



Ralph Stelzer (Hrsg.)

ENTWERFEN ENTWICKELN ERLEBEN 2016

Beiträge zur virtuellen Produktentwicklung
und Konstruktionstechnik

Ralph Stelzer (Hrsg.) **ENTWERFEN ENTWICKELN ERLEBEN** 2016
Beiträge zur virtuellen Produktentwicklung und Konstruktionstechnik

Ralph Stelzer (Hrsg.)

ENTWERFEN ENTWICKELN ERLEBEN 2016

Beiträge zur virtuellen Produktentwicklung
und Konstruktionstechnik

Dresden · 30. Juni – 1. Juli 2016

Programmkomitee Virtuelle Produktentwicklung und Konstruktionstechnik

Prof. Dr. Ralph Stelzer, TU Dresden

Prof. Dr. Michael Abramovici, Ruhr-Universität Bochum

Prof. Dr. Reiner Anderl, TU Darmstadt

Prof. Dr. Martin Eigner, Universität Kaiserslautern

Prof. Dr. Detlef Gerhard, TU Wien

Prof. Dr. Jivka Ovtcharova, KIT Karlsruhe

Prof. Dr. Rainer Stark, TU Berlin

Prof. Dr. Sandor Vajna, Universität Magdeburg

Prof. Dr. Sandro Wartzack, Universität Erlangen

Entwickeln – Entwerfen – Erleben 2016.
Beiträge zur Virtuellen Produktentwicklung und Konstruktionstechnik
Herausgeber: Ralph Stelzer

Bibliografische Information der Deutschen Nationalbibliothek
Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der
Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind
im Internet über <http://dnb.d-nb.de> abrufbar.

Bibliographic information published by the Deutsche Nationalbibliothek
The Deutsche Nationalbibliothek lists this publication in the Deutsche
Nationalbibliografie; detailed bibliographic data are available in the
Internet at <http://dnb.d-nb.de>.

ISBN 978-3-95908-062-0

© 2016 w.e.b. Universitätsverlag & Buchhandel
Eckhard Richter & Co. OHG
Bergstr. 70 | D-01069 Dresden
Tel.: 0351/47 96 97 20 | Fax: 0351/47 96 08 19
<http://www.tudpress.de>

TUDpress ist ein Imprint von w.e.b.

Alle Rechte vorbehalten. All rights reserved.
Layout und Satz: Technische Universität Dresden.
Umschlaggestaltung: TU Dresden, Illustration © 2016 TU Dresden
Printed in Germany.

Erscheint zugleich auf QUCOSA der SLUB Dresden
<http://nbn-resolving.de/urn:nbn:de:bsz:14-qucosa-203878>





KOMPLEXITÄT DREHT SICH IMMER NUR IM KREIS.



EINFACH TRIFFT ENTSCHEIDUNGEN.

Komplexität bremst Ihr Business aus. Denn je gewaltiger die Informationsflut, desto schwieriger die Entscheidungsfindung. SAP arbeitet daran, Dinge zu vereinfachen. Damit aus Daten Wissen und aus Wissen fundierte Entscheidungen werden, die Ihr Unternehmen weiterbringen. Finden Sie heraus, wie gemeinsam einfach möglich wird auf sap.de/runsimple

Prozessgebundene Berechnungs-Baugruppen: Ein Ansatz zur Lösung komplexer Entscheidungs- und Berechnungsabläufe

Denis Polyakov · Willi Gründer

Zusammenfassung

Prozessgebundene Berechnungs-Baugruppen bieten Konstrukteuren die Möglichkeit, ihre zum Teil mehrstufigen Berechnungen durch den Einsatz modularer Funktionsbausteine ablaforientiert, verbindlich, nachvollziehbar und vor allem zeitsparender zu gestalten. Die Grundlage dieser neuen Methode bilden Ansätze, die sich bereits in der Informationstechnik und in der Konstruktionsmethodik bewährt haben. Vom Anforderungsmanagement bis zur Validierung kann dabei auf eine Bibliothek modularer Berechnungsobjekte in Form von Prozess-Bausteinen zugegriffen werden, deren Schnittstellen und Datenstrukturen ausnahmslos einheitlichen Definitionen entsprechen. Gemeinsam mit der in Anlehnung an den eCI@ss - Standard entwickelten Merkmalsstruktur der Prozess-Bausteine wird so eine hohe Wiederverwendbarkeit in unterschiedlichen Berechnungskonfigurationen erzielt. Ihre Klassifizierung orientiert sich an konstruktionssystematischen Gesichtspunkten. Die Methode wird beispielhaft an Berechnungen für Getriebekomponenten erläutert.

Konstruktionssystematik und Digitalisierung

Die Entwicklung von Getrieben ist heute ohne leistungsfähige Computerprogramme nicht mehr denkbar. Die Berechnungsprogramme sowohl für die tägliche Routinearbeit als auch für Spezialaufgaben ermöglichen neben verbesserten Leistungskennwerten höhere Lebensdauern, verringerte akustische Belastungen und geringere Bauräume. Hierfür muss zwischen zahlreichen Merkmalen bzw. Anforderungen ein optimales Verhältnis erzielt werden. Basis dieser Optimierungen sind Parameterstudien für Einzelkomponenten, Unterbaugruppen oder Baugruppen wie z.B. der Berechnung einer Getriebestufe oder der Dimensionierung einer Welle-Nabe-Verbindung.

Diese Auslegungen und Nachrechnungen erfolgen nur noch vereinzelt mit selbstprogrammierten Insellösungen. Doch auch die heute überwiegend eingesetzten fertigen Zukauf-Applikationen sind bezüglich ihrer Merkmale, Eingabemöglichkeiten und der zu berücksichtigenden Randbedingungen nur selten mit anderen am Markt angebotenen Teillösungen kompatibel. Überdies sind die verwendeten Algorithmen oftmals anbietergebunden, kaum nachvollziehbar und nur mit erheblichem Aufwand an besondere Aufgabenstellungen anzupassen. Andererseits hat „die Industrie zwischenzeitlich damit begonnen, auch ihre technischen Workflows in vollem Umfang mit Softwaretools zu unterstützen. Man kann davon ausgehen, dass in absehbarer Zeit alle wertschöpfenden Prozesse entlang einer Produktentstehung... weitestgehend digital vollzogen werden können...“ (Huber 2013)

Während die Fertigung bereits in den Phasen der Mechanisierung und Automatisierung weitgehend fließend organisiert wurde und aufgrund der dabei erzielten Transparenz und Modularität neue Technologien jederzeit integrieren kann, hat die Produktentwicklung das volle Potenzial von Workflows bei weitem noch nicht erkannt und ausgenutzt. Denn auch sie erfordert eine flexible, vielen Aufgabenstellungen gerechte Umgebung zur schnellen Konfiguration von Prozessen, die den Lösungsraum für die angestrebten Entwicklungsziele umreißen, die möglichen prinzipiellen Lösungen identifizieren und schließlich ein optimales Auslegungsergebnis präsentieren. Zu berücksichtigen sind dabei Anforderungen der Usability, der Vernetzung mit der übrigen IT-Welt und der Entwicklungs-Community sowie mit der Dokumentation und Qualitätssicherung.

Mehr Transparenz durch grafische Repräsentation

In der Geschäftsprozessentwicklung unterscheidet man grundsätzlich in textuelle und graphische Vorgehensweisen. Textuelle Beschreibungen können einerseits zwar sehr präzise sein, sie sind andererseits aber auch durch das Sprachverständnis des Lesers unterschiedlich interpretierbar. Außerdem sind umfangreiche Programmquellen nicht geeignet, die darin enthaltene Information schnell und zutreffend zu erfassen. Aus diesem Grund ersetzen graphische Definitionen für die Prozess- und Systemdarstellung allmählich die rein textlichen Beschreibungssprachen. Zu den wichtigsten zählen die Business Process Management Notation BPMN und die Unified Modeling Language UML. BPMN findet sich vor allem im Bereich der Geschäftsprozessmodellierung, wohingegen UML in der Softwareentwicklungsbranche angesiedelt ist. Letztere wurde für Engineeringaufgaben unter der Bezeichnung Systems Markup Language SysML weiterentwickelt. Sie bildet heute in der virtuellen Produktplanung einen wichtigen Ausgangs-

punkt für die strukturelle und funktionale Vernetzung von Merkmalen, Zuständen und Informationsflüssen.

Systemansicht mit Systems Engineering

Die graphische Sprache SysML beinhaltet eine Reihe von Diagrammtypen, um das System aus verschiedenen Sichten erfassen zu können. Für die hier beschriebene Aufgabe der prozessorientierten Beschreibung der Berechnungsabläufe sind zwei Diagrammtypen von besonderem Interesse: das Aktivitätsdiagramm und das interne Blockdiagramm (siehe Abbildung 1).

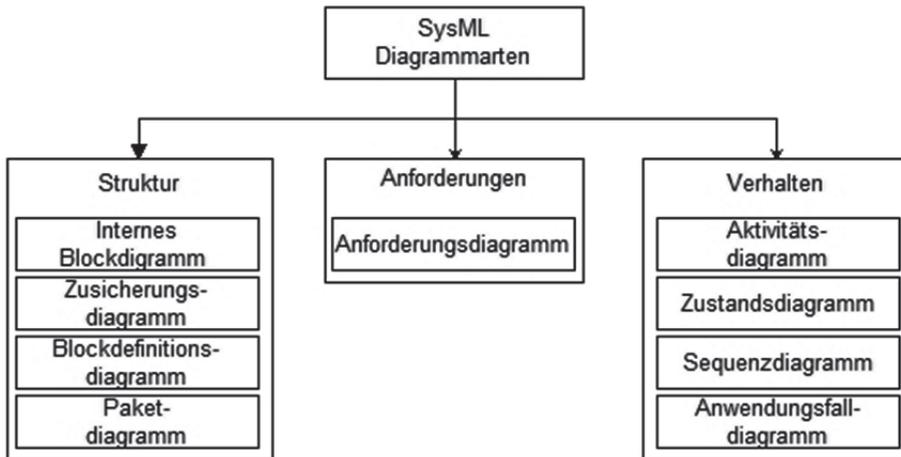


Abbildung 1: Diagrammtypen in SysML

Mit Hilfe dieser Funktionsdiagramme lassen sich nicht nur Strukturen, sondern auch zeitliche Abläufe definieren. Damit ist es möglich, Prozesse und Algorithmen auf verschiedenen Abstraktionsebenen übersichtlich und verständlich zu beschreiben, darunter auch sehr komplexe Abläufe mit vielen Varianten, Entscheidungen, und Wiederholungen. Besonders hilfreich ist dies bei textgebundenen Wissensszenarien, wie sie in Normen und Richtlinien anzutreffen sind. Durch die graphische Darstellung werden die darin versteckten Informationen und Aussagen zu ansprechbaren Elementen im Prozess-Modell.

Abbildung 2 zeigt den einfachen Ablauf einer Folge von Aktivitäten. Dabei ist die der Ausführungsreihenfolge der Aktionen (Vierecke mit abgerundeten Kanten) durch die gerichteten Pfeile angegeben. Der Startpunkt eines Prozesses wird durch einen ausgefüllten schwarzen Kreis gekennzeichnet.

Ein solcher Kreis mit einem weiteren Kreis als Umrandung beschreibt dabei den Endpunkt eines Prozesses. Mit den Rauten kann eine Verzweigung bzw. eine Zusammenführung realisiert werden. Im Diagramm der Abbildung 2 wird somit zuerst die Aktion 1 ausgeführt dann die Aktion 2 und anschließend abhängig von der Entscheidung an dem Kontrollknoten (Raute) entweder die Aktion 3 oder die Aktion 4. Anschließend wird der Prozess beendet (Kleuker 2009).

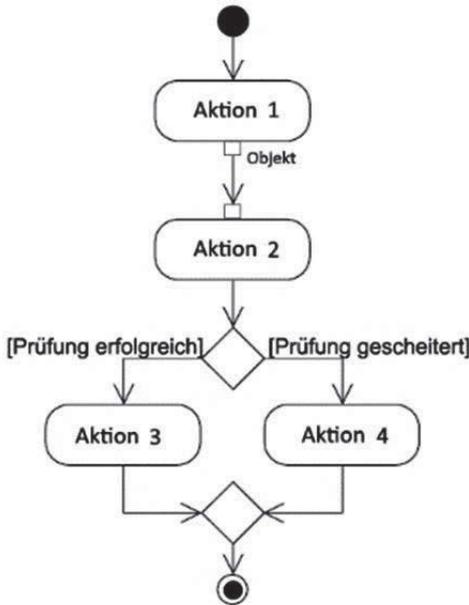


Abbildung 2: Beispiel eines Aktivitätsdiagramm (Kleuker 2009)

In Bezug auf die Berechnungsabläufe ähnelt eine Aktion einem elementaren mathematischen Ausdruck $y = f(x)$. Die Ein- und Aufgabepins entsprechen dann den Funktionsparametern bzw. dem Funktionswert. Darstellung einer einfachen Berechnung in Form eines Aktivitätsdiagramms ist in Abbildung 3 dargestellt.

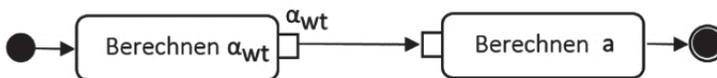


Abbildung 3: Berechnung als Aktivitätsdiagramm

Die Berechnung des Achsabstandes erfordert als Input den Wert des Stirneingriffswinkels ω_{wt} , welcher durch die davorstehende Funktion (Berechnen ω_{wt}) bereitgestellt wird. Diese Art des Datenaustausches nennt man direkten Datenaustausch. Bei komplexen Zusammenhängen kann aber die explizite Darstellung des Datenflusses zur Unlesbarkeit des Diagramms führen. Eine Alternative, die für transparente Abläufe sorgt, ist dagegen der indirekte Datenaustausch. Dieser kann zum Beispiel mit Hilfe eines gemeinsamen Speichers realisiert werden. Die Datenflüsse werden dabei nicht explizit modelliert, sondern ergeben sich automatisch durch Zugriffe der Funktionen auf die Prozessvariablen (siehe Abbildung 4). Ein lesender bzw. schreibender Zugriff auf eine Prozessvariable A durch eine Funktion f1() ist dann gegeben, wenn A als aktueller Eingabe- bzw. Ausgabeparameter in der durch A realisierten Schnittstelle vorkommt. Wird A als aktueller Ausgabeparameter von Funktion f1() und als aktueller Eingabeparameter von Funktion f2() verwendet, so existiert ein Datenfluss von f1() nach f2() (Wörzberger 2010).

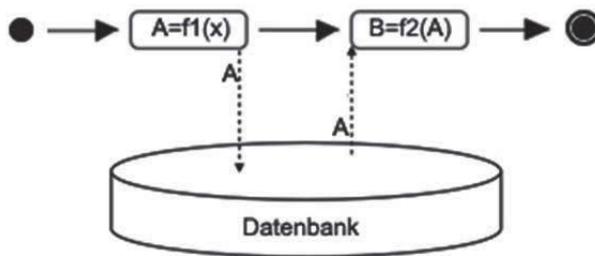


Abbildung 4: Expliziter Datenfluss

Die Beschreibung der Struktur eines Systems bzw. der an dem Prozess beteiligten Objekte kann mit Hilfe von Strukturdiagrammen durchgeführt werden. Dabei unterscheidet man in SysML zwischen den Diagrammtypen Blockdefinitionsdiagramm (bdd) und Internes Blockdiagramm (ibd). Die Bestandteile des Systems, die sogenannten Blöcke, werden im bdd definiert, der Zusammenhang zwischen den Blöcken dagegen im ibd. Diese Trennung ist auf die Objektorientierung der Notation zurückzuführen und wurde eingeführt, um den Systemingenieur nicht mit den Klassen, Objekten, Vererbung und etc. zu belasten.

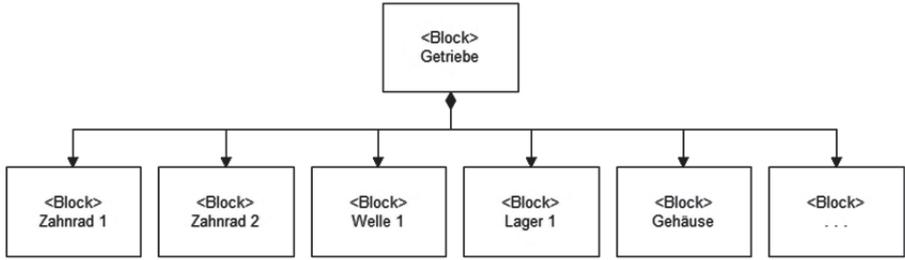


Abbildung 5: Beispiel für ein Blockdefinitionsdiagramm

Die Strukturdiagramme bdd und ibd können gut an einem Getriebe und seinen Komponenten veranschaulicht werden, denn alle Getriebekomponenten lassen sich in einem Blockdefinitionsdiagramm darstellen (siehe Abbildung 5). Die Beziehung von zum Beispiel einer Welle und einem Zahnrad kann dann in einem Internen Blockdiagramm beschrieben werden (siehe Abbildung 6).

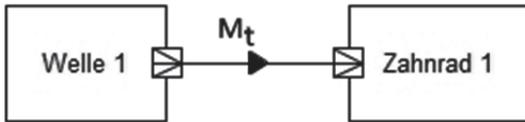


Abbildung 6: Beispiel einer Beziehung zwischen den Blöcken in ibd

Hier wird sofort ersichtlich, welche Bausteine miteinander verbunden sind und welche Informationen sie austauschen. Im Allgemeinen können die Blöcke über alle möglichen Energie-, Stoff und Informationsbeziehungen verknüpft werden.

Von der Modularisierung zur Standardisierung

Da in der Konstruktion für viele Fragenstellungen und Teilaufgaben bereits bewährte Lösungen existieren, die in Form von Normen, Richtlinien und anderen Vorschriften vorliegen, können diese als Grundlage für eine Bibliothek von fertigen Berechnungsbausteinen dienen. Die nach dem Grundprinzip der Datenverarbeitung (Abbildung 7) beschriebene Funktion bildet dann als Grundelement einer Berechnung den Funktionsbaustein.

Beschreibung der Funktion		
Eingabe (Parameter)	Transformation	Ausgabe (Parameter)

Abbildung 7: Funktionsaufbau nach dem EDV-Prinzip

Somit hätten wir zum Beispiel gemäß der Norm 3960 für einen Achsabstand gleich mehrere Funktionsbausteine:

$$a = a_d \cdot \frac{\cos \alpha_t}{\cos \alpha_{wt}} = \frac{m_n \cdot (z_1 + z_2)}{2 \cdot \cos \beta} \cdot \frac{\cos \alpha_t}{\cos \alpha_{wt}}$$

Name	Input	Output	Transformation
FB1	$a_d, \alpha_t, \alpha_{wt}$	a	$a_d \cdot \cos(\alpha_t) / \cos(\alpha_{wt})$
FB2	$m_n, z_1, z_2, \alpha_t, \alpha_{wt}$	a	$m_n \cdot (z_1 + z_2) / (2 \cdot \cos(\beta)) \cdot \cos(\alpha_t) / \cos(\alpha_{wt})$

Abbildung 8: Beispiel der Berechnungsbausteine

Wobei: a – Achsabstand in [mm], a_d – Null-Achsabstand in [mm], m_n – Normalmodul in [mm], m_t – Stirnmodul in [mm], $z_{1,2}$ – Anzahl der Zähne vom Ritzel und Rad in [-], $d_{1,2}$ – Teilkreisdurchmesser vom Ritzel und Rad in [mm], β – Schrägungswinkel in [°], α_t – Stirneingriffswinkel in [°], α_{wt} – Betriebseingriffswinkel in [°].

Zusammengefasst repräsentieren die Funktionsbausteine die elementaren Berechnungen zur Ermittlung von geometrischen oder physikalischen Eigenschaften von Maschinenelementen.

Bei der Anwendung eines gemeinsamen Datenspeichers für die Verwaltung von Prozessvariablen (Eingabe- und Ausgabeparametern der Funktionsbausteine) müssen die Schnittstellen zwischen den Funktionsbausteinen eindeutig definiert und formalisiert werden, damit die Kommunikation unter denen reibungslos und vor allem automatisiert erfolgen kann. Um außerdem die Datenkonsistenz des Abläufe zu gewährleisten, müssen sie flexibel und nach bestimmten Regeln in einander überführbar sein, damit es nicht zur Einschränkung bei der Modellierung kommt. Das kann mit einer Klassifizierung erreicht werden.

Klassifizierung der Berechnungsobjekte

Bei der Klassifikation sollen die Objekte und Klassen durch technische Eigenschaften beschrieben werden, so dass der Ingenieur nicht mit vollkommen abstrakten Objekten arbeiten muss. Ein weiterer wichtiger Aspekt ist die Interdisziplinarität, mit der ein Ingenieur während seiner Arbeit konfrontiert wird. Deshalb muss bei der Klassifikation die Vielfalt der Fachbereiche berücksichtigt werden, denn es sollen nicht nur Maschinenelemente, sondern auch alle anderen Fachgebiete und Objekte mit der gleichen Logik in die Berechnungsprozesse eingebunden werden. Für diese Aufgabe benötigt man einen Standard, in dem alle für uns notwendigen Daten fachbezogen aufbereitet sind. Hier kann der eCI@ss Standard herangezogen werden.

Der eCI@ss Standard ist ein hierarchisches System zur Gruppierung von Produkten und Dienstleistungen. Es besteht aus vier Hierarchieebenen (Klassen): Sachgebiet (Ebene 1), Hauptgruppe (Ebene 2), Gruppe (Ebene 3) und Untergruppe (Ebene 4). Die Knoten der Baumstruktur werden zusammen als Materialklassen bezeichnet. Auf der 4. Ebene (Untergruppe) stellt eCI@ss sogenannte Merkmalsleisten zur Verfügung. Merkmale ermöglichen die detaillierte Beschreibung von Produkten und Dienstleistungen und vereinheitlichen auf diese Weise die Geschäftsprozesse in Produktentwicklung und Materialwirtschaft (Herfurth 2014).

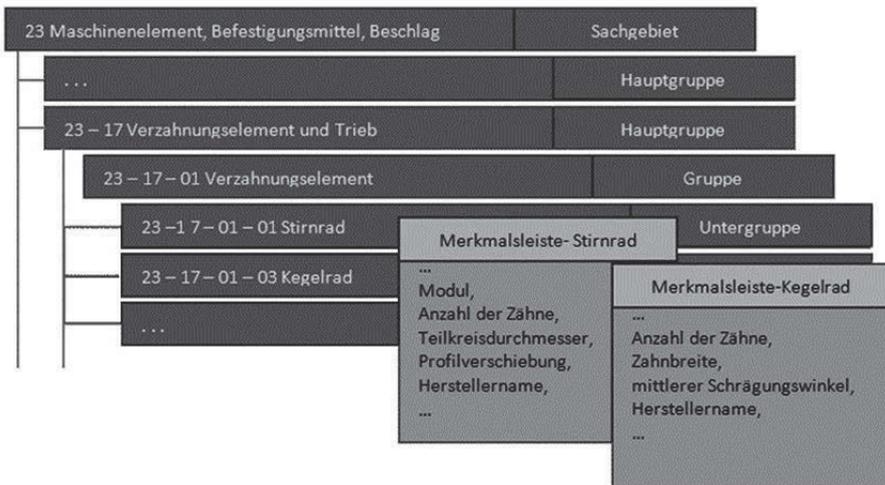


Abbildung 9: E-Class-Aufbau

Das Ergebnis der Klassifizierung gilt gleichermaßen für die Funktionsbausteine. Da ein Funktionsbaustein in der Regel als Berechnungsergebnis den Wert eines bestimmten Merkmals hat, sind die Funktionsbausteine schon nach dem Typ des Ausgangsports eindeutig den Merkmalen zugeordnet (siehe Abbildung 10). Sie bedürfen daher keiner zusätzlichen Klassifizierung.

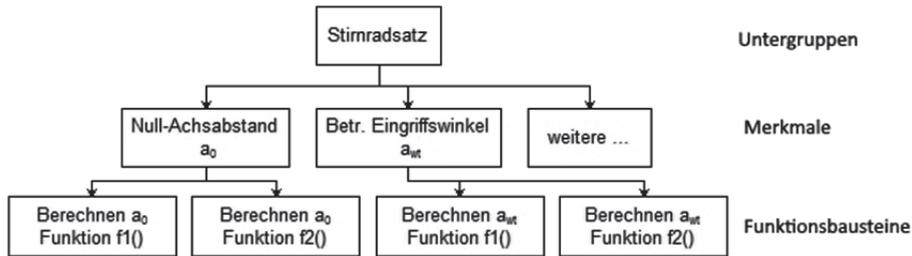


Abbildung 10: Hierarchie der Datenstruktur

Würde man den kompletten Inhalt der DIN 3960 bzw. der neuen ISO 21771 modularisieren, könnte daraus eine Bibliothek der auf einander abgestimmten Funktionsbausteine für die Berechnung der Verzahnungsgeometrie entstehen.

Prozessgebundene Getriebeentwicklung

Bei einer Getriebeentwicklung werden üblicherweise zunächst die Anforderungen an das zukünftige Getriebe in Form eines Lastenheftes zusammengefasst. Zu den wichtigsten Merkmalen gehören Antriebsart, Lastannahmen, Umgebungsbedingungen sowie die gewünschten Zielgrößen wie Drehmoment, Drehzahl und Lebensdauer. Getriebeart, Anzahl der Stufen (bei Zahnradgetrieben) und deren Ausführung (Stirnradstufe oder Planetenradstufe) werden meistens im Vorfeld geklärt, wobei diese Entscheidungen oft auf den Erfahrungen der Ingenieure und dem unternehmensinternen Know-how basieren. Diese Annahmen sind aber insofern von großer Bedeutung, als die Auswahl aller weiteren Getriebeelemente darauf aufbaut und nachträgliche Änderungen dieser Eigenschaften zu völlig neuen Getriebe-konzeptionen führen kann.

Bei der Erstausslegung der Getriebeelemente wird meistens auf firmeninterne Richtlinien oder Erfahrungen zurückgegriffen. Weil diese sehr unterschiedlich sein können, können sie die zahlreichen national sowie auch international standardisierten Rechenverfahren (DIN, ISO, AGMA, usw.) oft nicht direkt nutzen, da diese keine Auslegungsmethoden enthalten, sondern für die Nachrechnung bestimmt sind. Allerdings richtet sich auch das Unter-

nehmens Know-how an den Normen aus, müssen die erzielten Merkmale am Ende doch den anerkannten Regeln der Technik entsprechen. Die Auslegung von Maschinenelementen und Maschinenbau-Baugruppen ist deshalb meistens durch iterative Verfahren und Parameterstudien geprägt (siehe Abbildung 11).

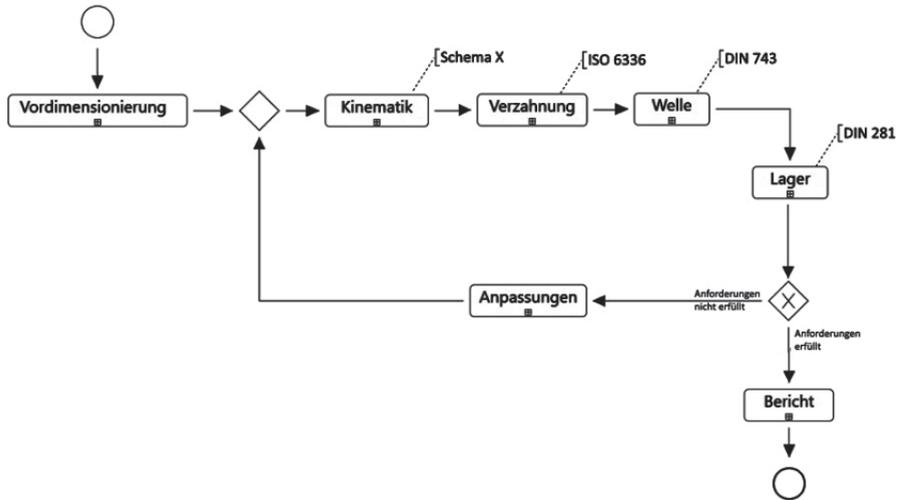


Abbildung 11: Getriebeauslegung iterativ

Eine Alternative zum klassischen iterativen Vorgehen bei der Auslegung bietet die Methode des expliziten Entwurfsmodells (Parow et al. 2016). Dabei müssen die gewünschten Modellparameter zunächst hergeleitet werden. Die inversen Berechnungen der genormten Zusammenhänge bilden die Basis dieser Methode, wobei die Anpassung der Parameter nicht mehr nach ingenieurstechnischen Gesichtspunkten, sondern durch feste mathematische Funktionen erfolgt. Die Umsetzung solcher Methoden ist zwar rechenintensiver als die klassische iterative Vorgehensweise, das ist aber bei den heutigen Rechnerleistungen nicht mehr relevant. Ausschlaggebend für die Anwendung solcher Methoden ist, dass die häufig fehlende Erfahrung durch die vordefinierten mathematischen Zusammenhänge (fachliches Wissen) ergänzt werden kann. Da viele Bewertungsregeln und Änderungsalgorithmen bereits in einer formalen mathematischen Sprache vorliegen, lassen sie sich ebenso wie die analytisch formulierten Normen gut automatisieren. (Parow et al. 2016)

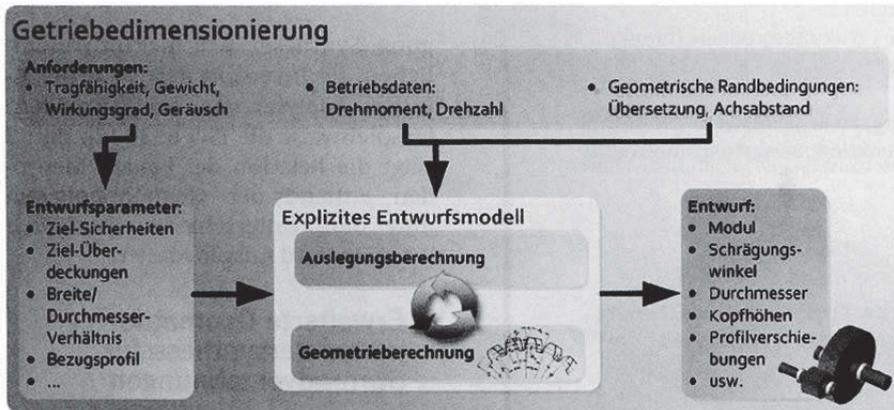


Abbildung 12: Explizites Entwurfsmodell (Parow et al. 2016)

Beide Vorgehensweisen lassen sich mit den prozessgebundenen Berechnungsbaugruppen modellieren, da sowohl die benutzergebundene iterative Korrektur der Modellparameter als auch ausgewählte numerische Verfahren für das Lösen von nicht linearen Gleichungen und Gleichungssystemen unterstützt werden.

Modellierungswerkzeug

Der eigentliche Prozess der Baugruppenbildung erfolgt mit einem Prozess-Designer, der die Berechnungsobjekte miteinander verknüpft, indem er die zugehörigen Informationsobjekte, Datenbankzugriffe und Algorithmen in die Benutzeroberfläche und in den Prozess einbindet.

Der an den Designer angeschlossene Prozess-Interpreter überführt anschließend die graphisch modellierten Zusammenhänge (visuelles Modell) in die maschinenlesbare Sprache XML. Dieses Beschreibungsformat des Prozesses kann dann von einer Prozess-Engine ausgeführt werden (Lütke-meier & Thöne 2001). Anschließend initialisiert der Interpreter die Dokumenten-Engine, die ihrerseits die Aufgabe hat, eine Eingabemaske zu generieren, die für den Start der Berechnung erforderlich ist. Diese minimal notwendigen Startinformationen werden anschließend in die Projektdatei geschrieben und an die Prozess-Engine übergeben. Sie koordiniert und synchronisiert die Ausführung der einzelnen Funktionsbausteine sowie die Aufrufe der externen Applikationen und sorgt so für den Ablauf des kompletten Prozesses. Nach erfolgreichem Abschluss des Prozesses werden die Ausgabeinformationen in die Projektdatei geschrieben. Die Dokumenten-

Engine stellt sie dann mit den Eingangsdaten übersichtlich in Form von Tabellen zusammen (siehe Abbildung 14).

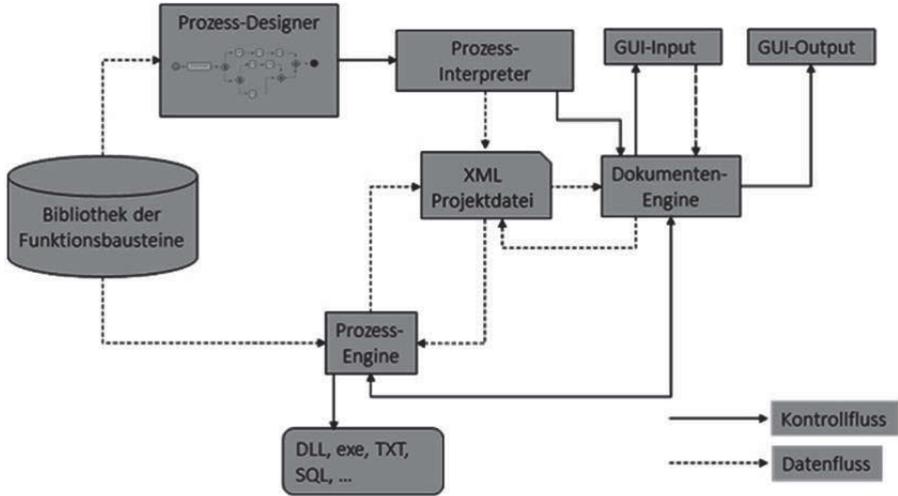


Abbildung 13: Schema des Modellierungswerkzeugs

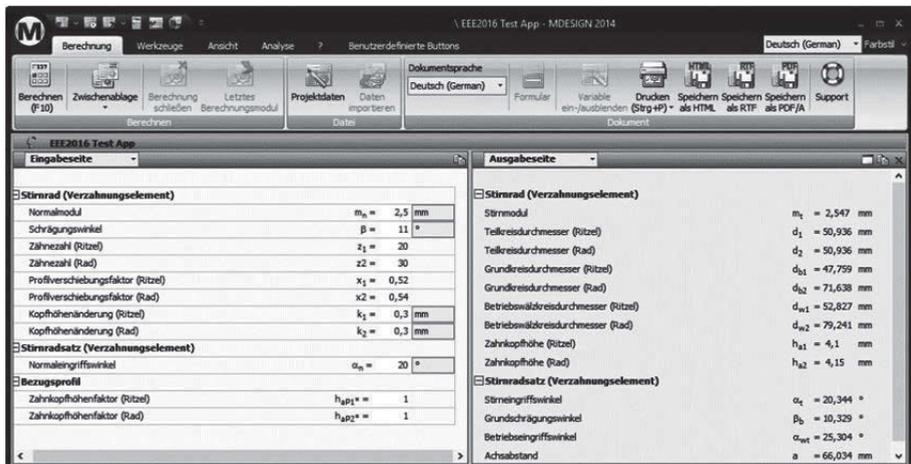


Abbildung 14: Dynamisch erstellte Ein- und Ausgabemaske

Fazit

Modellbasierte und prozessorientiert gestaltete Konstruktionsabläufe sind für die Digitalisierung der Produktplanung von großer Bedeutung. Sie ermöglichen eine interdisziplinäre Systembeschreibung der Strukturen, Funktionen und des Verhaltens eines neuen Erzeugnisses und damit virtuelle, in den Entwicklungsprozess integrierte Überprüfungen der geforderten Merkmale und Funktionen. Als wesentlicher und integraler Bestandteil der Konstruktion und Entwicklung müssen analytische und numerische Berechnungen zukünftig in diese automatisierten Prozesse nahtlos eingebunden werden. Mit dem vorgestellten Ansatz der prozessorientierten Berechnungs-Baugruppen und den entwickelten Modellierungswerkzeugen können nach dem Baukastenprinzip aufgebaute Wissensobjekte für Auslegungen, Optimierungen und Nachweise zur Verfügung gestellt und von den Entwicklungsingenieuren zu beliebigen Berechnungsprozessen verknüpft werden. Nach Beendigung der grafisch unterstützten Modellierung der Berechnung werden die für die Eingabe und Ausgabe freigegebenen Systemparameter automatisch auf der Eingabe- bzw. Ausgabeseite zusammengefasst. Der modellierte Prozess kann nach Belegung aller Eingabeparameter auf Knopfdruck ausgeführt werden. Somit erhält der Anwender ein interaktives Modellierungswerkzeug für die Gestaltung von Entscheidungs- und Berechnungsprozessen, das es ihm erlaubt, auch sehr komplexe Berechnungsgänge schnell und regelgerecht in digitale Wissensobjekte umzusetzen. An Hand des grafischen Ablaufdiagramms können die Aufgabenstellungen und Lösungswege einer prozessgebundenen Berechnungs-Baugruppe jederzeit nachvollzogen und modifiziert werden.

Literaturverzeichnis

- Parow, J., Otto M. & Stahl, K. 2016, Vom Lastenheft zur Verzahnung – Anwendungsflexible Dimensionierung von Zahnradgetrieben mittels expliziten Entwurfsmodell, Konstruktion (Ausgabe März), 64-69
- Kokurina E. 2012: Qualitätssicherung von Terminologie im eCI@ss-Standard – am Beispiel des Sachgebiets 22 Bautechnik in der deutschen, englischen und russischen Fachsprache, Masterarbeit, Wien: Universität Wien
- Herfurth M. 2014: Entwicklung von prozessorientierten Informationssystemen für die industrielle Dienstleistungsbeschaffung, Dissertation, Karlsruhe: Karlsruher Institut für Technologie (KIT)
- Kleuker S. 2009: Grundkurs Software-Engineering mit UML, 11, Wiesbaden: Springer Vieweg
- Wörzberger R. 2010: Management dynamischer Geschäftsprozesse auf Basis statischer Prozessmanagementsysteme, Dissertation, Aachen: Rheinisch-Westfälischen Technischen Hochschule Aachen

Lütkemeier B. & Thöne S. 2001: Prozessorientierte Integration von Softwarekomponenten durch XML-basierte Workflow-Modelle, Diplomarbeit, Paderborn: Universität Paderborn

DIN 3960 1987: Begriffe und Bestimmungen für Stirnräder (Zylinderräder) und Stirnradpaare (Zylinderradpaare) mit Evolventenverzahnung, Berlin: Deutsche Institut für Normung e. V.

DIN ISO 21771 2014, Zahnräder – Zylinderräder und Zylinderradpaare mit Evolventenverzahnung - Begriffe und Geometrie (ISO 21771: 2007), Berlin: Deutsche Institut für Normung e. V.

Kontakt

Dipl.-Ing. Denis Polyakov

TEDATA GmbH, NL DD

Wettiner Platz 10

01067 Dresden

www.tedata.com

Prof. h. c. Dr.-Ing. Willi Gründer

TEDATA GmbH

Königsallee 45

44789 Bochum

www.tedata.com