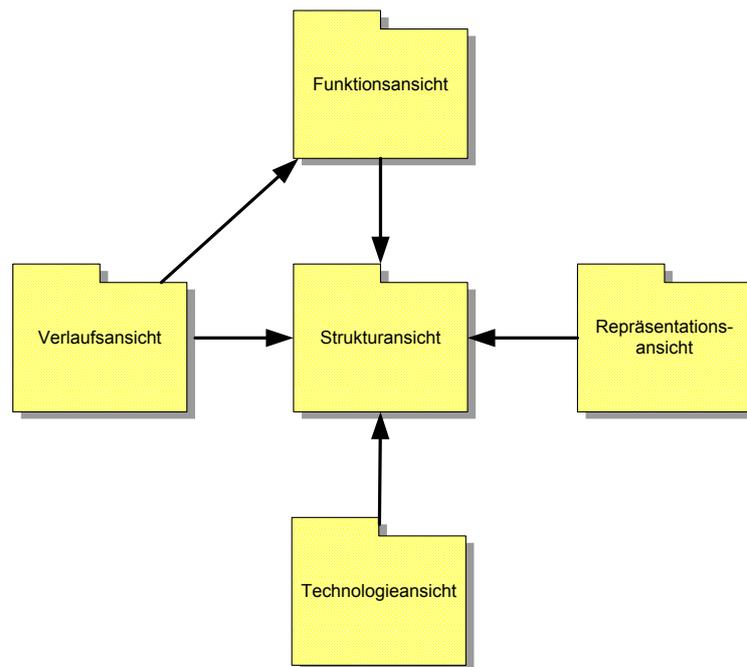


Abbildung 3-4: Beziehungen zwischen dem MML Produktansichten (angelehnt an [MOK12e])

Strukturansicht

Definiert die Zerlegung einer Produktstruktur in Bauteile, Baugruppen und Eigenschaften. Diese Ansicht kann zur Darstellung einer physikalischen, logischen oder konzeptuellen Struktur in jeder Phase der Konstruktion verwendet werden. (vgl. [MOK12e])

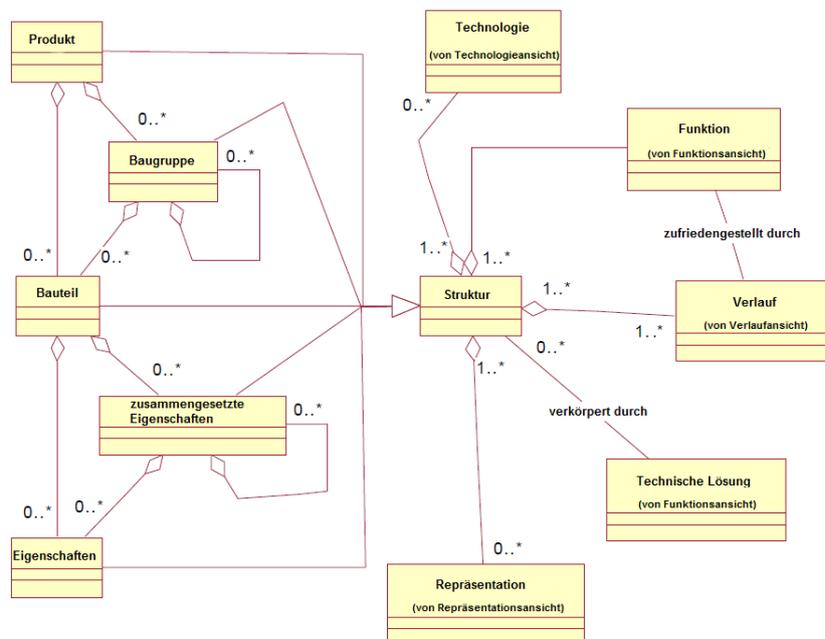


Abbildung 3-5: Strukturansicht (angelehnt an [MOK12e])

Funktionsansicht

Bestimmt die funktionelle Zerlegung des Produktes und stellt fest wie die Funktionen technisch gelöst werden können. (vgl. [MOK12e])

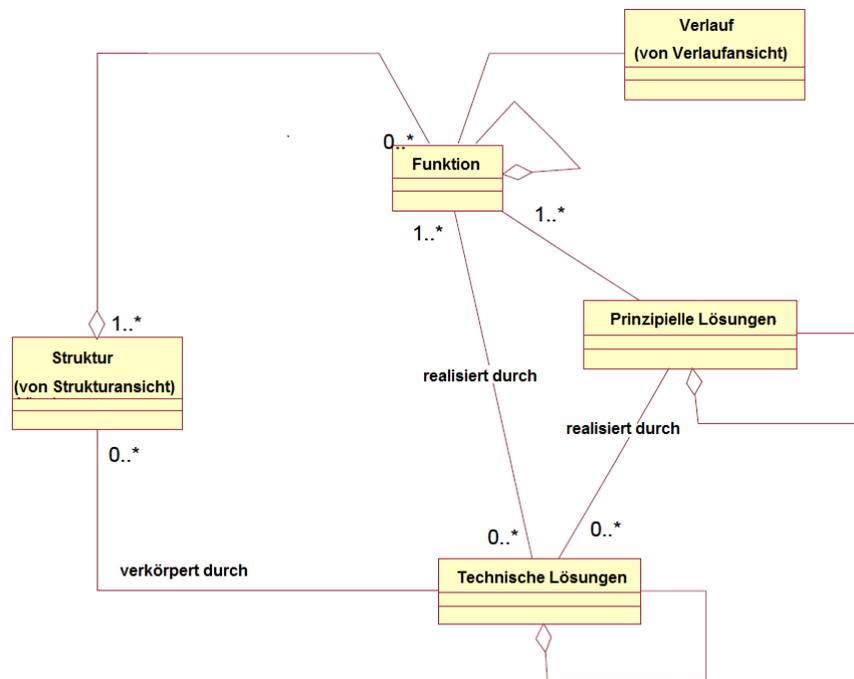


Abbildung 3-6: Funktionsansicht (angelehnt an [MOK12e])

Verlaufsansicht

Enthält eine Phasenmodellansicht für die verschiedenen Phasen eines Produktes und den Übergang von einer Phase auf die Nächste. (vgl. [MOK12e])

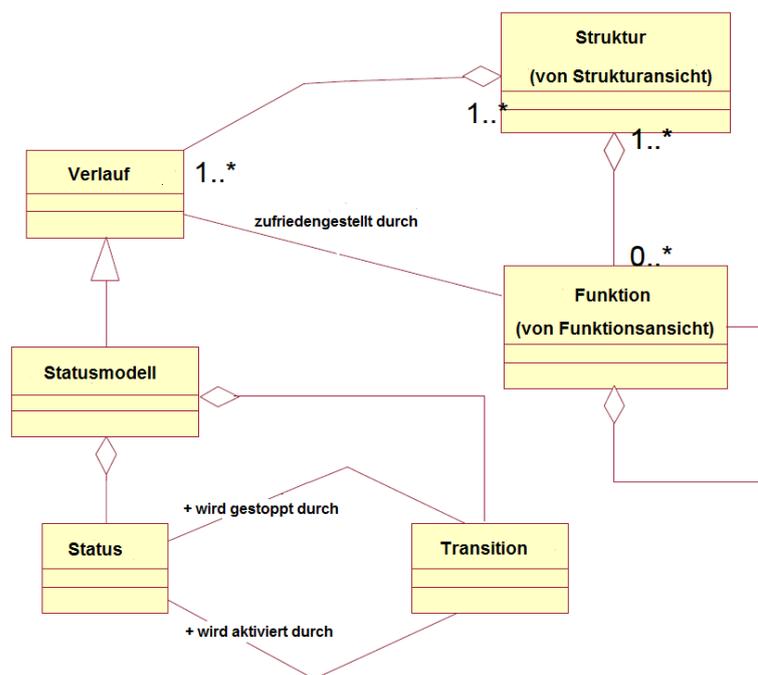


Abbildung 3-7: Verlaufsansicht (angelehnt an [MOK12e])

Technologieansicht

Umfasst eine Material- und auch eine Herstellungsprozessansicht. (vgl. [MOK12e])

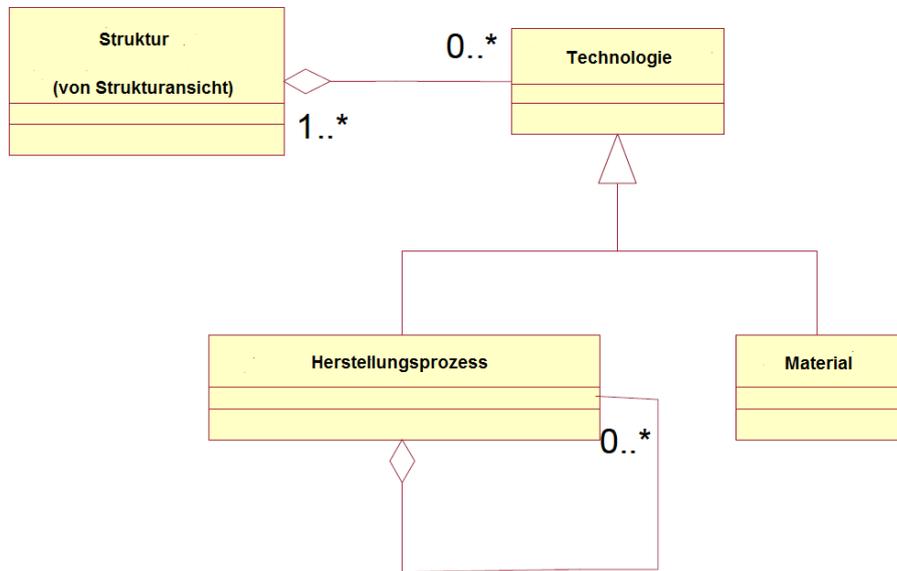


Abbildung 3-8: Technologieansicht (angelehnt an [MOK12e])

Repräsentationsansicht

Beinhaltet die Geometrie des Produktes und ist vom Benutzer erweiterbar. (vgl. [MOK12e])

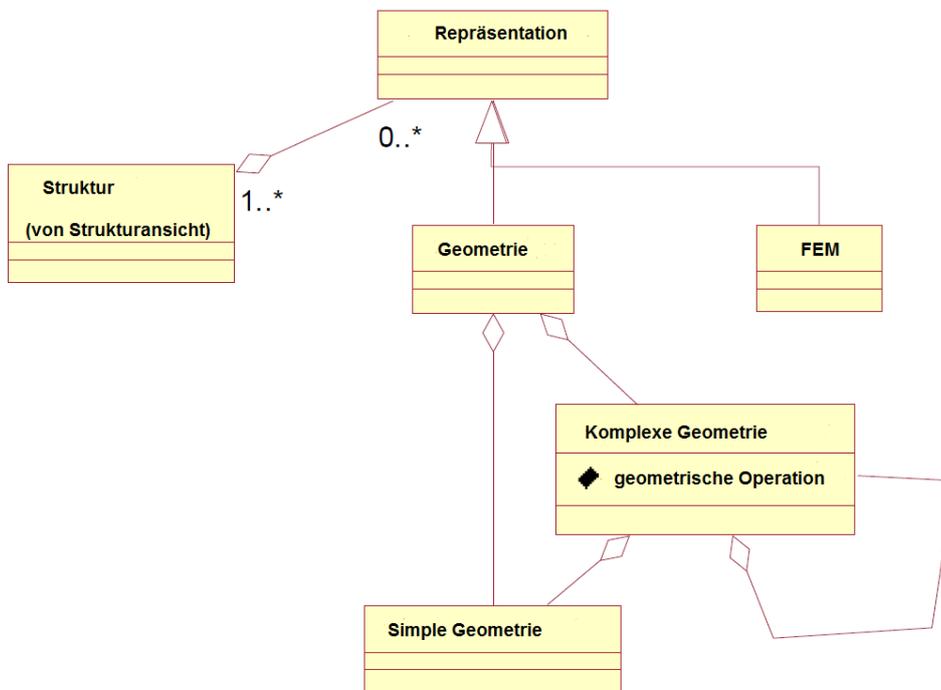


Abbildung 3-9: Repräsentationsansicht (angelehnt an [MOK12e])

Die *Produktmodellklassen* identifizieren und beschreiben die Komponenten der Produktfamilie. Diese werden allgemein beschrieben, bei der Instanziierung einer Klasse wird ein Objekt in der endgültigen Produktmodellinstanz erstellt. (vgl. [MOK12e])

Die *Produktmodellklassenattribute*, drücken die Merkmale des Produktmodelles aus. Diese Klassenattribute besitzen mindestens einen Namen, eine Type und optional einen Wert mit der damit verbunden Beschränkung. (vgl. [MOK12e])

Die *Produktmodellbeschränkungen* repräsentieren die Konstruktionsbedingungen, welche erfüllt werden müssen. Als Beispiel sind die Abmessungen eines Bauteils häufig von den Abmessungen anderer Bauteile abhängig. MML klassifiziert die Produktmodellbeschränkungen danach wie sie innerhalb der Produktmodellidiagramme modelliert wurden: (vgl. [MOK12f])

- **Attributsgrenzen**
Beschränkungen gerichtet auf die Grenzen eines Attributs z.B. $X < 10$
- **Klassengrenzen**
Beschränkungen gerichtet auf die Grenzen einer Klasse
z.B. Klasse X < Klasse Y
- **Assoziationsgrenzen**
Beschränkungen gerichtet auf die Verbindung zwischen den Klassen
z.B. Klasse A.X < Klasse B.Y
- **Existenzbeschränkungen**
Beschränkungen verbunden zur Existenz der Objekte innerhalb der endgültigen Produktmodellinstanz (vgl. [MOK12f])

MML Prozessmodellierung

Der MOKA Designprozess beschreibt die Schritte zur Realisierung der Produktmodellinstanz, die von einem Produktmodell abgeleitet werden. Es definiert die Lösung der Produktauswahl, die den Produktbeschränkungen unterliegen und in welcher Reihenfolge die Konstruktionsschritte und Konstruktionsentscheidungen ausgeführt werden sollen. Die MML Prozessmodellierung umfasst die Punkte statische und dynamische Prozesse, das Prozessmodell sowie die Prozessdiagramme. (vgl. [MOK12e])

Der Ablauf wie Konstruktionsaktivitäten durchgeführt werden sollen, ist nicht immer fixiert und verfolgt unterschiedliche Strategien. Dies führt zu einer differentiellen Ausführungsreihenfolge, welche von den Laufzeitbedingungen abhängig sind. MOKA unterscheidet zwischen dem statischen und dem dynamischen Konstruktionsprozess. (vgl. [MOK12e])

- *Statischer Konstruktionsprozess:* Beim statischen Konstruktionsprozess ist die Reihenfolge der Aktivitäten fix festgelegt, d.h. vordefiniert.
- *Dynamischer Konstruktionsprozess:* Beim dynamischen Konstruktionsprozess wird die Reihenfolge der Ausführungen nicht fixiert, sondern wird erst während der Laufzeit bestimmt. (vgl. [MOK12e])

Das *Prozessmodell* ist das zugrundeliegende Metamodell des Prozesses, unabhängig von der Informationsart werden sie dem Benutzer präsentiert. Es erfasst die Aktivitätenzerlegung, den seriellen und parallelen Ausführungsablauf, den Verzweigungsgrad, die Synchronisierung, den statischen und dynamischen Prozess. Das UML Aktivitäten Modell wurde um die zwei neuen Klassen Verbindungsaktivität und Elementaraktivität erweitert. Verbindungsaktivität ist eine Spezialisierung der UML-Klassen und wird als Aktivitätenphase bezeichnet, die für die Repräsentation der Phase verwendet wird und eine Reihenfolge von Aktivitäten beinhalten. Elementaraktivitäten sind eine Spezialisierung der UML-Klassen und werden als Aktionsphase bezeichnet, die für die Repräsentation der Phase verwendet wird und die eine Einzelaktivität beinhalten. (vgl. [MOK12e])

Die *Prozessdiagramme* können in unterschiedliche Ausführungsmöglichkeiten die Konstruktionsprozessmodelle für den Benutzer repräsentieren. Ein naheliegender Ansatz ist es, Klassendiagramme wie bei dem Produktmodell zu verwenden. UML bietet hier eine Alternative, die als Aktivitätsdiagramme bezeichnet werden. Sie repräsentieren die Aktivitäten und Abläufe in einer ähnlichen Art und Weise wie es bei Flussdiagrammen üblich ist. Der Klassendiagrammansatz wird nicht von MOKA empfohlen, kann aber bei Tools, die die Aktivitätsdiagramme nicht unterstützen, eingesetzt werden. (vgl. [MOK12e])

3.1.2 KOMPRESSA-Methodik

Die KOMPRESSA-Methodik wurde im Zuge des Projektes REFIT im Jahre 1999 von der Coventry Universität entwickelt. Die Abkürzung KOMPRESSA bedeutet Knowledge-Oriented Methodology for the Planning and Rapid Engineering of Small-Scale Applications. Der Projektname REFIT steht für Revitalisation of Expertise in Foundries using Information Technology. Ziel des Projektes ist es, die Wettbewerbsfähigkeit von klein- und mittelständischen Unternehmen in der Gießereiindustrie zu verbessern. Die Prinzipien und Praktiken, die bei REFIT zum Einsatz kommen, eignen sich auch für andere Fachbereiche im Ingenieurwesen. (vgl. [LIB00])

3.1.2.1 KOMPRESSA-Merkmale

Der gesamte Entwicklungslebenszyklus wird von der KOMPRESSA-Methodik abgedeckt. Die Beteiligung des Kunden wird maximiert und sie bietet den Vorteil, dass sie sehr flexibel ist. KOMPRESSA eignet sich besonders gut für kleinere Unternehmen, kleinere Projekte, Ingenieurbüros und für unerfahrene Benutzer. (vgl. [KOM12])

KOMPRESSA bietet laut (vgl. [KOM12]):

- Richtlinien für die Analyse, um die Aktivitäten des Produktes zu strukturieren.
- Anleitungen für die Konstruktion, um die Praktiken zu standardisieren und die Wiederverwendbarkeit zu verbessern.
- Formate für die Modellierung, um das komplexe Wissen zu repräsentieren, manipulieren und zu verwalten.
- Techniken für die Implementierung, um die Fähigkeiten von erfahrenem Personal auf neue Mitarbeiter zu übermitteln.
- Hinweise für die Dokumentation, um die Planung, Überwachung und die Steuerung zu unterstützen.

3.1.2.2 KOMPRESSA-Aktivitätsgruppen

Die Aktivitätsgruppen der KOMPRESSA-Methodik liefert die Klarstellung der organisatorischen Strukturen und identifiziert das Anwendungsgebiet für mögliche KBE-Lösungen. KOMPRESSA setzt sich aus neun Aktivitätsgruppen zusammen, die in der Abbildung 3-10 dargestellt werden. (vgl. [KOM12])

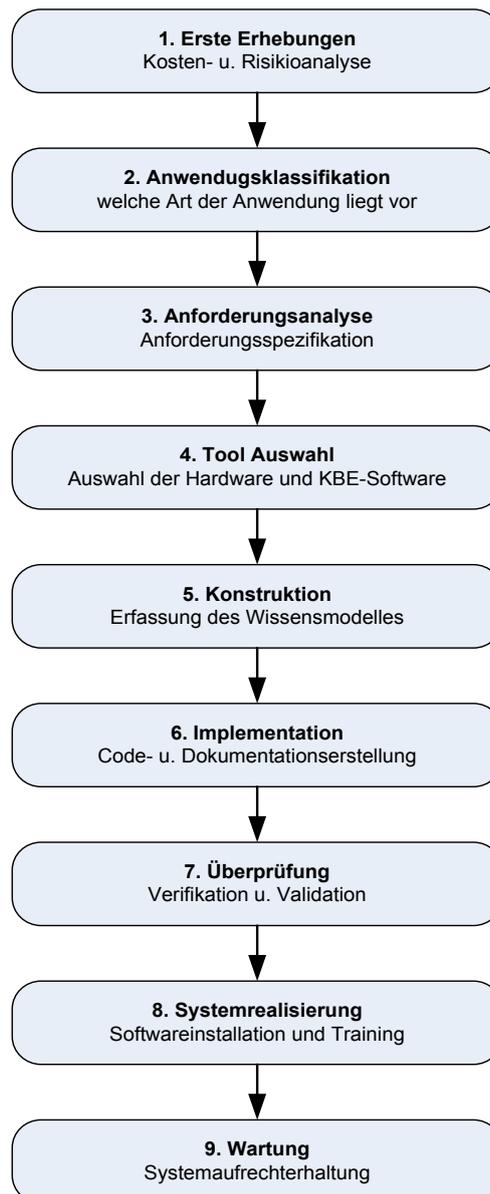


Abbildung 3-10: Aktivitätsgruppen von KOMPRESSA (angelehnt an [KOM12])

Erste Erhebungen

In der Aktivitätsgruppe „erste Erhebungen“ werden für die Abklärung der Geschäftsbedürfnisse strategische Diskussionen durchgeführt. Interviews finden in dieser Phase auch statt, damit die gesamte organisatorische Struktur bestimmt werden kann. Des Weiteren werden Kosten/Nutzen- und Risikoanalysen erstellt. Als Abschluss findet eine Projektautorisierung statt, welche über die Fortsetzung des Projektes entscheidet. (vgl. [LIB00])

Anwendungsklassifikation

Nach der Klassifizierung der Anwendung liegt ein völlig ausgefüllter Fragebogenkatalog vor, der die Anwendungsart spezifiziert hat. (vgl. [LIB00])

Anforderungsanalyse

Bei diesem Abschnitt werden die Anforderungsanalysen für die Funktion, Benutzeroberfläche, Daten und die Wissenserhebung erstellt. Diese Analysen lie-

fern eine Funktionsstrukturtafel, einen Benutzeroberflächenentwurf, eine Datenstrukturbeschreibung und einen Wissenserhebungsplan. Die Durchführung dieser Analysen bietet einen Gesamtüberblick über die Anforderungen, der als Entscheidungshilfe für die Projektfortführung dienen sollen. (vgl. [LIB00])

Tool Auswahl

Hier werden Untersuchungen über die zu verwendende Hardware und KBE-Software durchgeführt, welche zu einer anforderungsgerechten Entscheidung für die Hardware- und Softwarewahl führen soll. (vgl. [LIB00])

Konstruktion

In der Konstruktionsebene werden die Benutzeroberfläche und das Datenmodell entworfen und es erfolgt die Erstellung und Rationalisierung des Wissensmodells. Des Weiteren wird eine Strategie für die Überprüfung festgelegt. Das abschließende Konstruktionsprotokoll dient als Entscheidung für den Übergang in die nächste Phase. (vgl. [LIB00])

Implementation

Die Aktivitätsgruppe Implementation ist für die Erstellung des Programmcodes und Programmdokumentation verantwortlich. (vgl. [LIB00])

Überprüfung

Bei der Überprüfungsphase erfolgt die Durchführung der Tests an das System. Dabei wird die Verifikation und Validation durchgeführt. (vgl. [LIB00])

Systemrealisierung

Die Softwareinstallation erfolgt in dieser Ebene, der Aktivitätsgruppen. Nach der Installation werden die Benutzer auf das System geschult und es wird die Dokumentation und das Benutzerhandbuch erstellt. (vgl. [LIB00])

Wartung

Für die Aufrechterhaltung des Systems, muss es in regelmäßigen Abständen gewartet werden. (vgl. [LIB00])

3.1.3 DEE-Methodik

Die Design und Engineering Engine kurz DEE ist ein erweitertes Konstruktionsgerüst und wurde von der Gruppe für Flugperformance und Antrieb an der TU Delft im Jahre 2002 entwickelt. Dabei wird der Konstruktionszyklus in Synthese, Analyse und Bewertung aufgeteilt, diese sind in ein formelles Gerüst für die Unterstützung der Konstruktionsaktivitäten von multidisziplinärem Konstruktionsoptimierungen eingegliedert. (vgl. [COO11] , S.27)

3.1.3.1 DEE-Komponenten und Prozessablauf

Die Hauptkomponenten der Design und Engineering Engine sind der Initiator, der Multi-Modell Generator, das Analysewerkzeug, die Konvergenz- und Bewertungseinheit sowie das Kommunikationsgerüst. Die Anordnung der Komponenten und der Prozessablauf werden in der Abbildung 3-11 dargestellt. (vgl. [LAR11] , S.41)

Initiator

Das Initiatormodul besteht aus einer Sammlung von Dimensionierungswerkzeugen, die die Initialwerte für den Multi-Modell Generator, kurz MMG, festlegen. D.h., hier werden manuell Startwerte für die Berechnung und Generierung gesetzt und es erfolgt eine Kalkulation des Produktmodelles. Der Initiator erstellt für den MMG eine anfängliche Machbarkeitsprüfung, bevor man mit der multidisziplinären Analyse und dem Optimierungsprozess starten kann. Er hat auch die Fähigkeit die erforderlichen Geometrieinformationen für den Machbarkeitsprozess zu extrahieren. Des Weiteren besitzt er eine Optimierungstechnik und beschäftigt die Konvergenz- und Bewertungseinheit. (vgl. [LAR11] , S. 41 f)

Multi-Modell Generator

Der MMG verwendet die KBE-Technologie für die zentrale parametrische Modellierung. Aus den gesetzten Eingangsparameterwerten, die im Initiator festgelegt wurden, wird im MMG das parametrische Modell generiert. Der MMG bietet die Möglichkeit, z.B., ein Flugzeugmodell zu einem bestimmten Parametersatz zu erstellen, jedoch verfügt er über keinerlei autonomes Kenntnis über die Auswahl dieser Parameterwerte. Danach werden automatisch die Reportdaten für die physikalischen Analysewerkzeuge erzeugt wie z.B., die Finite Elemente- und die Computational Fluid Dynamics-Berechnungen. (vgl. [COO11] , S.41)

Disziplinäre Analysewerkzeuge

Die disziplinären Analysewerkzeuge sind computerunterstützte Systeme, die Analysen mit geringer oder großer Genauigkeiten liefern können. Sie werden entweder intern entwickelt oder von Fremdfirmen als fertiges Tool zur Verfügung gestellt. Die DEE enthält keine fixe Produktserie von Analysewerkzeugen. Je nach Projektanforderungen können die jeweiligen Analysewerkzeuge eingesetzt werden. Die Abbildung 3-11 zeigt, dass die disziplinären Analysewerkzeuge in separaten Depots angeordnet sind und nicht direkt im Multi-Modell Generator eingebettet sind. Zum Beispiel verwendet ein strukturelles Analysewerkzeug die Lastfälle, die aus einem aerodynamischen Analysemodul berechnet wurden. Im Fall von einem stark gekoppelten Analysewerkzeugdepot

kann es als ein multidisziplinäres Instrument betrachtet werden. (vgl. [LAR11] , S.42)

Konvergenz- und Bewertungseinheit

Die Konvergenz- und Bewertungseinheit hat die Aufgabe zu überprüfen, ob die verschiedenen Analysewerkzeuge die Konvergenz erreicht haben und ob die Leistungen und Merkmale der bewerteten Konstruktion die Ziele, die vom Konstrukteur gesetzt wurden, erreicht sind. Dieses Modul empfängt die Ergebnisse, die von den unterschiedlichen Disziplinberechnungen erstellt wurden und generiert daraus neue Parameter für die Einspeisung in den Multi-Modell Generator, die mit dem implementierten Optimierungsalgorithmus übereinstimmen müssen. Der MMG modifiziert darauffolgend das Modell und aktualisiert die Daten für die Durchführung einer neuen Analyseschleife. Im Falle einer unmöglichen Erfüllung der Initialanforderungen beendet die Konvergenz- und Bewertungseinheit die Interaktion mit dem MMG und ruft erneut den Initiator auf, damit eine andere Konfiguration festgelegt werden kann. (vgl. [LAR11] , S.42)

Kommunikationsgerüst

Das Kommunikationsgerüst, dargestellt durch den Verbindungs- und Verknüpfungssatz der verschiedenen Komponenten in der DEE (wie in Abbildung 3-11 abgebildet), kümmert sich um den Daten und Informationsfluss zwischen den verschiedenen Konstruktions- und Analysewerkzeugen und verwaltet den gesamten Konstruktionsprozessablauf. Eine innovative agentbasierende Architektur wurde implementiert, damit ein hoher Grad an Flexibilität der DEE gewährleistet werden kann. Das Agentensystem ermöglicht eine sehr schnelle Einstellung und Rekonfiguration des gesamten multidisziplinären Optimierungsverfahrens, für den konkreten Konstruktionsfall und dem Standort des verfügbaren Konstruktions- und Analysewerkzeuges. Die Agenten nutzen das Internet, um Gruppen von heterogenen Konstruktion und Algorithmen zu entwickeln, welche an verschiedenen Servern und Computern installiert werden können und die Möglichkeit haben, an unterschiedlichen Betriebssystemen zu laufen und zu separaten Netzwerken gehören können. (vgl. [LAR11] , S.42 f)

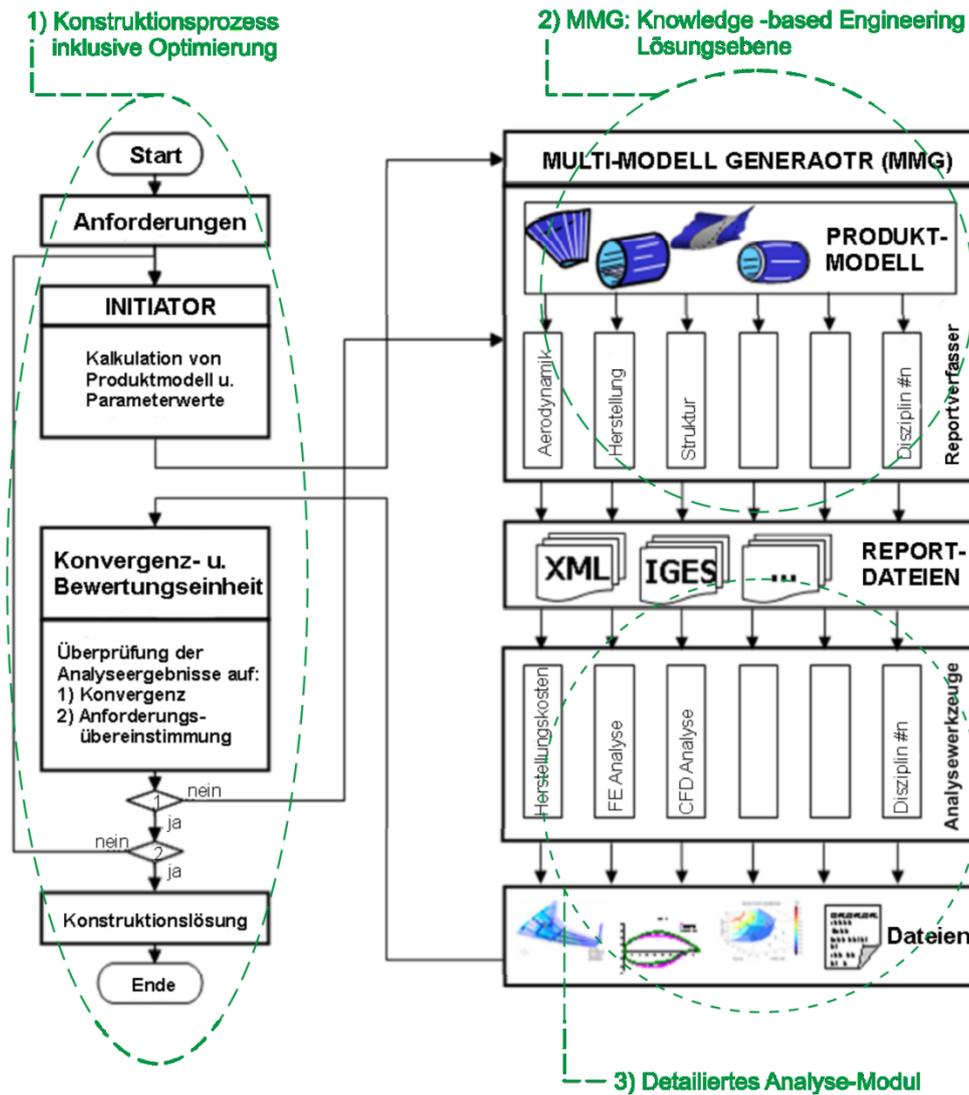


Abbildung 3-11: Prozessablauf der DEE (angelehnt an [LAR11], S.43)

3.1.3.2 DEE-Agenten System

Das agentenbasierende Kommunikationsgerüst wurde von den Herrn Berendes, van Tooren und Schut in den Jahren 2007 bis 2008 entwickelt. Es zielt auf die Imitation einer organisatorischen Struktur für ein Konstruktionserstellungsteam ab, worin die verschiedenen Projektbeteiligten in einen flexiblen serviceorientierten Ansatz arbeiten und interagieren, als das in einem fest vorgegebenen prozeduralen Schema der Fall wäre. Die Abbildung 3-12 zeigt eine formelle und vereinfachte Darstellung des Konstruktionserstellungsteams. In der Softwareimplementierung des Gerüsts müssen alle DEE-Spezialisten-Werkzeuge in eine Softwareagenten eingepackt sein, welche die Verbindung mit anderen Werkzeugen ermöglicht. Jedes Spezialistenwerkzeug kann durch ein anderes ausgewechselt werden oder durch eine aktuellere Version desselben Werkzeuges ersetzt werden, wenn die Schnittstelle konsistent gehalten wird. Dadurch ermöglichen die Agenten eine Plug & Play Einstellung für die Design and Engineering Engine. Das Gerüst verwendet eine Art Peer-to-Peer/Server-Client Archi-

tektur, wobei die DEE-agenteneingepackten Werkzeuge als Server ernannt werden, während all die anderen die Clients sind. Jede Agentenwerkzeugkombination, welche der DEE beitreten will, muss sich am Server registrieren. Von Diesem erhalten sie die aktuelle Liste von all den anderen zurzeit registrierten Clients. Die Rolle des Servers wird von dem „höchsten Expertenagenten“ eingenommen (d.h., wer die längste Aktivität in der DEE vorzuweisen hat), aber je nach Notwendigkeit kann jeder Agent in der DEE gleichermaßen als Server fungieren. Wenn sich der ernannte Serveragent von dem Netzwerk trennt, übernimmt der zweitälteste Agent die Rolle des Servers um die Integrität der DEE-Struktur und die Kontinuität des Konstruktionsprozesses zu gewährleisten. Das agentenbasierende Gerüst benötigt keine vordefinierte Beschreibung des Prozessablaufes wie die meisten derzeit verfügbaren Systemintegrationen, jedoch wird ein abrufgetriebener oder serviceorientierter Ansatz für die Selbstkonfiguration des Prozessablaufes verwendet. Jeder Agent hat die Möglichkeit, wenn die Ausgabe des eingepackten Werkzeuges angefordert wird entweder von einem anderen Agenten oder von dem Konstrukteur, den Konstruktionsprozess zu erstellen. Seit sich jeder Agent bewusst ist, dass eine Eingabe benötigt wird, um sein internes Werkzeug zu starten, sind Anfragesendungen für die Eingabe aller Agenten notwendig und diese machen die DEE lebendig. Jeder Agent ist sich auch über die Ausgabeeinflüsse seines eingebetteten Werkzeuges im Klaren. Im Falle, dass ein Agent in der Lage ist, die Anfragesendung zu beantworten, wird eine Peer to Peer Verbindung mit den anfragenden Agenten aufgebaut. Die erforderlichen Eingabedaten werden generiert und ihr Speicherort wird dem nachfragenden Client für den Abruf mitgeteilt. (vgl. [LAR11] , S.44)

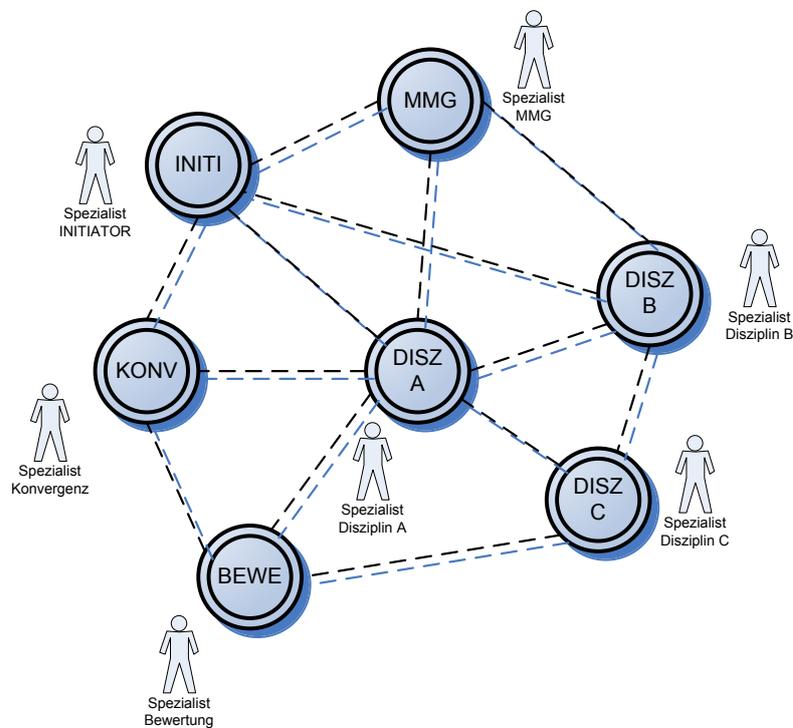


Abbildung 3-12: DEE Agenten-System (angelehnt an [LAR11] , S.44)

3.1.4 KNOMAD-Methodik

Die Abkürzung KNOMAD steht für Knowledge Nurture for Optimal Multidisciplinary Analysis and Design. Da die MOKA-Methodik nur darauf abzielt, eine KBE-Anwendung zu entwickeln und nicht dafür bestimmt ist, eine größere multidisziplinäre Konstruktionsanwendung wie die DEE abzudecken, wurde die KNOMAD-Methodik an der TU Delft entwickelt. Während MOKA für jegliche KBE-Anwendungsentwicklungen eingesetzt werden kann, ist die KNOMAD-Methodik speziell für die Unterstützung des DEE-Konzeptes entwickelt worden. (vgl. [COO11], S. 39)

3.1.4.1 KNOMAD-Merkmale

Für eine analytische Verwendung eines multidisziplinären Ingenieurwissens innerhalb des Konstruktionsprozesses sind für die KNOMAD-Methodik folgende Merkmale entscheidend. (vgl. [CVT10])

- Ein Ansatz für eine multidisziplinäre Konstruktionsoptimierung.
- Eine Vorgehensweise für Wissenserfassung und –formalisierung.
- Eine signifikante Begründung der Methodik-Phasen beinhalten Beispiele, Tools und Techniken.
- Eine ausreichende Flexibilität und Skalierbarkeit für die disziplinspezifische Anwendung, z. B., könnte die Fertigungsdisziplin ein Anwendungsfall für die Instanziierung einer neuen Methodik sein.

3.1.4.2 KNOMAD-Phasen

Die KNOMAD-Methodik erfordert bestimmte Implementierungsphasen, die unternommen werden müssen und die wiederholt werden können, als Teil des Wissenslebenszyklus. In diesem Kontext KNOMAD pflegt das gesamte Wissensmanagement durch den Lebenszyklus wie auch in jeder speziellen Phase. Die Abbildung 3-13 stellt die KNOMAD Phasen dar, die folgend lauten: (vgl. [CVT10])

- Wissenserfassung
- Normierung
- Organisation
- Modellierung
- Analyse
- Bereitstellung

Wissenserfassung

Die Wissenserfassungsphase beinhaltet die Identifikation des Einsatzbereiches, die Zielsetzung und Annahme des betreffenden Projektes. Generell sind anfängliche Perspektiven des Einsatzbereiches sowie Zielsetzungen und Annahmen von dem Projekt Kunde oder von einer anfänglichen definierten Mission Bedürfnisaufstellung verfügbar. Zur Erreichung einer endgültigen Identifizierung des Einsatzbereiches der Zielsetzung und der Annahme, die von den Projektbeteiligten unterstützt werden, kommt eine Kombination von Wissenserhebung und Systemtechniken zum Einsatz. Zum Beispiel werden Verfahren wie Brainstorming und Mind-Mapping zur Erhaltung einer befriedigenden Perspektive auf die involvierten Aspekte, verwendet. Funktionelle Strukturgliederungen, funktionelle Ablaufdiagramme, Organigramme und N2-Tabellen sind nur eine kleine Auswahl von einer Vielzahl von System-Engineering-Techniken, für die Struktur-

rierung der beteiligten Aspekte. Am Ende des Prozesses werden der Einsatzbereich, die Zielsetzungen und die Annahmen mit den Projektbeteiligten abgestimmt. Auf dieser Grundlage werden die erforderlichen Wissensquellen identifiziert. Diese können explizites und implizites Wissen enthalten. Anschließend können Wissenserwerbstechniken angewendet werden, um das Wissen zu erfassen. Eine breite Palette von Wissenserwerbstechniken steht dabei zur Verfügung wie Interviews, Prozess-Mapping, Laddering, Zustandsdiagramm-Mapping, Concept Mapping, Beobachtung und Kommentierung, diese sind nur einige Beispiele dafür. (vgl. [CVT10])

Normierung

Das Ziel der Phase Normierung ist die Erreichung eines ausreichend hohen Qualitätsniveau der Wissensinhalte, damit in den anschließenden Phasen wie Organisation, Modellierung und Analyse das Wissen reibungslos genutzt werden kann. Um dies zu erreichen, werden drei verschiedene Aktivitäten durchgeführt. Zuerst wird das Wissen auf die folgenden anwendbaren Qualitätskriterien geprüft wie Rückverfolgbarkeit, Besitz, Genauigkeit und Zuverlässigkeit.

- **Rückverfolgbarkeit:** Das Wissen muss über die verschiedenen Iterationen hinweg zu seiner Quelle zurückverfolgt werden können. Dies ermöglicht die Erfassung der zugehörigen Konstruktionsschritte, während es Einblicke in den gesamten Wissenslebenszyklus bietet. Davon profitiert die Konstruktion besonders in der frühen Phase, in der eine hohe Anzahl an Iterationen durchgeführt werden. Konstruktionsentscheidungen können zu den speziellen Wissensinstanzen verfolgt werden, dadurch werden schnelleres Lernen und mehr Einblick in den gesamten Konstruktionsprozess geschaffen.
- **Besitz:** Das Wissen muss zu seinem Eigentümer verknüpft werden. Dieser Eigentümer übernimmt die Verantwortung für die Genauigkeit und Zuverlässigkeit des Wissens. Dies ist eine Voraussetzung für ein effizientes Wissensmanagement, vor allem, wenn implizites Wissen beteiligt ist. Dies kann zu erheblichen Vorteilen in den späteren Phasen führen (Organisation, Modellierung und Analyse), falls weitere Erklärungen zu dem Wissen von Nöten sind, kann der Besitzer leicht kontaktiert werden.
- **Genauigkeit:** Der Punkt Genauigkeit stellt die Frage, ob das erfasste Wissen genau genug für die anschließende Verwendung in der Modellierung und Analyse ist. Die Modellierung und Analyse von Parametern stellt Anforderungen an die Genauigkeit, erhebliche Schwankungen oder Unsicherheiten können die Modellierungs- und Analyselösung verhindern.
- **Zuverlässigkeit:** Dieser Punkt wirft die Frage auf wie zuverlässig das erfasste Wissen ist. Besonders während der frühen Konstruktionsphase, Parameter werden abgeschätzt oder sie basieren auf nicht geprüften Annahmen. Natürlich ist es dem Konstrukteur lieber, überprüfte und bestätigte Parameter heranzuziehen, jedoch ist das in der frühen Konstruktionsphase nicht immer möglich. Wenn die Parameter hohen Schwankungen unterliegen, sollten diese Unsicherheiten in den nachfolgenden Schritten berücksichtigt werden. In der Tat kann die Schwankung so hoch sein, dass der Parameter zu unklar ist für die weitere Modellierung und Analyse. Dieser Ansatz bietet den Vorteil, den gesamten Prozess auf die Ressource Wissen auf „würdig“ zu begrenzen und die Unsicherheiten in einem Bericht zu nehmen, welcher zu einer zuverlässigeren Analyseabschätzung führt.

Zum Abschluss der Normierungsphase werden die Aktivitäten, die innerhalb dieser Phase durchgeführt wurden, dokumentiert. Die Verwendung eines speziellen Wissensmanagements, erleichtert das Erstellen der Dokumentation. Solch ein System kann von der Organisation des Wissens in einer geeigneten Struktur unterstützt werden, welche in der nächsten KNOMAD-Phase beschrieben wird. (vgl. [CVT10])

Organisation

Die Organisation des Wissens ist ein wesentlicher Schritt in Richtung Wissensnutzung in der Modellierung und Analyse. Eine Wissensstruktur muss, sofern die Projektbeteiligten von den verschiedenen Disziplinen es erlauben, automatisch auf das notwendige Wissen zugreifen, damit es in der Modellierung und der Analyse verwendet werden kann. Für diese Anforderung könnte eine Ontologie verwendet werden. Eine Ontologie bildet eine formalisierte Wissensstruktur (durch den Einsatz von Begriffsdefinitionen und Zusammenhängen), welche die Instanziierung des erfassten Wissens innerhalb der Struktur unterstützt. Die KNOMAD- Organisationsphase muss Techniken, Tools und Methodiken, für die Unterstützung der Ontologieerstellung bieten. (vgl. [CVT10])

Modellierung

Der nächste Schritt in der KNOMAD-Methodik ist die Modellierung des Produktes und Prozesses. Dazu verwendet KNOMAD den Multi-Model Generator (MMG) Ansatz, der auch in der Design and Engineering Engine eingesetzt wird. Für die Unterstützung des MMG-Ansatzes müssen die verschiedenen Disziplinen, für die ein Produkt oder ein Prozess modelliert werden sollte, gut verstanden werden. Die verschiedenen Projektbeteiligten aus den verschiedenen Disziplinen in dem Konstruktionsprozess, haben unterschiedliche Ansichten über dieselben Bauteilkomponenten. Die spezifischen Anforderungen einer Disziplin kann dazu verwendet werden, ein disziplinspezifisches Modell zu erzeugen, welche für die unterschiedlichen Bedürfnisse der Projektbeteiligten bestimmt sind. (vgl. [CVT10])

Analyse

Analog zum DEE-Ansatz beschäftigt die Analysephase von KNOMAD die Reportdateien, die vom MMG generiert wurden. Die Reportdateien enthalten die Informationen von dem Produkt- und Prozessmodell, die für die detaillierten Analysemodule benötigt werden. Wie zuvor in der DEE-Methodik erwähnt, berechnen die Analysewerkzeuge die Konstruktionsauswirkungen für die jeweilige Disziplin. Analysen können für die unterschiedlichsten Anliegen durchgeführt werden. Kostenabschätzung ist eine primäre Bestrebung sowie die Analyse für die Herstellbarkeit und die Abschätzung der Terminierung. (vgl. [CVT10])

Bereitstellung

Die Bereitstellungsphase von KNOMAD beginnt mit der Überprüfung der Lösung im Vergleich zu den Anforderungen, die zu Beginn des Konstruktionsprozesses festgelegt wurden. Wird diese Überprüfung von der Lösung bestanden, können die detaillierten Analyseergebnisse bereitgestellt und die Auswirkungen auf die Ressourcen ausgewertet werden. Beispielsweise können Prozesszeiten und –kosten, die in der Analysephase berechnet wurden, direkt der nachfolgen-

den Analyse wie der „Diskreten Ergebnissimulation“ bereitgestellt werden, damit die Auswirkungen auf die Ressourcen von der Konstruktion erforscht werden können. Bei der Verwendung von „Diskreten Ergebnissimulationen“ kann die logistische Auswirkung einer Konstruktion auf die Produktionshalle sowie den Grundriss dafür, bis ins Detail geplant werden. Des Weiteren ist eine gute ontologische Repräsentation wie in der Organisationsphase erwähnt, wesentlich mit inkludierten Ressourceauswirkungen entweder als Klasse für sich selbst mit Beziehungen zu den Produkten und Prozessen oder als eine Eigenschaft der bereits existierenden Klassen. Diese Eigenschaft begleitet die instanziierten Objekte in die darauffolgenden KNOMAD-Phasen wie Modellierung und Analyse, damit die finanziellen Auswirkungen, die daraus resultieren, für jedes Objekt bestimmt werden können. (vgl. [CVT10])

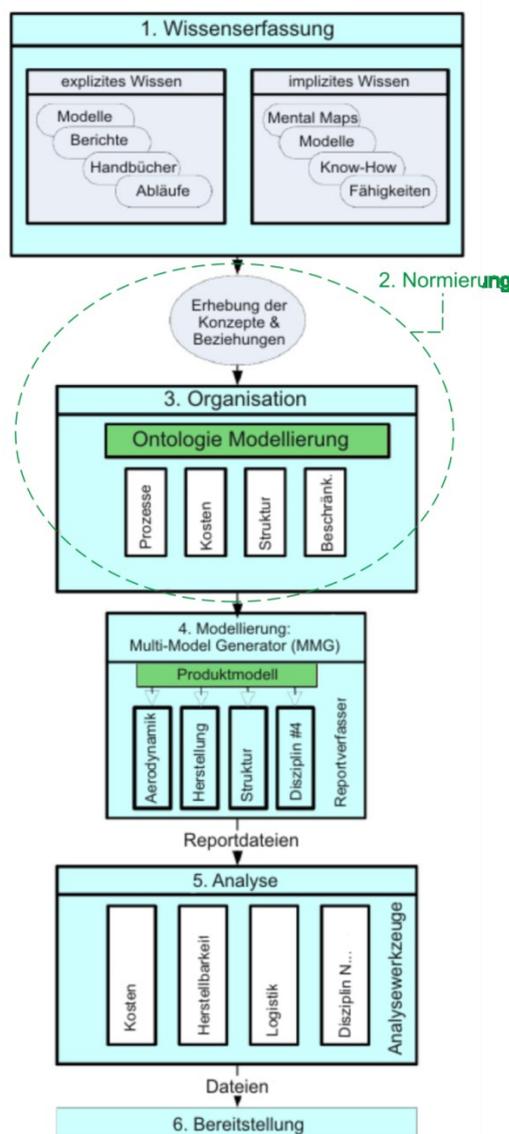


Abbildung 3-13: KNOMAD Phasen (angelehnt an [CVT10])

Im fünften Kapitel erfolgen die Ergebnisse einer Internetrecherche, welche eine aktuelle Anwendung der KBE-Methodiken untersuchte. Des Weiteren erfolgen eine Bewertung und ein Vergleich der KBE-Methodiken im sechsten Kapitel.

3.2 MDE (Model Driven Engineering)

Bei der Literaturrecherche über KBE-Systeme trat das Model Driven Engineering, kurz MDE, in Augenschein. Da bei der Erstellung von KBE-Anwendungen die Softwareentwicklung einen Kernpunkt darstellt, erfolgt in diesem Teil der Arbeit eine Beschreibung über MDE. Das MDE wird in der Softwareentwicklung eingesetzt, wobei aus einem erstellten Modell automatisch ein Code generiert wird. Die Abkürzung MDSD steht für Model Driven Software Development und kann als Synonym von MDE betrachtet werden. Die folgende Erläuterung soll den Unterschied zwischen der modellbasierten und der modellgetriebenen Softwareentwicklung aufzeigen.

3.2.1 Modellbasierte Softwareentwicklung

Bei der Softwareentwicklung bestehen die Anwendungen aus einer Reihe von Konstruktionsparadigmen, wobei die Art der Implementierung einen direkten Einfluss auf Qualitätsmerkmale wie Wiederverwendbarkeit und Wartbarkeit, hat. Diese Muster sind auf dem Niveau des Quellcodes nicht mehr eindeutig erkennbar, da sie wegen des geringen Abstraktionsgrades durch viele Details verschachtelt sind. Deshalb versucht man die interne Struktur in Modellen zu dokumentieren. Dabei besteht jedoch das Problem, dass zwischen Modell und Code nur ein gedanklicher Zusammenhang existiert, der darauffolgend von einem Entwickler manuell codiert werden muss. Das Modell liefert dadurch nur eine Art von Dokumentation. Diese Vorgehensweise bezeichnet man modellbasiert. Da sich der Nutzen der Modelle für die Projektbeteiligten nicht eindeutig erkennen lässt, werden sie aus Zeitmangel sehr oft nicht erstellt bzw. werden nicht aktualisiert und dadurch unbrauchbar. (vgl. [WEI06], S.6)

3.2.2 Modellgetriebene Softwareentwicklung

In der modellgetriebenen Softwareentwicklung besitzen die erstellten Modelle denselben Nutzen wie der Quellcode. Mithilfe eines Generators und einer Reihe von Transformationsvorschriften erfolgt die automatische Erstellung des Quellcodes aus dem Modell. Diese Vorgehensweise wird modellgetriebene Softwareentwicklung oder MDSD bezeichnet. Hier werden die Modelle abstrakt dargestellt, da die Implementierungsdetails weggelassen werden. Die Modelle sind formal, weil sie sich an einer Domain Specific Language kurz DSL anlehnen. Die Domäne bei der modellgetriebenen Softwareentwicklung stellt einen abgegrenzten Problemraum dar, in der die Modelle im Kontext stehen. Für die Konzepterstellung einer Domäne wird die DSL herangezogen. Eine der Hauptkomponenten der DSL ist das Metamodell der Domäne, welches in dem nachfolgenden Punkt beschrieben wird. Zur Vereinfachung der Generierung und damit der erstellte Quellcode nicht auf einer „rohen“ Programmiersprache aufgebaut werden muss, werden domänenspezifische Plattformen verwendet. Eine Plattform besteht aus Komponenten und Frameworks. In der Abbildung 3-14 werden die Zusammenhänge, die bei der Entwicklung einer MDSD auftreten, dargestellt. Der Quellcode eines beliebigen Mitglieds einer Systemfamilie liefert auf den ersten Blick den Eindruck, dass es sich dabei um ein Unikat handelt. Wird der Code jedoch umstrukturiert und kategorisiert, entstehen dadurch drei Bereiche. Der erste Bereich beinhaltet den generischen Codeanteil, der bei allen Anwendungen

einer Systemfamilie gleich ist. Im zweiten Bereich besitzt jedes „Familienmitglied“ ein schematisch wiederholendes und bestimmtes Designmuster. Der dritte und letzte Bereich besteht aus einem individuellen Codeanteil, welcher bei jeder Anwendung unterschiedlich ist. Aufgrund der Aufgliederung des Quellcodes in die drei Bereiche erscheint es sinnvoll, den gemeinsamen generischen Codeanteil wiederzuverwenden. Der schematische Codeanteil wird hauptsächlich durch eine „Copy & Paste“ Arbeitsweise mit einer anschließenden Nachbearbeitung umgesetzt. Ein Generator ermöglicht eine Automatisierung dieser Arbeitsschritte, wobei er drei Dinge benötigt. Zum einen ein Modell der Applikation, zum anderen eine DSL an der sich die Modelle orientieren können und als drittes eine Reihe von Transformationsvorschriften für die Ausgabensteuerung. Ein Entwickler muss den individuellen Teil des Codes manuell hinzufügen. (vgl. [WEI06], S.7 f)

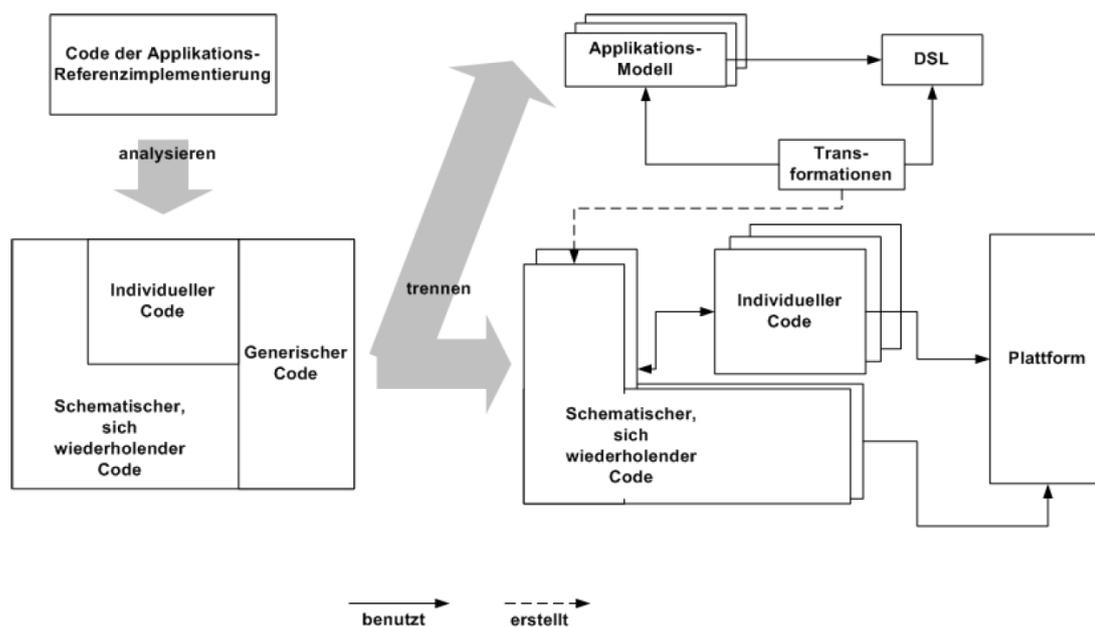


Abbildung 3-14: Modellgetriebene Softwareentwicklung ([WEI06], S.8)

3.2.2.1 Modellierung

Für die Modellierung werden eine Domäne, ein Metamodell und eine domänenspezifische Sprache benötigt, die Definition dieser Begriffe erfolgt darunter.

Domäne

„Eine Domäne stellt ein begrenztes Wissensgebiet dar. Sie spiegelt den zu bearbeitenden Problemraum wieder und ist somit der Startpunkt der Modellierung. Bevor ein Metamodell bzw. eine DSL zur Beschreibung der Konzepte einer Domäne entworfen werden kann, muss diese abgegrenzt sein. Diesen Vorgang nennt man Domain Scoping.“ ([WEI06], S.10 f)

Metamodell

„Ein Metamodell formalisiert die Strukturen einer Domäne und definiert somit jene Konstrukte, die später im Rahmen der Anwendungsmodelle zum Einsatz

kommen. Es bildet daher die Grundlage für eine Automatisierung im Rahmen der Entwicklung.“ ([WEI06] ; S.11)

Domänenspezifische Sprache

„Eine Domänenspezifische Sprache oder Domain Specific Language, kurz als DSL bezeichnet, benötigt neben einem Metamodell noch eine Definition der (dynamischen) Semantik. Mit anderen Worten, der Modellierer muss die Konstrukte der Sprache verstehen. Dazu gibt es zwei Möglichkeiten: Entweder ist die Bedeutung eines Konstruktes intuitiv verständlich oder es existiert eine natürliche sprachliche Beschreibung.“ ([WEI06] , S.12)

3.2.2.2 Ziele der modellgetriebenen Softwareentwicklung

In der folgenden Aufzählung werden die Ziele, die mit MDSD laut dem Autor (vgl. [WEI06] , S.9 f) erreicht werden sollen, angeführt.

- **Steigerung der Entwicklungsgeschwindigkeit:**
Aufgrund der automatischen Erstellung von sich wiederholenden Codeanteilen, kann ein Implementierungsrahmen oder lauffähiger Code erstellt werden.
- **Steigerung der Softwarequalität:**
Die automatisierten Transformationen bewirken eine gleichförmige Implementierung einer zuvor festgelegten Architektur. Dadurch entsteht keine Abhängigkeit zwischen den Fähigkeiten des Entwicklers und der Implementierungsqualität.
- **Zentrale Fehlerbehebung:**
Die im schematischen Codeanteil auftretenden Fehler können direkt bei den Transformationsvorschriften behoben werden.
- **Steigerung der Wiederverwendung:**
Die Architektur, die DSL und die Transformationsvorschriften werden nach der einmaligen Erstellung ständig verbessert.
- **Handhabbarkeit von Komplexität durch Abstraktion:**
Die problemspezifizierten Modellierungssprachen stellen eine deutlich höheren Abstraktionsgrad als die Programmiersprachen, dar.

3.3 Knowledge-based Systems – Expertensysteme

Im Kapitel Grundlagen wurde das Wissensmanagement erläutert, bekannte Wissensmanagementwerkzeuge sind die Expertensysteme, welche in diesem Punkt genauer betrachtet werden. Des Weiteren erfolgt die Beschreibung eines weit verbreiteten speziellen Expertensystems, welches den Namen Common-KADS trägt. (vgl. [COD12])

Wenn für die Lösung eines Problems Expertenwissen benötigt wird, sich dieses Wissen für den Computer aufbereiten lässt und sich auf diese Weise die Lösung von dem Computer bestimmt werden kann, kommen Expertensysteme zur Anwendung. Die Expertensysteme werden für ganz spezielle Einsatzbereiche verwendet, dabei soll der menschliche Experte ersetzt werden, jedoch soll er auch bei alltäglichen Routineaufgaben unterstützt werden. Der Anlass für die Entwicklung solcher Expertensysteme waren die hohen Kosten eines Experten sowie die lange Dauer für die Ausbildung einer Person zum Experten. Das Expertenwissen entsteht meist erst über die Jahre hinweg, das auch als Erfahrungswissen bezeichnet wird. Des Weiteren wird Expertenwissen oft zeitgleich an verschiedenen Orten benötigt. Der Benutzer von Expertensystemen muss selbst kein Experte des jeweiligen Fachbereiches sein. (vgl. [ROS12] , S.5 f)

Aufgabenbereiche von Expertensystemen

Die Aufgabenbereiche der Expertensysteme werden in drei Bereiche unterteilt. Einerseits in die Klassifikation, welche auch Analyse genannt wird, andererseits die Konstruktion welche auch als Synthese bezeichnet wird und den dritten Bereich die Simulation. In der Klassifikation wird eine Lösung aus einer Menge von bereits vorhandenen Lösungsmöglichkeiten aufgrund der Eingabe ausgewählt. Zum Beispiel bei der Erstellung von Diagnosen wie etwa die Krankendiagnose, basierend auf ihrer Symptome im medizinischen Bereich oder in der Steuerungstechnik. Das Ansteuern eines Reglers aufgrund der gemessenen Daten des Sensors. In dem Aufgabenbereich Simulation wird aus dem Ausgangszustand und der Eingabe der Folgezustand festgelegt. Dabei können die Auswirkungen simuliert werden, die durch Aktivitäten in einem bestimmten Zustand hervorgerufen werden. In dem Teilgebiet Konstruktion werden neue Lösungen erstellt, die nicht auf bereits existierenden Lösungen basieren. Als Beispiel bei der Konfiguration und der Planung von Aktionsreihenfolgen zur Erfüllung bestimmter Ziele sowie die Produktionsplanung in einem Industriebetrieb. (vgl. [ROS12] , S.6)

Architektur von Expertensystemen

In den Expertensystemen erfolgt eine Unterteilung nach dem vorhandenen Expertenwissen und den Problemlösungsstrategien, welche sich in der grundlegenden Architektur von Expertensystemen widerspiegelt. Sie besteht aus den beiden Hauptkomponenten Wissensbasis und Steuersystem, wie in der Abbildung 3-15 dargestellt. (vgl. [ROS12] , S.6 f)

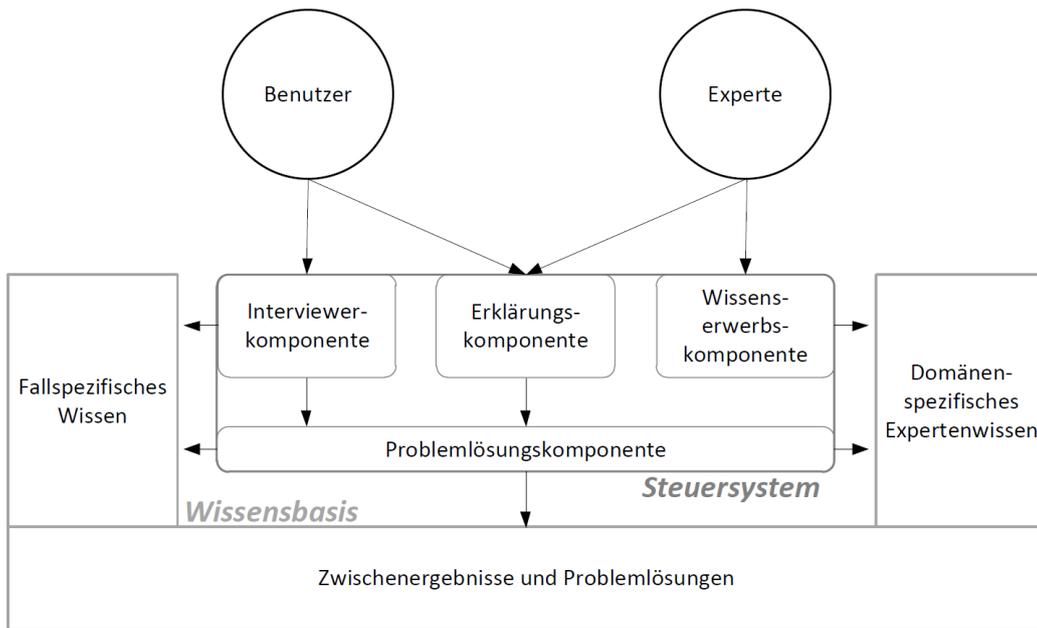


Abbildung 3-15: Architektur eines Expertensystems ([ROS12] , S.7)

3.3.1 CommonKADS-Methodik

CommonKADS ist die führende Methodik zur Unterstützung von Knowledge-based Systems bzw. Expertensystemen. Sie wurde von vielen Unternehmen und Universitäten im Rahmen des europäischen Forschungsprogramms ESPRIT entwickelt und getestet. CommonKADS bildet den europäischen Standard für die Knowledge-based Systems-Entwicklung und wurde in europäischen, amerikanischen und japanischen Unternehmen eingesetzt. Sie liefert ebenso eine Methode für eine detaillierte Analyse der Wissensaufgaben und der Prozesse. CommonKADS stammt aus dem KADS-I-Projekt, KADS steht für Knowledge Acquisition Design System und baut die Knowledge-based Systems im Grunde durch Modellierungsaktivitäten auf. Eines der wichtigsten Merkmale von KADS ist, unabhängig von der Implementierung, die Erstellung eines vollständigen Wissensmodelles. Im Winter des Jahres 1990 begann die Entwicklung einer neuen Methodik die wirtschaftlich machbar war und den gesamten Knowledge-based Systems-Lebenszyklus abdeckt. Das Resultat war die Methodik CommonKADS. Die Abbildung 3-16 zeigt die Entwicklung der Knowledge-based Systems Methodiken. (vgl. [COD12])

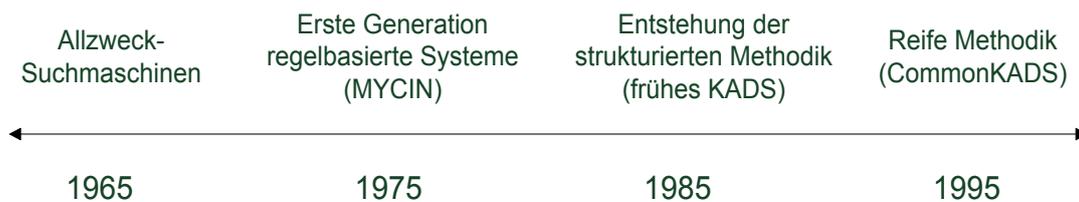


Abbildung 3-16: Entwicklung der KBS (angelehnt an [COD12])

3.3.1.1 CommonKADS-Modellreihenfolge

Die CommonKADS-Methodik beinhaltet eine Modellreihenfolge, die den Kern des wissensbasierten Systems bildet. Die Modelle werden in drei Gruppen untergliedert, da es im Wesentlichen drei Arten von Fragen gibt, die beantwortet werden müssen. (vgl. [GAA00], S.17 f)

1. Warum?

Warum ist ein Wissenssystem eine mögliche Hilfe? Für welches Problem? Welche Vorteile, Kosten und organisatorischen Auswirkungen hat es? Das Verstehen des organisatorischen Umfeldes ist in diesem Punkt das Wichtigste.

2. Was?

Was für eine Art und Struktur von Wissen ist involviert? Was für eine Art und Struktur der dazugehörigen Kommunikation herrscht vor? Die Hauptaufgabe dieses Punktes ist die konzeptionelle Beschreibung des Wissens in einer angewendeten Aufgabe.

3. Wie?

Wie muss das Wissen in einem Computersystem implementiert werden? Wie sehen die Software-Architektur und die rechnerischen Mechanismen aus? Dieser Punkt betrachtet schwerpunktmäßig die technischen Aspekte der Computerrealisierung.

Die Modellreihenfolge von CommonKADS beinhaltet das Organisations-, das Aufgaben-, das Agenten-, das Wissens-, das Kommunikations- und das Konstruktionsmodell, deren nähere Erklärung erfolgt darunter und die Abbildung 3-17 zeigt die Anordnung der einzelnen Modelle.

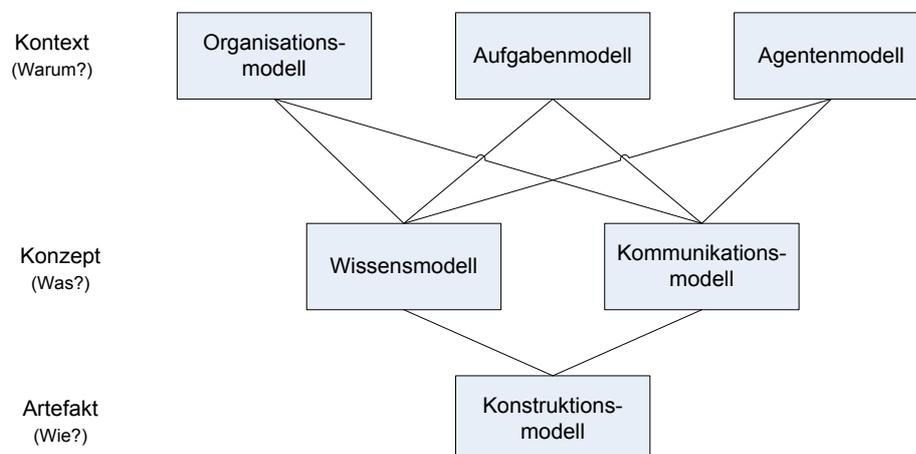


Abbildung 3-17: Modellreihenfolge (angelehnt an [GAA00], S.18)

Organisationsmodell

Das Organisationsmodell unterstützt die Analyse der wichtigsten Funktionen einer Organisation, damit Probleme und Möglichkeiten für Wissenssystem entdeckt werden, die für die Begründung der Machbarkeit und die Bewertung der Auswirkungen auf die Organisation der beabsichtigten Wissensaktionen herangezogen werden können. (vgl. [GAA00], S. 18)

Aufgabenmodell

Aufgaben sind die relevanten Unterpunkte eines Geschäftsprozesses. Das Aufgabenmodell analysiert den globalen Aufgabenentwurf, ihre Ein- und Ausgänge, Voraussetzungen und Leistungskriterien sowie die benötigten Ressourcen und Kompetenzen. (vgl. [GAA00] , S.18)

Agentenmodell

Agenten sind die Vollzieher einer Aufgabe. Ein Agent kann eine Person, ein Informationssystem oder jede andere Instanz sein, die in der Lage ist, eine Aufgabe auszuführen. Die Agentenmodelle beschreiben die Eigenschaften der Agenten, insbesondere ihrer Kompetenzen, ihrer Autorität und ihrer Einschränkungen zu handeln. (vgl. [GAA00] , S.18 f)

Wissensmodell

Der Zweck des Wissensmodelles ist die nähere Erläuterung der Wissensart und Wissensstruktur bei der Durchführung einer Aufgabe. Es bietet eine implementierungsunabhängige Beschreibung der unterschiedlichen Wissenskomponenten einer Problemlösung, die für einen Menschen verständlich sind. Dies macht das Wissensmodell zu einem wichtigen Instrument für die Kommunikation zwischen den Experten und den Anwendern über die Problemlösungsaspekte eines wissensbasierten Systems, während der Entwicklungs- und Systemausführung. (vgl. [GAA00] , S.19)

Kommunikationsmodell

Da mehrere Agenten an einer Aufgabe beteiligt sein können, ist es wichtig, die kommunikativen Transaktionen zwischen den beteiligten Agenten zu modellieren. Dies erfolgt durch das Kommunikationsmodell in einer konzeptionellen und implementierungsunabhängigen Weise wie beim Wissensmodell. (vgl. [GAA00] , S.19)

Konstruktionsmodell

Die zuvor erwähnten CommonKADS-Modelle können zusammen als ein rechtswirksames Pflichtenheft für das Wissenssystem, aufgeteilt in unterschiedlichen Aspekten, gesehen werden. Basierend auf diesen Anforderungen gibt das Konstruktionsmodell die technischen Systemspezifikationen in Bezug auf die Architektur, der Implementierungsplattform, der Softwaremodule der repräsentativen Konstrukte und der numerischen Mechanismen an, die für die Implementierung der zuvor im Wissens- und Kommunikationsmodell festgelegten Funktionen benötigt werden. (vgl. [GAA00] , S.19)

3.3.1.2 CommonKADS-Prozesspersonen

Es ist sehr wichtig die unterschiedlichen Personen, die im Wissensmanagementprozess und im Ingenieureprozess eine Rolle spielen, zu identifizieren. Die CommonKADS-Methodik unterscheidet nach sechs verschiedenen Prozesspersonen, welche in der Abbildung 3-18 grafisch dargestellt sind. Bei kleineren Projekten kann eine bestimmte Person mehrere Rollen übernehmen. Die Bezeichnungen der einzelnen Prozesspersonen sind der Wissensanbieter/-spezialist, der Wissensingenieur/-analyst, der Wissenssystementwickler, der Wissensanwender, der Projektmanager und der Wissensmanager. (vgl. [GAA00] , S.20)

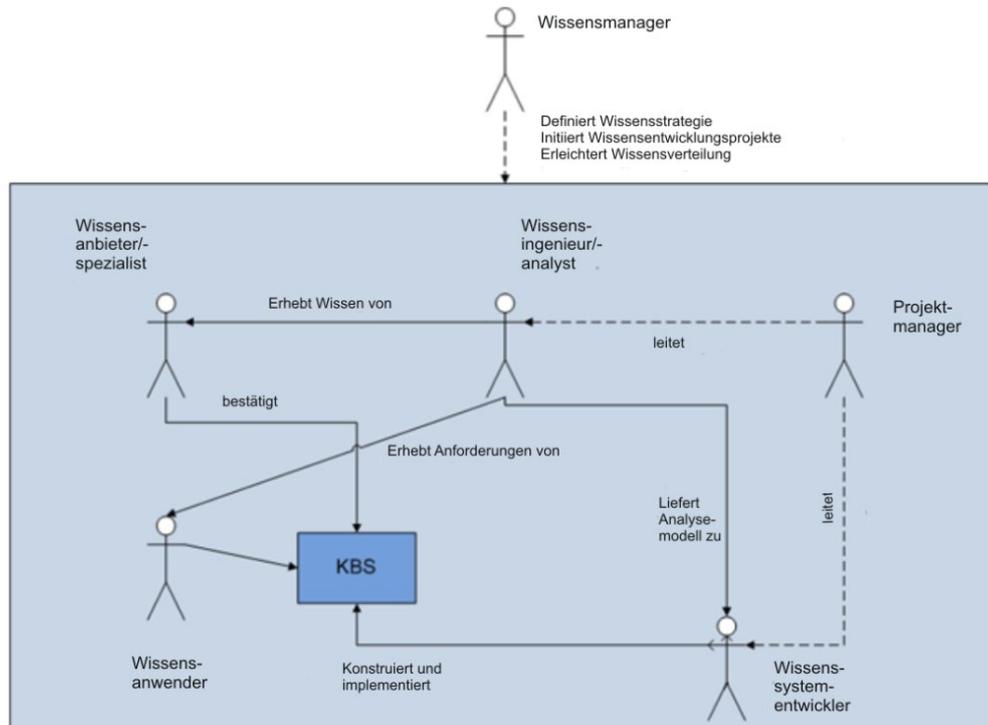


Abbildung 3-18: Prozesspersonen (angelehnt an [GAA00] , S.21)

Wissensanbieter/-spezialist

Eine wichtige Rolle in dem Prozess wird durch den menschlichen „Besitzer“ des Wissens gespielt. Dies ist traditionell ein „Experte“ in der Anwendungsdomäne, es könnte jedoch auch eine andere Person in der Organisation die nicht den „Experten“ Status besitzt die Rolle übernehmen. Eine wichtige Aufgabe für den Wissensingenieur ist es einen „echten“ Experten zu finden. Scheinexperten können sich sehr schädlich auf das Projekt auswirken. (vgl. [GAA00] , S.20)

Wissensingenieur/-analyst

Obwohl streng genommen der „Wissensingenieur“ in allen Phasen des Entwicklungsprozesses beteiligt ist, wird die Bezeichnung i.d.R. für Systemanalysearbeiten vorbehalten. Die Wissensanalyse wurde von Anfang an als der Flaschenhals in der Wissenssystementwicklung wahrgenommen. CommonKADS bietet dem Wissensingenieur eine Reihe von Methoden und Werkzeugen, für die Standardisierung der wissensintensiven Analyse (z.B. Bewertung). (vgl. [GAA00] , S.20)

Wissenssystementwickler

Bei kleinen Projekten wird die Implementierung von der Person übernommen die die Analyse durchführt. Der Wissenssystementwickler ist verantwortlich für die Konstruktion und die Implementation. Er benötigt ein Basiswissen über die Analysemethoden, damit er die Anforderungen die vom Wissensanalysten formuliert wurden versteht. In der Praxis hat sich herausgestellt, dass bei der Wissenssystementwicklung die wichtigsten wissensbezogenen Probleme durch den Wissensanalysten gelöst wurden. Daher benötigt er dieselben Fähigkeiten wie ein Softwareentwickler. (vgl. [GAA00] , S.20 f)

Wissensanwender

Die Wissensanwender benutzen direkt oder indirekt ein wissensbasiertes System. Die Einbeziehung des Wissensanwenders von Anfang an, ist hier deutlich mehr von Bedeutung als bei gewöhnlichen Softwareentwicklungsprojekten. Die Automatisierung der wissensintensiven Aufgaben beeinflusst unweigerlich die Arbeit der Personen die damit involviert sind. Für die Konstruktion und Implementation ist es wichtig, sicherzustellen, dass sie mit ihrer eigenen Schnittstellenrepräsentation mit dem System interagieren. Der Wissensingenieur muss auch in der Lage sein die Analyseergebnisse dem potentiellen Wissensanwender zu präsentieren. Dies erfordert besondere Aufmerksamkeit. Eine der Erfolgsfaktoren von CommonKADS ist, dass die Wissensanalyse von dem Wissensanwender verstanden wurde. (vgl. [GAA00] , S.21 f)

Projektmanager

Der Wissensprojektmanager ist zuständig für die Leitung der Projektentwicklung des Wissenssystems. Das typische Projekt ist klein bis mittelgroß mit vier bis sechs Projektbeteiligten. Der Projektleiter profitiert von dem strukturierten Ansatz von CommonKADS. Die Modellreihenfolge bietet ein leistungsstarkes und flexibles Instrument zur Projektplanung und -steuerung. Das Hauptrisiko für den Projektmanager liegt in der schwer fassbaren Art der wissensbezogenen Probleme. Deshalb sind die Anforderungen an die Überwachung des Projektes während der Laufzeit von zentraler Bedeutung. (vgl. [GAA00] , S.22)

Wissensmanager

Der Wissensmanager ist nicht direkt in das Wissensentwicklungsprojekt involviert. Er formuliert eine Wissensstrategie auf wirtschaftlicher Ebene und initiiert Wissensentwicklungs- und Wissensverteilungsaktivitäten. (vgl. [GAA00] , S.22)

3.4 Matrixbasierte Methoden

In dem Abschnitt Produktentwicklung wurden Methoden erwähnt, die eine systematische Vorgehensweise bei der Produktentwicklung ermöglichen, ein Behelfsinstrument sind die matrixbasierten Methoden, deren Beschreibung darunter erfolgt.

Matrixbasierte Methoden werden eingesetzt, um die Komplexität von technischen Systemen begreiflich zu machen und überschaubar darzustellen. Die Komplexität der Systeme entsteht aufgrund der großen Anzahl von Elementen (d.h., den Bestandteilen von Systemen, wie Komponenten und Funktionen) und deren Beziehungen zueinander. Die matrixbasierten Methoden für den Umgang mit komplexen Systemen haben ihren Ursprung neben der Graphentheorie vor allem in der Entwicklung der Design Structure Matrix. (vgl. [RS12] , S.681 ff)

3.4.1 DSM (Design Structure Matrix)

Im Jahre 1981 erschien zum ersten Mal in einer Publikation von Don Steward das Verfahren der Design Structure Matrix. Eine Design Structure Matrix, abgekürzt DSM, ist eine quadratische Matrix, d.h., eine Matrix mit einer gleichen Anzahl von Zeilen und Spalten. Sie bietet eine systematische Beziehungszuordnung zwischen den Elementen. Diese Elemente können physische Produktkomponenten, Leistungsmerkmale, Konstruktionsanforderungen oder Prozessaufgaben sein. Die Elementbezeichnungen werden auf der Ordinate sowie auf der Abszisse in derselben Reihenfolge angeführt. Die Diagonalzellen der Matrix werden meist nicht berücksichtigt und bleiben deshalb leer, oder werden geschwärzt. Die markierte Zelle bestimmt die Abhängigkeit zwischen zwei Elementen. Die Leserichtung der Matrix nach dem europäischen System wird dadurch bestimmt, dass die Zeile die Spalte beeinflusst (beim amerikanischen System beeinflusst die Spalte die Zeile). Wenn sich zwei Elemente gegenseitig (bidirektional) beeinflussen, werden die Zellen der Elemente oberhalb und unterhalb der Diagonale markiert. Die markierten Zellen zwischen den Elementen geben nur an, dass die Elemente in Abhängigkeit stehen und in welcher Richtung sie sich beeinflussen. Mehr Information über die Beziehungen ist in der numerischen DSM enthalten. Dabei wird an Stelle der Markierung ein Faktor gesetzt, welcher die quantitative Abhängigkeit zwischen zwei Elementen bestimmt. Die DSM kann nur die Beziehungen von Elementen innerhalb einer Domäne abbilden. Die Abbildung 3-19 zeigt an der linken Seite die DSM und an der rechten Seite ein Flussdiagramm, welches aus der DSM abgeleitet werden kann. (vgl. [MAU07] , S.54 f)

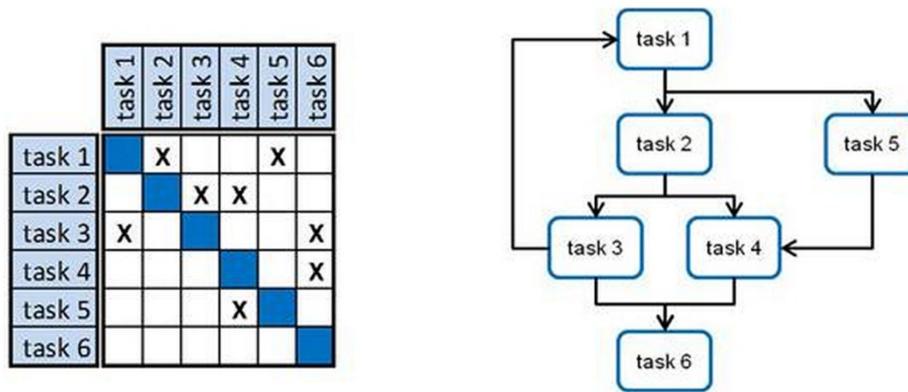


Abbildung 3-19: Design Structure Matrix [DSM12]

3.4.1.1 DSM Typen

Die DSM kann für vier unterschiedliche Bereiche angewendet werden und wird deshalb in vier verschiedene DSM-Typen untergliedert: (vgl. [DSM12])

- **Komponentenbasierte DSM (Produkt):**
Eine komponentenbasierte DSM dokumentiert die Abhängigkeiten zwischen den Elementen in einer komplexen Systemarchitektur. Sie findet Anwendung in der Entwicklung und Konstruktion.
- **Teambasierte DSM (Organisation):**
Die teambasierte DSM beinhaltet die organisatorische Analyse und den Informationsfluss zwischen den Organisationseinheiten. Einzelpersonen und Gruppen an einem Projekt sind die Elemente die analysiert werden. Sie wird für die Organisationsgestaltung, Schnittstellenmanagement und Teamintegration verwendet.
- **Aktivitätenbasierte DSM (Prozess):**
Die aktivitätenbasierte DSM betrachtet eine Reihenfolge von Aufgaben innerhalb eines Prozesses. Diese Aufgaben müssen aufeinander abgestimmt sein, damit das Ziel des gesamten Prozesses erreicht wird. Sie wird für die Prozessverbesserung und Projektplanung angewendet.
- **Parameterbasierte DSM (unteres Prozessniveau):**
Eine parameterbasierte DSM analysiert die Beziehungen des Konstruktionsprozesses auf der Ebene der Parameter. Sie wird auf einem niedrigeren Niveau der Aktivitätensequenzierung und der Prozesskonstruktion sowie für die Sequenzierung der Konstruktionsentscheidungen verwendet.

3.4.1.2 DSM Analysealgorithmen

Die DSM-Analysealgorithmen konzentrieren sich auf die Charakterisierung der gesamten Matrixstruktur. Sie basieren auf der Neuordnung der Elemente in der Matrix, um strukturelle Merkmale hervorzuheben. Wenn umfangreiche Systeme untersucht werden, ist eine Softwareunterstützung erforderlich, da bei einer großen Anzahl an Elementen und Abhängigkeiten der Rechenaufwand zu groß wird. Eine Beschreibung der bekanntesten Analysealgorithmen für die DSM erfolgt in den nächsten Seiten. (vgl. [MAU07] , S.130)

Partitioning

Das Partitioning beschreibt die Umordnung einer DSM (Zeilen und Spalten), mit dem Ziel der Anordnung aller bestehenden Abhängigkeiten an einer Seite der Diagonale. Wenn eine solche Ausrichtung gefunden werden kann, sind keine Rückkoppelungsschleifen in der Struktur vorhanden. Jedoch kann die Menge und die Entstehung der Schleifen nicht weiter spezifiziert werden (es soll dabei erwähnt werden, dass stark verbundene Elemente innerhalb der Struktur mit Partitioning spezifiziert werden können). In Prozess- oder Aktivitätensnetzwerken zeigt die Anwendung des Partitioning eine entsprechende Reihenfolge von Prozessschrittausführung. Die Identifizierung der Abhängigkeitsausrichtungen an einer Seite der Diagonale kennzeichnet einen Arbeitsablauf ohne erforderliche Iterationsschritte. Komplexe Strukturen besitzen oft Rückkoppelungsschleifen, die keine ideale Ausrichtung zulassen. In diesem Fall versucht Partitioning ein Minimum von Abhängigkeitsmarkierungen unter der Diagonale zu positionieren und alle Abhängigkeitsmarkierungen so nahe wie möglich an der Diagonale auszurichten d.h., Rückkoppelungsschleifen besitzen nur eine minimale Länge. Diese Optimierung macht nur Sinn in zeitbasierten Strukturen für die Erreichung einer Zeitersparnis. Andere Bezeichnungen für Partitioning sind Sequencing oder Triangularization. Diese Algorithmen liefern jedoch nicht immer optimale Resultate. Eine potentielle Möglichkeit ist die Kombination aus gemeinsamen deterministischen Algorithmen mit Permutation. Die Abbildung 3-20 zeigt Links die Ausgangsmatrix und Rechts die partitionierte Matrix (vgl. [MAU07], S.133)

	A	B	C	D	E	F	G
A			X				
B			X	X			
C	X				X		
D	X						X
E							
F			X		X		X
G		X	X				

	F	B	D	G	C	A	E
F				X	X		X
B			X		X		
D				X		X	
G		X			X		
C						X	X
A					X		
E							

Abbildung 3-20: Partitioning einer DSM [DSM12]

Banding

Das Banding ist eine Erweiterung des Partitioning und stellt voneinander unabhängige Elemente in einer gesamten Struktur dar. Dies erfolgt durch eine Neuordnung der Elemente und durch die Darstellung von hellen und dunklen Streifen in der Matrix. Wenn die Elemente nebeneinander angeordnet sind und dieselbe Farbe besitzen, können sie unabhängig voneinander bearbeitet werden. Banding bezieht sich auf die unterschiedlichen hierarchischen Ebenen, die durch das Partitioning identifiziert werden, dabei wird jedoch die Erkennung von Rückkoppelungsschleifen ignoriert. Aus diesem Grund ist die nützliche Anwendung von Banding zweifelhaft, da die Analyse wichtige Strukturmerkmale vernachlässigt. Dieser Analysealgorithmus wurde für Prozess- und Aktivitätensnetzwerke konzipiert. Das Ziel dabei ist es, eine geringe Anzahl an Streifen in

einer Matrix wie möglich zu erhalten, was zu einem hohen Grad der Parallelität der Prozessschritte führt. Die Abbildung 3-21 zeigt eine Matrix nach dem Banding. (vgl. [MAU07] , S.130 f)

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

Abbildung 3-21: Banding einer DSM [DSM12]

Clustering

Das Clustering ermöglicht die Identifizierung von spezifischen Teilmengen einer Struktur. Ziel des Clustering ist es, Untergruppen zu erkennen, die viele interne Abhängigkeiten und wenige Abhängigkeiten zu externen Elementen in der Struktur besitzen. Aus diesem Grund bietet sich die komponentenbasierte DSM als eine geeignete Anwendung für Clustering an, welche Erstellung von Produktmodulen ermöglicht. Eine Anpassung eines Elementes an das Cluster verursacht viele Auswirkungen auf die anderen Elemente in dieser Untergruppe, während des Austauschs des gesamten Clusters (Modul) nur wenige Abhängigkeiten bzw. Auswirkungen auf die anderen Produktteile haben. Die Graphentheorie bietet verschiedene Clusteringalgorithmen, einige davon basieren auf der Anwendung des Ähnlichkeitsprinzips. Zusätzlich können Cluster durch das Kriterium der stark verbundenen Elemente identifiziert werden. Für das DSM-Clustering kann der Ansatz des genetischen Algorithmus angewendet werden, beispielsweise unter der Verwendung des Minimum Description Length (MDL)-Verfahrens. Die Abbildung 3-22 zeigt links die Ausgangsmatrix und rechts die Matrix nach dem Clustering. (vgl. [MAU07] , S.131)

	1	2	3	4	5	6	7
1						X	
2	X		X	X			X
3				X			X
4		X	X		X		X
5	X			X		X	
6	X				X		
7		X	X	X			

	1	6	5	4	2	3	7
1		X					
6	X		X				
5	X	X		X			
4			X		X	X	X
2	X			X		X	X
3				X			X
7				X	X	X	

Abbildung 3-22: Clustering einer DSM [DSM12]

Im Punkt 3.6.5 wird ein Softwaretool vorgestellt, welches sich für die Erstellung einer DSM eignet. Des Weiteren wurde in dieser Arbeit eine parameterbasierte Design Structure Matrix für den Fördertyp Rollenförderer erstellt, welche in dem Punkt 4.6.1 genauer beschrieben wird.

3.4.2 DMM (Domain Mapping Matrix)

Die DSM liefert Mittel für die Lösung vieler Probleme in der Produktentwicklung, jedoch besteht bei der DSM nur die Möglichkeit in einer Domäne zu arbeiten. Es entsteht die Komplexität der Produktentwicklung im Zusammenspiel verschiedener Domänen, wie zwischen der Organisationsstruktur und Prozessen, Produkte und Funktionen, Zulieferern und Produktion. Aufgrund dieser Problemstellung wurde die Domain Mapping Matrix kurz DMM eingeführt, die die Möglichkeit bietet zwei Domänen miteinander zu koppeln. Zur Analyse solcher Strukturen wird das DMM-Clustering eingesetzt. Es beruht auf dem Ähnlichkeitsprinzip für unsymmetrische Matrizen, welches auch bei der DSM eingesetzt wird. Die DMM ist jedoch nur auf die Analyse des Clustering beschränkt, andere Analysemethoden können bei der DMM nicht angewandt werden. Die Abbildung 3-23 zeigt ein Beispiel einer DMM in dem die Abhängigkeiten zwischen Personen und Aufgaben dargestellt werden. (vgl. [RS12] , S.686 ff)

	task 1	task 2	task 3	task 4	task 5	task 6
person 1	X				X	
person 2		X				
person 3			X			X
person 4				X		

Abbildung 3-23: Domain Mappig Matrix [DSM12]

3.4.3 MDM (Multiple Domain Matrix)

Die Multiple Domain Matrix, abgekürzt MDM, besteht aus mehreren DSMs und MDMs, damit ein ganzes System modelliert werden kann. Das System beinhaltet mehrere Domänen, von denen jede mehrere Elemente enthält, die durch verschiedene Beziehungstypen miteinander verbunden sind. MDMs erlauben die Analyse einer Systemstruktur über mehrere Domänen. Die Abbildung 3-24 veranschaulicht das Konzept einer MDM. Sie zeigt, dass eine MDM im Grunde eine Erweiterung der DSM und MDM ist, mit detaillierten DSMs entlang ihrer Diagonale und DMMs außerhalb der Diagonale. Sie zeigt auch, wie verschiedene Beziehungstypen mehrere Darstellungen der gesamten MDM erstellen. (vgl. [DSM12])

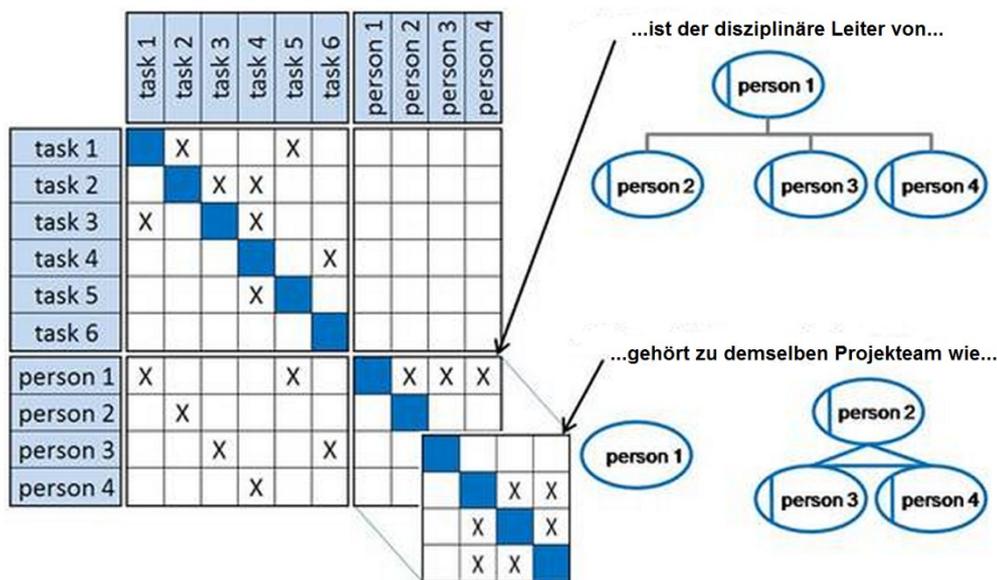


Abbildung 3-24: MDM in Anlehnung an [DSM12]

3.5 Axiomatic Design

Dieser Teil der Arbeit beschreibt das Axiomatic Design, welches im Kapitel Grundlagen unter Vorgehensmodelle in der Produktentwicklung erwähnt wurde.

Das Axiomatic Design wurde 1980 von Nam Pyo Suh am Massachusetts Institut of Technology entwickelt. Dabei soll wie bei der Mathematik, die Produktentwicklung als ein System aufgebaut werden, welches auf Axiomen basiert. Es soll damit eine wissenschaftlichere Herangehensweise bei der Konstruktion technischer Systeme ermöglicht werden. (vgl. [LEY06] , S.107)

Funktionelle Beschreibung

Axiomatic Design bietet eine methodische Vorgehensweise bei der Suche und Zuordnung passender Lösungen für vorherig bestimmte Anforderungen. Dabei erfolgt eine Aufgliederung der Anforderungen und Lösungen auf vier verschiedenen Domänen. Jeder Entwicklungsprozess muss dabei alle vier Domänen durchlaufen. Die Anforderungen werden in der Vorgängerdomäne definiert und die Folgedomäne definiert die korrespondierten Lösungen. Mittels Matrizen erfolgt eine Zuordnung von Anforderungen zu den korrespondierenden Lösungen. Axiome beschreiben die Richtlinien für den Entwicklungsprozess und bilden Bewertungsgrundlagen für die gewählten Entscheidungen. (vgl. [WIK12])

Domänen

Jeder Entwicklungsprozess durchläuft die vier Domänen - Kundendomäne, Funktionsdomäne, physikalische Domäne und Prozessdomäne - deren Anordnung in der Abbildung 3-25 dargestellt wird. Jede Domäne verfügt über unterschiedliche Variablengruppen. Die Kundendomäne beinhaltet eine Ansammlung der gewünschten Merkmale des Produktes (customers needs, CN), welche in die Produktfunktionsanforderungen (functional requirements, FR) transformiert werden müssen. In der darauffolgenden physikalischen Domäne werden die Lösungsmerkmale (design parameters, DP) aufgrund der Funktionsanforderungen festgelegt. Die Prozessdomäne bildet den Abschluss der Reihenfolge, in der die Fertigungsprozesse des Produktes entwickelt und die durch die Prozessvariablen (PV) beschrieben werden. In jeder Domäne müssen Restriktionen (constraints) eingehalten werden. (vgl. [LEY06] , S.108)

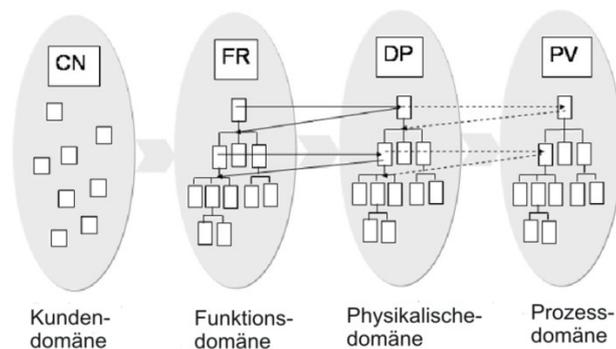


Abbildung 3-25: Domänen des Axiomatic Design (angelehnt an [GB04])

Constraints

Die Constraints auch als Restriktionen bezeichnet, beschränken den Raum der möglichen Lösungen. Dabei erfolgt eine Unterscheidung von zwei Restriktionsarten. Zum einen die Eingangsrestriktionen, welche die Planungsaufgaben bestimmen. Zum anderen die Systemrestriktionen, die die Randbedingungen bilden, in welcher die Lösungen funktionieren müssen. Constraints können unternehmensexterner Natur sein, wie Gesetze, Kundenwünsche oder unternehmensinterner Natur, etwa die Wahl des Unternehmensstandortes. Die Constraints müssen in jeder Phase des Entwurfsprozesses eingehalten werden. (vgl. [LEY06], S.109)

Matrizen

Die Konstruktion kann mithilfe eines Zahlenverhältnisses zwischen den Designparametern und den Funktionsanforderungen bewertet werden. Im Falle einer Parameteranzahlabweichung herrscht keine optimale Konstruktionslösung vor. Es wird nach drei verschiedenen Fällen unterschieden. Der erste Fall tritt ein, wenn $DP < FR$, Gründe dafür sind die Nichterfüllung aller Anforderungen oder das Vorherrschen von unerwünschten Wechselbeziehungen. Dies wird auch als „coupled design“ bezeichnet. Der zweite Fall ergibt sich wenn $DP > FR$, dabei existiert eine redundante Konstruktion. Der gewünschte Idealfall wäre, wenn $DP = FR$. Mithilfe von Matrizen werden die Verbindungen zwischen den einzelnen Ebenen beschrieben. Die Entwurfsmatrix Gl. 3-1 definiert die Beziehung zwischen den Funktionsanforderungen FR und den Designparametern DP. (vgl. [LEY06], S.109)

$$\begin{pmatrix} FR_1 \\ FR_2 \\ \dots \\ FR_n \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{pmatrix} \begin{pmatrix} DP_1 \\ DP_2 \\ \dots \\ DP_n \end{pmatrix} \quad \text{Gl. 3-1}$$

Wenn der Designparameter entworfen werden kann, ohne die Funktionsanforderung zu beeinflussen, wird das Matrixelement null gesetzt. Wenn dies nicht der Fall ist, wird ein x eingetragen. Dadurch ergeben sich drei mögliche Entwurfsmatrixformen: (vgl. [LEY06], S.110)

1. „coupled design“

Einzelne Elemente können oberhalb oder auch unterhalb der Diagonale ungleich null sein.

$$\begin{pmatrix} FR_1 \\ FR_2 \\ FR_3 \end{pmatrix} = \begin{pmatrix} x & 0 & 0 \\ x & x & x \\ x & x & x \end{pmatrix} \begin{pmatrix} DP_1 \\ DP_2 \\ DP_3 \end{pmatrix} \quad \text{Gl. 3-2}$$

2. „decoupled design“

Alle Elemente sind entweder unterhalb oder oberhalb der Diagonale gleich null.

$$\begin{pmatrix} FR_1 \\ FR_2 \\ FR_3 \end{pmatrix} = \begin{pmatrix} x & 0 & 0 \\ x & x & 0 \\ x & x & x \end{pmatrix} \begin{pmatrix} DP_1 \\ DP_2 \\ DP_3 \end{pmatrix} \tag{Gl. 3-3}$$

3. „uncoupled design“

Nur auf der Diagonale kommen Werte ungleich null vor.

$$\begin{pmatrix} FR_1 \\ FR_2 \\ FR_3 \end{pmatrix} = \begin{pmatrix} x & 0 & 0 \\ 0 & x & 0 \\ 0 & 0 & x \end{pmatrix} \begin{pmatrix} DP_1 \\ DP_2 \\ DP_3 \end{pmatrix} \tag{Gl. 3-4}$$

Das „uncoupled design“ stellt die optimale Lösung dar, weil ein Designparameter genau eine Funktionsanforderung erfüllt und dabei keine anderen Funktionsanforderungen beeinflusst. Das „decoupled design“ bietet eine annehmbare Lösung, weil die Parameter sich nur in Richtung eines Zieles beeinflussen. In diesem Fall kann ein FR/DP-Kombination nur von hierarchisch übergeordneten Parametern beeinflusst werden, aber keinesfalls umgekehrt. Bei dem „coupled design“ herrschen zahlreiche Abhängigkeiten in beiden Richtungen vor und bietet daher keine brauchbare Lösung. (vgl. [LEY06] , S.111)

Vorgehensweise („Zigzagging“)

Auf jeder Ebene des Gesamtproduktes bestehen Verbindungen zwischen Funktionsanforderungen und den Designparametern. Die Vorgangsweise der hierarchischen Verbindungen der Variablen aus der Funktionsdomäne und der physikalischen Domäne wird als „Zigzagging“ bezeichnet. Die Abbildung 3-26 veranschaulicht die systematische Vorgehensweise. Als erstes werden die Anforderungen auf der ersten Ebene in Designparameter projiziert, danach geht man zurück in die Funktionsdomäne, in der die zweite Ebene analysiert wird und die physikalische Domäne projiziert. Durch das „Zigzagging“ erfolgt eine Produktstrukturierung, die aufgrund einer rekursiven Projektion von Anforderungen auf Produktfunktionen hervorgerufen wird. (vgl. [LEY06] , S.111)

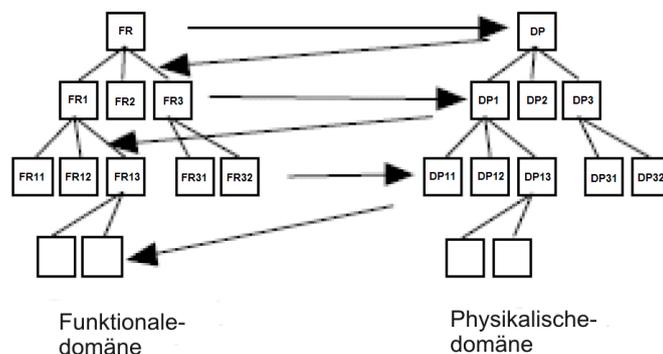


Abbildung 3-26: Vorgehensweise „zigzagging“ (angelehnt an [LEY06] , S.111)

Axiome

Für die Bewertung und für den Vergleich von Konstruktionslösungen werden die Axiome herangezogen. Nam P. Suh definiert zwei maßgebende Axiome:

1. Unabhängigkeitsaxiom: Dabei soll die Unabhängigkeit der Funktionsanforderung gewährleistet sein. Wenn mehrere Funktionsanforderungen existieren, muss die Konstruktion so konzipiert sein, dass alle Funktionsanforderungen erreichbar sind und bei Veränderungen der Funktionsanforderungen keine anderen beeinflusst werden.
2. Informationsaxiom: Dabei soll der Informationsgehalt der Konstruktion minimiert werden.

Das Unabhängigkeitsaxiom wird zur Anzeige der funktionalen Abhängigkeit in der Organisationseinheit verwendet. Das Informationsaxiom eignet sich für die Qualitätsbewertung der alternativen Konstruktionslösungen, in dem der Informationsinhalt verglichen wird. (vgl. [CEK00])

Ablaufdiagramm

Die Abbildung 3-27 stellt die schematische Ablauffolge beim Axiomatic Design dar.

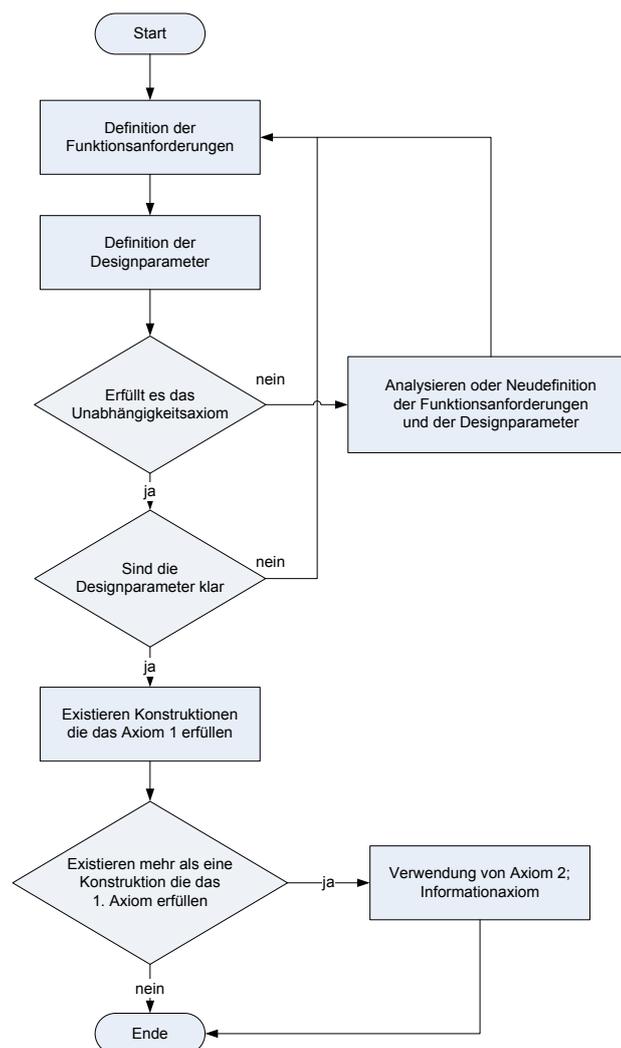


Abbildung 3-27: Ablaufdiagramm des Axiomatic Design (angelehnt an [CK10])

3.6 Wissensrepräsentation mit Softwaretools

Im Kapitel Grundlagen wurde das Thema Wissensrepräsentation erläutert, in diesem Teil der Arbeit werden einige Wissensrepräsentations- und Wissensvisualisierungstools für unterschiedliche Anwendungen betrachtet. Es erfolgt eine kurze Beschreibung der Tools in den nachfolgenden Punkten.

3.6.1 Compendium

Das Softwaretool Compendium hat seinen Ursprung im Jahre 1992 und wurde die folgenden Jahre bis 2006 laufend weiter entwickelt. Auf der Homepage von „CompendiumInstitute“ steht das Softwaretool kostenlos zur Verfügung. Einführungsvideos sowie Übungsanleitungen sind auf der Homepage ebenso publiziert.

Beschreibung

Compendium wurde für den Ideenaustausch sowie für die Schaffung von Artefakten und für das Aufbrechen von Grenzen zwischen Dialog, Artefakt, Wissen und Daten konzipiert. Es ist ein Softwaretool, das eine flexible Schnittstelle für die Verwaltung der Verbindungen zwischen Informationen und Ideen bietet. Compendium ist das Resultat von fünfzehn Jahren Forschung und Entwicklung von der Schnittmenge Hypertext, kooperative Modellierung, organisatorischer Speicherung und computerunterstützte Argumentation. (vgl. [KMI12])

Merkmale

Compendium bietet einen Weg für die Zusammenarbeit in Gruppen und in Projektteams. Es gibt Kontrolle über Wissens Elemente und kann auf unterschiedlicher Weise verwendet werden wie z.B., in Meetings oder für Besprechungen in der sich Personen an unterschiedlichen Orten befinden. Mit Compendium kann das Wissen bzw. Ideen in einer Mappe graphisch dargestellt und mit anderen Benutzern computerunterstützt diskutiert werden. Es können dabei alle „gewöhnlichen“ Dateiformate mittels „drag und drop“ in die Mappe eingefügt werden. Compendium bietet die Möglichkeit, die erstellte Wissensmappe in ein browserlesbares Dokument zu exportieren. In dem Softwaretool existieren drei unterschiedliche Verbindungstypen zwischen den Ideen bzw. Wissens Elementen:

1. Assoziation:
Verbindungen zwischen Ideen und Wissens Elementen die sich in derselben Mappe befinden.
2. Transklusion:
Verbindungen zwischen Ideen und Wissens Elementen die sich in unterschiedlichen Mappen befinden.
3. Kategorie:
Verbindungen zwischen Ideen und Wissens Elementen die sich in derselben Kategorie befinden.

Beispiel Compendium

Die Abbildung 3-28 zeigt die Anwendung von Compendium für eine Literaturanalyse, in der sich zahlreich verlinkte Notizen für die Forschung und Analyse von wissenschaftlicher Literatur befinden.

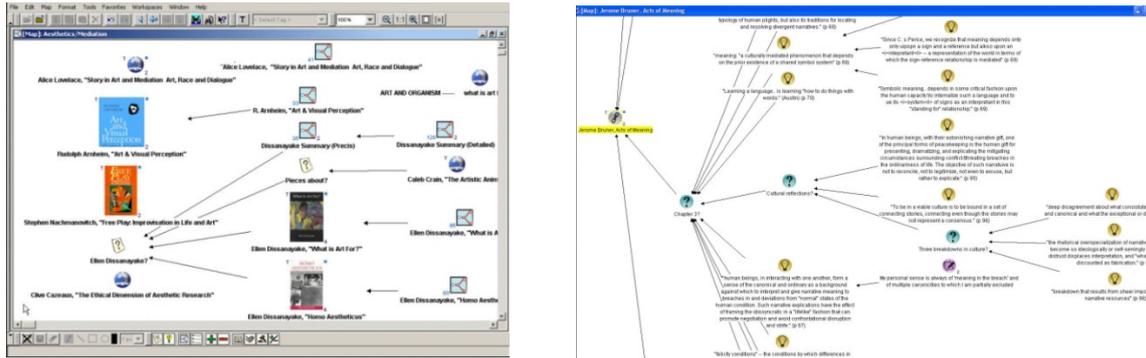


Abbildung 3-28: Literaturanalyse mit Compendium [COM12]

3.6.2 CmapTools

Das Wissensrepräsentationstool CmapTools wurde am Institute for Human & Machine Cognition in Florida entwickelt, es kann von der Homepage des Institutes „IHMC“ kostenlos heruntergeladen werden. Des Weiteren ist auf der Internetseite von „IHMC“ ein umfangreiches Anleitungsdocument veröffentlicht.

Beschreibung

CmapTools ist entwickelt worden um Concept Maps zu erstellen, welche das Verständnis für ein Wissensgebiet repräsentieren sollen. Im Falle einer großen Domäne oder einer detaillierten Repräsentation einer Domäne, kann eine einzelne Concept Map für den Anwender unüberschaubar, schwer zu verstehen und schwer zu manipulieren sein. Für die Konstruktionserleichterung einer großen Repräsentation erlaubt CmapTools die Aufsplitterung in eine Sammlung von Concept Maps. Zusätzlich kann der Benutzer Verbindungen zu anderen Informationsquellen wie Videos oder Bilder etc., erzeugen. Eine Reihe von Concept Maps und deren zugehörigen Informationsquellen über ein bestimmtes Wissensgebiet bezeichnet man als Wissensmodell. (vgl. [CHL02])

Merkmale

Die erstellten Concept Maps können auf den CmapServer hochgeladen und von anderen autorisierten Personen beurteilt und verändert werden. An einer Concept Map können mehrere Personen zeitgleich arbeiten, wobei die Änderungen allen Benutzern sofort angezeigt werden. Daher ist CmapTools sehr gut für Gruppenarbeiten geeignet. Wie bei dem vorigem Tool bietet dieses auch die Möglichkeit die Concept Maps als browserlesbares Dokument zu exportieren. Nebenbei beinhaltet es die Funktion zur Erstellung von Präsentationen.

Beispiel CmapTools

Die Abbildung 3-29 zeigt mehrere geöffnete Fenster eines Wissensmodelles über den Mars. In diesem Wissensmodell sind verschiedene Informationsquellen wie Webseiten, Bilder und Videos enthalten.

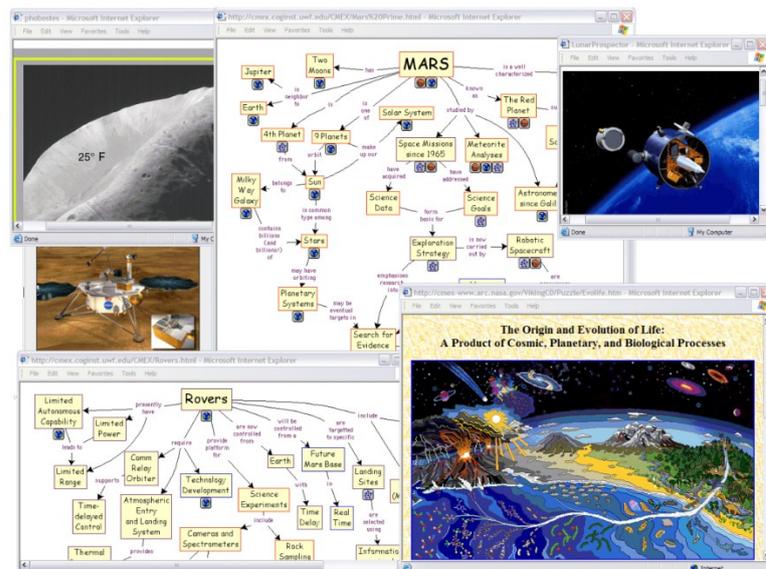


Abbildung 3-29: Wissensmodell über den Mars [CHL02]

3.6.3 VUE (Visual Understanding Environment)

Das Visual Understanding Environment-Tool kurz VUE, wurde an der Universität Tuft in Boston entwickelt. Das Tool kann von der Homepage „<http://vue.tufts.edu/>“ kostenlos bezogen werden. Zusätzlich bietet es den Lizenzcharakter Open Source. Anleitungsbeschreibungen wurden unter dem Menüpunkt „Help“ publiziert.

Beschreibung

Im Kern ist das VUE-Tool eine Concept Map Anwendung, entwickelt für die Organisationsunterstützung der Lehre und der Forschung. Es soll damit den Projektbeteiligten ein einfacherer Zugang zu den digitalen Informationen verschaffen werden. Mit einer visuellen Darstellung, bestehend aus Entitäten und Verbindungen, können die Projektbeteiligten Beziehungen zwischen den Ideen und den digitalen Inhalten zuordnen. VUE bietet eine flexible visuelle Umgebung für die Strukturierung, Präsentation und Teilung von digitalen Informationen. (vgl. [VUE12])

Merkmale

Die erstellten Concept Maps werden in einem Depot gespeichert. Das VUE-Tool bietet die Möglichkeit auf unterschiedlichen Ebenen zu arbeiten und die einzelnen Entitäten können gruppiert werden. Es unterstützt auch den Import von CSV-Dateien. Bei dem Einfügen einer Internetadresse bietet es die Möglichkeit zu einer automatischen Semantikanalyse. Die erstellten Concept Maps können in eine browserlesbare Datei exportiert werden und die Funktion zur Erstellung einer Präsentation ist ebenfalls in dem Tool enthalten.

Beispiel VUE

Die Abbildung 3-30 zeigt eine im VUE-Tool erzeugte Concept Map des U-Bahn Netzes in Boston mit Sightseeinginformationen zu den jeweiligen Plätzen.

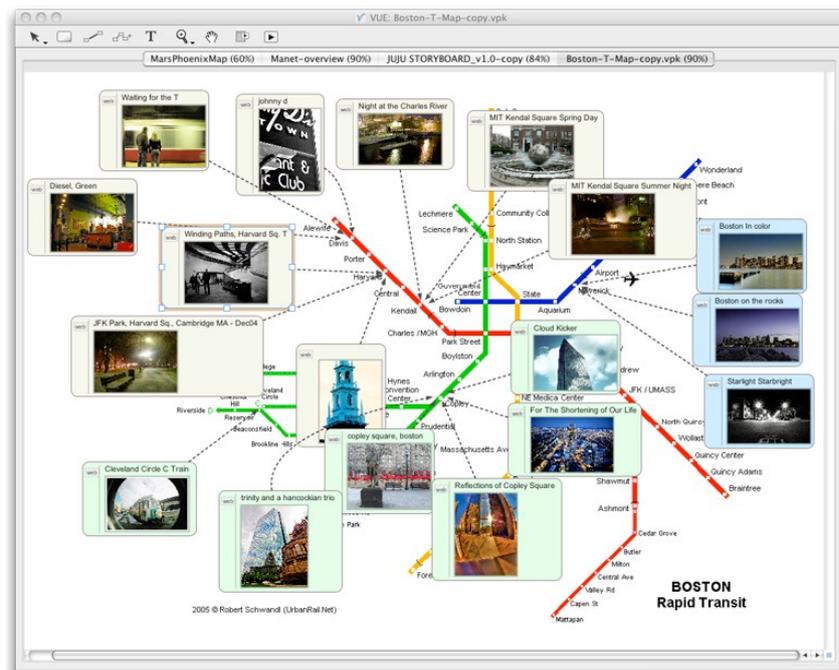


Abbildung 3-30: Concept Map des U-Bahn-Netzes in Boston [VUE12]

3.6.4 iMapping

Das iMapping Tool ist das Resultat der Doktorarbeit von Heiko Haller, zurzeit kann es noch kostenlos von der Homepage „<http://www.imapping.info/>“ heruntergeladen werden. Des Weiteren bietet es die Lizenzzeichnung eines Open Source-Tools. Das Tool wird in der Doktorarbeit von dem Autor [HAL11] detailliert dokumentiert.

Beschreibung

Ziel des iMapping Tools ist es, große, zoombare Wissenslandkarten zu erstellen. Es soll als ein visuelles Werkzeug für die Verwaltung des persönlichen Wissens dienen. Mit dem iMapping Tool können komplexe Zusammenhänge übersichtlich dargestellt werden und es eignet sich besonders gut für Forscher und Journalisten, die mit der Zeit sehr viele Informationen ansammeln und diese strukturieren möchten. Man kann sich das Tool wie eine große verschachtelbare Pinnwand vorstellen, in der man die Notizen übersichtlich organisieren kann. (vgl. [IMA12])

Merkmale

Mit dem iMapping Tool können sehr einfach Wissenslandkarten erstellt werden. Die Beziehungen zwischen den Entitäten können wahlweise ein- und ausgeblendet werden, welches zur Übersichtlichkeit der Mappe beiträgt. Dabei besteht die Möglichkeit alle Beziehungen oder nur die Beziehungen der ausgewählten Entität anzuzeigen. Es können auch versteckte Beziehungen erstellt werden, die wiederum je nach Bedürfnis angezeigt werden können. Des Weiteren bietet es eine Suchfunktion, damit die einzelnen Entitäten schneller gefunden werden können. Das Tool bietet jedoch nicht die Möglichkeit, dass mehrere Benutzer zeitgleich an einer Mappe arbeiten. Es können nur Bild-Dateien eingefügt werden. Das Einfügen von URLs und Videos ist nicht möglich. Zwei oder mehrere Mappen können auch nicht miteinander in Verbindung gesetzt werden.

Beispiel iMapping

Die Abbildung 3-31 stellt eine Wissenslandkarte über Quadrocopter dar, die in dem iMapping Tool erstellt wurde.

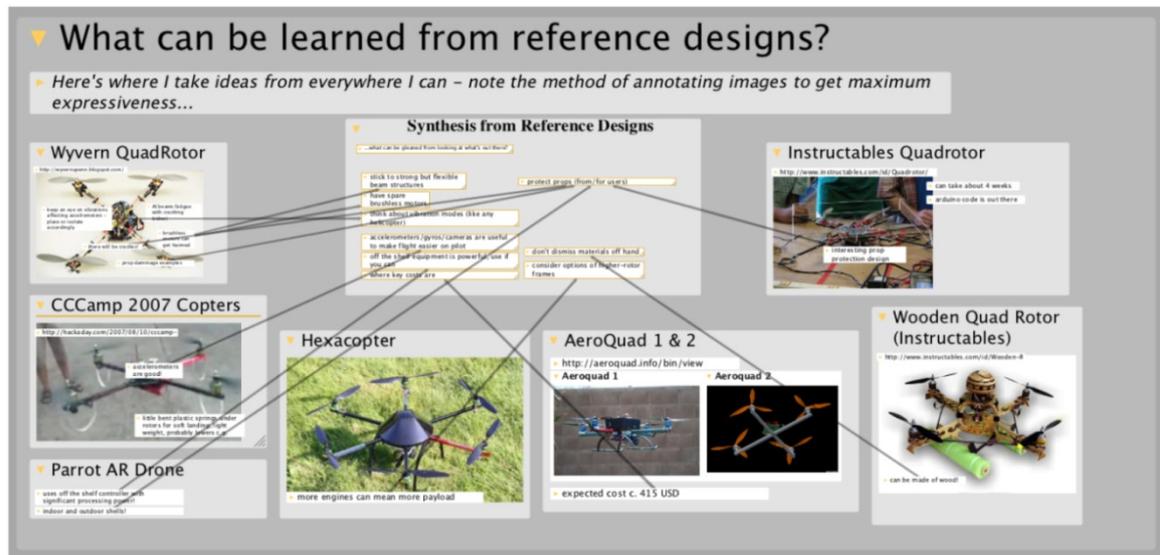


Abbildung 3-31: Wissenslandkarte eines Quadrocopters ([HAL11], S.137)

3.6.5 CAM (Cambridge Advanced Modeller)

Der Cambridge Advanced Modeller, abgekürzt CAM, wurde an der Universität Cambridge entwickelt. Das Softwaretool steht unter der Webseite „<http://www.edc.eng.cam.ac.uk/cam>“ für akademische Zwecke frei zur Verfügung. Ebenfalls ist unter der Adresse eine Anleitung für den CAM veröffentlicht.

Beschreibung

Das Softwaretool wurde für die Modellierung und Analyse der Abhängigkeiten von komplexen Systemen entwickelt. Mit dem CAM werden Produkte, Prozesse und Organisation modelliert und in Relation gesetzt. Er beinhaltet ein Diagramm-, ein Simulations- und ein DSM-Tool. (vgl. [CAM12])

Merkmale

In dieser Arbeit wurde nur das DSM-Tool näher betrachtet. Mit dem CAM kann sehr einfach und schnell eine DSM erstellt werden. Jedoch ist dabei zu beachten, dass sich die Matrix auf das amerikanische System bezieht, d.h., die Spalte beeinflusst die Zeile. Er bietet die Möglichkeit CSV-Dateien zu exportieren und zu importieren. Nach der Erstellung der DSM und der Bestimmung der Abhängigkeiten der einzelnen Parameter, bietet er die Möglichkeit, die Parameter manuell zu gruppieren. Im CAM sind auch die Funktionen zum automatischen Partitioning, Banding und Clustering enthalten. Für eine grafische Darstellung der Abhängigkeiten zwischen den Parametern kann eine Netzgrafik erzeugt werden. In dem Punkt 4.6.1 wurde der CAM für die Erstellung einer DSM für Rollenförderer angewendet.

Beispiel CAM

Die Abbildung 3-32 zeigt an der linken Seite die DSM und an der rechten Seite die Netzgrafik eines Staubsaugers die mit dem Cambridge Advanced Modeller erstellt wurden.

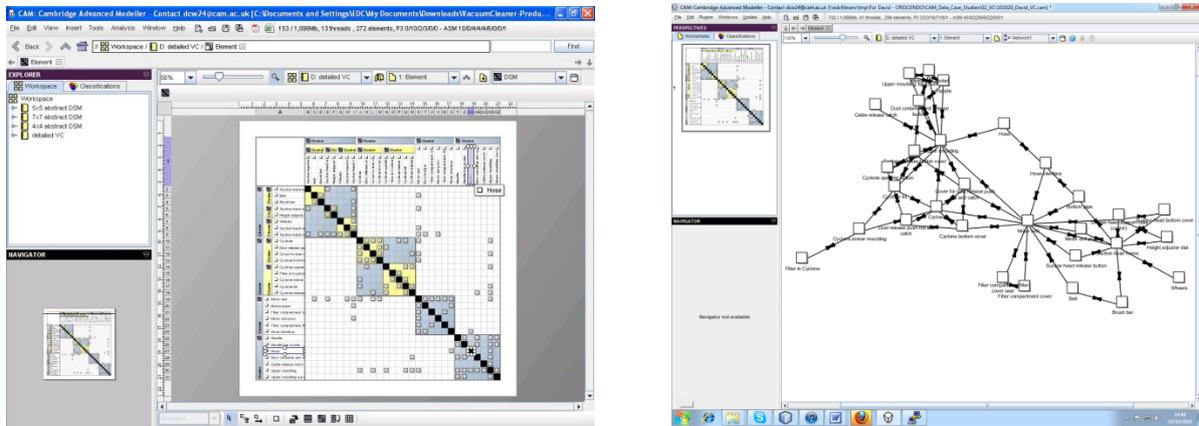


Abbildung 3-32: DSM und Netzgrafik eines Staubsaugers [CAM12]

3.6.6 Set Visualiser

Das Softwaretool Set Visualiser wurde an der Universität Cambridge von David F. Wyatt entwickelt und kann nach einer Registrierung von der Internetseite „http://www-edc.eng.cam.ac.uk/tools/set_visualiser/“ kostenlos bezogen werden. Eine Anleitung für das Tool befindet sich auch unter der Adresse.

Beschreibung

Der Set Visualiser wurde für die visuelle Darstellung einer Anzahl von Elementen entwickelt, die sich in eine oder mehrere Kategorien befinden. Diese Einstufung erfolgt mithilfe eines Schnittmengendiagramms. Der Vorteil dieses Ansatzes ist, dass die Darstellung nicht in einer hierarchischen Klassifikation angeordnet wird, sondern die Elemente, die zu mehreren Kategorien gehören, durch eine Schnittmenge dargestellt werden können. (vgl. [EDC12])

Merkmale

Die Handhabung des Set Visualiser ist sehr simpel, nachdem man für die gewünschte Domäne eine CSV-Datei (Comma-separated values) erstellt hat, in der man die einzelnen Elemente zu den Oberbegriffen zuordnet, kann die CSV-Datei in den Set Visualiser importiert und grafisch dargestellt werden. Das hiermit erzeugte Schnittmengendiagramm kann in die gängigsten Bildformate und Vektorgrafiken exportiert werden. Der Set Visualiser wurde im Punkt 4.6.3 für die Darstellung eines Schnittmengendiagrammes von Rollenfördererparameter angewendet.

Beispiel Set Visualiser

In der Abbildung 3-33 wird ein Mengendiagramm, welches mit dem Set Visualiser erstellt wurde, dargestellt. In dem Diagramm werden Früchte- und Gemüsesorten kategorisiert. Des Weiteren erfolgt eine Kategorisierung nach ihrer jeweiligen Farbe.

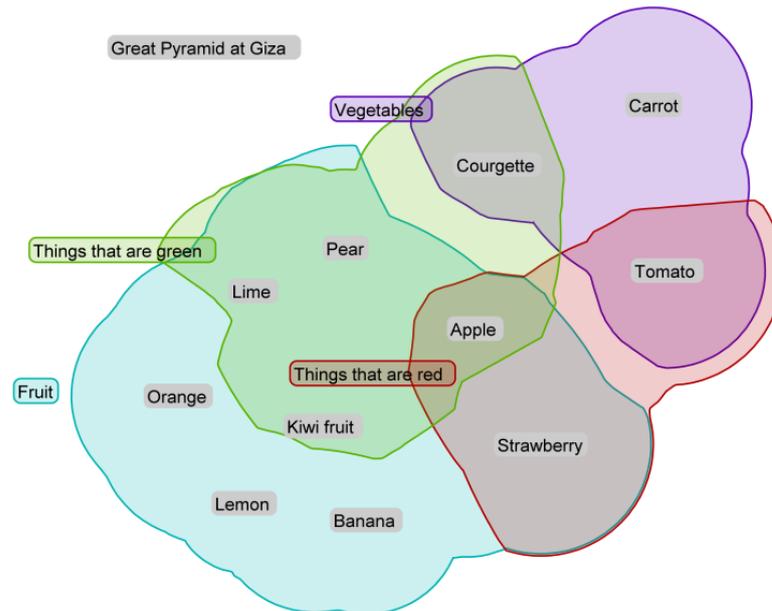


Abbildung 3-33: Mengendiagramm von Früchte- und Gemüsesorten [EDC12]

Im sechsten Kapitel unter Punkt 6.3 erfolgt eine Bewertung der Wissensrepräsentationstools und anschließend befindet sich ein Fazit der Softwaretools.

4 Eigene Ansätze – Ontologie und DSM am Beispiel eines Rollenförderers

Basierend auf den davor behandelten Themen Ontologie bzw. Wissensrepräsentation und Design Structure Matrix, wurde ein Anwendungsbeispiel durchgeführt. Maßgebend dafür war, dass das Einsatzfeld in der Fachdisziplin der Technischen Logistik angesiedelt wird. Die Fördertechnik stellt einen Schwerpunkt in diesem Gebiet dar. Die Definition der Fördertechnik nach [VDI70] lautet: „Das Fortbewegen von Gütern in beliebiger Richtung über begrenzte Entfernungen durch technische Hilfsmittel sowie die Ortsveränderung von Personen, soweit diese nicht in den Bereich der Verkehrstechnik fällt, einschließlich der Lehre der Fördermittel selbst.“

Wobei der Oberbegriff Fördermittel für alle Maschinen und Hilfsmittel in der Fördertechnik steht. Die Unterteilung der Fördermittel erfolgt in Unstetig- und Stetigförderer. Unstetigförderer sind definiert durch einen diskontinuierlichen Förderprozess. Bei diesen Fördergeräten erfolgt i.d.R. nach einem Lastspiel ein Leerspiel (z.B. Stapler oder Kräne). Die Stetigförderer zeichnen sich durch einen kontinuierlichen Förderprozess aus, Transportgüter werden über einen längeren Zeitraum (ca. 10min) gefördert (z.B. Gurtbandförderer oder Rollenförderer). (vgl. [JOD12] , S.140 ff)

Für dieses Anwendungsbeispiel wurde auf die Stetigförderer näher eingegangen. Im konkreteren Fall wurde der Fördertyp Rollenförderer bearbeitet. Eine fiktive Rollenförderstrecke aufgrund von Eingangsparametern wurde erstellt. Die erarbeitete Grundagentheorie über Rollenförderer und die erstellte Förderstrecke wurden in das Softwaretool xKBE-App eingespeist. Die xKBE-App wird am Institut für Technische Logistik entwickelt, sie bietet ein unterstützendes Werkzeug für die Konstruktion mit integrierter Wissensspeicherung, genauere Erläuterung erfolgt im Punkt 4.5. Des Weiteren wurde eine Design Structure Matrix aufgestellt, welche die Parameterabhängigkeiten eines Rollenförderers aufzeigt und grafisch darstellen lässt.

4.1 Definition Stetigförderer

Stetigförderer werden für den Transport von Schütt- und Stückgut als auch für den Personentransport eingesetzt. Sie arbeiten kontinuierlich über einen längeren Zeitraum hinweg. Mit den unterschiedlichen Fördertypen sind waagrechte, geneigte oder senkrechte Förderungen über gerade oder gekrümmte Strecken möglich. Abseits der Bewältigung üblicher Förderaufgaben werden Stetigförderer auch als Bindeglied für die technologischen Prozesse in der Serien- und Massenproduktion verwendet. Sie können die technologischen Prozesse auch selbst übernehmen (z.B. Kühlbänder oder Durchlauföfen). Die Definition von Stetigförderern sowie die Einteilung der Fördertypen mit ihren Zubehörgeräten sind in der DIN 15201 enthalten. (vgl. [MRW08] , S.187 f)

Unterteilung der Stetigförderer

Die Unterteilung der Stetigförderer kann auf zwei Arten erfolgen. Einmal wird das Kriterium nach dem konstruktiven Aufbau und dem Funktionsprinzip herangezogen, zum anderen kann die Gliederung nach dem Transportgut festgelegt werden. Diese Einteilungen werden in der Abbildung 4-1 und Abbildung 4-2 dargestellt. (vgl. [MAR06] , S.130)

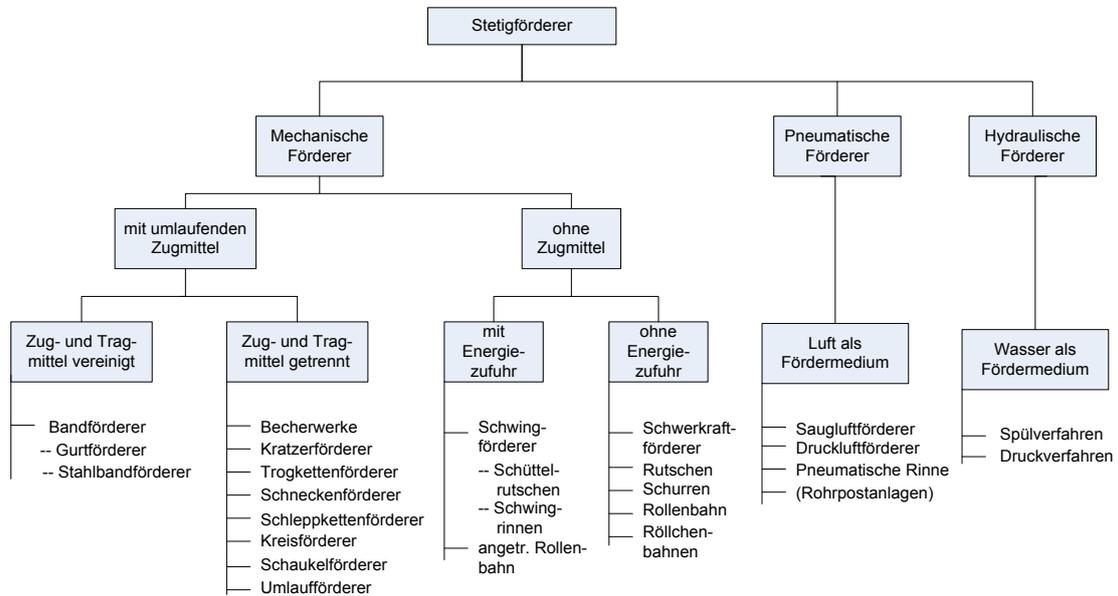


Abbildung 4-1: Einteilung der Stetigförderer nach konstruktiven und funktionalen Gesichtspunkten (angelehnt an [MAR06] , S.130)

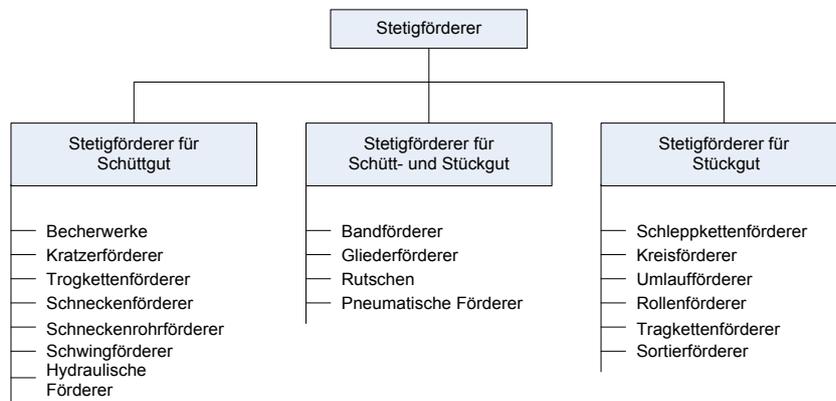


Abbildung 4-2: Einteilung der Stetigförderer nach der Art des Transportgutes (angelehnt an [MAR06] , S.130)

Mithilfe von Baukastensystemen und Geschwindigkeitsänderung können Kapazitäts- oder Anlagenerweiterungen durchgeführt werden. Stetigförderer werden angewendet beim An- und Abtransport von Materialien, beim Bergbau, beim Tagebau, bei der metallverarbeitenden Industrie, bei Kraftwerken, beim Fertigungsablauf, im Lagerbereich, bei der Automobilmontage oder zum Verbinden von Produktionsprozessen. (vgl. [MAR06] , S.129)

4.2 Berechnungsgrundlagen Stetigförderer

Die maßgebenden Auslegungsgrößen für Stetigförderer sind die Fördermenge und die benötigte Antriebsleistung bei gegebener Streckenführung und Bauart. Prinzipiell wird nach Schüttgut- und Stückguttransport unterschieden. Die folgenden Berechnungsvorschriften stammen aus der Literatur (vgl. [MRW08] , S.187 ff).

Fördermenge bei Schüttguttransport

Volumenstrom bei fließender Förderung:

$$\dot{V} = A * v_F \quad \text{Gl. 4-1}$$

Volumenstrom bei Förderung in Einzelgefäßen:

$$\dot{V} = \frac{V}{e_{st}} * v_F \quad \text{Gl. 4-2}$$

Massenstrom:

$$\dot{m} = \dot{V} * \rho_s \quad \text{Gl. 4-3}$$

A Gutquerschnitt
 v_F Fördergeschwindigkeit
 \dot{V} Volumen des Einzelgefäßes
 e_{st} Abstand der Einzelgefäße
 ρ_s Schüttdichte

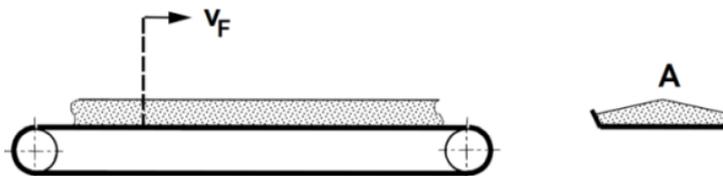


Abbildung 4-3: Schüttguttransport ([JOD12] , S.145)

Fördermenge bei Stückguttransport

Massenstrom:

$$\dot{m} = \frac{m}{e_{st}} * v_F \quad \text{Gl. 4-4}$$

Stückstrom [#h]:

$$\dot{m} = \frac{v_F}{e_{st}} \quad \text{Gl. 4-5}$$

m Masse des zu fördernden Einzelstücks
 e_{st} Abstand der Einzelstücke
 v_F Fördergeschwindigkeit

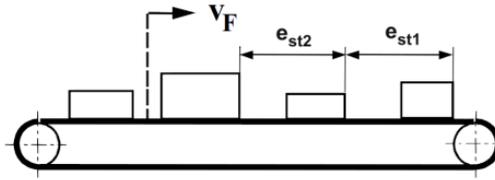


Abbildung 4-4: Stückguttransport ([JOD12], S.145)

Antriebsleistung

Die Antriebsleistung eines Steigförderers mit Zugmittel ist abhängig von dem Gesamtwiderstand F_W und der Fördergeschwindigkeit v_F .

Der Gesamtwiderstand ergibt sich aus allen Bewegungswiderständen, die von der Antriebstrommel bzw. dem Antriebsrad zu überwinden sind. Die Summe aus allen Hub- und Reibungswiderständen, die das Zugmittel aufbringt, ist der Gesamtwiderstand.

Der Hubwiderstand wird hervorgerufen durch Förderung eines Massenstroms auf die Höhe h .

Hubwiderstand:

$$F_{WH} = m_{IG} * g * h \quad \text{Gl. 4-6}$$

Gutlast (m_{IG}) bezogen auf die Längeneinheit [kg/m]:

$$m_{IG} = \frac{\dot{m}}{v_F} \quad \text{Gl. 4-7}$$

\dot{m} .. Massenstrom

g ... Fallbeschleunigung

h ... Förderhöhe

v_F .. Fördergeschwindigkeit

Bei ausschließlich horizontaler Förderung ist der Hubwiderstand null.

Der Reibungswiderstand kann mit der Gesamtwiderstandsmethode überschlägig ermittelt werden. Dabei wird die Gesamtreibungszahl μ_{ges} zur Hilfe genommen. Je nach Stetigfördererbauarten kann die Gesamtreibungszahl μ_{ges} grob schwanken.

Reibungswiderstand:

$$F_{WR} = \mu_{ges} * l * g * (m_{IF} + m_{IG}) \quad \text{Gl. 4-8}$$

Daraus folgt der

Gesamtwiderstand:

$$F_W = F_U = \mu_{ges} * l * g * (m_{IF} * m_{IG}) \pm m_{IG} * g * h \quad \text{Gl. 4-9}$$

- l Horizontalprojektion der Förderlänge
- μ_{ges} Gesamtreibungsbeiwert der Anlage
- m_{lF} reibungskrafterzeugende Eigenlast bezogen auf die Längeneinheit [kg/m]
- „+“ Aufwärtsförderung
- „-“ Abwärtsförderung

Zusätzliche Widerstände, die durch Gutauf- und -abgabe, Umlenkung usw. hervorgerufen werden, sind wegen ihres geringen Einflusses vernachlässigbar.

Für die Antriebsdimensionierung ist die Nennleistung entscheidend:

Antriebsleistung:

$$P_N = \frac{F_U \cdot v_F}{\eta} \qquad \text{Gl. 4-10}$$

- F_U Umfangskraft der Antriebstrommel/-rad
- v_F Fördergeschwindigkeit
- η Wirkungsgrad des Antriebs

Bei schnelllaufenden, sehr langen und schwer belasteten Stetigförderern muss auch die Anlaufleistung bzw. Beschleunigungsleistung überprüft werden. Die dynamischen Kräfte können bei Kettenförderer aufgrund der geringen Fördergeschwindigkeiten vernachlässigt werden. (vgl. [MRW08] , S.187 ff)

4.3 Definition Rollenförderer

Rollenförderer bestehen aus Rollenbahnen und werden nur für den Stückguttransport eingesetzt. Die Rollenbahnen, dargestellt in der Abbildung 4-5, sind aus vielen nacheinander angeordneten, frei drehbaren und zwischen zwei Profilen befestigten Tragrollen, zusammengesetzt. Der Achsabstand muss auf die Transportgutlänge abgestimmt sein. Je größer die Anzahl der tragenden Rollen ist, desto gleichmäßiger erfolgt der Fördervorgang. Das Transportgutgewicht unterteilt den Rollenförderer nach leichten Stückgutförderer (bis ca. 50kg, z.B. Behälter und Pakete) und schweren Stückgutförderer (bis 1t und mehr, z.B. Europaletten etc.). Die Rollenbahnen können angetrieben und nicht angetrieben ausgeführt werden. Der angetriebene Förderer besitzt einen Motor, sehr häufig werden Trommelmotoren eingesetzt, dabei wird ein Elektroantrieb in einer Rolle eingebaut. Dies bietet die Möglichkeit für eine staub- und wasserdichte Ausführung. Die Kraftübertragung bei leichtem Stückgut erfolgt im Allgemeinen kraftschlüssig über einen Riemen. Beim Untergurtantrieb pressen Druckrollen den Riemen an die Unterseite der Tragrolle. Eine andere Möglichkeit des Übertriebs wäre mit Keilriemen, bei der die Rollen mit Keilriemenscheiben oder Nuten ausgeführt sind. Fördergeschwindigkeiten von 0,3m/s bis 1,0m/s sind üblich, in Ausnahmefällen sind auch Geschwindigkeiten bis zu 2m/s möglich. Bei den Schwerlastrollenbahnen erfolgt der Antrieb über eine Kette, wobei nur Fördergeschwindigkeiten bis zu ca. 0,3m/s möglich sind. Die Rollenbahnen werden ausschließlich nach dem Baukastenprinzip hergestellt und in fertigmontierten Baugruppen geliefert. Sie bestehen aus den Funktionsgruppen Rollkörper (Achse,

Lagerung, Abdichtung), Rahmenkonstruktion und Aufständigung bzw. Aufhängung. Zur Förderrichtungsänderung werden Rollenkurven eingesetzt, dargestellt in der Abbildung 4-5, die vorwiegend aus konischen Rollen bestehen, welche die unterschiedlichen Geschwindigkeitskomponenten des Förderguts ausgleichen. Zur Fördergutabgabe- bzw. -aufnahme werden Bauelemente wie Ein- und Ausschleusweichen, Drehtische, Verschiebewagen und Hub-, Absenk- sowie Transfereinrichtungen, eingesetzt. Staurollenförderer ermöglichen eine Pufferung der Transportgüter, die bei unterschiedlichen zu- und abfließenden Strömen erforderlich ist. Es gibt zwei Varianten von Staurollenförderern: Zum einen den staudruckarmen Förderer, der während des Stauvorgangs einen Restdruck auf das Fördergut ausübt und zum anderen den staudrucklosen Förderer, welcher das Fördergut mit Abstand, ohne jeglichen Druck, staut. Nähere Informationen über Staurollenförderer stehen in der DIN 15201. Eine andere Variante für den Rollenfördereraufbau ist die Aufteilung des Förderabschnitts in Segmente von ca. 4-5 Rollen, deren Antrieb über einen Trommelmotor erfolgt. Eine Rolle treibt die anderen Rollen im Segment mittels eines Gurtes an. Die einzelnen Antriebsrollen können getrennt voneinander angesteuert werden, diese Bauweise liefert den Vorteil einer automatischen Staufunktion. Aufgrund der einfachen Bauweise, der Robustheit und der geringen Investitions- und Betriebskosten werden Rollenförderer sehr oft eingesetzt. Ihr Anwendungsgebiet liegt in der Lagervorzone, im Kommissionierbereich und in der Produktion. (vgl. [HSN07] , S.131 ff)

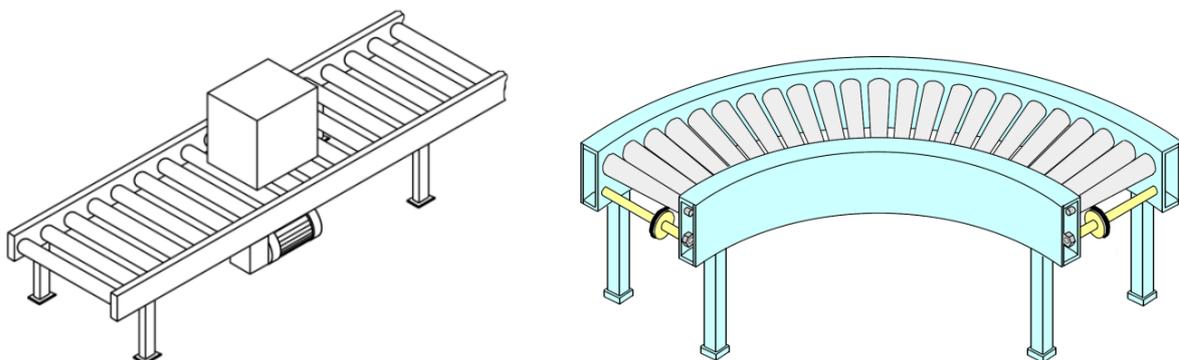


Abbildung 4-5: Rollenbahn und Rollenbankkurve ([HSN07] S.131; [JOD12] S.262)

4.4 Erstellung einer Rollenfördererstrecke

Aufgrund der bis jetzt erwähnten Theorie über Stetigförderer und im speziellen über Rollenförderer, wurde eine Förderstrecke für einen Rollenförderer mit Eingangsparametern (wie Transportgutabmessungen, Transportgutgewicht und Durchsatz) angenommen. Damit ist ein konkreter Anwendungsfall für die Erstellung einer Ontologie in der xKBE-App und die Aufstellung einer DSM über Rollenförderer möglich. In der Abbildung 4-6 wird eine Prinzipskizze der fiktiven Förderstrecke veranschaulicht. Die Förderstrecke wurde aus Fördermodulen von dem Hersteller Interroll zusammengestellt, die aus dem Fördermodulkatalog von Interroll [INT12a] ausgewählt wurden. Dabei wurden die Module Rollenbahngerade, Rollenbahnkurve, Schrägeinschleuser, Riemenausschleuser, Endstück und Standfüße verwendet. Die folgende Aufzählung enthält die gewählten Eingangsparameter der Förderstrecke.

Eingangsparameter:

- Transportgutmaterial:..... Kunststoffboxen
- Transportgutbreite: 550mm
- Transportgutlänge:..... 300mm
- Transportguthöhe:..... 150mm
- Transportgutgewicht:..... 30kg
- Durchsatz:..... 7200Stk/h
- Gemittelter Stückgutabstand: 500mm

Mit der Formel für den Durchsatz ergibt sich die Fördergeschwindigkeit:

$$\lambda = \frac{v_F}{s}$$

λDurchsatz

v_FFördergeschwindigkeit

sgemittelter Stückgutabstand

$$v_F = \lambda * s = \frac{7200 \frac{\text{Stk}}{\text{h}}}{3600} * 0,5\text{m} = 1,0\text{m/s}$$

- Fördergeschwindigkeit:..... 1,0m/s

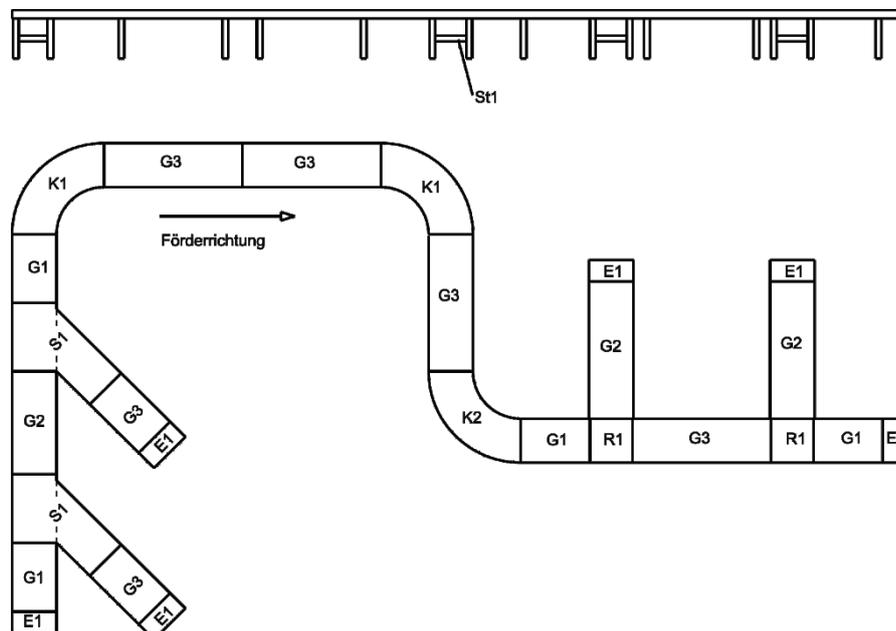


Abbildung 4-6: Prinzipskizze der Förderstrecke

Aufgrund der Eingangsparameter und der errechneten Fördergeschwindigkeit, fiel die Entscheidung für den Antrieb auf eine Motorrolle (RollerDriveEC310, Typenbezeichnung von Interroll) und für das Antriebsmedium wurde ein Keilrippenriemen (PolyVEE Riemen) gewählt. Zur genauen Spezifizierung der verwendeten Fördermodule werden in der folgenden Auflistung die Bestelldaten angegeben, mit den technischen Datenblättern, die sich im Anhang befinden, kann hiermit das genau gewählte Fördermodul bestimmt werden.

Bestelldaten der Fördermodule:

- Gerade G1: 5504-660-1200-75-2-EC310-L-ZE-PR-1,0-150/150
- Gerade G2: 5504-660-1800-75-3-EC310-L-ZE-PR-1,0-150/150
- Gerade G3: 5504-660-2400-75-4-EC310.L-ZE-PR-1,0-150/150
- Kurve K1: 5530-660-90-2-RH-EC310-ZE-P-1,0
- Kurve K2: 5530-660-90-2-LH-EC310-ZE-P-1,0
- Schrägeinschleuser S1: 5550-660-1200-45-75-RH-EC310-P-1,0-150/150
- Riemeneinschleuser R1: 5591-660-740-75-135-225-225-75-
O/O/C-EC310.P-1,0
- Endstück E1: 5574-660-5504-150/150
- Standfuß St1: 5703-660-700

4.5 Erstellung einer Ontologie in der xKBE-App

Die xKBE-App ist ein Softwaretool, das zurzeit am Institut für Technische Logistik entwickelt wird. Mit der App kann man eine Ontologie über eine Domäne erstellen, ihr Anwendungsgebiet zielt jedoch auf den Konstruktionsbereich im Maschinenbau ab. Es soll damit möglich sein, Komponenten einer Baugruppe und ihre gegenseitigen Abhängigkeiten in Form einer Ontologie darzustellen. Zusätzlich können Informationen zu den jeweiligen Bauteilen verknüpft werden wie Normen, VDI –Richtlinien, technische Datenblätter oder Grundagentheorie über einen bestimmten Fachbereich. Sie soll als unterstützendes Werkzeug für den Konstrukteur und als Wissensspeicherung dienen, mit der man das bereits erarbeitete Wissen festhalten und anderen Mitarbeiter zur Verfügung stellen kann. Die Daten der Ontologie sowie die Informationen über die Theorie werden dabei in einer Datenbank gespeichert.

4.5.1 Einpflegen der Förderstrecke in die xKBE-App

Zuerst wurden die unterschiedlichen Fördertypen als Entität in der App erstellt, die in der Abbildung 4-7, auf der rechten Seite dargestellt sind. Danach wurden sie mit dem Oberbegriff Stetigförderer in Relation gesetzt, die Relationsart wurde zwischen der Entität Stetigförderer und den einzelnen Entitäten der Fördertypen mit dem Begriff „abgeleitet“ festgelegt, d.h. der Rollenförderer ist eine Ableitung vom Stetigförderer. Danach erfolgte die Erfassung der zuvor bestimmten Förderstrecke, welche in der Abbildung 4-7 auf der linken Seite angesiedelt ist. Dabei wurden die einzelnen Fördermodule als Entitäten erzeugt und mit der Relationsart „zusammengebaut“ zu der Entität Förderstrecke verknüpft. Das Fördermodul bzw. die Entität Gerade besteht aus den Entitäten Rolle und Seitenprofile, die Entität Rolle besteht wiederum aus den Entitäten Rollenlager und Achse usw. Dies erleichtert dem Benutzer die Zusammenhänge der Bauteile mit der Baugruppe und dem gesamten Förderer zu erkennen. Den jeweiligen Entitäten wurden Attribute hinzugefügt, z.B. wurden dem Fördermodul Gerade die Attribute Fördergeschwindigkeit, max. Traglast, Rollenteilung, Einsatztemperatur, lichte Weite und Zonenanzahl angefügt. Des Weiteren wurden den Entitäten Dokumente angehängt wie z.B., technische Datenblätter bei den Fördermodulen oder eine Streckenskizze bei der Entität Förderstrecke.

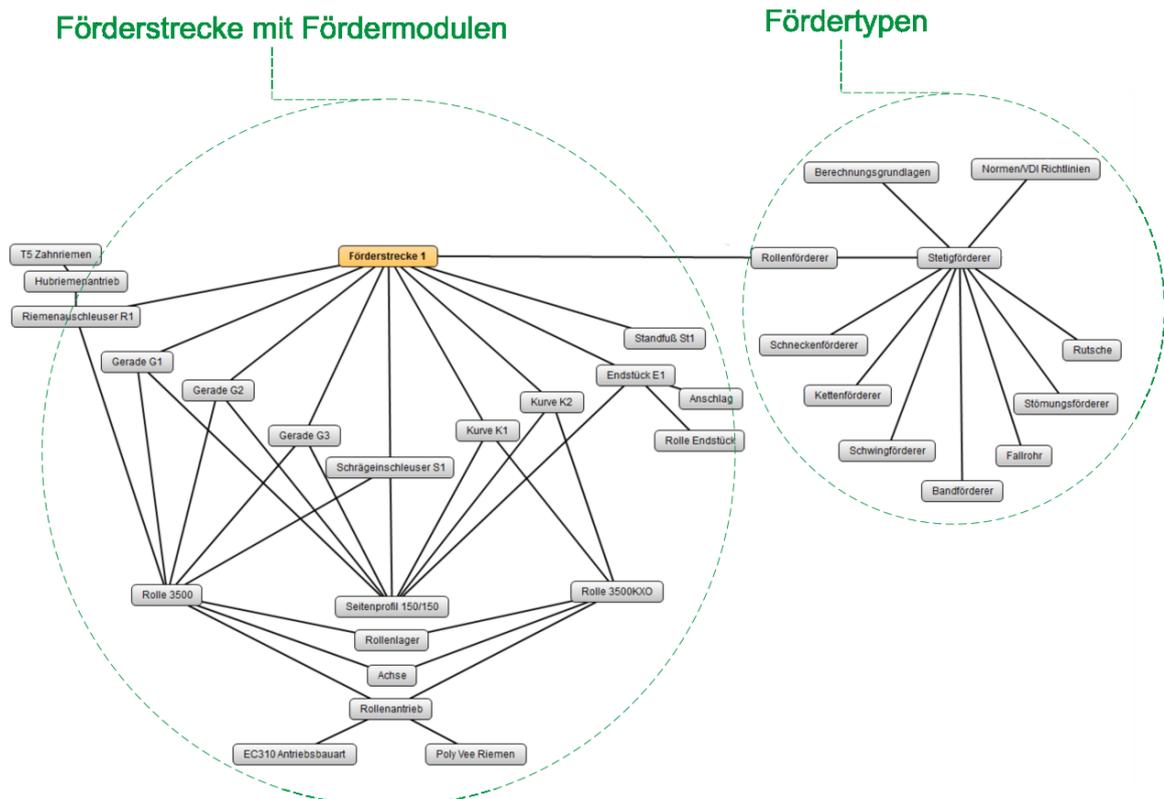


Abbildung 4-7: Ontologie der Förderstrecke

4.5.2 Einbetten von Normen und Theorie in die xKBE-App

Zu der bis jetzt erzeugten Ontologie wurde noch Theorie über Rollenförderer hinzugefügt. Allgemeine Informationen wie Eigenschaften der unterschiedlichen Antriebsmedien sowie Informationen für die Auslegung eines Rollenförderers, z.B. der Einfluss des Transportgutes auf die Fördermodule. Die Rollenförderertheorie befindet sich in der Abbildung 4-8 auf der linken Seite. Die jeweiligen Theoriekomponenten wurden dann mit den Bauteilen der Förderstrecke verknüpft, z.B., das Fördermodul Kurve in der Förderstrecke wurde mit dem Theorie teil Kurvenmerkmale, in Relation gesetzt. In diesem Teil sind unter anderen Informationen wie die Berechnungsgrundlage für die Kurvenradien, enthalten. Dem Benutzer dieses Programmes bleibt so zeitaufwändiges Suchen nach Informationen erspart. Die Ontologie wurde noch um Berechnungsgrundlagen sowie Normen und VDI- Richtlinien über Stetigförderer erweitert, die in der Abbildung 4-8 rechts angesiedelt sind, die wiederum mit der Rollenförderertheorie in Relation gesetzt wurden. Zum Beispiel wurde die DIN 15201, in der die Staufunktion definiert ist, mit der Theorieentität Stauvorgang verknüpft. Den jeweiligen Entitäten wurden noch Informationsdokumente im pdf-Format angehängt.

4.5.3 Funktionsumfang der xKBE App

Erstellung von Entitäten und Relationen:

Die Erstellung von Entitäten funktioniert mit einem rechten Mausklick innerhalb des Arbeitsbereiches, danach wählt man „Add Entity“. Zur Umbenennung oder um Kommentare der Entität hinzuzufügen, klickt man mit der rechten Maustaste auf die Entität und wählt in der Auswahlliste die Option „Details“ aus. Für die Relationserstellung zwischen zwei Entitäten, gilt dasselbe wie für die Umbenennung, nur wird nun in der Auswahlliste die Option „Add Relation“ gewählt, dabei bestimmt jedoch die Reihenfolge der Entitätenauswahl, die Richtung der Relation, welche sich aber nachträglich im Relationsmanager jederzeit ändern lässt, aber dazu wird darunter Näheres beschrieben.

Erzeugen von Attributen:

Um den Entitäten Attribute hinzuzufügen, klickt man wieder mit der rechten Maustaste auf die Entität und wählt danach „Details“, in dem „Entity-Details“ Fenster wählt man oben in der Menüleiste „Attribute“. Mit dem Button „Add new“ lässt sich nun ein Attribut erzeugen, welches mit einem Wert und einer Einheit bestimmt wird. Die Attribute lassen sich nachträglich durch den Button „Modify“ jederzeit ändern.

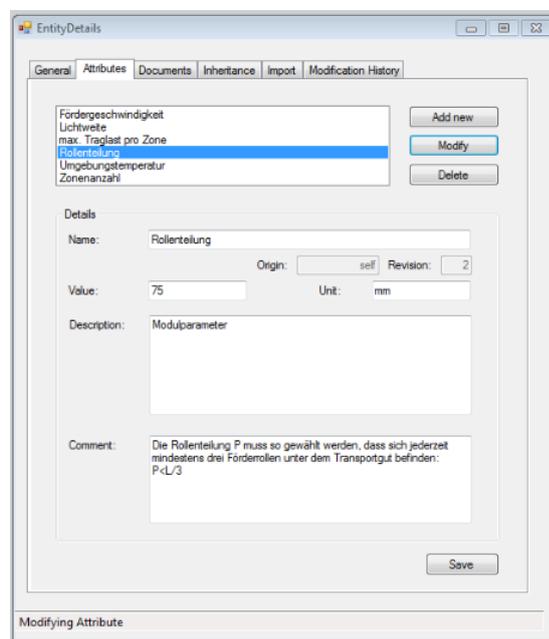


Abbildung 4-9: Attributerstellung

Anheften von Dokumenten:

Für die Anheftung von Dokumenten wählt man in der Menüleiste des Fensters „Entity-Details“ „Documents“ und durch den Button „Hinzufügen“ lassen sich pdf-Dateien oder jegliche Grafikformat hinzufügen.

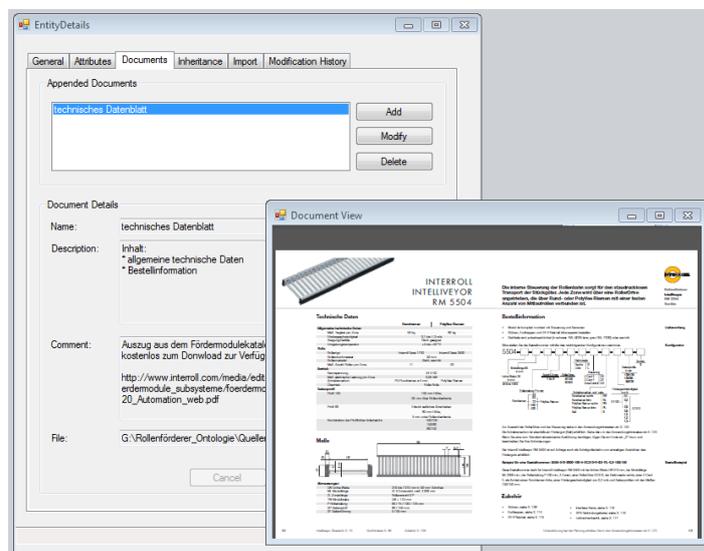


Abbildung 4-10: Dokumentenanheftung

Verwalten von Relationen:

Mit einem Doppelklick auf die Relationslinie kommt man zur Relationsliste, davon wählt man die jeweilige Relation aus. In dem Fenster „relationForm“ kann die Richtung der Relation und die zuvor festgelegten Relationstypen verändert werden. Links unten in dem Fenster „relationForm“ befindet sich der Button „Manage Relation Types“ damit wird das Fenster „Relation Types“ geöffnet, hier können neue Relationstypen hinzugefügt, modifiziert oder gelöscht werden.

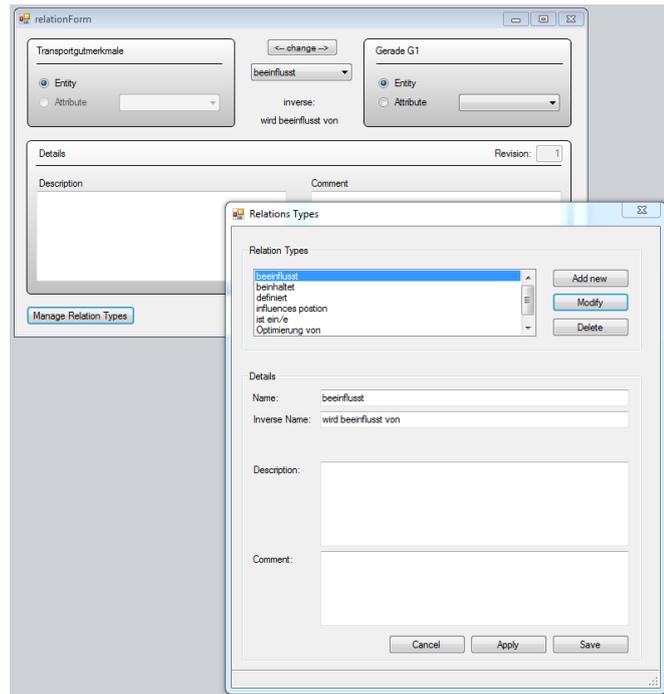


Abbildung 4-11: Relationsverwaltung

Stücklistenanzeige:

Wenn man die Entitäten mit der Relationsart „zusammengebaut“ verknüpft, kann eine Stückliste angezeigt werden. Diese Darstellung erfolgt durch einen rechten Mausklick auf die Entität und darauffolgend wählt man die Option „Show BOM“. Danach öffnet sich das Fenster „BOM“ in der angezeigt wird, welche Bauteile in der Baugruppe verbaut sind.

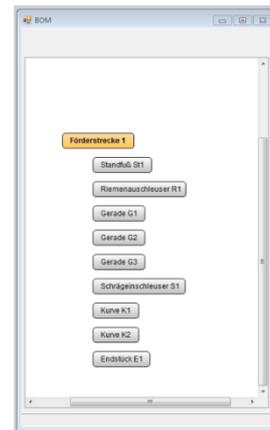


Abbildung 4-12: Stücklistenanzeige

Klassenbrowser:

Im Klassenbrowser werden alle erzeugten Entitäten angezeigt. Entitäten, die mit der Relationsart „abgeleitet“ verknüpft wurden, sind der Entität von der sie abgeleitet wurden, untergeordnet.

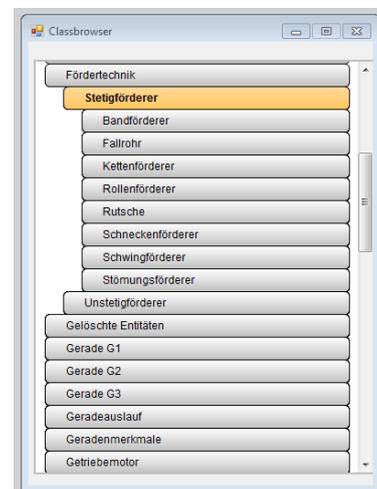


Abbildung 4-13: Klassenbrowser

4.5.4 Möglichkeiten und Nutzen der xKBE App

Die xKBE App bietet die Möglichkeit zur Erstellung einer Ontologie über eine Domäne. Der Konstrukteur kann die Zusammenhänge der Bauteile zu den Baugruppen in der xKBE App abbilden. Dadurch wird ein besserer Überblick der Komplexität eines Systems verschafft. Durch die Zuteilung von Attributen zu den Bauteilen können wichtige technische Daten, sowie allgemeine Informationen festgehalten werden. Des Weiteren können Relationen zwischen den Bauteilen und einer allgemeinen Theorie über einen Fachbereich, sowie zu Normen und VDI-Richtlinien erstellt werden, was eine schnelle Informationsfindung bietet. Es können jeweilige Relationsarten erstellt werden, die wiederum im Relationsmanager verwaltet werden können. Zukünftig sollte die Möglichkeit bestehen direkt aus der xKBE App CAD-Bauteile und CAD-Baugruppen aufzurufen. Diese Eigenschaften der xKBE App bietet dem Konstrukteur eine Systemumgebung Informationen und Zusammenhänge von Systemen festzuhalten und zu verwalten. Des Weiteren liefert die xKBE App die Möglichkeit anderen und neuen Mitarbeitern eine schnellere und leichtere Einarbeitung eines bereits eingepflegten Systems in der xKBE App.

4.6 Design Structure Matrix für Rollenförderer

Für die Darstellung der Abhängigkeiten von Parametern des Fördertyps Rollenförderer in einer Design Structure Matrix wurden zuerst die Parameter für den Rollenförderer bestimmt, die einzelnen Parameter sind im Anhang unter der Parameternauflistung angeführt. Die Informationen über die Parameter stammen aus der Literatur [INT12a] ; [INT12b] ; [JOD12] .

4.6.1 Erstellung einer DSM für Parameter eines Rollenförderers

Für die Erstellung einer Design Structure Matrix wurde der Cambridge Advanced Modeller verwendet, der in dem Punkt 3.6.5 beschrieben wurde. Die zuvor bestimmten Parameter wurden durch Hinzufügen von „DSM Element“ in die Matrix eingefügt und durch die kleinen markierten Kästchen werden die Abhängigkeiten zwischen den Parametern festgelegt. Die Matrix des Cambridge Advanced Modeller wird nach dem amerikanischen System gelesen. Die Spalte beeinflusst die Zeile, im Gegensatz zu dem europäischen System in der die Zeile die Spalte beeinflusst. Eine bidirektionale Beziehung zwischen zwei Parametern wird durch zwei Kästchen bestimmt, einem unter der Diagonale und einem über der Diagonale. Die Parameter wurden aus Übersichtlichkeitsgründen zuerst manuell nach Umgebungs-, Eingangs-, Rohr-, Modul-, Förderungs-, Antriebs-, Rahmen- und Stützenparameter gruppiert. Die Abbildung 4-14 zeigt die erstellte und manuell gruppierte DSM.

	Umgebung	Eingangsparameter	Rohr	Modul	Förderung	Antrieb	Rahme	Stützen																																						
Umgebung	Umgebungstemperatur	Feuchte Umgebung	Verschmutzung	Lärmentwicklung	Transportgutbreite	Transportgutlänge	Transportguthöhe	Transportgutgewicht	Transportgutmaterial	Fördergeschwindigkeit	Durchsatz	Einschleusvorgang	Ausschleusvorgang	Lastverteilung	Rohrdurchmesser	Rohrwandstärke	Rohrlänge	Rohrwerkstoff	Rohrüberzug	Rohrverzinkung	max. Rohrtraglast	Lichte Weite Gerade	Lichte Weite Kurve	Rollenteilung	Rollenlager	Geradeauslauf	Kippgefahr	Mitnahmekraft	Rundlaufgenauigkeit	Antriebsleistung	Motordrehmoment	Antriebsmedium	Rahmenwerkstoff	Rahmenprofil	max. Rahmentraglast	Stützenwerkstoff	Stützenprofil	Stützenabstand	max. Stützentragslast							
Eingangsparameter																																														
Rohr																																														
Modul																																														
Förderung																																														
Antrieb																																														
Rahme																																														
Stützen																																														

Abbildung 4-14: Design Structure Matrix der Rollenfördererparameter

4.6.2 Analysefunktionen des CAM

Es wurden noch die automatischen Analysefunktionen zum Partitioning, Banding und Clustering, die der Cambridge Advanced Modeller bietet, ausgeführt. In den folgenden Punkten erfolgt die Beschreibung der automatischen Umstrukturierung der Matrizen. Die umstrukturierten DSMs sind in dem Anhang abgebildet.

Partitioning:

Durch die automatische Analysefunktion Partitioning des Cambridge Advanced Modeller werden die Parameterelemente so umstrukturiert, dass fast alle markierten Zellen an einer Seite der Diagonale ausgerichtet sind, es bleiben jedoch einige markierte Zellen übrig, die an der anderen Seite positioniert sind. Die übrig gebliebenen Zellen werden, wie in dem Punkt 3.4.1.2 erwähnt, sehr nahe an der Diagonale ausgerichtet. Die ersten elf Parameterelemente in der Matrix werden durch keine anderen Parameter beeinflusst. Diese Anordnung gibt eine Hilfestellung bei der Konzeption eines neuen Rollenförderers. In dem man die Parameter in der Reihenfolge wie sie in der Matrix angeordnet sind, für die Konstruktionsentscheidungen des Rollenförderers bestimmt. Die markierten Zellen an der anderen Seite gibt die Iterationsabhängigkeit zwischen zwei Parametern an und sollten bei der Festlegung der Parameter als nächstes aufeinander abgestimmt werden, damit die Iterationsschleifen kurz gehalten werden können.

Banding:

Die Bandinganalyse des Cambridge Advanced Modellers, liefert keine Streifen wie im Punkt 3.4.1.2 beschrieben, sondern wie beim Clustering werden quadratische Kästchen erzeugt, es erfolgt auch keine Umstrukturierung der Zeilen und Spalten. Deshalb ist es schwierig daraus Rückschlüsse zu ziehen, welche Parameter parallel bzw. gleichzeitig bestimmt werden können.

Clustering:

Die Analysefunktion Clustering gruppiert die Parameter nach der Abhängigkeitsähnlichkeit und wird laut (vgl. [RS12] , S.698; [MAU07] , S.57) gewöhnlich für komponenten- und teambasierte DSMs eingesetzt, damit Module eines Systems oder ein Arbeitsteam bestimmt werden können. Jedoch liefert das Clustering der parameterbasierten DSM über Rollenförderer, eine effektive Gruppierung der Parameter, die sehr eng miteinander in Beziehung stehen. Bei der Bestimmung der Parameter sollten diese gruppierten Parameter gleichzeitig betrachtet werden.

4.6.3 Parameterdarstellung im Set Visualiser

Zur visuellen Darstellung der Rollenfördererparameter wurde die partitionierte DSM mithilfe des Set Visualiser in einem Schnittmengendiagramm veranschaulicht. Der Cambridge Advanced Modeller bietet die Möglichkeit CSV-Dateien zu exportieren, welche in dem Set Visualiser importiert wurden. Der Set Visualiser wurde im Punkt 3.6.6 beschrieben. Die Abbildung 4-16 zeigt das erstellte Schnittmengendiagramm der Rollenfördererparameter. Die Parameter

die links außerhalb der Blasen angeordnet sind, werden durch keine anderen Parameter beeinflusst. Die Parameter, die sich in denselben Blasen mit denselben Farben befinden, weisen sehr starke Abhängigkeiten zueinander auf.

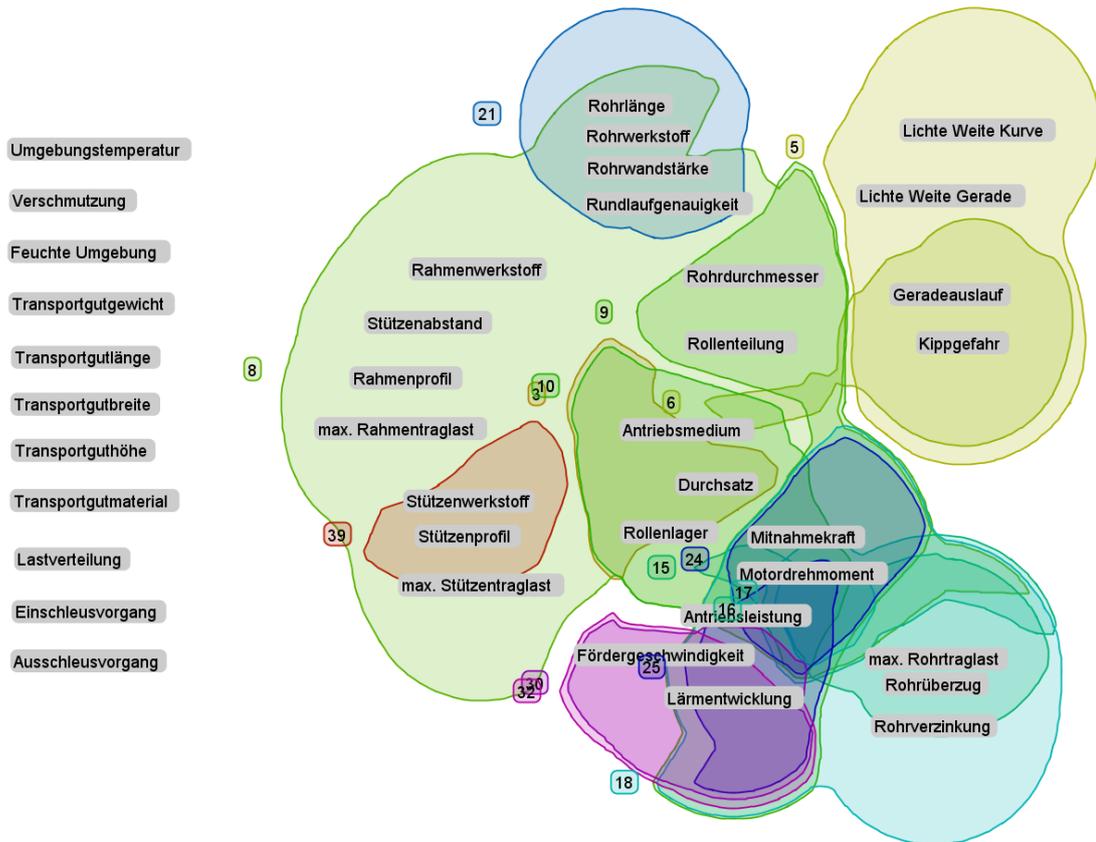


Abbildung 4-16: Darstellung der Rollenfördererparameter mit dem Set Visualiser

5 Rechercheergebnisse der Anwendung von KBE-Methodiken – Erkenntnisse des Anwendungsbeispiels

In diesem Kapitel erfolgen die Ergebnisse einer Internetrecherche, welche eine aktuelle Anwendung der KBE-Methodiken, die im dritten Kapitel beschrieben wurden, untersuchte. Des Weiteren enthält das Kapitel die Erkenntnisse die aus der Erstellung einer Ontologie in der xKBE-App und die Erzeugung einer parameterbasierten DSM, für den Fördertyp Rollenförderer gewonnen werden konnten.

5.1 Anwendung der KBE-Methodiken in der Praxis

Anwendung der MOKA-Methodik

Die MOKA-Methodik wurde anhand eines kleinen Beispiels in dem Artikel [MOK12g] vor Augen geführt. Er bietet leider nur einen sehr begrenzten Einblick in die Praxisanwendung der Methodik. Des Weiteren wurden Testanwendungsfälle mit Industriepartnern durchgeführt, die in dem Artikel [MOK12c] angeführt sind. Bei dem Partner BAE Systems wurde das Fahrwerk eines Flugzeuges als Testfall herangezogen und bei Aerospatiale Matra die Flugzeugrumpfkonstruktion. Eine Wasserpumpenkonstruktion wurde bei dem Unternehmen Daimler Chrysler getestet und bei dem Konzern PSA Peugeot Citroën wurde die Kurbelwellenkonstruktion für die Anwendung von MOKA getestet. Laut dem Bericht [MOK12c] war das Resultat dieser Testanwendungen eine deutliche Verbesserung der Wissensmodellierung und –managements. Nähere Informationen über die Testanwendungsfälle sind nicht publiziert worden. Laut dem Autor [MIL08] wird die MOKA-Methodik von den beiden Konzernen Airbus und Fokker verwendet. Nach einer umfangreichen Internetrecherche scheint aber nirgends auf, dass die MOKA-Methodik zurzeit in der Praxis eingesetzt wird, weder von den Industriepartnern noch von anderen Unternehmen.

Anwendung der KOMPRESSA-Methodik

Die Anwendung der KOMPRESSA-Methodik zielt auf kleine und mittelgroße Projekte und Unternehmen ab. Literatur über diese Methodik ist nur in sehr spärlichem Ausmaß vorhanden. Die Praxisanwendung der Methodik wurde in dem Projekt REFIT, welches im dritten Kapitel erwähnt wurde, untersucht. Durch die Internetrecherche stellte sich heraus, dass eine kommerzielle Verwendung dieses Verfahrens in der Industrie sich zurzeit noch nicht durchgesetzt hat.

Anwendung der DEE-Methodik

Da die DEE-Methodik an der TU Delft im Fachbereich für Flugzeugbau entwickelt wurde, sind die Testanwendungsfälle hauptsächlich für diese Sparte durchgeführt worden. In dem Artikel [RKT02] wird die Methodik für den Konstruktionsprozess eines Flugzeugflügelrumpfes beschrieben und in einem weiteren Artikel [CBT05] wird die Anwendung der DEE an einem Flugzeugheck veranschaulicht. Die Testanwendung wurde auch bei einem Flugzeugquerruder in

dem Artikel [LT05] angeführt. Abseits von der Flugzeugdisziplin kam die DEE-Methodik noch in den Projekten Upwind und Pegasus zum Einsatz.

In dem Projekt Upwind wurden Windturbinen mit einer Leistung von 10-20MW von 2006 bis 2011 erforscht. Dabei wurde die DEE-Methodik für den Konstruktions- und Berechnungsprozess eingesetzt. Nähere Informationen über das Upwind Projekt kann man im Endbericht [UPW12] nachlesen. Das Projekt Pegasus ist in der Automobilindustrie angesiedelt, dabei liegt das Hauptziel des Projektes bei der Entwicklung einer integrierten Entwicklungs- und Konstruktionsumgebung. Europäische Klein- und Mittelbetriebe sollen zusammengeführt werden, damit die Kommunikation untereinander und mit dem Kunden verbessert werden kann. Sie sollen als ein gemeinsamer Anbieter auftreten. Die entwickelte Plattform von Pegasus soll die Integration von Funktionen, Prozessen und Materialien zu einem einzigen Prozess realisieren. Der Zeitrahmen dieses Projektes verlief von 2006 bis 2010. [PEG12] Wie bei den beiden Methodiken zuvor ergab eine Internetrecherche, außer den bereits angeführten Test- und Projektanwendungen, keinen Einsatz der DEE-Methodik in der Industrie.

Anwendung der KNOMAD-Methodik

Die KNOMAD-Methodik wurde wie die DEE-Methodik von der TU-Delft in der Fachdisziplin für Flugzeugbau entwickelt. Daher sind die Anwendungsfälle wieder in diesem Fachbereich angesiedelt. In der Dissertation von [COO11] wurde die KNOMAD-Methodik für die Flügelkonstruktion, die aus Metallfaserlaminaten bestehen, angewendet. In dem Artikel [CVT10] wurde die KNOMAD-Methodik durch das Anwendungsbeispiel eines Flugzeugruders vor Augen geführt. Die Internetrecherche nach weiteren Anwendungsfällen oder der Verwendung der Methodik in der Industrie ergab, dass die KNOMAD-Methodik nur für die beiden zuvor erwähnten Anwendungsfällen eingesetzt wurde.

5.2 Erkenntnisse aus dem Anwendungsbeispiel

Erkenntnisse aus der Ontologieerstellung

Zur Erstellung einer Ontologie über den Fördertyp Rollenförderer mussten zuerst Informationen über diesen Fördertyp gesammelt werden. Aufgrund dieser Problematik stellte sich heraus, dass sehr viele Hersteller für Rollenförderer am Markt existieren, jedoch war das Informationsmaterial darüber nur in sehr geringem Ausmaß vorhanden. Die Firma Interroll bietet eine sehr große Produktpalette von Fördermodulen, sowie ein weites Spektrum von Informationsunterlagen auf ihrer Homepage zur Verfügung. Deshalb wurden die Fördermodule von diesem Hersteller für das Anwendungsbeispiel herangezogen. Die Erstellung einer Ontologie in der xKBE-App stellte sich aufgrund der sehr einfachen Erstellung von Entitäten und ihren Relationen zueinander als sehr benutzerfreundlich heraus. Durch die einfache Handhabung des Relationsmanager können sehr schnell jegliche Relationstypen erzeugt und verändert werden. Das Durchforsten von Dokumenten für die jeweilig gewünschte Information wird durch das anhängen von Dokumenten an die Entitäten vermieden. Der Benutzer der xKBE-App hat auch keine Beschränkung bei der Erstellung einer Ontologie, er kann die Ontologie ganz nach seinen Bedürfnissen erstellen. Der Nachteil einer sol-

chen Ontologiedarstellung ist sicherlich die Unübersichtlichkeit für Benutzer, die die Ontologie nicht selbst erstellt haben.

Erkenntnisse aus der DSM –Erstellung

Für die Bestimmung der Rollenfördererparameter mussten zuerst wieder Informationen über diese Parameter und ihren Abhängigkeiten erarbeitet werden, dabei waren die Informationsunterlagen des Herstellers Interroll sowie die Quelle [JOD12] , sehr hilfreich. Für die Erstellung der DSM über Rollenförderer wurde der Cambridge Advanced Modeller verwendet. Er bietet die Möglichkeit sehr schnell eine Design Structure Matrix zu erzeugen. Die Abhängigkeitsauflistung mithilfe einer DSM liefert eine sehr gute Strukturierungsmöglichkeit der Beziehungen zwischen den Parametern. Bei der Abbildung von sehr umfangreichen Baugruppen oder Anlagen mit einer großen Anzahl von Relationen wird es sehr schwierig den Überblick solcher DSMs zu behalten, sowie die Darstellung der Netzgrafik wird bei sehr vielen DSM-Elementen und Abhängigkeiten unübersichtlich. Die automatischen Analysefunktionen, die der Cambridge Advanced Modeller bietet, liefert bei der Funktion Partioning und Clustering sehr brauchbare Ergebnisse für die Bestimmung der Reihenfolge von den Konstruktionsentscheidungen. Aus der Funktion Banding, des Cambridge Advanced Modellers ist es jedoch schwierig herauszufinden, welche Parameter unabhängig bzw. parallel voneinander bestimmt werden können. Die Darstellung der Rollenfördererparameter in einem Schnittdiagramm mithilfe des Set Visualiser liefert einen groben Überblick der Parameterabhängigkeiten. Es ist nicht genau zu erkennen welche Parameter aufgrund ihrer Abhängigkeiten einer Menge zugeordnet wurden.

6 Bewertung

Dieses Kapitel der Arbeit enthält eine Bewertung der Literatur, der KBE-Methodiken und der Softwaretools für die Wissensrepräsentation, sowie einen Vergleich der KBE-Methodiken und ein Fazit über die Softwaretools. Dabei sei zu erwähnen, dass bei einer Bewertung von nur einer einzelnen Person sich ein gewisser Grad an Subjektivität nicht vermeiden lässt.

6.1 Bewertung der Literatur

Die folgende Matrix bewertet die verwendeten Literaturquellen der Themen KBE-Methodiken, MDE, Expertensysteme, DSM und Axiomatic Design, welche im dritten Kapitel behandelt wurden.

Literaturquellen	Literaturthemen							
	MOKA	KBE-Methodiken		KNOMAD	MDE	Expertensysteme	DSM	Axiomatic Design
		KOMPRESSA	DEE					
[VBD11]	~	-	-	-	-	-		
[MOK12a]	~							
[MOK12b]	~							
[MOK12c]	+							
[MOK12d]	~							
[MOK12e]	+							
[MOK12f]	~							
[LAR11]	+		+			+		
[MIL08]	~		-			+		
[LIB00]		~						
[KOM12]		-						
[COO11]	-		+	+				
[CVT10]	~		+	+		~		
[WEI06]					+			
[ROS12]						+		
[COD12]						+		
[GAA00]						+		
[RS12]							~	
[MAU07]							+	
[DSM12]							+	
[LEY06]							~	+
[CEK00]								+
[CK10]								~

Tabelle 6-1: Bewertungsmatrix der Literaturquellen

+ sehr gut

~.....befriedigend

-.....mangelhaft

■..... in dieser Quelle wird das jeweilige Thema nicht behandelt

6.2 Bewertung der KBE Methodiken

Die KBE-Methodiken, welche im Punkt 3.1 detailliert beschrieben wurden, werden in der folgenden Matrix hinsichtlich einiger Kriterien bewertet, des Weiteren erfolgt ein Vergleich der Methodiken bezüglich ihrer Ähnlichkeiten und Unterschiede.

	KBE-Methodiken			
	MOKA	KOMPRESSA	DEE	KNOMAD
Dokumentation	+	-	+	+
Aktualität	-	-	+	+
Wissenserfassung	+	+	-	+
Theorielastig	-	~	-	-
Multidisziplinärer Ansatz	-	-	+	+
Praxisanwendung	-	-	~	-
EDV-lastig	-	~	-	-
Realisierungsaufwand	-	~	-	-

Tabelle 6-2: Bewertungsmatrix der KBE-Methodiken

+ sehr gut ~.....befriedigend -.....mangelhaft

Vergleich der KBE-Methodiken

Die Methodiken KOMPRESSA und MOKA sind in ihrem Ablauf sehr ähnlich, sie wurden auch ungefähr zum selben Zeitpunkt entwickelt. Die Risikoanalyse ist bei KOMPRESSA gegenüber von MOKA detaillierter ausgeführt. Des Weiteren werden bei KOMPRESSA vermehrt Aktivitätsdiagramme für die organisatorische Leitung eingesetzt. Literatur über die KOMPRESSA-Methodik ist nur in sehr geringem Ausmaß vorhanden, was eine Beurteilung der Methodik erschwert. (vgl. [VBD11])

Die Wissenserfassungs- und Normierungsphase von KNOMAD weisen starke Ähnlichkeiten mit der Phase „Problem- und Möglichkeitenidentifikation“, sowie der Wissenserfassung von MOKA auf. Der Unterschied zwischen MOKA und KNOMAD ist der multidisziplinäre Ansatz von KNOMAD in der Modellierungs- und Analysephase. Zusätzlich bietet KNOMAD gegenüber MOKA in der Normierungsphase eine genauere Qualitätsprozedur bei der Wissenserfassung. (vgl. [CVT10])

Die Ähnlichkeit zwischen den Methodiken KNOMAD und DEE besteht darin, dass KNOMAD den Multi-Model Generator (MMG) von DEE übernommen hat. Zusätzlich wurde die Kommunikationsverknüpfung zwischen der Modellierungs-, Analyse- und Bereitstellungsphase von der DEE-Methodik adaptiert. KNOMAD unterscheidet sich von DEE durch die explizite Wissenserfassung, -normierung und -organisation. Zudem ist die Phase Bereitstellung der KNOMAD-Methodik gegenüber der DEE-Methodik ausgereifter. (vgl. [CVT10])

Der Multi-Model Generator (MMG), der KNOMAD- und DEE-Methodik beinhaltet die gesamte Wissensformalisierung von der MOKA-Methodik, sprich das Formelle Modell von MOKA inklusive der MLL. (vgl. [CVT10] ; [LT04])

6.3 Bewertung der Wissensrepräsentationstools

Die Tabelle 6-3 bewertet die im dritten Kapitel angeführten Softwaretools zur Wissensrepräsentation nach ihren Eigenschaften und Funktionen.

	Softwaretools					
	Compendium	CmapTools	VUE	iMapping	CAM	Set Visualiser
Freeware	+	+	+	+	+	+
Dokumentation	+	+	~	~	~	~
Aktualität	-	+	+	+	+	+
Bedienbarkeit	~	+	+	+	+	+
Graphische Darstellung	~	~	+	+	~	~
Übersichtlichkeit	-	+	+	+	~	+
Suchfunktion	-	-	+	+	-	-
Gruppierungsmöglichkeit	-	+	+	+	+	+
Datenimport	~	~	+	-	+	+
Datenexport	+	~	+	-	+	+
Kollaborationsmöglichkeit	+	+	+	-	-	-
Semantikanalyse	-	-	+	-	-	-

Tabelle 6-3: Bewertungsmatrix der Wissensrepräsentationstools

+ sehr gut ~.....befriedigend -.....mangelhaft

Fazit aus den Wissensrepräsentationstools:

Für die Erstellung von Wissenslandkarten eignet sich aus dem im dritten Kapitel beschriebenen Softwaretools das VUE (Visual Understanding Environment) am besten, da es sehr übersichtlich und leicht zu bedienen ist. Es bietet auch umfangreichere Gestaltungsmöglichkeiten der Wissenslandkarte als die anderen Tools. Des Weiteren unterstützt es den Import von CSV-Dateien. Das VUE bietet als einziges Tool eine automatische Semantikanalyse. Mit dem VUE-Tool besteht auch die Möglichkeit zur Teamarbeit und es enthält auch eine Suchfunktion für die schnelle Auffindung von Entitäten.

Als zweite Wahl würde das CmapTools in Frage kommen. Es ist dem VUE-Tool sehr ähnlich. Es bietet jedoch weniger Gestaltungsmöglichkeiten und beinhaltet keine automatische Semantikanalyse und keine Suchfunktion.

Das Compendium-Tool ist nicht so übersichtlich wie die beiden zuvor erwähnten Tools. Ein weiterer Nachteil ist, dass es im Jahre 2006 das letzte Mal aktualisiert wurde. Das Einsatzfeld des Compendium-Tools liegt nicht so sehr in der Erstellung von Wissenslandkarten, sondern in der computerunterstützten Diskussion von neuen Ideen und Lösungsvarianten.

Das iMapping-Tool besitzt keine Möglichkeit zur Teamarbeit und es können auch keine unterschiedlichen Wissenslandkarten in Verbindung zueinander gesetzt werden. Der Datenimport ist ausschließlich auf Bildformate begrenzt. Positiv zu dem iMapping-Tool sei erwähnt, dass es sehr übersichtlich und leicht zu bedienen ist. Jedoch ist es eher für die Organisation von persönlichen Notizen geeignet.

Der Cambridge Advanced Modeller und der Set Visualiser sind mit den anderen Tools nicht direkt vergleichbar, da sie für eine andere Anwendung konzipiert

sind. Dennoch sind sie in der Bewertungsmatrix aus Vollständigkeitsgründen angeführt.

Der CAM (Cambridge Advanced Modeller) liefert eine schnelle Möglichkeit zur Erstellung einer DSM, jedoch kann damit keine DMM oder MDM erstellt werden. Des Weiteren sind die automatischen Funktionen zum Partitioning, und Clustering, sehr nützlich. Durch die Darstellung der Netzgrafik im CAM können die Abhängigkeiten der Parameter leichter nachvollzogen werden.

Der Set Visualiser bietet nur die Möglichkeit zur graphischen Darstellung einer Schnittmenge von Entitäten. Er beinhaltet außer dem Export des Mengendiagramms in eine Vektorgrafik oder in ein Bildformat keine weiteren Funktionen und ist deshalb nur für diese Anwendung geeignet.

7 Zusammenfassung

Die Arbeit beschäftigte sich mit der „Wissensbasierten Konstruktion“ (engl. KBE). Darunter versteht man die Automatisierung von Konstruktionsschritten aufgrund einer Regelverarbeitung in einem CAD-System. Zunächst wurde die Thematik KBE und die beiden angrenzenden Themen Wissensmanagement und Produktentwicklung grundlegend behandelt. Im Kapitel systematische Analyse zur Verwendung von KBE wurden existierende KBE-Methodiken detailliert erläutert. Des Weiteren wurde auf das Wissensmanagementinstrument Expertensysteme näher eingegangen. Zur methodischen Vorgehensweise bei der Produktentwicklung wurde das Verfahren Axiomatic Design und die matrixbasierten Methoden, im speziellen die Design Structure Matrix behandelt, mit welcher Abhängigkeiten von Bauteilkomponenten oder Parameter innerhalb einer Domäne abgebildet werden können. Zur Wissensrepräsentation wurde im Rahmen dieser Arbeit einige Softwaretools vorgestellt. Im vierten Kapitel wurde ein Anwendungsbeispiel im Bereich der Technischen Logistik dokumentiert, dabei wird die Erstellung einer Ontologie als Wissensrepräsentation über den Fördertyp Rollenförderer beschrieben. Das Anwendungsbeispiel beinhaltet zusätzlich eine Erstellung einer parameterbasierten Design Structure Matrix über Rollenförderer. Der darauffolgende Teil der Arbeit enthält die Ergebnisse einer Recherche die darauf abzielte, zu erfahren ob die KBE-Methodiken zurzeit in der Praxis eingesetzt werden. Das Resultat der Recherche ergab, dass die KBE-Methodiken nur in ein paar Testanwendungsfällen und Forschungsprojekten eingesetzt wurden. Zusätzlich beinhaltet dieser Teil der Arbeit die Erkenntnisse aus dem durchgeführten Anwendungsbeispiel. Bei der Ontologierstellung wurde die institutsentwickelte Software, die den Namen xKBE-App trägt, verwendet. Dabei stellte sich heraus, dass die App dem Anwender jeglichen Spielraum für den Aufbau einer Ontologie bietet, außerdem zeichnete sie sich durch ihre Benutzerfreundlichkeit aus. Die Anwendung einer Design Structure Matrix zeigte eine gute Möglichkeit die Parameterabhängigkeiten zu strukturieren und visuell darzustellen. Im letzten Teil der Arbeit erfolgte eine Bewertung der Literatur, die im Rahmen dieser Arbeit verwendet wurden. Zusätzlich enthält dieser Teil eine Beurteilung und einen Vergleich der KBE-Methodiken, welcher zum Vorschein brachte, dass die KBE-Methodiken in ihrer Systematik sehr ähnlich sind. Des Weiteren beinhaltet der letzte Punkt eine Bewertung der Wissensrepräsentationstools mit einem anschließenden Fazit der Softwaretools.

8 Verzeichnisse

8.1 Literaturverzeichnis

[ALC12] Alcyon (Hrsg.): URL

<http://www.alcyon.co.in/whatiskbe.pdf> - Abrufdatum: 07.11.2012.

[CAM12] Cambridge Advanced Modeller (Hrsg.): URL

<http://www-edc.eng.cam.ac.uk/cam> - Abrufdatum: 19.10.2012

[CBT05] C., Cerulli; J.P.T.J., Berendes; M.J.L., van Tooren; J.W., Hofstee: *Parametric Modeling for Structural Dynamics Investigations in Preliminary Design*. 2005.

[CEK00] David S., Cochran; Walter, Eversheim; Gerd, Kubin; Marc L., Sesterhenn: *The Application of Axiomatic Design and Lean Management Principles in the Scope of Production System Segmentation*. In: The International Journal of Productuin Research, Vol-38 No. 6. 2000.

[CHL02] Alberto J., Canas; Greg, Hill; James, Lott: *Support for Knowledge Models in CmapTools*. Technical Report IHMC CmapTools. 2002

[CK10] Selcuk, Cebi; Cengiz, Kahrama: *Extensions of axiomatic design principles under fuzzy environment*. In: Experts Systems with Applications. 37, 2010.

[COD12] Codeproject (Hrsg.): URL

<http://www.codeproject.com/Articles/43474/Knowledge-Based-Systems-with-the-CommonKADS-Method#kbs1.1> - Abrufdatum: 11.10.2012.

[COM12] CompendiumInstitute (Hrsg.): URL

<http://compendium.open.ac.uk/institute/index.htm> - Abrufdatum: 17.10.2012.

[COO11] Cory A., Cooper: *Development of a Methodology to Support Design of Complex Aircraft Wings*. Dissertation, TU Delft, Fakultät Aerospace Engineering, 2011.

[CVT10] Richard, Curran; Wim J.C., Verhagen; Michel J.L., van Tooren; Ton. H., Van der Laan: *A multidisciplinary implementation methodology for knowledge based engineering: KNOMAD*. In: Expert Systems with Applications 37. 2010.

[DSM12] dsmweb.org (Hrsg.): URL

<http://www.dsmweb.org/> - Abrufdatum: 24.10.2012.

[EDC12] EDC Engineering Design Center (Hrsg.): URL

http://www-edc.eng.cam.ac.uk/tools/set_visualiser/ - Abrufdatum: 19.10.2012.

[EM07] C.L., Emberey; N.R., Milton: *Application of Knowledge Engineering methodologies to Support Engineering Design Application Development Aerospace*. In: 7th AIAA Aviation Technology, Integration and Operations Confernece (ATIO). 2007.

[ES09] Martin, Eigner; Ralph, Stelzer: *Product Lifecycle Management Ein Leitfaden für Product Development und Lifecycle Management*, Dordrecht/ Heidelberg/London/New York: Springer, 2009. - ISBN 978-3-540-68401-5.

[FAA04] Faatz, Andreas: *Ein Verfahren zur Anreicherung fachgebietsspezifischer Ontologien durch Begriffsvorschläge*. Dissertation, TU Darmstadt, Fakultät Informatik, 2004.

[GAA00] Guus, Schreiber: *Knowledge Engineering and Management The CommonKADS Methodology*. Cambridge/Massachusetts/London: The MIT Press, 2000. - ISBN 0-262-19300-0.

- [GAA10] Andreas, Gaag: *Entwicklung einer Ontologie zur funktionsorientierten Losungssuche in der Produktentwicklung*. Dissertation, TU München, Lehrstuhl für Produktentwicklung, 2010.
- [GB04] M. D., Guenov; S.G., Barker: *Application of Axiomatic Design and Design Structure Matrix to the Decomposition of Engineering Systems*. 2004.
- [GRU95] Thomas R., Gruber: *Toward principles for the design of ontologies used for knowledge sharing*. In: International Journal of Human Computer Studies. 1995.
- [HAL11] Heiko, Haller: *User Interfaces for Personal Knowledge Management with Semantic Technologies*. Dissertation, Karlsruher Instituts für Technologie, Fakultät für Wirtschaftswissenschaften, 2011.
- [HSA12] Hochschule für angewandte Wissenschaften Augsburg (Hrsg.): URL <http://www.hs-augsburg.de/~schmidmi/forschung/wissensbasiertekonstruktion/index.html> - Abrufdatum: 14.11.2012.
- [HSN07] Michael, ten Hompel; Thorsten, Schmid; Lars, Nagel: *Materialflusssystem Förder- und Lagertechnik*, Berlin/Heidelberg/New York: Springer, 2007. - ISBN 978-3-540-73235-8.
- [IMA12] iMapping (Hrsg.): URL <http://www.imapping.info/> - Abrufdatum: 19.10.2012.
- [INT12a] Interroll (Hrsg.): URL http://www.interroll.com/media/editorial_media/interroll_group/downloads/foerdermodule_subsysteme/foerdermodule_katalog/2012-02-20_Automation_web.pdf - Abrufdatum: 18.09.2012.
- [INT12b] Interroll (Hrsg.): URL http://www.interroll.com/media/editorial_media/interroll_group/downloads/antriebe_foerderrollen/drivers_rollers/conveyor_rollers_catalogue/FR_DE_2011_WEB.pdf - Abrufdatum: 18.09.2012.
- [JOD12] Dirk, Jodin: *Materialflusstechnik (Skriptum zur Vorlesung)*. Graz, Institut für Technische Logistik. 2012.
- [KMI12] KMI Knowledge Media Institute (Hrsg.): URL http://people.kmi.open.ac.uk/selvin/examples/history/A_Brief_History_of_Compendium_12820925921132166278811.html1162662646610.html - Abrufdatum: 17.10.2012.
- [KOM12] Slidefinder (Hrsg.): URL http://www.slidefinder.net/k/kbe_design_methodology/342des-2bkbedesignmethodology/9309657 - Abrufdatum: 03.07.2012.
- [LAR11] Gianfranco, La Rocca: *Knowledge Based Engineering Techniques to Support Aircraft Design Optimization*. Dissertation, TU Delft, Fakultät Aerospace Engineering, 2011.
- [LEE05] Helen, Leemhuis: *Funktionsgetriebene Konstruktion als Grundlage verbesserter Produktentwicklung*. Dissertation, TU Berlin, Fakultät V für Verkehrs- und Maschinensysteme, 2005.
- [LEY06] Gerhard Leykauf: *Modularität und vertikale Desintegration Gütertechnik als Determinante der Wertschöpfungstiefe*. Dissertation, Universität Erlangen-Nürnberg, 2006.
- [LIB00] P.J., Lovett; A., Ingram; C.N., Bancroft: *Knowledge-based engineering for SMEs-a methodology*. In: Journal of Materials Processing Technology 107. 2000.

- [LJ12] Christian, Landschützer; Dirk, Jodin: *Knowledge-Based Engineering (KBE) für Hubseiltrommeln*. - 20. Kranfachtagung Veranstalter TU Dresden, Veranstaltungsort Dresden 21. März 2012.
- [LT04] A.H., van der Laan; M.L.J., van Tooren: *Parametric Modeling Of Moveables For Structural Analysis*. In: 45th Structures, Structural Dynamics and Material Conference. 2004.
- [LT05] A.H., van der Laan; M.J.L., van Tooren: *Manufacturability analysis of aircraft movables in a multi disciplinary design environment*. In: 1st International Conference on Innovation and Integration in Aerospace Sciences. 2005.
- [MAR06] Martin, Heinrich: *Transport- und Lagerlogistik Planung, Struktur, Steuerung und Kosten von Systemen der Intralogistik*. Wiesbaden: Vieweg, 2006. - ISBN 3-8348-0168-2.
- [MAU07] Maik S., Maurer: *Structural Awareness in Complex Product Design*. Dissertation, TU München, Lehrstuhl für Produktentwicklung, 2007.
- [MIL08] Nick, Milton: *Knowledge Technologies*. Monza/Milan: Polimetrica, 2008. - ISBN 987-88-7699-100-4.
- [MMA08] Jaby, Mohammed; Jared, May; Ali, Alavi: *Application of Computer-Aided Design (CAD) In Knowledge Based Engineering*. In: Proceedings of The 2008 IAJC-IJME International Conference. 2008.
- [MOK12a] MOKA (Hersg.): URL
<http://web1.eng.coventry.ac.uk/moka/project.htm> - Abrufdatum: 11.05.2012.
- [MOK12b] MOKA (Hersg.): URL
<http://web1.eng.coventry.ac.uk/moka/Documents/reports/publicreport99.pdf> - Abrufdatum: 12.05.2012.
- [MOK12c] MOKA (Hrsg.): URL
<http://web1.eng.coventry.ac.uk/moka/Documents/consortium/d4-3.pdf> - Abrufdatum: 30.05.2012.
- [MOK12d] MOKA (Hrsg.): URL
<http://web1.eng.coventry.ac.uk/moka/Documents/Papers/cirp.pdf> - Abrufdatum: 18.06.2012.
- [MOK12e] MOKA (Hrsg.): URL
http://web1.eng.coventry.ac.uk/moka/Documents/consortium/mig_def.pdf - Abrufdatum: 25.06.2012.
- [MOK12f] MOKA (Hrsg.): URL
<http://web1.eng.coventry.ac.uk/moka/Documents/Papers/ekaw2000.pdf> - Abrufdatum: 25.06.2012.
- [MOK12g] MOKA (Hrsg.): URL
<http://web1.eng.coventry.ac.uk/moka/Documents/consortium/d3-4.pdf> - Abrufdatum: 26.09.2012.
- [MRW08] Heinrich, Martin; Peter, Römisch; Andreas, Weidlich: *Materialflusstechnik Auswahl und Berechnung von Elementen und Baugruppen der Fördertechnik*, Wiesbaden: Vieweg, 2008. - ISBN 978-3-8348-0313-9.
- [MUL07] Stefan, Muller: *Methodik für die entwicklungs- und planungsbegleitende Generierung und Bewertung von Produktionsalternativen*. Dissertation, TU München, Lehrstuhl für Werkzeugmaschinen und Fertigungstechnik, 2007.
- [PEG12] Pegasus (Hersg.): URL
<http://www.pegasus-eu.net/language/de-DE/Home.aspx> - Abrufdatum: 16.07.2012

- [RKT02] G., La Rocca; L., Krakers; M.J.L., van Tooren: *Development of an ICAD Generative Model for Blended Wing Body Aircraft Design*. In: 9th AIAA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, 2002.
- [ROS12] Mirko, Rose: *Entwicklung eines Expertensystems zur automatischen Erstellung effizienter Information-Extraction-Pipelines*. Diplomarbeit, Universität Paderborn, Fakultät Informatik, 2012.
- [RS12] Frank, Rieg; Rolf, Steinhipler: *Handbuch Konstruktion*. München/Wien: Hanser, 2012. - ISBN 978-3-446-43403-5.
- [SCH06] Stefan, Schulte: *Integration von Kundenfeedback in die Produktentwicklung zur Optimierung der Kundenzufriedenheit*. Dissertation, Ruhr-Universität Bochum, Fakultät Maschinenbau, 2006.
- [STA12] Stilian, Stanev: *Methodik zur produktionsorientierten Produktanalyse für die Wiederverwendung von Produktionssystemen - 2REUSE*. Dissertation, Karlsruher Instituts für Technologie, Fakultät Maschinenbau, 2012.
- [STE09] Florian, Stendel: *VEsonomy - Konzeption und Entwicklung eines Social-Tagging-Systems für den Virtual Engineering Prozess*. Diplomarbeit, Universität Magdeburg, Fakultät Informatik, 2009.
- [UPW12] UpWind (Hrsg.): URL
http://www.ewea.org/fileadmin/ewea_documents/documents/upwind/21895_UpWind_Report_low_web.pdf - Abrufdatum: 16.07.2012.
- [VBD11] Wim J.C., Verhagen; Pablo, Bermell-Garcia; Reinier E.C., van Dijk; Richard, Curran: *A critical review of Knowledge-Based Engineering: An identification of research challenges*. In: Advanced Engineering Informatics 26, 2011.
- [VDI09] VDI-FML (Hrsg.): *VDI-Richtlinie 5610 - Wissensmanagement im Ingenieurwesen Grundlagen, Konzepte, Vorgehen*. VDI-Handbuch Informationstechnik Band 1, Berlin : Beuth Verlag, 2009.
- [VDI70] VDI-FML (Hrsg.): *VDI-Richtlinie 2411 - Begriffe und Erläuterungen im Förderwesen*. VDI/AWF-Handbuch Förderwesen, Berlin : Beuth Verlag, 1970.
- [VDI93] VDI-FML (Hrsg.): *VDI-Richtlinie 2221 - Methodik zum Entwickeln und Konstruieren technischer Systeme und Produkte*. VDI-Handbuch Konstruktion, Berlin : Beuth Verlag, 1993.
- [VUE12] VUE (Hrsg.): URL
<http://vue.tufts.edu/> - Abrufdatum: 18.10.2012.
- [WEI06] Bendikt, Weismann: *Architekturzentrierte Modellgetriebene Softwareentwicklung - Fallbeispiel und Evaluierung*. Diplomarbeit, TU Wien, Institut für Softwaretechnik und Interaktive Systeme, 2006.
- [WIK12] Wikipedia (Hrsg.): URL
http://de.wikipedia.org/wiki/Axiomatic_Design - Abrufdatum: 29.10.2012.

8.2 Abbildungsverzeichnis

Abbildung 2-1: Erreichbare Zeiteinsparung durch KBE –Einsatz (angelehnt an [VBD11])	7
Abbildung 2-2: KBE-System Entwicklungszeit versus herkömmliche Konstruktionszeit (angelehnt an [EM07])	7
Abbildung 2-3: Wissenstreppe von North ([LEE05], S.11)	9
Abbildung 2-4: Operative Wissensselemente des Wissensmanagements ([MUL07], S.18)	10
Abbildung 2-5: Semiotisches Dreieck ([GAA10], S.55)	11
Abbildung 2-6: Beispiel einer Ontologie ([STE09], S.29)	12
Abbildung 2-7: Einstufung der Produktentwicklung im Produktlebenszyklus ([LEE05], S7)	13
Abbildung 2-8: Vorgehensmodell nach VDI 2221 ([GAA10], S.17)	14
Abbildung 2-9: Erweiterte Integrationstiefe durch PLM ([ES09], S.37)	15
Abbildung 3-1: Hauptphasen des KBE-Lebenszyklus (angelehnt an [LAR11], S.243)	18
Abbildung 3-2: Spezifizierung der vier Hauptphasen (angelehnt an [LAR11], S.246)	20
Abbildung 3-3: Die Wissensverarbeitungskette von dem informellen zum formellen Modell (angelehnt an [MOK12d])	22
Abbildung 3-4: Beziehungen zwischen dem MML Produktansichten (angelehnt an [MOK12e])	24
Abbildung 3-5: Strukturansicht (angelehnt an [MOK12e])	24
Abbildung 3-6: Funktionsansicht (angelehnt an [MOK12e])	25
Abbildung 3-7: Verlaufsansicht (angelehnt an [MOK12e])	25
Abbildung 3-8: Technologieansicht (angelehnt an [MOK12e])	26
Abbildung 3-9: Repräsentationsansicht (angelehnt an [MOK12e])	26
Abbildung 3-10: Aktivitätsgruppen von KOMPRESSA (angelehnt an [KOM12])	30
Abbildung 3-11: Prozessablauf der DEE (angelehnt an [LAR11], S.43)	34
Abbildung 3-12: DEE Agenten-System (angelehnt an [LAR11], S.44)	35
Abbildung 3-13: KNOMAD Phasen (angelehnt an [CVT10])	39
Abbildung 3-14: Modellgetriebene Softwareentwicklung ([WEI06], S.8)	41
Abbildung 3-15: Architektur eines Expertensystems ([ROS12], S.7)	44
Abbildung 3-16: Entwicklung der KBS (angelehnt an [COD12])	44
Abbildung 3-17: Modellreihenfolge (angelehnt an [GAA00], S.18)	45
Abbildung 3-18: Prozesspersonen (angelehnt an [GAA00], S.21)	47
Abbildung 3-19: Design Structure Matrix [DSM12]	50
Abbildung 3-20: Partitioning einer DSM [DSM12]	51
Abbildung 3-21: Banding einer DSM [DSM12]	52
Abbildung 3-22: Clustering einer DSM [DSM12]	53
Abbildung 3-23: Domain Mappig Matrix [DSM12]	53
Abbildung 3-24: MDM in Anlehnung an [DSM12]	54
Abbildung 3-25: Domänen des Axiomatic Design (angelehnt an [GB04])	55
Abbildung 3-26: Vorgehensweise „zigzagging“ (angelehnt an [LEY06], S.111) ..	57
Abbildung 3-27: Ablaufdiagramm des Axiomatic Design (angelehnt an [CK10]) ..	58
Abbildung 3-28: Literaturanalyse mit Compendium [COM12]	60

Abbildung 3-29: Wissensmodell über den Mars [CHL02]	61
Abbildung 3-30: Concept Map des U-Bahn-Netzes in Boston [VUE12]	62
Abbildung 3-31: Wissenslandkarte eine Quadrocopter ([HAL11] , S.137)	63
Abbildung 3-32: DSM und Netzgrafik eines Staubsaugers [CAM12]	64
Abbildung 3-33: Mengendiagramm von Früchte- und Gemüsesorten [EDC12]	65
Abbildung 4-1: Einteilung der Stetigförderer nach konstruktiven und funktionalen Gesichtspunkten (angelehnt an [MAR06] , S.130).....	67
Abbildung 4-2: Einteilung der Stetigförderer nach der Art des Transportgutes (angelehnt an [MAR06] , S.130)	67
Abbildung 4-3: Schüttguttransport ([JOD12] , S.145).....	68
Abbildung 4-4: Stückguttransport ([JOD12] , S.145)	69
Abbildung 4-5: Rollenbahn und Rollenbankurve ([HSN07] S.131; [JOD12] S.262)	71
Abbildung 4-6: Prinzipskizze der Förderstrecke	72
Abbildung 4-7: Ontologie der Förderstrecke	74
Abbildung 4-8: Ontologie über Rollenförderer	75
Abbildung 4-9: Attributserstellung	76
Abbildung 4-10: Dokumentenanheftung	76
Abbildung 4-11: Relationsverwaltung.....	77
Abbildung 4-12: Stücklistenanzeige	77
Abbildung 4-13: Klassenbrowser	77
Abbildung 4-14: Design Structure Matrix der Rollenfördererparameter	79
Abbildung 4-15: Netzgrafik der Rollenfördererparameter	80
Abbildung 4-16: Darstellung der Rollenfördererparameter mit dem Set Visualiser.....	82

8.3 Tabellenverzeichnis

Tabelle 6-1: Bewertungsmatrix der Literaturquellen	86
Tabelle 6-2: Bewertungsmatrix der KBE-Methodiken	87
Tabelle 6-3: Bewertungsmatrix der Wissensrepräsentationstools	88

9 Anhang

Der Anhang enthält die technischen Datenblätter der Fördermodule für die erstellte Rollenfördererstrecke, die Parameterauflistung zur Erstellung einer DSM über einen Rollenförderer und die umstrukturierten DSMs, die mit den automatischen Analysefunktionen des Cambridge Advanced Modellers erzeugt wurden.

9.1 Technische Datenblätter

Die folgenden Seiten enthalten die technischen Datenblätter der Fördermodule, die zur Erstellung der Förderstrecke verwendet wurden.

Fördermodul: Gerade

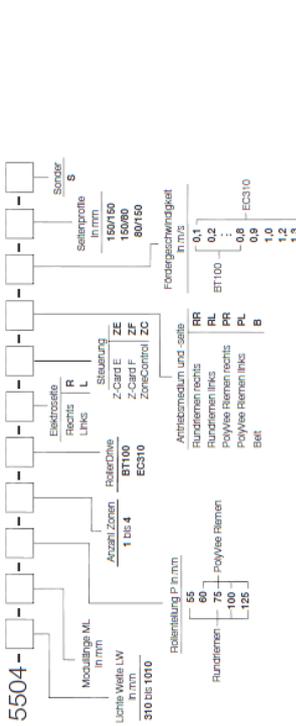


Die interne Steuerung der Rollenbahn sorgt für den staudrucklosen Transport der Stückgüter. Jede Zone wird über eine RollerDrive angetrieben, die über Rund- oder PolyVee Riemen mit einer festen Anzahl von Mitlaufrollen verbunden ist.

Bestellinformation

- Modul ist komplett montiert mit Steuerung und Sensoren
- Stützen, Endkappen und 24-V-Netzteil bitte separat bestellen
- Stahlteile sind pulverbeschichtet (in schwarz RAL 9005 bzw. grau RAL 7030) oder verzinkt

Bitte stellen Sie die Bestellnummer mithilfe des nachfolgenden Konfigurators zusammen.



Zur Auswahl der RollerDrive und der Steuerung siehe in den Anwendungshinweisen ab S. 122. Als Antriebsmedium ist ebenfalls ein Fördergurt (Belt) erhältlich. Siehe dazu in den Anwendungshinweisen ab S. 122. Wenn Sie eine vom Standard abweichende Ausführung benötigen, fügen Sie am Ende ein „S“ hinzu und beschreiben Sie Ihre Anforderungen.

Der Intelliveyor RM 5504 ist auf Anfrage auch als Schrägrollenbahn zum einseitigen Ausrichten des Förderguts erhältlich.

Beispiel für eine Bestellnummer: 5504-510-2000-100-4-EC310-R-ZE-RL-0,2-150/150

Diese Bestellnummer steht für Intelliveyor RM 5504 mit der lichten Weite LW 150 mm, der Modullänge ML 2000 mm, der Rollenlänge P 100 mm, 4 Zonen, einer RollerDrive EC310, der Elektronik rechts, einer Z-Card E, als Antrieb einen Rundriemen links, einer Fördergeschwindigkeit von 0,2 m/s und Seitenprofilen mit den Maßen 150/150 mm.

Zubehör

- Stützen, siehe S. 108
- Endkappen, siehe S. 114
- 24-V-Netzteil, siehe S. 115
- Interface-Karte, siehe S. 116
- SPS-Verbindungskabel, siehe S. 116
- Lichtschrankenkette, siehe S. 117

Unterstützung bei der Planung erhalten Sie in den Anwendungshinweisen ab S. 122.

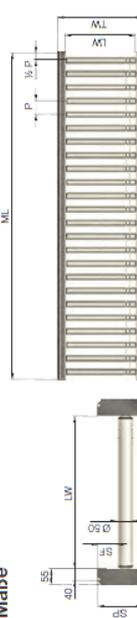


INTERROLL
INTELLIVEYOR
RM 5504

Technische Daten

Allgemeine technische Daten	Rundriemen	PolyVee Riemen
Max. Traglast pro Zone	50 kg	80 kg
Fördergeschwindigkeit	0,1 bis 1,3 m/s	
Steuerung/Cs-Steuerung	Nicht geeignet	
Umgebungstemperatur	+5 bis +40 °C	
Rolle	Interrail Serie 1700	Interrail Serie 3500
Rolldurchmesser	60 mm	60 mm
Rollenmaterial	Stahl, verzinkt	20
Antrieb	Max. Anzahl Rollen pro Zone	11
Nennspannung	24 V DC	
Max. elektrische Leistung pro Zone	0,05 kW	
Antriebsmedium	PUL-Rundriemen o 5 mm	PolyVee Riemen
Übertrieb	Roller-Rolle	
Seitenprofil	Profil 150	150 mm Höhe, 66 mm über Rollenkante
	Profil 80	Erlaubt seitliches Abschieben 80 mm Höhe, 5 mm unter Rollenkante
	Kombination der Profile links/rechts	150/150 160/160 80/150

Maße



Abmessungen	310 bis 1010 mm in 50-mm-Schritten
LW Lichte Weite	ZL x Zonenanzahl, max. 3.000 mm
ML Modullänge	ZL Zonenlänge
ZL Zonenlänge	LW + 110 mm
P Rollenlänge	60 / 75 / 100 / 125 mm
SP Seitenprofil	80 / 150 mm
SF Seitenführung	5 / 65 mm

Fördermodul: Kurve



INTERROLL
INTELLIVEYOR
RM 5530



Rollenförderer
Intelliveyor
RM 5530
Kurve

Die Rollenbahnkurven ändern die Transportrichtung von Fördergütern. Durch konische Rollen wird die Ausrichtung der Fördergüter zwischen den Seitenwangen beibehalten.

Technische Daten

Allgemeine technische Daten	
Max. Traglast pro Zone	50 kg
Fördergeschwindigkeit	0,1 bis 1,3 m/s
Steigung/Cosfälle	Nicht geeignet
Umgebungstemperatur	+5 bis +40 °C
Rolle	
Rollentyp	Interroll Serie 1700KXO Interroll Serie 3500KXO
Rollendurchmesser	50 mm
Rollermaterial	Stahl, verzinkt mit konischen KU-Elementen
Max. Anzahl Rollen pro Zone	g
Antrieb	
Nennspannung	24 V DC
Max. elektrische Leistung pro Zone	0,05 kW
Antriebsmedium	PU-Rundriemen ø 5 mm PolyVee Riemen
Übertrieb	Rolle-Rolle
Seitenprofil	
Profil 150	150 mm Höhe, 65 mm über Rollenkante
Kombination der Profilhöhen links/rechts	
150/150	

Maße



Abmessungen	310 bis 1 010 mm in 50-mm-Schritten
LW Lichtwele	LW + 110 mm
TW Modulbreite	30° / 45° / 60° / 90°
alpha Winkel	-f(0,087 mm x LW) + P1
P Rollenteilung, außen	-72 mm
P1 Rollenteilung, innen	150 mm
SP Saitenprofil	65 mm
SF Saitenführung	

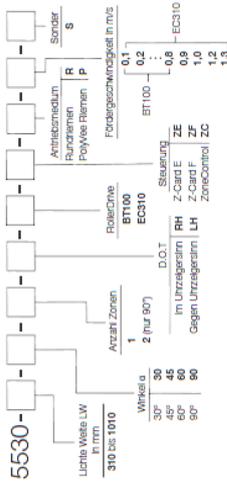
Intelliveyor Übersicht S. 10

Gurtförderer S. 86

Zubehör S. 106

Bestellinformation

- Modul ist komplett montiert mit Steuerung und Sensoren
 - Stützen, Endkappen und 24-V-Netzteil bitte separat bestellen
 - Stahlteile sind pulverbeschichtet (in schwarz RAL 9005 bzw. grau RAL 7030) oder verzinkt
- Bitte stellen Sie die Bestellnummer mithilfe des nachfolgenden Konfigurators zusammen.



Zur Auswahl der RollerDrive siehe in den Anwendungshinweisen ab S. 122.

Wenn Sie eine vom Standard abweichende Ausführung benötigen, fügen Sie am Ende ein „S“ hinzu und beschreiben Sie Ihre Anforderungen.

Beispiel für eine Bestellnummer: 9530-510-90-2-LH-EC310-R-0.2

Diese Bestellnummer steht für Interroll Intelliveyor RM 5530 mit der lichten Weite LW 510 mm, dem Winkel alpha 90°, 2 Zonen, einer Laufrichtung gegen den Uhrzeigersinn, der RollerDrive EC310, Antriebsmedium Rundriemen und einer Fördergeschwindigkeit von 0,2 m/s.

Zubehör

- Stützen, siehe S. 106
- Endkappen, siehe S. 114
- 24-V-Netzteil, siehe S. 115
- Interface-Karte, siehe S. 116
- SPS-Verbindungskabel, siehe S. 116
- Lichtschrankenkit, siehe S. 117

Bestellbeispiel

Unterstützung bei der Planung erhalten Sie in den Anwendungshinweisen ab S. 122.

Fördermodul: Schrägeinschleuser



INTERROLL INTELLIVEYOR TRANSFER RM 5550

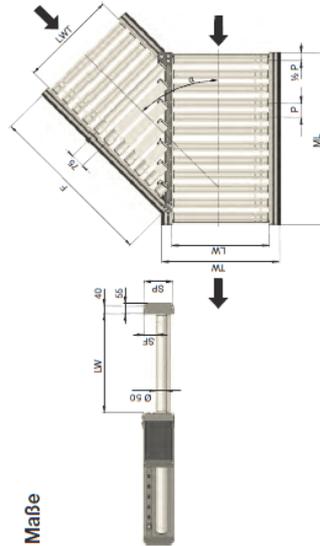


Die Intelliveyor Einschleuserrollenbahn führt zwei Förderlinien zusammen und nutzt dazu die Lücken im Förderfluss.

Technische Daten

Allgemeine technische Daten	Rundriemen	PolyVee Riemen
Max. Transport pro Zone	35 kg	50 kg
Fördergeschwindigkeit	0,1 bis 1,3 m/s	Nicht geeignet
Steigung/Gefälle	Nicht geeignet	+5 bis +40 °C
Umgebungstemperatur		
Rolle	Interroll Serie 1700	Interroll Serie 3500
Rolldurchmesser	50 mm	50 mm
Rollenmaterial	Stahl, verzinkt	
Antrieb		
Nennspannung	24 V DC	
Max. elektrische Leistung pro Zone	0,05 kW	
Antriebsmedium	PU-Rundriemen ø 5 mm	PolyVee Riemen
Übertrieb	Rolle-Rolle	
Seitenprofil		
Profil 150	150 mm Höhe, 65 mm über Rollenoberkante	150/150
Kombination der Profiltypen links/rechts		

Maße



Abmessungen

LW	Lichte Weile	310 bis 1 010 mm in 50-mm-Schritten
LWT	Lichte Weile Transfer	310 bis 1 010 mm in 50-mm-Schritten
ML	Max. Modullänge	1500 mm
TW	Modulbreite	LW + 110 mm
SP	Spurweite	Siehe Anwendungshinweise
P	Polenstellung	30°/45°
SF	Seitenprofil	75 mm
SF	Seitenführung	150 mm
SF	Seitenführung	65 mm

Intelliveyor Übersicht S. 10

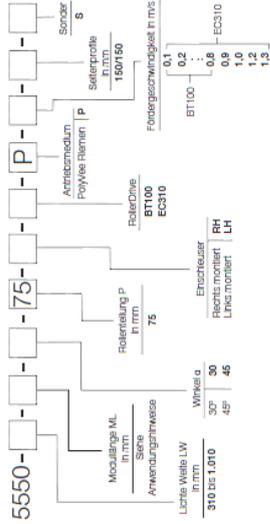
Gurtförderer S. 86

Zubehör S. 106

Bestellinformation

- Modul ist komplett montiert mit Steuerung und Sensoren
- Stützen, Endkappen und 24-V-Netzteil bitte separat bestellen
- Stahlteile sind pulverbeschichtet (in schwarz RAL 9005 bzw. grau RAL 7030) oder verzinkt

Bitte stellen Sie die Bestellnummer mithilfe des nachfolgenden Konfigurators zusammen.



Lieferumfang

Konfigurator

Bestellbeispiel

Zur Auswahl der RollerDrive siehe in den Anwendungshinweisen ab S. 122. Wenn Sie eine vom Standard abweichende Ausführung benötigen, fügen Sie am Ende ein „S“ hinzu und beschreiben Sie Ihre Anforderungen.

Beispiel für eine Bestellnummer: 5550-510-1200-45-100-RH-EC310-P-0,2-150/150

Diese Bestellnummer steht für Intelliveyor Transfer RM 5550 mit der lichten Weile LW 510 mm, der Modulänge ML 1200 mm, einem Winkel α von 45°, der Rollenlänge P 100 mm, rechts montiertem Einschleuser, einer RollerDrive EC310, dem Antriebsmedium PolyVee Riemen, einer Fördergeschwindigkeit von 0,2 m/s und Seitenprofilen mit den Maßen 150/150 mm.

Zubehör

- Stützen, siehe S. 108
- Endkappen, siehe S. 114
- 24-V-Netzteil, siehe S. 115
- Interface-Karte, siehe S. 116
- SPS-Verbindungskabel, siehe S. 116
- Lichtschranke, siehe S. 117

Unterstützung bei der Planung erhalten Sie in den Anwendungshinweisen ab S. 122.

Fördermodul: Riemeneinschleuser



INTERROLL
INTELLIVEYOR
TRANSFER RM 5591



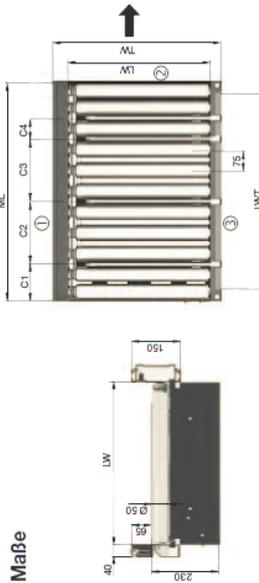
Rollenförderer
Intelliveyor
RM 5591
Transfer

Der Intelliveyor Transfer 90° leitet Fördergüter von einer geraden Strecke im rechten Winkel um und kann sowohl zum Einschleusen als auch zum Ausschleusen genutzt werden. Der elektrische Hub der Transferriemen gewährleistet einen sicheren Transport.

Technische Daten

Allgemeine technische Daten	
Max. Traglast pro Zone	50 kg
Fördergeschwindigkeit	0,1 bis 1,3 m/s
Stütz- und Gestellmaterial	Nicht geölt
Umgebungstemperatur	-5 bis +40 °C
Rolle	
Rollentyp	Interroll Serie 3500
Rollendurchmesser	50 mm
Rollenmaterial	Stahl, verzinkt
Antrieb (Rollerbahn)	
Nennspannung (RollerDrive)	24 V DC
Max. elektrische Leistung pro Zone	0,05 kW
Antriebsmedium	Polyäther-Riemen
Übertrieb	Roller-Rolle
Antrieb Transfer (Riemen und Hub)	
Nennspannung (RollerDrive)	24 V DC
Max. elektrische Leistung	2,05 kW
Kraftübertragung Transfer	Zahnriemen TS
Hubhöhe	15 mm über Rolleroberkante

Maße



Abmessungen

LW	Lichte Weile
LWT	Lichte Weile Transfer
ML	Modüllänge
LWT + 80 mm	LWT + 80 mm
TW	Modüllänge
LW + 95 mm	LW + 95 mm
α	Winkel
C1	Erster Riemenabstand
C2	Riemenabstand
C3	Riemenabstand
C4	Riemenabstand
C5	Riemenabstand

Intelliveyor Übersicht S. 10

Gurtförderer S. 86

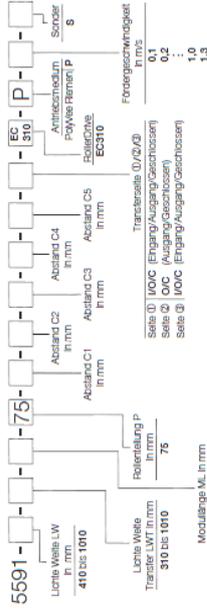
Zubehör S. 106

20

Bestellinformation

- Modul ist komplett montiert mit Steuerung und Sensoren
- Stützen, Endkappen und 24-V-Netzteil bitte separat bestellen
- Stahlteile sind pulverbeschichtet (in schwarz RAL 9006 bzw. grau RAL 7030) oder verzinkt

Bitte stellen Sie die Bestellnummer mithilfe des nachfolgenden Konfigurators zusammen.



Lieferumfang

Konfigurator

Wenn Sie eine vom Standard abweichende Ausführung benötigen, fügen Sie am Ende ein „S“ hinzu und beschreiben Sie Ihre Anforderungen.

Beispiel für eine Bestellnummer: 5591-510-710-700-75-105-225-225-75-C1/O/O-EC310-P-0,6

Diese Bestellnummer steht für Interroll Intelliveyor Transfer 5591 mit der lichten Weile LW 510 mm, der lichten Transferweite LWT 710 mm, der Modüllänge ML 700 mm, der Rollenlänge P 75 mm, dem Abstand C1 bis zum ersten Riemen von 135 mm, den jeweiligen Abständen zu den nächsten Riemen von 225 mm, 225 mm und 75 mm, der 1. Transferseite geschlossen, der 2. und 3. Transferseite als Ausgang, einer RollerDrive EC310, dem Antriebsmedium Polyäther und der Fördergeschwindigkeit von max. 0,6 m/s.

Zubehör

- Stützen, siehe S. 108
- Endkappen, siehe S. 114
- 24-V-Netzteil, siehe S. 115
- Interface-Karte, siehe S. 116
- SPS-Verbindungskabel, siehe S. 116
- Lichtschrankenkit, siehe S. 117

21

Unterstützung bei der Planung erhalten Sie in den Anwendungshinweisen ab S. 122.

Fördermodul: Endstück



INTERROLL
END STOP RM 5774



Der Endstopp ist ein fest montierter mechanischer Abschluss einer Förderlinie, der ankommendes Fördergut anhält und aufstaut. Typischerweise wird der Endstopp am Ende von Gefälle- oder Friktionsrollenbahnen oder zum sicheren Abschluss aller anderen Förderlinien verwendet.

Technische Daten

Allgemeine technische Daten
Max. Staudruck 100 N
Seitenprofil Kombination der Profiltypen links/rechts Für alle Profilkombinationen geeignet

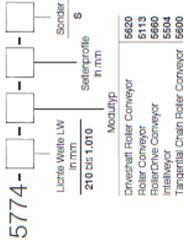
Maße



Abmessungen
LW/Lichte Weile 210 bis 1010 mm

Bestellinformation

- Modul ist komplett montiert
 - Stahlteile sind pulverbeschichtet (in schwarz RAL 9005 bzw. grau RAL 7030) oder verzinkt
- Bitte stellen Sie die Bestellnummer mithilfe des nachfolgenden Konfigurators zusammen.



Wenn Sie eine vom Standard abweichende Ausführung benötigen, fügen Sie am Ende ein „S“ hinzu und beschreiben Sie Ihre Anforderungen.

Beispiel für eine Bestellnummer: 5774-510-5620-150/150

Diese Bestellnummer steht für Interroll End Stop RM 5774 mit der lichten Weile LW 510 mm, dem Modultyp Interroll Drivehaft Roller Conveyor RM 5620 und der Seitenprofilkombination 150/150 mm.

Bestellbeispiel

Fördermodul: Standfuß



Zubehör
RM 5703
Stütze

Die Interroll Support Stands RM 5703 bestehen aus einem tragfähigen Aluminiumprofil, auf dem die Fördermodule montiert werden. Die Stützen sind mit einem einstellbaren Fuß ausgestattet und werden mit einer Winkelhalterung am Seitenrahmen des Förderers befestigt.

Bestellinformation

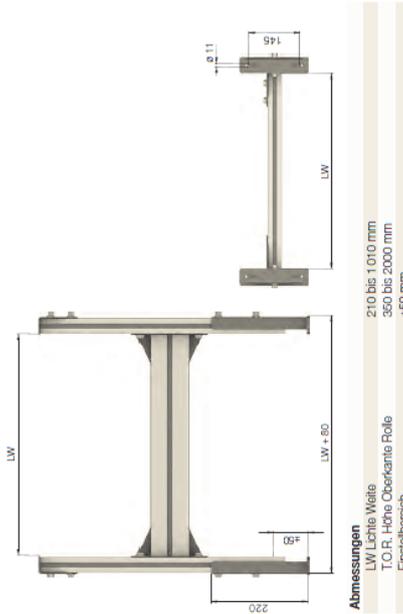
- Modul ist komplett montiert
 - Stahlteile sind pulverbeschichtet (in schwarz RAL 9005 bzw. grau RAL 7030) oder verzinkt
- Bitte stellen Sie die Bestellnummer mithilfe des nachfolgenden Konfigurators zusammen.



Technische Daten

Allgemeine technische Daten	200 kg
Max. Traglast	
Seitenprofil	Für alle Profilkombinationen geeignet
Kombination der Profilhöhen links/rechts	1 bei 350 bis 800 mm Höhe Oberkante Rolle
Anzahl Querstreben	2 bei 800 bis 1.400 mm Höhe Oberkante Rolle
	3 bei 1.400 bis 2.000 mm Höhe Oberkante Rolle

Maße



Bestellbeispiel

Wenn Sie eine vom Standard abweichende Ausführung benötigen, fügen Sie am Ende ein „S“ hinzu und beschreiben Sie Ihre Anforderungen.

Beispiel für eine Bestellnummer: 5703-S10-700

Diese Bestellnummer steht für Interroll Support Stands RM 5703 mit der lichten Weile LW 510 mm und der Höhe Oberkante Rolle T.O.R. 700 mm.

9.2 Parameterauflistung

In der folgenden Parameterauflistung sind die einzelnen Parameter, die zur Erstellung einer Design Structure Matrix für den Fördertyp Rollenförderer verwendet wurden, angeführt. Die Informationen der Parameter stammen aus der Literatur [INT12a] ; [INT12b] ; [JOD12] .

Umgebungsparameter:

- Umgebungstemperatur:
Die Fördermodule und Bauteile von Interroll können bei Temperaturen von +5° bis +40°C ohne Bedenklichkeiten eingesetzt werden.
- Feuchte Umgebung:
Wenn die relative Luftfeuchtigkeit von 65% überschritten wird, herrscht feuchte Umgebung vor, z.B. im Hafbereich.
- Verschmutzungsgrad
- Lärmentwicklung:
Die von der Anlage hervorgerufene Lautstärke wird als Lärmentwicklung bezeichnet.

Eingangsparameter:

- Transportgutbreite b siehe Abbildung Seite 107
- Transportgutlänge l siehe Abbildung Seite 107
- Transportguthöhe h siehe Abbildung Seite 107
- Transportgutgewicht:
Die Firma Interroll klassifiziert nach Gewichtsklassen (30kg Light, 100kg Medium, 250 kg Heavy, 1500kg Palett)
- Transportgutmaterial:
Für einen störungsfreien Transport ist die Bodenbeschaffenheit entscheidend. Behältermaterialien: Stahl, Kunststoff, Holz, Karton (hart/weich)
- Fördergeschwindigkeit
- Durchsatz
- Einschleusvorgang:
Bei Förderstrecken, in die Fördergut eingeschleust wird, muss die lichte Weite um mindestens 100mm größer sein als die Transportgutbreite. Durch den Einschleusvorgang verringert sich der Durchsatz.
- Ausschleusvorgang:
Ebenso wie beim Einschleusvorgang wird der Durchsatz verringert.
- Lastverteilung:
Je ungleichmäßiger die Gewichtsverteilung ist, desto schwieriger ist ein zuverlässiger Transport.

Rohrparameter:

- Rohrdurchmesser siehe Abbildung Seite 107
- Rohrwandstärke
- Rohrlänge siehe Abbildung Seite 107
- Rohrwerkstoff:
Der Rohrwerkstoff kann aus Stahl, Aluminium oder Kunststoff bestehen.
- Rohrüberzug:
Für die Optimierung der Rohroberfläche werden Rohrüberzüge verwendet. Damit werden Eigenschaften wie Geräuschdämpfung und Erhöhung der Mitnahme von Transportgütern, erreicht.

- Rohrverzinkung:
Die Rohrverzinkung bietet einen kostengünstigen Korrosionsschutz für Stahlrohre. Anwendung nur bei normal temperierten und trockenen Bereichen.

- max. Rohrtraglast:
Die Biegespannung des Rohres muss unter den Materialgrenzwerten liegen.

$$\sigma_{b(\text{Rohr})} = \frac{M_{b(\text{Rohr})}}{W_{(\text{Rohr})}} \leq \sigma_{zul(\text{Rohr})}$$

$\sigma_{b(\text{Rohr})}$ Biegespannung des Rohres

$M_{b(\text{Rohr})}$ Biegemoment des Rohres (abhängig von Belastungswert/ -art und Einbaulänge bzw. lichte Weite)

$W_{(\text{Rohr})}$ Widerstandsmoment für das Rohr

(abhängig vom Rohrdurchmesser und Rohrwandstärke)

Fördermodulparameter:

- lichte Weite Gerade siehe Abbildung Seite 107
- lichte Weite Kurve:
Sie muss in Kurven größer sein als bei geraden Förderstrecken.
Bei gegebenem Innenradius wird der Außenradius folgend berechnet.

$$R_a = 50 + \sqrt{(R_i + b)^2 - \left(\frac{l}{2}\right)^2}$$

Die Differenz aus Außenradius und Innenradius ergibt die lichte Weite.

R_a Außenradius der Kurve

R_i Innenradius der Kurve

b Transportgutbreite

l Transportgutlänge

LW lichte Weite

- Rollenteilung: siehe Abbildung Seite 107
Laut Interroll müssen sich immer drei Förderrollen unter dem Fördergut befinden.
- Rollenlager:
Rollenlagerparameter (Werkstoff und Tragzahl)
Interroll verwendet vorzugsweise Kugellager aus Chromstahl oder Niro-Stahl. Die mittlere nominelle Lebensdauer eines Wälzlagers wird wie folgend berechnet.

$$L_{10h} = \frac{10^6}{60 * n} * \left(\frac{C}{P}\right)^p$$

L_{10h} Lebensdauer des Wälzlagers

n Drehzahl (abhängig von Fördergeschwindigkeit und Rohrdurchmesser
 $v=r*\omega$ und $n= \omega/(2*\Pi)$)

C Tragzahl

P äquivalente Lagerbelastung

p Lebensdauerexponent

Förderungsparameter:

- Geradeauslauf:
Er wird durch das Verhältnis von Länge zu Breite bestimmt, je größer dieses Verhältnis ist, desto besser ist der Geradeauslauf von den Transportgütern.
- Kippgefahr
Je größer die Höhe des Transportgutes im Verhältnis zu seiner Aufstandsfläche ist, desto kippgefährdeter ist es beim Fördern
- Mitnahmekraft:

Die Mitnahmekraft ist abhängig von dem Transportgutgewicht, Transportgutmaterial, von der Temperatur und von der Luftfeuchtigkeit.

- Rundlaufgenauigkeit:

Die Rundlaufgenauigkeit des Rohres ist die maximale radiale Abweichung des Rohrdurchmessers vom perfekten Kreis. Bei Kunststoffrohren steigt die Rundlaufgenauigkeit mit der Länge des Rohres überproportional an.

Antriebsparameter:

- Die Antriebsleistung für Rollenförderer berechnet sich laut Formel:

$$P = \frac{F_W * v_F}{\eta_{ges}}$$

PAntriebsleistung

F_W Gesamtwiderstand (Summe aus Rollwiderstand, Lagerreibungswiderstand und Beschleunigungswiderstand)

η_{ges} Gesamtwirkungsgrad des Antriebes (Produkt der Einzelwirkungsgrade)

- Rollwiderstand:

Die Formel für den Rollwiderstand lautet:

$$F_{WR} = \frac{2 * m_{st} * g * f}{D}$$

F_{WR} Rollwiderstand

m_{st} Masse der Stückgüter

g Erdbeschleunigung

f Hebelarm der rollenden Reibung = 0,5 bis 1mm(St/St)

D Rohrdurchmesser

- Lagerreibungswiderstand

Die Formel für den Lagerreibungswiderstand lautet:

$$F_{WL} = \frac{\mu_L * g * d}{D} * (m_{st} + m_R * z)$$

F_{WL} Lagerreibungswiderstand

μ_L Lagerreibungszahl (Wälzlager=0,002 bis 0,003; Gleitlager=0,08)

d Lagerinnendurchmesser

m_R Masse der drehenden Teile einer Rolle

z Rollenanzahl je Gutstück

- Beschleunigungswiderstand:

Die Formel für den Beschleunigungswiderstand lautet:

$$F_{WB} = \frac{m_R * (v_F^2 - v_0)}{s_R}$$

F_{WB} Beschleunigungswiderstand

m_R Masse der drehenden Teile einer Rolle

v_F Fördergeschwindigkeit

v_0 Anfangsgeschwindigkeit

s_R Rollenteilung

- Motordrehmoment:

Ergibt sich aus dem Gesamtwiderstand mal dem Rohrdurchmesser.

Das Motordrehmoment steht mit der Leistung nach: $P=M * \omega$ in Relation.

- Antriebsmedium/Übertrieb:

Je nach Fördergeschwindigkeit kommen unterschiedliche Antriebsmedien zum Einsatz.

Rahmenparameter:

- Rahmenprofil

- Rahmenwerkstoff

Als Rahmenwerkstoff wird vorwiegend Stahl oder Aluminium verwendet.

- max. Rahmentraglast:

Die Biegespannung des Rahmens muss unter den Materialgrenzwerten liegen.

$$\sigma_{b(\text{Rahmen})} = \frac{M_{b(\text{Rahmen})}}{W_{(\text{Rahmen})}} \leq \sigma_{zul(\text{Rahmen})}$$

$\sigma_{b(\text{Rahmen})}$... Biegespannung des Rahmens

$M_{b(\text{Rahmens})}$. Biegemoment des Rahmens (abhängig von Belastungswert, -art und Stützenabstand)

$W_{(\text{Rahmen})}$ Widerstandsmoment des Rahmens
(abhängig von der Rahmengenometrie)

Stützenparameter:

- **Stützenwerkstoff:**
Stahl oder Aluminium werden vorzugsweise verwendet.
- **Stützenprofil:**
Als Stützenprofil kann jegliche Geometrie in Frage kommen.
- **Stützenabstand**
- **max. Stützentraglast:**
Die Druckspannung der Stützen muss unter den Materialgrenzwerten liegen.

$$\sigma_d(\text{Stützen}) = \frac{F_{(\text{Stütze})}}{A_{(\text{Stütze})}} \leq \sigma_{zul(\text{Stütze})}$$

$\sigma_d(\text{Stützen})$ Druckspannung der Stützen

$F_{(\text{Stützen})}$ Stützenbelastung (abhängig von Belastungswert/ -art und Stützenabstand)

$A_{(\text{Stützen})}$ Querschnittsfläche der Stütze (abhängig von der Stützengeometrie)

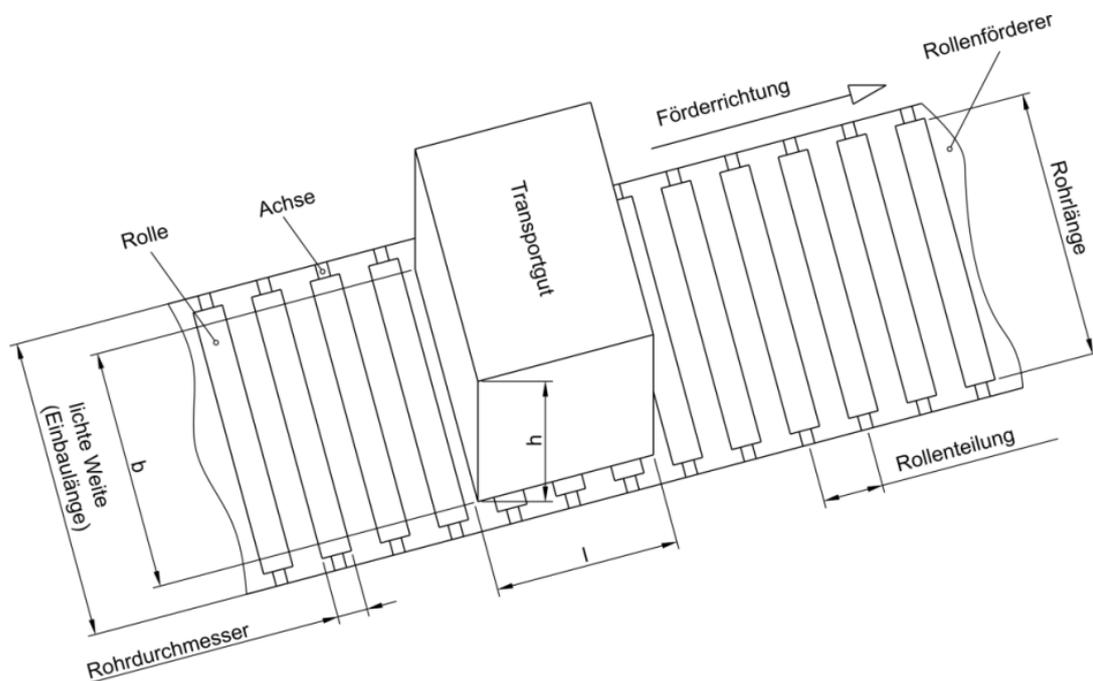


Abbildung: Transportgut- und Rollenfördererparameter

Banding

	Band 0	Band 1	Band 2	Band 3	Band 4	Band 5
<input type="checkbox"/> Umgebungstemperatur	<input type="checkbox"/>					
<input type="checkbox"/> Feuchte Umgebung	<input type="checkbox"/>					
<input type="checkbox"/> Verschmutzung	<input type="checkbox"/>					
<input type="checkbox"/> Lärmentwicklung	<input type="checkbox"/>					
<input type="checkbox"/> Transportgutbreite	<input type="checkbox"/>					
<input type="checkbox"/> Transportguthöhe	<input type="checkbox"/>					
<input type="checkbox"/> Transportgutgewicht	<input type="checkbox"/>					
<input type="checkbox"/> Transportgutmaterial	<input type="checkbox"/>					
<input type="checkbox"/> Fördergeschwindigkeit	<input type="checkbox"/>					
<input type="checkbox"/> Durchsatz		<input type="checkbox"/>				
<input type="checkbox"/> Einschleusvorgang		<input type="checkbox"/>				
<input type="checkbox"/> Ausschleusvorgang		<input type="checkbox"/>				
<input type="checkbox"/> Lastverteilung		<input type="checkbox"/>				
<input type="checkbox"/> Rohrdurchmesser		<input type="checkbox"/>				
<input type="checkbox"/> Rohrwandstärke		<input type="checkbox"/>				
<input type="checkbox"/> Rohrlänge		<input type="checkbox"/>				
<input type="checkbox"/> Rohrwerkstoff		<input type="checkbox"/>				
<input type="checkbox"/> Rohrüberzug		<input type="checkbox"/>				
<input type="checkbox"/> Rohrverzinkung		<input type="checkbox"/>				
<input type="checkbox"/> max. Rohrtraglast		<input type="checkbox"/>				
<input type="checkbox"/> Lichte Weite Gerade		<input type="checkbox"/>				
<input type="checkbox"/> Lichte Weite Kurve		<input type="checkbox"/>				
<input type="checkbox"/> Rollenteilung		<input type="checkbox"/>				
<input type="checkbox"/> Rollenlager		<input type="checkbox"/>				
<input type="checkbox"/> Geradeauslauf		<input type="checkbox"/>				
<input type="checkbox"/> Kippgefahr		<input type="checkbox"/>				
<input type="checkbox"/> Mitnahmekraft		<input type="checkbox"/>				
<input type="checkbox"/> Rundlaufgenauigkeit		<input type="checkbox"/>				
<input type="checkbox"/> Antriebsleistung		<input type="checkbox"/>				
<input type="checkbox"/> Motordrehmoment		<input type="checkbox"/>				
<input type="checkbox"/> Antriebsmedium		<input type="checkbox"/>				
<input type="checkbox"/> Rahmenwerkstoff		<input type="checkbox"/>				
<input type="checkbox"/> Rahmenprofil		<input type="checkbox"/>				
<input type="checkbox"/> max. Rahmentraglast		<input type="checkbox"/>				
<input type="checkbox"/> Stützenwerkstoff		<input type="checkbox"/>				
<input type="checkbox"/> Stützenprofil		<input type="checkbox"/>				
<input type="checkbox"/> Stützenabstand		<input type="checkbox"/>				
<input type="checkbox"/> max. Stützentraglast		<input type="checkbox"/>				

Clustering

	Einschleusvorgang	Ausschleusvorgang	Rundlaufgenauigkeit	Cluster 3	Cluster 4	Cluster 5	Cluster 7	Cluster 9	Cluster 10
<input type="checkbox"/> Einschleusvorgang	<input checked="" type="checkbox"/>								
<input type="checkbox"/> Ausschleusvorgang		<input checked="" type="checkbox"/>							
<input type="checkbox"/> Rundlaufgenauigkeit			<input checked="" type="checkbox"/>						
<input type="checkbox"/> Rollenlager			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>					
<input type="checkbox"/> Transportgutgewicht			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>					
<input type="checkbox"/> Fördergeschwindigkeit			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>					
<input type="checkbox"/> Durchsatz			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>					
<input type="checkbox"/> Antriebsleistung			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>					
<input type="checkbox"/> Motordrehmoment			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>					
<input type="checkbox"/> Stützenabstand				<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>				
<input type="checkbox"/> Stützenprofil				<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>				
<input type="checkbox"/> Stützenwerkstoff				<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>				
<input type="checkbox"/> max. Stützentragslast				<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>				
<input type="checkbox"/> Lärmentwicklung			<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>				
<input type="checkbox"/> Transportgutmaterial					<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>			
<input type="checkbox"/> Mitnahmekraft					<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>			
<input type="checkbox"/> Rohrwerkstoff					<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>			
<input type="checkbox"/> Rohrüberzug					<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>			
<input type="checkbox"/> Verschmutzung						<input checked="" type="checkbox"/>			
<input type="checkbox"/> Antriebsmedium						<input checked="" type="checkbox"/>			
<input type="checkbox"/> Lichte Weite Kurve						<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
<input type="checkbox"/> Rohrlänge						<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
<input type="checkbox"/> Rohrwandstärke						<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
<input type="checkbox"/> Rohrdurchmesser						<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
<input type="checkbox"/> max. Rohrtragslast						<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
<input type="checkbox"/> Transportgutbreite						<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
<input type="checkbox"/> Lichte Weite Gerade						<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
<input type="checkbox"/> Feuchte Umgebung							<input checked="" type="checkbox"/>		
<input type="checkbox"/> Umgebungstemperatur							<input checked="" type="checkbox"/>		
<input type="checkbox"/> Rohrverzinkung							<input checked="" type="checkbox"/>		
<input type="checkbox"/> Lastverteilung							<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
<input type="checkbox"/> Rahmenprofil							<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
<input type="checkbox"/> Rahmenwerkstoff							<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
<input type="checkbox"/> max. Rahmentraglast							<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
<input type="checkbox"/> Geradeauslauf								<input checked="" type="checkbox"/>	
<input type="checkbox"/> Transportguthöhe								<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<input type="checkbox"/> Transportgutlänge								<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<input type="checkbox"/> Rollenteilung								<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<input type="checkbox"/> Kippgefahr								<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>