

ANSÄTZE FÜR EINE FUNKTIONALE REPRÄSENTATION MULTIDISZIPLINÄRER PRODUKTE

Kristin Paetzold

Zusammenfassung

Der Beitrag widmet sich den besonderen Herausforderungen die sich bei der Unterstützung der Prozesse bei der interdisziplinären Produktentwicklung ergeben. Die zunehmend notwendige Kooperation verschiedener Domänen im Entwicklungsprozess (z. B. Maschinenbau, Elektrotechnik, Informationstechnik) verlangt parallel zu den angepassten Vorgehensweisen und Methoden geeignete Beschreibungsformen für die zu entwickelnden Produkte, um Aussagen über den Prozessfortschritt zu gewährleisten. Ziel ist daher die Zusammenstellung von Produktdaten im Sinne eines domänenübergreifenden Produktdatenmodells für einen Functional Mock-up soweit aufzubereiten, dass eine effektiven und effizienten Zusammenarbeit aller Disziplinen möglich wird.

1 Einleitung

Die Entwicklung interdisziplinärer Produkte stellt für viele Unternehmen eine große Herausforderung dar. Dies ist z.T. auf die Schwierigkeiten bei der organisatorischen Einbindung der einzelnen Domänen zurückzuführen, wird aber auch durch deren verschiedenartigen Herangehensweisen bei der Problemlösung geprägt. Auf Basis von bekannten Entwicklungsmethoden aus den einzelnen Domänen und einem allgemeingültigen gesamtheitlichen Ansatz für mechatronische Produktentwicklung [1] ist es zunächst möglich, eine Detaillierung einzelner Prozessschritte unter Berücksichtigung der Verwendung typischer Methoden zur Lösungsfindung zu erarbeiten, ein einheitliches Verständnis für die Vorgehensweisen in den Domänen lässt sich daraus aber nicht ableiten. Zur Beschreibung des Prozessfortschritts und damit eng verbunden natürlich auch des Produktes selbst, muss es das Ziel sein, eine einheitliche Beschreibungsform in einer Art zu finden, die für alle Domänen akzeptierbar ist. Als Ansatz dazu soll die Zusammenstellung der Daten über ein herkömmliches Produktdatenmodells soweit erweitert werden, dass auch für den Prozessfortschritt entsprechende Aussagen ableitbar sind.

Für die Produktentwicklung sind Prozessunterstützungen, die auf der Festlegung starrer vorgegebener Prozessabläufe beruhen, nicht anwendbar. Der Produktentwicklungsprozess entfaltet sich erst in seinem Verlauf. Daher bedarf es vielmehr Adhoc-Ansätze oder hybrider Ansätze zur Modellierung der Prozesse, die vor allem die Entscheidungsfindung für das weitere Vorgehen unterstützen sollen. Unter Berücksichtigung des Kenntnisstandes über das Produkt, die Zielerfüllung und freie Kapazitäten im Unternehmen können dann die nächsten Prozessschritte vom Entwickler geplant werden. Da die Produktentwicklung immer auch als Optimierungsprozess zu betrachten ist, benötigt man zudem für den Iterationsprozess Methoden, die zunächst eine qualitative Bestimmung von Gütekriterien erlauben, die es im Weiteren dann anhand der Anforderungen zu konkretisieren gilt.

2 Multidisziplinäre Sichtweisen auf ein Produkt

Speziell in den Ingenieurwissenschaften lassen sich drei prinzipielle Sichtweisen auf ein Produkt unterscheiden, die wiederum für die spezifischen Entwicklungsmethodiken die Grundlagen liefern. Ausgangspunkt einer jeden Entwicklung, egal in welcher Domäne diese durchgeführt wird, bildet immer das Finalitätsprinzip. Das bedeutet, Produktentwicklung verfolgt immer einen Zweck. Das Produkt wird dabei als die geplante Wirkung auf eine mögliche Ursache, die in der Zukunft auftreten könnte, hin entwickelt [2]. Diese definierte Gesamtfunktion bildet im Allgemeinen die Grundlage und die Initialisierung eines Entwicklungsprozesses (Bild 1), sie wird weiter strukturiert, indem die Gesamtfunktion in Teilfunktionen zerlegt wird. Die Funktion beschreibt die teleologischen Zusammenhänge in einem Produkt, klärt also die Frage, welche Wirkung letztlich erreicht werden soll.



Bild 1: Funktionsorientierte Black-Box-Darstellung

Unter anderem auch durch den Trend hin zu mehr Intelligenz in den technischen Systemen tritt verstärkt die Frage in den Vordergrund, welches Verhalten letztlich erforderlich ist, um den Systemzweck zu erfüllen. Das Verhalten beschreibt, wie und auf welche Art und Weise das Produkt diesen erfüllen soll. Bezogen auf die klassischen Ansätze aus der Produktentwicklung ist dies mit einer näheren Charakterisierung der Teilfunktionen gleichzusetzen. Während eine Teilfunktion z.B. heißen kann „Geschwindigkeit ändern“ ist im Kontext der Mechatronik sofort die Frage zu stellen wie dies geschehen soll, also z.B. kontinuierlich, sprunghaft, nach einem bestimmten Verlauf oder ähnlichem. Bezogen auf die im Bild 1 eingeführte Black-Box-Betrachtung liegt für die Verhaltensperspektive in der Produktentwicklung der Fokus auf den Ein- und Ausgängen in die Black-Box. Das Verhalten des Systems ist dadurch beschrieben, dass auf Basis der gegebenen Eingangsdaten diese so in die erforderlichen Ausgangsdaten transformiert werden, sodass sie letztlich den Systemzweck erfüllen. Diese Betrachtungsweise liegt z.B. der Reglerentwicklung und der Entwicklungsmethodik integrierter Schaltkreise zugrunde. Letztlich müssen alle Entwicklungsmethoden, egal aus welcher Domäne, auf Basis solcher Verhaltensbetrachtungen eine strukturelle Umsetzung erhalten.

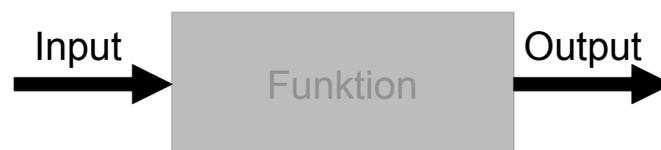


Bild 2: Verhaltensorientierte Black-Box-Darstellung

Der dritte Blickwinkel, aus dem ein technisches System heraus betrachtet werden kann, ist dessen Struktur. Das Verhalten des Systems hängt sehr stark von der inneren Struktur der Black-Box ab. Damit wird beschrieben, aus welchen Komponenten sich das technische System zusammensetzt und wie deren geometrische und relationale Beziehungen untereinander aussehen. So wird letztlich aus einer Black-Box-Betrachtung eine Glas-Box-Betrachtung, weil durch die Festlegung der Struktur auf Basis der zu erfüllenden Funktion ganz klar gesagt wird, wie die Eingangsgrößen in Ausgangsgrößen zu wandeln sind. Diese Perspektive steht üblicherweise im sehr stark geometrieorientierten Maschinenbau im Fokus der Betrachtungen.

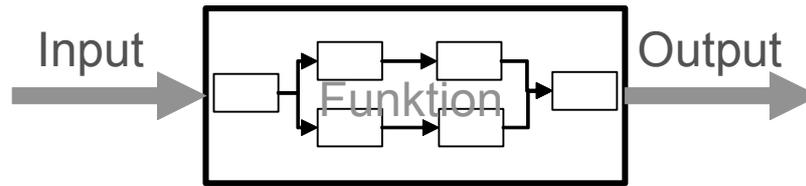


Bild 3: Strukturorientierte Black-Box-Darstellung

Die dargestellten Sachverhalte zeigen vor allem, dass eine Betrachtung einer der drei Sichten Funktion, Verhalten oder Struktur in einer isolierten Form nicht sinnvoll ist, da deren Beschreibungen immer eng mit den jeweils anderen Sichten verknüpft sind [3]. Eine prozessorientierte Betrachtungsweise zeigt aber, dass immer zunächst die Funktion definiert und in Unterfunktionen zerlegt wird, was die Basis für die Verhaltensbeschreibung und letztlich auch für die Strukturauslegung liefert. Die Strukturfestlegungen prägen essentiell das Verhalten eines Systems. Für die Zerlegung der Gesamtfunktion in Teilfunktionen existieren ebenfalls unterschiedliche Möglichkeiten, die sich aber entsprechend in ihrer Verhaltensaussprägung unterscheiden. Die genannten Aspekte sind unabhängig von domäneabhängig vorherrschenden Entwicklungsmethoden zu erkennen. Eine ganzheitliche Produktbetrachtung erfordert immer die Berücksichtigung aller drei Aspekte.

Da die Strukturbeschreibung in den Domänen sehr heterogen ist, erscheint es nicht sinnvoll, diese als Basis für die Prozessgestaltung heranzuziehen. Hier besteht die Gefahr, dass sich die Entwickler in den Domänen missverstehen, Schnittstellen können nicht klar definiert werden, weil die Domänen sehr unterschiedliche Repräsentationen mit unterschiedlicher Bedeutung nutzen. Angemessener erscheint es, die Verhaltensperspektive als Basis für eine Prozessunterstützung heranzuziehen, da zum einen eher ein gemeinsames Verständnis besteht und zum anderen der Zusammenhang zum Systemzweck klarer hervorgeht.

3 Allgemeine Prozessbetrachtungen

3.1 Der Charakter des Produktentwicklungsprozesses

Im Hinblick auf immer kürzer werdende Entwicklungszeiten gewinnen Ansätze zur Prozessunterstützung in der Produktentwicklung immer mehr an Bedeutung. Im Vergleich z.B. zu Geschäftsprozessen oder Fertigungsprozessen stellt dies aufgrund der Charakteristik des Produktentwicklungsprozesses eine große Herausforderung dar. Der Entwicklungsprozess ist ein sehr kreativer Prozess, um innovative Ideen in Produkte umzusetzen braucht der Ingenieur einen gewissen Spielraum. Fest vorgegebene Prozesse, wie dies für Workflow-Ansätze typisch ist, zwingen den Entwickler in ein starres Gerüst, welches seiner Wissensbasis und seinem Erfahrungshintergrund nicht gerecht wird. Sie belasten zudem den Entwickler mit Routinearbeiten, die ihn vom kreativen Denken abhalten. Ziel muss es vielmehr sein, den Ingenieur in seinen Entscheidungen innerhalb des Prozesses zu unterstützen, Wissen und Informationen situationsgerecht zur Verfügung zu stellen und ihn möglichst wenig durch formale Vorgaben einzuengen.

Erschwerend für die Suche nach Methoden zur Prozessunterstützung wirkt sich aus, dass der Entwicklungsprozess sehr unstrukturiert ist. In der Konzeptphase sind zunächst vorrangig strategische Entscheidungen zu treffen, im Verlauf der Entwicklung, also mit zunehmender Konkretisierung des Produktes wird die Anwendung von Richtlinien und Gestaltungsregeln möglich, der Prozess kann klarer strukturiert und damit ist der gezielte Methodeneinsatz besser planbar. Die strategischen Entscheidungen in den frühen Phasen sind abhängig von (sich ändernden) Kundenanforderungen, zur Verfügung stehendem Wissen und Informationen, dem Erfahrungshintergrund im Unternehmen, der Unternehmensphilosophie und unsicheren Randbedingungen, die sich erst im Verlauf der Produktentwicklung konkretisieren.

Dies führt zu Iterationen im Entwicklungsprozess, wodurch die Unstrukturiertheit zum einen zu begründen ist. Es wird aber deutlich, dass die genannten Faktoren im Verlauf der Entwicklung und mit zunehmender Produktkonkretisierung an Bedeutung verlieren.

Iterationen im Entwicklungsprozess resultieren aber auch aus einer mangelnden Kommunikation unter den Beteiligten. Interdisziplinäres Arbeiten unter Ingenieuren mit sehr unterschiedlichen Erfahrungshorizonten und Wissensbasen verschärfen die Kommunikationsproblematik, da diese meist nicht den gleichen Sprachgebrauch haben. Hieraus resultieren vor allem Unklarheiten und Unsicherheiten über die Schnittstellen, was letztlich dazu führt, dass Informationen unzureichend ausgetauscht bzw. erfragt und dann notwendige Prozessschritte nicht angestoßen bzw. rechtzeitig durchgeführt werden.

Ziel einer Prozessunterstützung ist es daher nicht, Methoden zu finden, um den Prozess zu strukturieren, was zu fest vorgebbaren Pfaden in der Problemstellung führen würde. Der Ingenieur wird während der Produktentwicklung immer wieder vor der Frage gestellt, welcher Schritt als nächstes notwendig ist. Er muss Entscheidungen treffen, was erforderlich ist, um den gewünschten Systemzweck zu erhalten und abzusichern. Dazu wendet er als kleinste Prozessgranularität Methoden an, von denen er der Meinung ist, dass sie ihn an das gewünschte Ziel führen. Eine verfeinerte Prozessbeschreibung hat demnach als kleinsten Baustein die Methode bzw. die Richtlinie. Eine weitere Verfeinerung ist nicht sinnvoll, weil dann der administrative Aufwand für den Entwickler wächst und ihm damit letztlich den Spielraum für die Ideenfindung und Lösungssuche einschränkt.

Die Prozessunterstützung soll daher darauf reduziert werden, dass sich nach Anwendung einer Methode eine Entscheidungssituation einstellt, die den weiteren Prozessfortschritt charakterisiert, Entscheidungen müssen in Abhängigkeit vom gegebenen Produkt- und Informationsstand und den Randbedingungen (Unternehmen, Kunde, Ingenieur) getroffen werden, um die Entwicklung effektiv voranzutreiben. Zu ergänzen ist bei einer interdisziplinären Entwicklung, dass genaue Schnittstellenbeschreibung zwischen den Domänen um Teilschritte parallel ablaufen lassen zu können und die Kommunikation zwischen den Beteiligten zu unterstützen.

3.2 Unterstützung des Entscheidungsprozesses

Basis für die Entscheidungsfindung sind prinzipielle Überlegungen zum Ablauf in der Produktentwicklung. Beschreibungen des Prozesses, wie dies aus der VDI 2221 [4] oder nach Pahl/Beitz [5] bekannt sind, erscheinen hier vor allem wegen der vergleichsweise groben Untergliederung nicht sinnvoll. Zur Unterstützung der Entscheidungsfindung soll daher der Property-Driven-Development Ansatz nach Weber [6] herangezogen werden. Dieser Ansatz zur Modellierung von Produkten und Produktentwicklungsprozessen basiert auf der Unterscheidung von Merkmalen und Eigenschaften, die im Bereich der Konstruktion Tradition hat. Nach Andreasen [7] definieren Merkmale ein Produkt und werden vom Entwickler festgelegt. Im Maschinenbau sind dies im Allgemeinen geometrische Parameter, in der Elektrotechnik können dies z.B. Bauelemente sein, in der Informatik Softwarecode. Eigenschaften dagegen beschreiben das Produktverhalten. Sie werden nicht direkt vom Entwickler festgelegt sondern resultieren aus den Merkmalen, müssen aber den geforderten Systemzweck widerspiegeln. Zwischen den Merkmalen und den Eigenschaften bestehen dabei keine eindeutigen Beziehungen. Ein Merkmal kann unterschiedliche Eigenschaften des Produktes beeinflussen, eine Eigenschaft ist im Allgemeinen von mehr als einem Merkmal abhängig.

Durch die Kopplung zwischen den Strukturparameter „Merkmal“ mit dem Verhaltensparameter „Eigenschaft“ gelingt eine ganzheitliche Sichtweise auf das Produkt. Doch auch für die Prozessgestaltung bietet der Ansatz Vorteile. Wird nach jeder Merkmalsfestlegung geprüft, wie sich dies auf bestehende Eigenschaften auswirkt, bzw. welche neuen Eigenschaften für

die Produktgestaltung zu berücksichtigen sind, können daraus Hinweise auf nachfolgend erforderliche Prozessschritte abgeleitet werden.

Ausgangspunkt der Entwicklung bildet neben dem Systemzweck die Anforderungsliste, die eine Ansammlung von Eigenschaften darstellt. Diese geforderten Eigenschaften können zum einen als Präzisierung des Systemzwecks aufgefasst werden, zum anderen beinhaltet die Anforderungsliste aber auch Rand- und Nebenbedingungen. Die Festlegung der Gesamtfunktion und die Untergliederung in Teilfunktionen orientiert sich am Systemzweck, wobei die Anforderungen zu berücksichtigen sind. Die Wissens- und Informationsbasis für das Produkt ist jedoch noch sehr lückenhaft und klein. Mit der Suche nach Lösungsprinzipien werden erste Merkmale festgelegt, die zum einen die geforderten Eigenschaften sicherstellen sollen, zugleich aber auch Eigenschaften hervorrufen, die bei der weiteren Bearbeitung zu berücksichtigen sind. Dieser Prozess macht deutlich, dass die Randbedingungen nicht per se vorgegeben sind, sondern tatsächlich erst im Laufe des Prozesses und in Abhängigkeit von der gewählten Lösung entstehen, weshalb die Informationsgrundlage als unvollständig und unsicher vorausgesetzt werden muss.

Im Rahmen der weiteren Konkretisierung werden in späteren Phasen Strukturentscheidungen getroffen. Der Einsatz gängiger Methoden führt zur Definition von Merkmalen, die immer auch definierbare Eigenschaften mit sich bringen (gewollte und ungewollte). Gleichzeitig ist mit der Konkretisierung eine Art der Hierarchisierung vom Groben ins Detail zu beobachten, da immer mehr Einzelheiten des zukünftigen Produktes herausgearbeitet werden. Die Wissensbasis über das Produkt wächst und zusätzliche Informationen lassen sich entsprechend berücksichtigen und kontextorientiert in die Wissensbasis integrieren.

Damit wird die These verstärkt, dass Strukturmerkmale zwar essentiell für die Produktbeschreibung sind, dabei aber aufgrund der sehr unterschiedlichen Möglichkeiten, was unter Struktur subsumiert werden kann, wenig Aussagegehalt liefert. Interessant auch und gerade für die Schnittstellenbeschreibung sind die Eigenschaften, die mit den Merkmalen verknüpft sind, da diese direkte Aussagen zum Produktverhalten erlauben. Diesem Aspekt gilt es Rechnung zu tragen.

Parallel zur Produktentwicklung werden Produktdaten mittels eines Produktmodells in ein Produktdaten-Management-System eingepflegt. Ziel ist es hierbei unter anderem, allen Beteiligten die aktuellsten Informationen über das Produkt zur Verfügung zu. PDM-Systeme bilden die Wissensbasis für das Produkt über den gesamten Entwicklungsprozess, mit dessen Hilfe ein Digital Mock-up (digital repräsentierte Produktbeschreibung) als Grundlage für das Handhaben, Darstellen und Analysieren für die bekannten Produktinformationen zur Verfügung steht. PDM-Systeme basieren auf CAD-Systemen. Abgelegt werden daher vorrangig geometrieorientierte Informationen. In jüngster Zeit sind die PDM-Systeme jedoch soweit erweitert wurden, dass auch andere Informationen, wie z.B. Toleranzen oder eben Produktdaten aus anderen Domänen, die nicht direkt geometriebezogen sind, als semantische Informationen abgelegt werden können. Damit sind im PDM-System nur Merkmalsdaten abgelegt, die resultierenden Eigenschaften ergeben sich maximal aus einer Analyse oder aus Simulationen auf Basis dieser Merkmale.

Der Digital Mock-up auf Basis des Produktmodells im PDM-System ist daher vorrangig geeignet, um eine Strukturrepräsentation wiederzugeben. Funktionale Zusammenhänge bzw. die tatsächlichen Eigenschaften des Produktes lassen sich maximal über zusätzliche Zwischenschritte ableiten. Daten aus anderen Domänen wie der Schaltkreisen oder Regelungen aus der Elektrotechnik oder Softwarecode aus der Informatik sind in die Repräsentation nicht integrierbar. Die dort gefundenen Merkmale lassen sich nur sinnvoll im Digital Mock-up verarbeiten, wenn geometrieorientierte Strukturinformationen damit verbunden werden können, andere Aspekte wie z.B. Datenflusspläne oder Schaltkreise finden sich zwar im PDM-System, verlieren aber an Aussagegehalt. Ein realistischer Abgleich des Produktmodells mit

der Anforderungsliste ist nicht möglich, weil über den Digital Mock-up aus den Merkmalen allein keine Aussage zum Produktverhalten ableiten lässt.

4 Übergang vom digital Mock-up zum Functional Mock-up

Daher erscheint es sinnvoll, den Digital Mock-up um einen Functional Mock-up zu erweitern, womit sich funktionale Aspekte einbringen lassen. Der Functional Mock-up stellt eine logische Weiterentwicklung des geometrieorientierten Digital Mock-up dar, mit welchem es möglich wird, physikalische Zusammenhänge zwischen den Produktmerkmalen darzustellen [8]. Während der Digital Mock-up vorrangig Merkmale abbildet, erfolgt beim Functional Mock-up die Verknüpfung dieser in der Art, dass Aussagen zu den Eigenschaften des technischen Systems möglich werden. Letztlich entspricht dies einer Abbildung des Verhaltens, welches sich für ein Produkt mittels der gegebenen Merkmale ableiten lässt.

Der Fokus der Betrachtung in der Entwicklung liegt dann nicht mehr allein auf der Struktur, sondern auf der Funktionalität des Produktes. Strukturbetrachtungen fließen jedoch ein, nicht zuletzt auch deshalb, weil der Digital Mock-up als Basis dient. Die Heterogenität der Betrachtung in den Domänen kann dabei aber umgangen werden. Sensor- oder Regelungsfunktionalität sowie die Funktionsweise von Software sind damit sinnvoller in die Problemlösung integrierbar. Strukturelle Betrachtungen treten zugunsten von Verhaltens- und funktionalen Betrachtungen zurück. Eine ganzheitliche Produktbeschreibung wiederum ist der Ausgangspunkt, um Prozessschritte auswählen, parallelisieren und aufeinander abstimmen zu können.

Mittels eines Functional Mock-up lassen sich nicht nur erweiterte Simulationen durchführen, interessant erscheint vor allem die Perspektive, das Produktverhalten prozessbegleitend zu analysieren und damit fortwährend zu prüfen, inwieweit der Systemzweck zufrieden stellend erfüllt ist. Zudem ergibt sich die Möglichkeit, die definierten Merkmale mit den damit verbundenen Eigenschaften zu koppeln, welche zur Beschreibung des Prozessfortschrittes mit der Anforderungsliste verglichen werden können. Die Anforderungsliste als Steuerungsinstrument für den Entwicklungsprozess gewinnt deutlich an Bedeutung. In den frühen Phasen besteht für diese im Allgemeinen noch Klärungs- und Präziserungsbedarf. Dies begründet sich zum einen aus den externen Faktoren (Kundenwünsche ändern sich). Andererseits werden Präziserungen oft erst möglich, wenn klar ist, welche Lösungsprinzipien eingesetzt werden sollen. Gelingt es, die Anforderungsliste während des gesamten Prozesses mit den externen und entwicklungsbedingten Parametern zu aktualisieren, erwächst daraus die Möglichkeit, den Entwicklungsstand auf Basis des Functional Mock-up mit dieser dynamischen Anforderungsliste zu vergleichen und damit Aussagen zum Reifegrad eines Produktes zu treffen. Als Reifegrad soll in diesem Kontext der Grad der Erfüllung der Anforderungen durch den Kunden (Kundensicht) unter Berücksichtigung zusätzlicher Informationen, die sich aus der Wahl des Lösungsprinzips ergeben (Ingenieurssicht) bezeichnet werden.

Der Reifegrad des Produktes spiegelt sich zudem im Detaillierungs- und Ergänzungsgrad der Anforderungsliste wieder. Um diese Anforderungsliste tatsächlich für die Prozessunterstützung nutzbar zu machen, bedarf es allerdings Mechanismen, die es erlauben, die prozessabhängige Fortschreibung der Anforderungsliste zu unterstützen. Um ein Produkt als ausgereift zu klassifizieren, sind zudem Gütekriterien von Nöten, welche z.B. auf einem Änderungsgradienten in der Anforderungsliste beruhen. Führen Veränderungen oder die Definition von Merkmalen nicht zu wesentlichen Änderungen der Eigenschaften oder werden damit Eigenschaften provoziert, die nicht für das Systemverhalten und den Systemzweck relevant sind bzw. dieses nachhaltig negativ beeinflussen, kann dies als Indikator dafür dienen, diesen Teilprozess abubrechen und eine neue Entscheidungssituation herauszufordern. Entsprechend lassen sich Iterationen gezielter steuern, was letztlich zu einer Optimierung des Entwicklungsprozesses beiträgt.

Für die Prozessunterstützung im Allgemeinen und die Unterstützung des Entwicklungsingenieurs in einer Entscheidungssituation im Besonderen erscheint es daher sinnvoll, den Digital Mock-up um einen Functional Mock-up zu erweitern, womit sich funktionale Aspekte einbringen lassen.

5 Literatur

- [1] VDI-Richtlinie 2206: Entwicklungsmethodik für mechatronische Systeme, 2003.
- [2] Hubka, V.: Theorie technischer Systeme. 2. Auflage, Springer Verlag, Berlin, 1984.
- [3] Gero, J.S.; Kannengiesser, U.: A Function-Behaviour-Structure ontology of processes, in JS Gero (ed), Design Computing and Cognition'06, Springer, pp. 407-422; 2006.
- [4] VDI-Richtlinie 2221: Methodik zum Entwickeln und Konstruieren technischer Systeme, 1993.
- [5] Pahl G. Beitz W.: Konstruktionslehre, Methoden und Anwendungen. Springer-Verlag, Berlin, 1997.
- [6] Weber, C.; Werner, H.: Klassifizierung von CAX-Werkzeugen für die Produktentwicklung auf der Basis eines neuartigen Produkt- und Prozessmodells. In: Meerkamm, H. (Hrsg): Beiträge zum 11. Symposium „Design for X“, S. 126-143, Neukirchen, 2000.
- [7] Andreasen, M.M.: System Modelling. PhD-Course on Design Theory and Research. Technical University of Denmark, Lyngby, 1995.
- [8] Krause, F.-L.; Rothenburg, U.: Advanced Product Validation using FunctionalDMU. Presentation, Deutsch-israelisches Symposium, Berlin 2005.

Dr.-Ing. Kristin Paetzold
Lehrstuhl für Konstruktionstechnik
FAU Erlangen-Nürnberg
Martensstraße 9, D-91058 Erlangen
Tel: +49-9131-85-23222
Fax: +49-9131-85-23223
Email: paetzold@mfk.uni-erlangen.de
URL: <http://www.mfk.uni-erlangen.de>

