

M. Eigner, T. Dickopf, H. Apostolov

Interdisziplinäre Konstruktionsmethoden und -prozesse zur Entwicklung cybertronischer Produkte – Teil 1

Interdisciplinary Design Methods and Processes to Develop Cybertronic Products – Part 1

Inhalt: Der unternehmerische Erfolg hängt letztlich davon ab, Ideen möglichst effizient und effektiv in technologisch hochwertige Produkte für globale Märkte zu überführen. Vor diesem Hintergrund kommt dem Produktentwicklungsprozess (PEP) eine zentrale Bedeutung zu. Verfolgt man die Produktentwicklung in den letzten 50 Jahren, so haben der Funktions- und damit auch der Komplexitätsumfang dramatisch zugenommen. Virtualisierung, Integration und Interdisziplinarität zwischen den Disziplinen Mechanik, Elektrik/Elektronik, Software und Dienstleistung sowie Zusammenarbeit zwischen den einzelnen Phasen des Produktlebenszyklus werden zur Grundlage eines modernen PEP. Dieser Artikel soll eine offene Diskussion über die Gestaltung eines solchen interdisziplinären PEP eröffnen.

Abstract: *The success of an enterprise depends on the capability to realize ideas as efficient and effective as possible into high-grade technological products for global markets. With this in mind, the product development process is of central importance. Looking back at the product development in the past 50 years, one can see that the functional capabilities and thereby the complexity of products has increased dramatically. Virtualization, integration and interdisciplinarity between mechanics, electric/electronics, software and services as well as collaboration throughout the different phases of the product development are becoming the basis of new development processes. This article aims to initiate an open discussion about shaping such interdisciplinary product development process.*

1 Einleitung

Statistiken der letzten Jahre bestätigen eine permanente Veränderung des Produktentwicklungsprozesses [1]. Diese Veränderungen resultieren aus veränderten Marktbedingungen sowie

aus neuen Anforderungen an das Produkt aus Kundensicht. Eine Zunahme der Produktkomplexität resultiert einerseits aus deutlich stärkeren „Multi-Market“-Produkten sowie Derivaten und Variantenvielfalt, und andererseits aus dem stetigen Anstieg von Elektronik und Software. Der Einsatz von Elektronik und Software ist in den letzten Jahren stetig gestiegen und liegt beispielsweise in der Automobilindustrie bei etwa 40 %. Wenn mechatronische Komponenten, Produkte oder Systeme nicht nur untereinander kommunizieren, sondern sich auch mit dem Internet der Dinge und Dienste vernetzen, spricht man von Cyber-Physical Systems (CPS) [2, 3] oder cybertronischen Systemen (CTS) [4]. Diese Trends führen zu einem vollständigen Umdenken bei Entwicklungsmethoden, -prozessen und IT-Infrastrukturen für die interdisziplinäre Produktentwicklung. Dieser „mind shift“ basiert auf organisatorischer und systemtechnischer

Unterstützung entlang allen Engineering-Aktivitäten über den gesamten Produktlebenszyklus hinweg, von den frühen Phasen der Anforderungserhebung bis zum Recycling des Produkts über alle Disziplinen (Mechanik, Elektrik/Elektronik, Software und Services) und darüber hinaus über die Grenzen eines Unternehmens entlang der Wertschöpfungskette. Für die disziplinübergreifende Entwicklung von Produkten und Systemen fehlen etablierte, industriell genutzte Methoden, Verfahren und Vorgehensmodelle. Die in den Disziplinen entwickelten heutigen, vollkommen disjunkten Entwicklungsmethoden und -prozesse, führen zu disziplinorientierten Silos, in denen wir ausgebildet werden, denken und arbeiten. Eine Übersicht über disziplinorientierte und mechatronische Vorgehensmodelle in der Produktentwicklung zeigt Bild 1.

Nach einem kurzen Überblick über aktuelle disziplinspezifische und inter-

Autoren

Prof. Dr.-Ing. Martin Eigner
Dipl.-Ing. Thomas Dickopf
Hristo Apostolov, M. Sc.
alle:

Lehrstuhl für Virtuelle Produktentwicklung
Fachbereich Maschinenbau und Verfahrenstechnik

Technische Universität Kaiserslautern
Gottlieb-Daimler-Str. 44
67663 Kaiserslautern
Tel.: 06 31 / 2 05-38 73
E-Mail: eigner@mv.uni-kl.de
www.mv.uni-kl.de

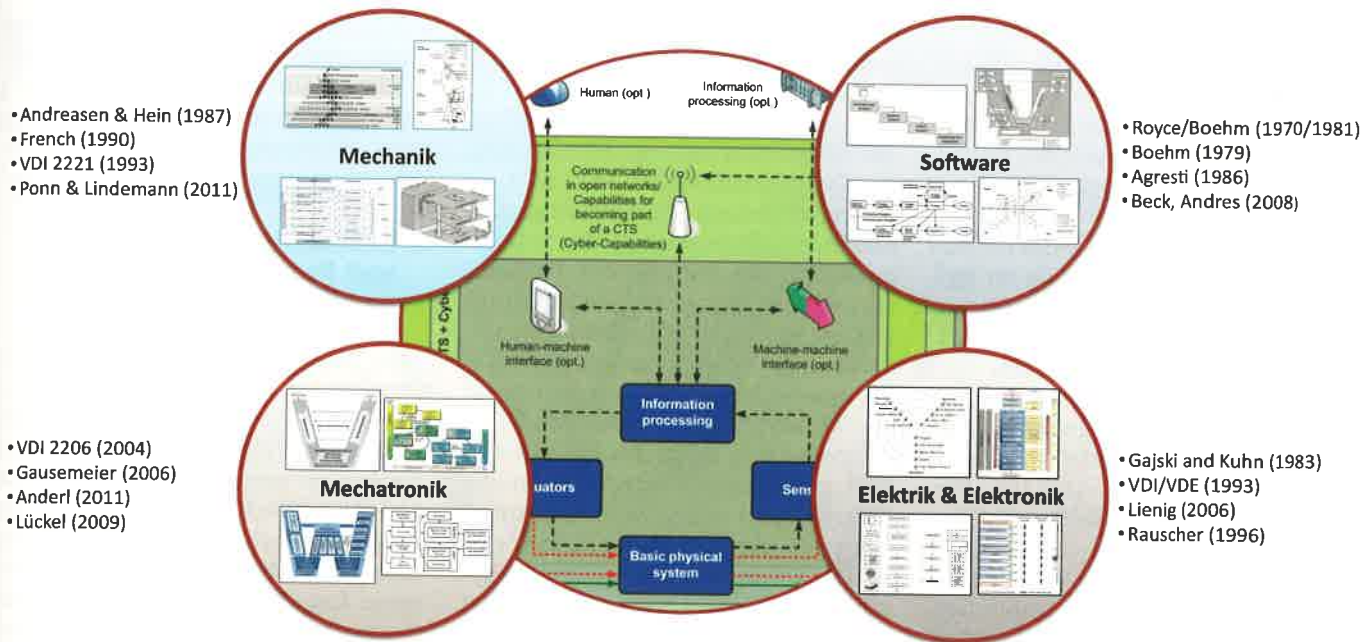


Bild 1

Disziplinen-orientierte Vorgehensmodelle in der Produktentwicklung

disziplinäre Entwurfsansätze und -standards aus den Bereichen Mechanik, Elektronik, Software, Mechatronik und Systems Engineering werden die Forschungsarbeiten des Lehrstuhls für Virtuelle Produktentwicklung (VPE) der TU Kaiserslautern zum Thema interdisziplinäre Produktentwicklung vorgestellt, welche sich zu einem ganzheitlichen Ansatz zur Entwicklung cybertonischer Systeme integrieren lassen.

2 Stand der Technik

Basierend auf einem funktionalen Ansatz wurden in Europa in den 60er und 70er Jahren ganzheitliche Entwurfsmethoden des **Maschinenbaus** vorgeschlagen. Typische Vertreter waren Franke, Kesselring, Rodenacker und insbesondere Pahl/Beitz [10]. Heutzutage basieren fast alle etablierten Entwurfsmethoden im Maschinenbau auf den Artefakten Anforderung, Funktion, Logik bzw. Wirkstruktur und Prinziplösung sowie auf vier wesentlichen Prozessschritten – Planung und Klärung der Aufgabe, Konzeption, Ausführungsentwurf und Detailentwurf. Nach Andreasen [6], French [7], Malmquist und Svensson [8], Ehrlenspiel und Meerkamm [9] sind die wesentlichen Schritte der Produktentwicklung die Definition von Funktionen und deren Realisierung nach prinzipiellen Lösungen. Diese Konzepte wurden auch Grundlage der

VDI-Richtlinie 2221 [11]. Einen anderen Ansatz stellt das Münchner Konkretisierungsmodell von Ponn/Lindemann [12] dar. Hier spielen die Anforderungen eine besondere Rolle, die alle Konkretisierungsebenen der Lösung (Funktions-, Wirk- und Bauebene) beeinflussen.

In der **Elektrotechnik und Elektronik** (E/E) ergibt sich vor allem auf Grund sehr verschiedener Anwendungsgebiete und eines rasanten Technologiewandels ein breiteres Bild an Konstruktionsansätzen. Ein wesentliches Klassifikationsmerkmal für die jeweiligen Entwurfsmodelle ist der Grad an Technologieunabhängigkeit. Zu diesen Modellen gehören zum Beispiel der Top-Down- und der Bottom-Up-Entwurf [5, 13] sowie das daraus hervorgegangene Jo-Jo-Modell. Zu den bekanntesten technologieunabhängigen Modellen zählt das durch Gajski und Kuhn entwickelte Vorgehensmodell, auch als Y-Diagramm bekannt [14]. Es beschreibt die Sichtweisen im Hardwareentwurf, insbesondere für die Entwicklung von integrierten Schaltkreisen in den Domänen Verhalten, Struktur, und Geometrie. Die Domänen sind als Achsen des „Y“ dargestellt. Ein weiterer pragmatischer Ansatz für den Entwurfsprozess integrierter Schaltungen wurde von Lienig entwickelt [15].

Im Bereich der **Softwareentwicklung** zeigen Entwicklungsmethodiken ähnlich der Vorgehensweise in der

Elektronik häufig andere Muster – etwa neben der Funktions- eine starke Verhaltensorientierung [16]. Mit dem Ziel die Softwareentwicklung produktiver und effizienter zu gestalten, wurden unter dem Begriff „Software Engineering“ mehrere phasenorientierte Modelle entwickelt, die den Software-designprozess unterstützen [17], zum Beispiel das Wasserfall-, Prototyp- oder Spiralmodell. Boehm führte 1979 einen Designansatz ein, dessen Konzept sich heute in Ansätzen aller Arten von Ingenieurdomänen durchgesetzt hat – das V-Modell [18]. Aufgrund des hohen Aufwands und der schwierigen Reproduzierbarkeit bei der Erstellung und Pflege komplexer Software erfolgt die Entwicklung zu den genannten Ansätzen in einem strukturierten, streng phasenorientierten und hoch formalisierten Verfahren, wobei der Entwicklungsprozess in überschaubare, zeitlich und inhaltlich begrenzte Phasen unterteilt wird. Durch die Entwicklung von objektorientierten und modellgetriebenen Konzepten im Software Engineering entstand die Unified Modeling Language (UML). Designprozesse, die die Fähigkeiten dieser Konzepte nutzen, sind der Unified Software Development Process (USDP) [19] und der Rational Unified Process (RUP) [20]. Designansätze, die auf schnelle Implementierung und Flexibilität während des Designprozesses abzielen und nicht auf spezifischen Designprozeduren basieren,

nennt man agile Softwareentwicklungsansätze [21].

Der Begriff **Mechatronik** wurde erstmals 1969 von Ko Kikuchi [22] verwendet. Anfänglich bezog sich der Begriff nur auf die elektrotechnische und elektronische Funktionserweiterung mechanischer Bauteile und Geräte. Die Software hat ihre Bedeutung in der Mechatronik erst viel später gewonnen [16]. Mechatronische Designansätze basieren auf einem allgemeinen Verständnis des Designprozesses [23], auf dem mechanischen Designprozess von Pahl und Beitz [24] oder auf Variationen des V-Modells [25][26]. Der etablierteste Ansatz ist die Designrichtlinie VDI 2206 [27].

Parallel wurde seit den 60er Jahren insbesondere bei der amerikanischen Luft- und Raumfahrt und in großen Militärprojekten **Systems Engineering** (SE) als interdisziplinärer, dokumentengetriebener Ansatz zur Entwicklung und Umsetzung komplexer, technischer Systeme und Projekte eingeführt. Systems Engineering beruht auf der Annahme, dass ein System in Hinsicht auf seine Funktionalität mehr ist als die Summe seiner Subsysteme, und dass aus diesem Grund der Fokus notwendigerweise auf die Betrachtung der Gesamtzusammenhänge gelegt werden sollte. Nach den Vorgaben der INCOSE ist das Systems Engineering eine Disziplin, deren Aufgabe die Erstellung und Ausführung eines interdisziplinären Prozesses ist, der garantieren soll, dass Kunden- und Stakeholder-Anforderungen qualitativ hochwertig, zuverlässig, kostengünstig und in vorgegebener Zeit über den gesamten Produktlebenszyklus erfüllt werden können [28]. Mehrere Standards definieren den SE-Prozess (IEEE 1220 [29], ANSI/EIA 632 [30]; ISO/IEC 15288 [31]). Weitere Entwicklungsansätze des Systems Engineering, welche in der Industrie und Forschung Beachtung gefunden haben sind der Harmony-SE-Ansatz des Softwareanbieters IBM [32], die Object-Oriented Systems Engineering Method (OOSEM) [33], die Vitech Model-based Systems Engineering Methodology [34], die JPL State Analysis [35], die Object-Process Methodology [36], das Zackman Framework [37] oder der Rational Unified Process for Systems Engineering (RUP SE) [38], welcher sich aus dem Rational Unified Process [19] der Softwareentwicklung ableitete.

Während klassische Methoden des Systems Engineering dokumen-

basiert sind, ermöglicht **Model-Based Systems Engineering** (MBSE) als Weiterführung des Systems Engineering, ein Entwicklungskonzept, welches auf die Integration von Modelle entlang des Systemlebenszyklus setzt [33]. Es basiert insbesondere auf entwicklungsphasenspezifischen, digitalen Systemmodellen, die entlang des Produktentwicklungsprozesses erstellt und integriert werden. Das Problem der Integration der Komponenten während des Entwicklungsprozesses kann durch die Verwendung formaler Modellierungssprachen möglichst früh in Angriff genommen werden, indem die Korrelationen zwischen Systemanforderungen, Funktionen, Verhalten und Struktur definiert werden. Eine Vielzahl dieser MBSE-Ansätze konzentriert sich insbesondere auf die Spezifikation der Systemarchitektur in der frühen Entwicklungsphase. Die Ansätze sind entweder auf die Anwendung der Systems Modeling Language (SysML) ausgelegt [39], wurden hinsichtlich einer Nutzung der SysML angepasst [40, 41] oder nutzen eine Modellierungssprache, welche sich aus der SysML ableitet [42, 43]. Die durchgängige, modellbasierte Entwicklung ist in der virtuellen Produktentwicklung von zentraler Bedeutung und ist somit auch eine wesentliche Herausforderung an die Optimierung des PEP für mechatronische und insbesondere für cybertronische Produkte beziehungsweise Systeme. Dies umfasst sowohl die Integration der Modelle der verschiedenen Entwicklungsphasen als auch die Integration der Modelle zur Überwachung und Analyse der operativen Nutzung des Systems (Digital Master, Digital Twin).

Zusammenfassend kann festgestellt werden, dass Ansätze für die Gestaltung interdisziplinärer und stark vernetzter Systeme existieren (vgl. Bild 1), aber je nach Forschungsrichtung immer noch eine starke Disziplinorientierung aufweisen [44]. Aktuelle interdisziplinäre Entwurfsansätze und -konzepte, welche auf dem MBSE-Gedanken basieren, bilden eine hervorragende Grundlage für die Entwicklung solcher Systeme, müssen jedoch neu überdacht und erweitert werden, um die komplexen Anforderungen zu erfüllen. Insbesondere können hier die Integration von Disziplinen – auch der vierten Disziplin, der Dienstleistung – das Management von Systeminformationen über den gesamten Lebenszyklus (\Rightarrow Digitalisierung der Prozess-

kette bis hin zum Digital Twin als eine Basis Service-orientierter Geschäftsmodelle) sowie die Gewährleistung einer kontinuierlichen Entwicklung ohne Medienbrüche und Informationsverlust genannt werden.

3 Entwicklungsmethoden und Prozessmodelle für das Engineering

Der Lehrstuhl für Virtuelle Produktentwicklung (VPE) der Technischen Universität Kaiserslautern (TUK) beschäftigt sich mit der Entwicklung und Integration neuer Methoden, Prozesse und IT-Architekturen zur Unterstützung der Entwicklung integrierter, interdisziplinärer und vernetzter Systeme. Um die steigende Komplexität dieser Systeme zu beherrschen und Durchgängigkeit sowie Rückverfolgbarkeit über alle Phasen des Lebenszyklus zu gewährleisten, sind im Rahmen der Forschungsaktivitäten des Lehrstuhls mehrere aufeinander aufbauende Vorgehensmodelle in den letzten Jahren entstanden, die im Folgenden kurz vorgestellt werden.

3.1 MVPE-Modell

Das MVPE-Modell [45] basiert auf dem interdisziplinären V-Modell der VDI-Richtlinie 2206 [27]. Es erweitert das systematische Vorgehen zur Entwicklung mechatronischer Systeme um Aspekte des Model-based Systems Engineering (MBSE) sowie um ein System Lifecycle Management (SysLM)-Konzept zur Verwaltung der entstehenden Daten entlang des gesamten Lebenszyklus (Bild 2). Die Erweiterung um Aspekte des MBSE betrifft insbesondere die linke Seite des V-Modells. Diese lässt sich aus Sicht der Modellbildung in die drei Abschnitte Modellbildung und Spezifikation, Modellbildung und erste Simulation sowie Disziplinspezifische Modellbildung unterteilen. Parallel zu diesen überlappenden Ebenen werden Informationsartefakte beziehungsweise Modellobjekte, die in den Autorensystemen und -sprachen modelliert werden, in Anforderung (A), Funktion (F), Element der logischen Architektur (L) und Element der physikalischen Ausprägung (P) unterteilt. Durch semantische Links zwischen den Modellierungselementen kann sowohl eine „horizontale“ als auch eine „vertikale“ Rückverfolgbarkeit sichergestellt werden. Die „vertikale“ Rückverfolgbarkeit

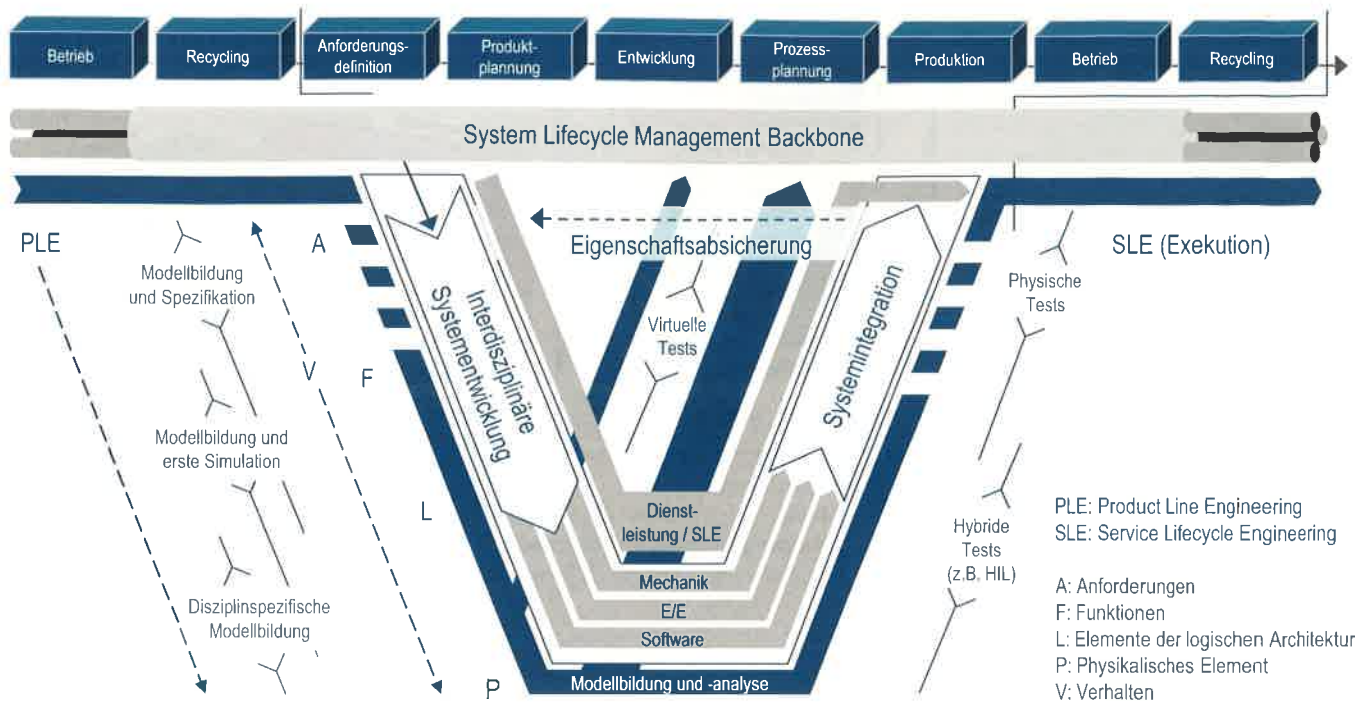


Bild 2
Das MVPE-Modell (nach [45])

erfolgt durch die hierarchische Untergliederung von Systemelementen. Die Verknüpfung zwischen den Modelltypen (A-F-L-P) erlaubt eine „horizontale“ Rückverfolgbarkeit über die Systemspezifikationsphasen [45]. Das MVPE-Modell ist ein übergeordnetes Vorgehensmodell, das seinen Schwerpunkt auf die Prozessebene und die IT-Architektur legt. Es enthält aber auch konstruktionsmethodische Anknüpfungspunkte.

Die Erweiterung des V-Modells umfasst darüber hinaus die Verwaltung lebenszyklusrelevanter Systeminformationen ab der ersten Phase der Produktentwicklung sowie zusätzliche Integrations- und Absicherungsäste, die nur auf digitalen Modellen basieren. Die Integration und Verwaltung der entstehenden Informationen aus der Produktentwicklung sowie allen weiteren Phasen des Lebenszyklus erfolgt über einen sogenannten System Lifecycle Management Backbone [45].

3.2 mecPro² Modellrahmenwerk

Im Rahmen des Forschungsprojektes mecPro² – Modellbasierter Entwicklungsprozess cybertronischer Produkte und Produktionssysteme [4] – wurde eine Beschreibungssystematik entwickelt, welche unter Verwendung der Modellierungssprache SysML das Ziel verfolgt, schrittweise die Zusammen-

hänge cybertronischer Systeme in einem Systemmodell abzubilden. Innerhalb der mecPro² Beschreibungssystematik bildet das mecPro²-Modellrahmenwerk (Bild 3, linke Seite) das Grundgerüst, entlang dessen das zu entwickelnde technische System beschrieben wird. Der Aufbau des Modellrahmenwerks basiert dabei auf den Grundgedanken ausgewählter Entwicklungsmethoden [12, 45, 47] verschiedenster Disziplinen [39].

Das entstehende Systemmodell bildet den Kern der Systemspezifikation und dient als Bindeglied zu den disziplinspezifischen Partialmodellen. Die Beschreibung des technischen Systems erfolgt über mehrere Ebenen mit zunehmender Lösungskonkretisierung. Die Ebenen sind dabei entlang der drei Achsen Detaillierung, Konkretisierung und Variabilität angeordnet. Die Achse der Detaillierung umfasst die Informationsanreicherung des Systemmodells ohne Einschränkung des Lösungsraums. Ist eine weitergehende Detaillierung ohne Lösungskonkretisierung jedoch nicht mehr möglich, erfolgt der Übergang auf eine tiefere Ebene. Da die Lösungskonkretisierung im Regelfall nicht ohne die Betrachtung von Alternativen erfolgen sollte [12], tritt hier zwangsläufig auch die dritte Achse der Variabilität zum Vorschein [39].

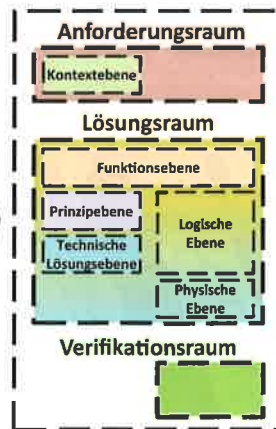
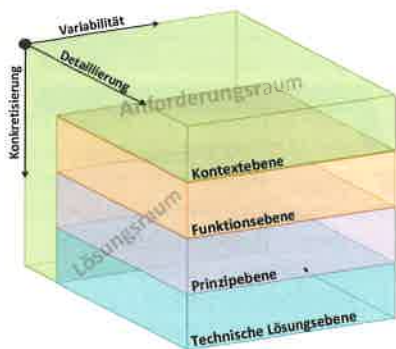
Das Modellrahmenwerk lässt sich zudem in einen Anforderungs- und einen

Lösungsraum unterteilen. Der Anforderungsraum wird durch die Kontextebene abgedeckt und dient der Systemanalyse. Neben der Übersetzung und Synchronisation von natürlichsprachlichen und modellbasierten Anforderungen, erfolgt hier vor allem auch die Betrachtung des zu entwickelnden technischen Systems hinsichtlich der variierenden Kontexte, in dem es als Teil eines oder mehrerer cybertronischer Systeme eingesetzt werden kann. Der Lösungsraum umfasst die nachfolgenden drei Ebenen und dient dem Systemdesign. Die Funktionsebene dient dabei der Erzeugung einer lösungsneutralen Beschreibung der Systemfunktionalität in Form einer funktionalen Systemarchitektur. Die Wahl des Konzepts zur Realisierung der gewünschten Systemfunktion erfolgt entlang der Prinzip- und technischen Lösungsebene. Während auf der erstgenannten Ebene das wesentliche Prinzip, also die Grundidee, zur Realisierung einer Funktion bestimmt wird, erfolgt auf der letzteren Ebene die Wahl der technischen Umsetzung dieser Idee.

3.3 Systemkonkretisierungsmodell nach Pfenning

Das Systemkonkretisierungsmodell (SKM) [46] (Bild 3, rechte Seite) baut wie das mecPro²-Modellrahmenwerk

mecPro² Modellrahmenwerk
als Bestandteil der mecPro² Beschreibungssystematik



Systemkonkretisierungsmodell
nach Pfenning

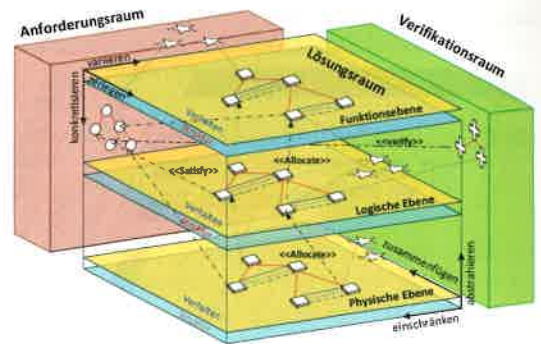


Bild 3

Gegenüberstellung des mecPro²-Modellrahmenwerks und des Systemkonkretisierungsmodells nach Pfenning

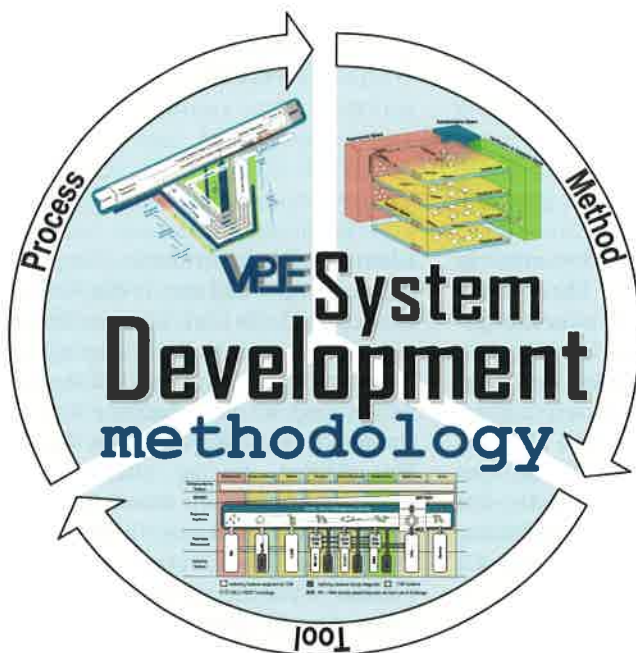


Bild 4
Die VPE System-
Development
Methodology

werden sogenannte Zwischenartefakte eingeführt, die in MBSE-spezifischen Modellierungsmethoden auftauchen und methodisch die Lösungsfindung auf allen Ebenen des SKMs und bei deren Übergang unterstützen. Diese Zwischenartefakte werden allerdings nicht wie die übrigen Lösungselemente (auch Anker-elemente genannt) in den PLM-Backbone übertragen, welcher eine rein aus AFLPT bestehende Ontologie vorsieht.

3.4 Vergleich der Ansätze und Ausblick

Wie in Bild 3 zu sehen, verfügen das mecPro²-Modellrahmenwerk und das SKM über eine große Menge an Anknüpfungspunkten zum MVPE-Modell. Bei den ersteren beiden ist vor allem die Aufteilung des Modells in einen Anforderungs- und einen Lösungsraum zu nennen, wobei insbesondere der Lösungsraum sich bei beiden Ansätzen in einen dreidimensionalen Raum entlang der Achsen zur Konkretisierung und Detaillierung des Systems sowie der Darstellung und Betrachtung von Lösungsvarianten aufspannen lässt. Beide Ansätze verfolgen wie auch das MVPE Modell den A-F-L-P-Ansatz, jedoch auf unterschiedlichen Detaillierungsstufen. Während das SKM und das MVPE-Modell den Ansatz 1:1 abdecken, befindet sich das mecPro²-Modellrahmenwerk auf einer detaillierteren Ebene. Zum einen werden die Anforderungen auf der Kontextebene aus einer natürlich-sprachlichen Form in eine modellbasierte überführt und zum anderen ist der Findungsprozess der logischen Lösungen über die Prinzip- und tech-

ebenfalls auf dem Münchner Produktkonkretisierungsmodell (MKM) nach Lindemann/Ponn auf [12], um hier vor allem das Konzept der Abstraktionsebenen, der orthogonalen Positionierung des Anforderungsraumes auf den Lösungsraum und die generelle Dreidimensionalität zur Realisierung der grafischen Einordnung der Achsen Zerlegen/Zusammenfügen, Variieren/Einschränken und Abstrahieren/Konkretisieren zu verwirklichen.

Darüber hinaus übernimmt es aus anderen disziplinspezifischen Vorgehensmodellen Konzepte, wie den Verifikationsraum, um der rechten Seite des V-Modells gerecht zu werden und einen Raum für ausführbare Modelle (Simulationsmodelle) zu schaffen. Zum anderen erkennt es wie im MBSE

oder auch im Vorgehensmodell nach Gajski [14] an, dass auf jeder Ebene sowohl Struktur (Architektur) als auch Verhalten beschrieben werden kann. Der Lösungsraum als solcher ist in UML/SysML beschrieben. Das SKM sieht zudem explizit die Verwendung von Modelltransformationen aus dem Systemmodell in Simulationsmodelle (akausale und kausale 1D-Simulation) vor. Neben der Tatsache, dass es sich beim SKM um ein interdisziplinäres Makrovorgehensmodell handelt, versucht es zudem eine Überführungsmöglichkeit aus feingranularen Graphen (Systemmodell) in einen eher grobgranularen AFLPT-Graphen (Anforderungen, Funktionen, Logische Elemente, Physische Elemente und Testfälle) zu verwirklichen. Hierzu

nische Lösungsebene gestreckt. Die physische Ausgestaltung der logischen Lösungen stand jedoch nicht mehr im Fokus des Forschungsprojektes und ist somit genau so wenig im Modellrahmenwerk zu finden, wie der Verifikationsraum des SKM, welcher wiederum die rechte Seite des „V“ des MVPE-Modells abdeckt. Allgemein ist anzumerken, dass sowohl das mecPro²-Modellrahmenwerk als auch das SKM das zu entwickelnde System eher hinsichtlich des Problemlösungsprozesses und aus einem methodischen Blickwinkel betrachten, während das MVPE-Modell den Fokus mehr auf die übergeordneten Prozessschritte und die oberste Ebene der IT-Architektur über den gesamten Lebenszyklus legt.

Aufgrund der großen Schnittmenge der Entwicklungsansätze entstand am Lehrstuhl für Virtuelle Produktentwicklung die Idee, diese in einem ganzheitlichen Ansatz – der „^{VPE}SystemDevelopment^{Methodology}“ – für die Entwicklung cybertronischer Systeme zusammenzuführen. Während das MVPE-Modell im Sinne der Definition einer Systems Engineering Methodologie nach Martin [48] für die Darstellung des Prozesses übernommen werden kann und lediglich hinsichtlich der Lebenszyklusphasen heutiger technischer Systeme angepasst wird, verschmelzen das mecPro²-Modellrahmenwerk sowie das SKM nach Pfenning zum Kaiserslauterner Systemkonkretisierungsmodell (KSKM), wel-

ches den methodischen Support des Prozesses insbesondere während der Systementwicklung bereitstellt. Die toolseitige Unterstützung und Umsetzung von Prozess und Methode erfolgt entlang des Fünf-Ebenen-IT-Architektur-Konzepts zur Bildung eines digitalen Modells in der Unternehmens-IT-Architektur [49][50]. Bild 4 zeigt die wesentlichen Bestandteile der neuen ^{VPE}SystemDevelopment^{Methodology}.

Eine detaillierte Beschreibung dieses ganzheitlichen Ansatzes zur Entwicklung und Verwaltung cybertronischer Systeme über alle Phasen des Lebenszyklus, sowie eine detaillierte Beschreibung des Zusammenspiels der einzelnen Komponenten erhalten Sie im zweiten Teil dieses Artikels.

Literatur

- [1] Porter, M.; Heppelmann, J.: Wie smarte Produkte den Wettbewerb verändern. In: Harvard-Business-Manager – das Wissen der Besten. Jhr. 36, HeftNr. 12 (2014), S. 34–60. – ISSN: 0945–6570.
- [2] Lee, E. A.: Cyber Physical Systems: Design Challenges. Technical Report No. UCB/EECS-2008–8, University of California, Berkeley, 2008
- [3] Broy, M.: Cyber-Physical Systems – Innovation durch softwareintensive eingebettete Systeme. Springer-Verlag, Berlin, 2010
- [4] Eigner, M.; Koch, W.; Muggeo, C.: Modellbasierter Entwicklungsprozess cybertronischer Systeme. Springer, Berlin, 2017
- [5] Ponn, J.; Lindemann, U.: Konzeptentwicklung und Gestaltung technischer Produkte – Systematisch von Anforderungen zu Konzepten und Gestaltungs-lösungen. Springer, Berlin, 2011
- [6] Andreasen, M.M.: Machine Design Methods Based on a Systematic Approach. Ph.D. Thesis, Lund University, 1980
- [7] French, M. J.: Conceptual design for engineers. Springer, London, 1999
- [8] Malmqvist, J.; Svensson, D. (Hrsg.): A Design Theory Based Approach Towards Including QFD Data In Product Models. Proceedings of the 1999 ASME Design Engineering Technical Conferences. Las Vegas, Nevada, USA, 1999
- [9] Ehrlenspiel, K.; Meerkamm, H.: Integrierte Produktentwicklung. Carl Hanser, München, 2013
- [10] Pahl, G.; Beitz, W.: Konstruktionslehre. Springer Vieweg, Berlin, 1977
- [11] VDI 2221 – Methodik zum Entwickeln und Konstruieren technischer Systeme und Produkte. Beuth, Berlin, 1994
- [12] Ponn, J.; Lindemann, U.: Konzeptentwicklung und Gestaltung technischer Produkte – Systematisch von Anforderungen zu Konzepten und Gestaltungs-lösungen. Springer-Verlag, Berlin, Heidelberg, 2011
- [13] Rauscher, R.: Entwurfsmethodik hochintegrierter anwendungsspezifischer digitaler Systeme. Pro-Universitate-Verlag, Sinzheim, 1996
- [14] Gajski, D.D.; Abdi, S.; Gerstlauer, A.; Schirner, G.: Embedded system design. Springer, Boston, 2009
- [15] Lienig, J.: Layoutsynthese elektronischer Schaltungen – Grundlegende Algorithmen für die Entwurfsautomatisierung. Springer Berlin Heidelberg, 2006
- [16] Andreasen, M. M.: Vorgehensmodelle und Prozesse für die Entwicklung von Produkten und Dienstleistungen. In: (Schäppi, B.; Andreasen, M. M.; Kirchgeorg, M.; Radermacher, F. J. Eds.): Handbuch Produktentwicklung, Carl Hanser, pages 247–263, München, 2005
- [17] Pomberger, G.; Pree, W.: Software-Engineering. Carl Hanser, München, 2004
- [18] Boehm, B.W.: Guidelines for Verifying and Validating Software Requirements and Design Specifications. Proceeding of the , 79 Euro IFIP Conference, London, September 25–28, North-Holland Pub. Co. Amsterdam, 1979
- [19] Rumbaugh, J.; Jacobson, I.; Booch, G.: The Unified Modeling Language Reference Manual, Pearson Education, 2010
- [20] Kruchten, P.: The rational Unified Process (Addison-Wesley object technology series). Addison-Wesley, Reading, 1999
- [21] Beck, K.; Andres, C.: Extreme Programming Explained: Embrace Change, 2nd Edition. Addison-Wesley Professional, 2005
- [22] Harashima, F.; Tomizuka, M.; Fukuda, T.: Mechatronics – What Is It, Why, and How? An editorial, IEEE/ASME Transactions on Mechatronics, Vol. 1 No. 1, 1996
- [23] Isermann, R.: Mechatronische Systeme. Grundlagen. Springer, Heidelberg, 2008
- [24] Gausemeier, J.; Lückel, J.: Entwicklungsumgebungen Mechatronik: Methoden und Werkzeuge zur Entwicklung mechatronischer Systeme. Heinz Nixdorf Institut, Paderborn, 2000
- [25] Bender, K.: Embedded Systems – qualitätsorientierte Entwicklung. Springer, Heidelberg, 2005
- [26] Nattermann, R.; Anderl, R.: Approach for a Data-Management-System and a Proceeding-Model for the Development of Adaptronic Systems. Proceedings of the IMECE2010, Vancouver, Canada, ASME, New York, 2010
- [27] VDI 2206: Entwicklungsmethodik für mechatronische Systeme – Design methodology for mechatronic systems. Beuth, Berlin, 2004
- [28] INCOSE Systems Engineering Handbook: A Guide for System Life Cycle Processes and Activities. Wiley, New Jersey, 2015
- [29] IEEE 1220–2005 – IEEE Standard for Application and Management of the Systems Engineering Process. IEEE, New York, N.Y., 2011
- [30] ANSI/EIA 632: Processes for Engineering a System. SAE International, USA, 2003
- [31] ISO/IEC/IEEE 15288:2015 Systems and Software Engineering –System Life Cycle Processes. 2015
- [32] Hoffmann, H.: Systems Engineering Best Practices with the Rational Solution for Systems and Software Engineering – Deskbook Release 4.1 – Model-Based Systems Engineering with Rational Rhapsody and Rational Harmony for Systems Engineering. IBM Corporation, 2011

M. Eigner, T. Dickopf, H. Apostolov

Interdisziplinäre Konstruktionsmethoden und -prozesse zur Entwicklung cybertronischer Produkte – Teil 2

Interdisciplinary Design Methods and Processes to Develop Cybertronic Products – Part 2

Inhalt: Der unternehmerische Erfolg hängt letztlich davon ab, Ideen möglichst effizient und effektiv in technologisch hochwertige Produkte für globale Märkte zu überführen. Vor diesem Hintergrund kommt dem Produktentwicklungsprozess (PEP) eine zentrale Bedeutung zu. Verfolgt man die Produktentwicklung in den letzten 50 Jahren, so haben der Funktions- und damit auch der Komplexitätsumfang dramatisch zugenommen. Virtualisierung, Integration und Interdisziplinarität zwischen den Disziplinen Mechanik, Elektrik/Elektronik, Software und Dienstleistung sowie Zusammenarbeit zwischen den einzelnen Phasen des Produktlebenszyklus werden zur Grundlage eines modernen PEP. Dieser Artikel soll eine offene Diskussion über die Gestaltung eines solchen interdisziplinären PEP eröffnen.

Abstract: *The success of an enterprise depends on the capability to realize ideas as efficient and effective as possible into high-grade technological products for global markets. With this in mind, the product development process is of central importance. Looking back at the product development in the past 50 years, one can see that the functional capabilities and thereby the complexity of products has increased dramatically. Virtualization, integration and interdisciplinarity between mechanics, electric/electronics, software and services as well as collaboration throughout the different phases of the product development are becoming the basis of new development processes. This article aims to initiate an open discussion about shaping such interdisciplinary product development process.*

1 Einleitung

Statistiken der letzten Jahre bestätigen eine permanente Veränderung des Produktentwicklungsprozesses [1]. Diese Veränderungen resultieren aus veränderten Marktbedingungen sowie

aus neuen Anforderungen an das Produkt aus Kundensicht. Eine Zunahme der Produktkomplexität resultiert einerseits aus deutlich stärkeren „Multi-Market“-Produkten sowie Derivaten und Variantenvielfalt, und andererseits aus dem stetigen Anstieg von Elektronik und Software. Der Einsatz von Elektronik und Software ist in den letzten Jahren stetig gestiegen und liegt beispielsweise in der Automobilindustrie bei etwa 40 %. Wenn mechatronische Komponenten, Produkte oder Systeme nicht nur untereinander kommunizieren, sondern sich auch mit dem Internet der Dinge und Dienste vernetzen, spricht man von Cyber-Physical Systems (CPS) [2, 3] oder cybertronischen Systemen (CTS) [4]. Diese Trends führen zu einem vollständigen Umdenken bei Entwicklungsmethoden, -prozessen und IT-Infrastrukturen für die interdisziplinäre Produktentwicklung. Dieser „mind shift“ basiert auf organisatorischer und sys-

temtechnischer Unterstützung entlang allen Engineering-Aktivitäten über den gesamten Produktlebenszyklus hinweg, von den frühen Phasen der Anforderungserhebung bis zum Recycling des Produkts über alle Disziplinen (Mechanik, Elektrik/Elektronik, Software und Services) und darüber hinaus über die Grenzen eines Unternehmens entlang der Wertschöpfungskette. Für die disziplinübergreifende Entwicklung von Produkten und Systemen fehlen etablierte, industriell genutzte Methoden, Verfahren und Vorgehensmodelle. Die in den Disziplinen entwickelten heutigen, vollkommen disjunkten Entwicklungsmethoden und -prozesse, führen zu disziplinenorientierten Silos, in denen wir ausgebildet werden, denken und arbeiten.

Dieser Artikel knüpft nahtlos an den ersten Teil dieser Beitragsserie an und liefert nach einer kurzen Definition des Begriffs „Methodologie“ eine de-

Autoren

Prof. Dr.-Ing. Martin Eigner
Dipl.-Ing. Thomas Dickopf
Hristo Apostolov, M. Sc.
alle:

Lehrstuhl für Virtuelle Produktentwicklung
Fachbereich Maschinenbau und Verfahrenstechnik
Technische Universität Kaiserslautern
Gottlieb-Daimler-Str. 44
67663 Kaiserslautern
Tel.: 06 31 / 2 05-38 73
E-Mail: eigner@mv.uni-kl.de
www.mv.uni-kl.de

taillierte Beschreibung der „^{VPE}System-Development_{Methodology}“, welche einen ganzheitlichen Ansatz zur Entwicklung und Verwaltung cybertronischer Systeme über alle Phasen des Lebenszyklus repräsentiert. Nach einer allgemeinen Einführung der ^{VPE}System-Development_{Methodology} folgt in den weiteren Kapiteln eine detaillierte Beschreibung der einzelnen Bestandteile und deren Zusammenhänge.

2 Definition des Begriffs Methodologie

Der im Folgenden genutzte Begriff „Methodologie“ – welcher im Rahmen dieses Beitrags einen ganzheitlichen Ansatz zur Entwicklung cybertronischer Systeme beschreibt – basiert auf den Definitionen nach Martin [5] und Estefan [6]. Martin [5] definiert Methodologie als eine Sammlung zusammengehöriger Prozesse, Methoden und Werkzeuge zur Lösung eines spezifischen Problems. Der Prozess beschreibt dabei eine logische Abfolge von Aufgaben, um ein bestimmtes Ziel zu erreichen. Er beschreibt somit „WAS“ zu tun ist. Die Methode hingegen beschreibt das „WIE“, also die Techniken, die genutzt werden sollen, um die Aufgaben des Prozesses auszuführen. Unter einem Werkzeug wird ein Instrument verstanden, welches die Effizienz der Ausführung einer Aufgabe verbessert, indem es für die Durchführung einer bestimmten Methode angewendet wird. Folglich verbessert es sowohl das „WAS“ als auch das „WIE“ des Systementwurfs. Als viertes essentielles Element für einen erfolgreichen Systementwurf sieht Martin [5] die Umwelt an. Sie umfasst alle externen Faktoren, welche das Handeln der Entwickler beeinflussen. Bild 1 verdeutlicht die Zusammenhänge der genannten Bestandteile einer Methodologie mit zusätzlichem Bezug auf die Themen Mensch und Technologie.

Estefans Definition einer MBSE-Methodologie [6] basiert auf den von Martin [5] genannten Merkmalen, adaptiert diese jedoch auf die Disziplin des Systems Engineering und den Entwurf eines Systems in einem modellbasierten beziehungsweise modellgetriebenen Kontext. Zudem sollte erwähnt werden, dass jede Methode wiederum ein Prozess sein beziehungsweise einen beinhalten kann. Denn sobald eine Methode aus einer Folge von Abläufen besteht, wird das

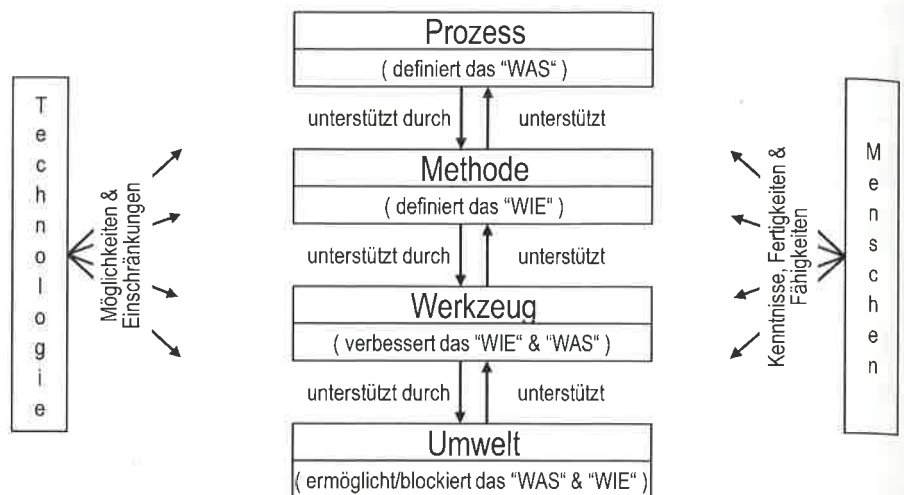


Bild 1

Bestandteile einer Methodologie in Bezug auf die Themen Mensch und Technologie. (eigene Abb. nach Martin [5])

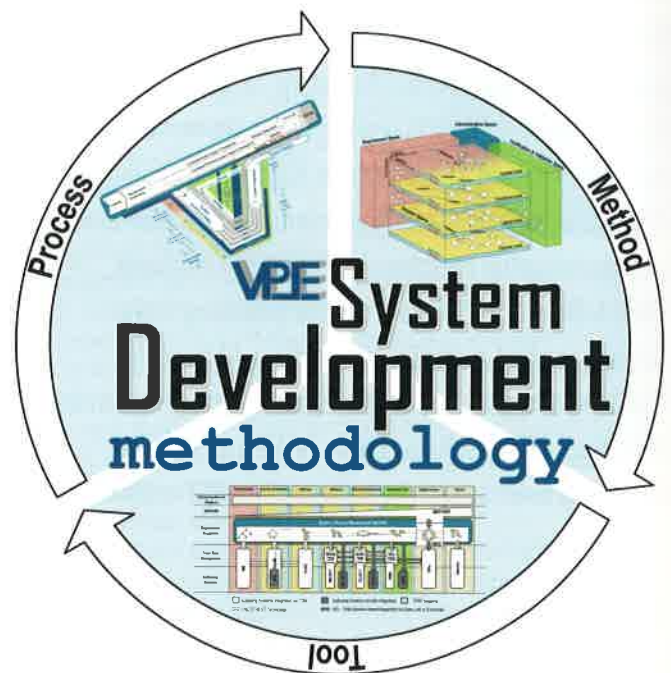


Bild 2

^{VPE}SystemDevelopment_{Methodology}

„WIE“ auf der höheren Abstraktions-ebene zum „WAS“ der nächstniedrigeren. [5, 6]

3 ^{VPE}SystemDevelopment_{Methodology}

Die ^{VPE}SystemDevelopment_{Methodology} beschreibt ein ganzheitliches Makro-Vorgehen für die Entwicklung cybertronischer Systeme im Kontext des Internet of Things. Mit Bezug auf die VDI-Richtlinie 2206 [7] wird im Kontext dieses Artikels unter einer sogenannten Makro-Methodologie ein allgemeiner Leitfaden für die makroskopische Bestimmung des Vorgehens zur Entwicklung cybertronischer Systeme verstanden. Die ^{VPE}SystemDevelopment_{Methodology} als Makro-Methodolo-

gie übernimmt und vereint dabei Konzepte verschiedener interdisziplinärer und disziplinspezifischer Entwicklungsansätze (Eine detaillierte Übersicht dieser Ansätze finden Sie im ersten Teil dieser Beitragsserie.) und passt diese an die Anforderungen und Rahmenbedingungen der Cybertronik an. Angelehnt an die Definitionen einer Methodologie nach Martin [5] und Estefan [6] besteht die ^{VPE}SystemDevelopment_{Methodology} aus den folgenden drei wesentlichen Bestandteilen (Bild 2):

- das für den Kontext des Internet of Things angepasste MVPE-Modell;
- das neue Kaiserslauterner Systemkonkretisierungsmodell (KSKM);
- ein 5-Ebenen-IT-Architekturkonzept.

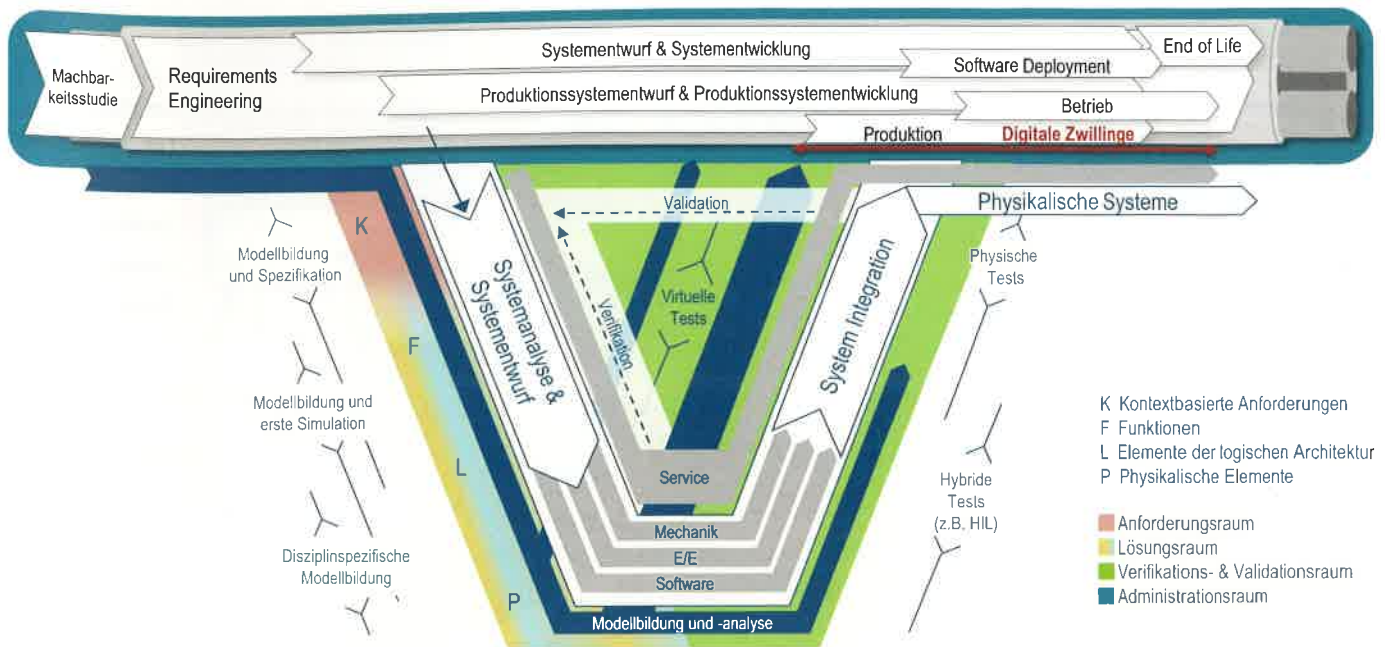


Bild 3
Das MVPE-Modell als Prozessbaustein der ^{VPE}SystemDevelopment Methodik

Während das MVPE-Modell – mit detailliertem Fokus auf den konzeptionellen Entwurf und die Entwicklung – die allgemeinen Lebenszyklusphasen eines Systems beschreibt (Prozess), bündelt das KSKM Techniken zur Erfüllung der Aufgaben, die während der Phasen des konzeptionellen Entwurfs und der disziplinspezifischen Entwicklung anfallen (Methode). Das fünfstufige IT-Architekturkonzept beschreibt einen ergänzenden Ansatz für die Tool-basierte Erstellung und das Management von Systemdaten entlang des gesamten Lebenszyklus (Tool).

4 MVPE-Modell

Erstmals eingeführt im Jahre 2012 [8], wurde das MVPE-Modell über die Jahre mehrmals angepasst und erweitert [9, 10]. Im Hinblick auf die zunehmende Digitalisierung in der Industrie und auf Basis gewonnener Erkenntnisse aus Forschungsprojekten im Kontext von Industrie 4.0, IoT und Industrial Internet [4, 11, 12] waren jedoch weitere Anpassungen und Erweiterungen notwendig. Diese betreffen im Allgemeinen alle Phasen des Lebenszyklus, jedoch im Detail die frühen Phasen der Entwicklung (Bild 3).

Roubanov [13], Cadet et al. [14], Sinnwell et al. [15] sowie Zafirov [16] haben gezeigt, dass die Verwendung von MBSE-Konzepten nicht nur hilft, die Komplexität heutiger Systeme zu

bewältigen, sondern auch um deren Produktionssysteme zu beschreiben. Dies ermöglicht zudem einen früheren Daten- und Informationsaustausch zwischen der Produktentwicklung und der Produktionssystementwicklung. Resultierend führt die Verwendung von MBSE in diesem Kontext nicht nur zu einem früheren Start der Produktionssystementwicklung, sondern ermöglicht folglich auch einen früheren Produktionsstart (SOP) des Systems (Bild 3). Dabei hilft MBSE, die komplexen Abhängigkeiten zwischen den verschiedenen Disziplinen und Abteilungen zu verstehen und den bereits genannten Austausch von Informationen und Daten zwischen diesen Abteilungen zu verbessern.

Wie bereits in der Einleitung erwähnt, resultiert die zunehmende Produktkomplexität heutiger Systeme unter anderem aus der stetigen Zunahme elektronischer Komponenten und der damit verbundenen eingebetteten Software [4]. Dies bringt, insbesondere in Betrachtung der Themen IoT und Industrial Internet, auch Vorteile mit sich. So lassen sich die Zustände und das Verhalten heutiger Systeme (und deren Elemente) während ihrer Verwendung überwachen und analysieren. Dies ermöglicht eine sofortige oder schnellere Reaktion auf auftretende Fehler, Ausfälle oder neue Anforderungen, indem Services ausgelöst, neue Software oder softwarebasierte Funktionen bereitgestellt und Hard-

ware-Komponenten angepasst oder ersetzt werden.

In Bezug auf die frühen Phasen des Entwicklungsprozesses (Bild 3, linke Seite des „V“) wurde der frühere RFLP-Ansatz zu einem KFLP-Ansatz modifiziert. Dies bedeutet nicht, dass Anforderungen ihre Relevanz im Entwicklungsprozess verloren haben. Es verdeutlicht vielmehr, dass der Kontext (K), in dem sich das System während seines Betriebs als Teil eines CTS bzw. CPS (zum Beispiel ein Automobil, das im Kontext des autonomen Parkens oder als Teil eines autonomen Ampelsystems) oder bereits schon während seiner Produktion befindet, immer mehr an Bedeutung gewinnt und Anforderungen ein Teil dieses Kontexts sind. Mit anderen Worten, die Systemanforderungen werden in Bezug auf den Kontext betrachtet, in dem das System agieren wird. Die hohe Bedeutung des Requirements Engineering entlang des gesamten Systemlebenszyklus ist in Bild 3 anhand seines Phasenpfeiles ersichtlich.

Bezogen auf die VPE-System-Development-Methodology bildet das MVPE-Modell als dessen Prozessbaustein das eigentliche Grundgerüst der Methodologie, da es vorgibt, was zu welcher Zeit durch wen zu erfüllen ist. Eine detaillierte Beschreibung der einzelnen Phasen zur Entwicklung cybertronischer Systeme ist im Rahmen des Forschungsprojekts mecPro² [4] entstanden.

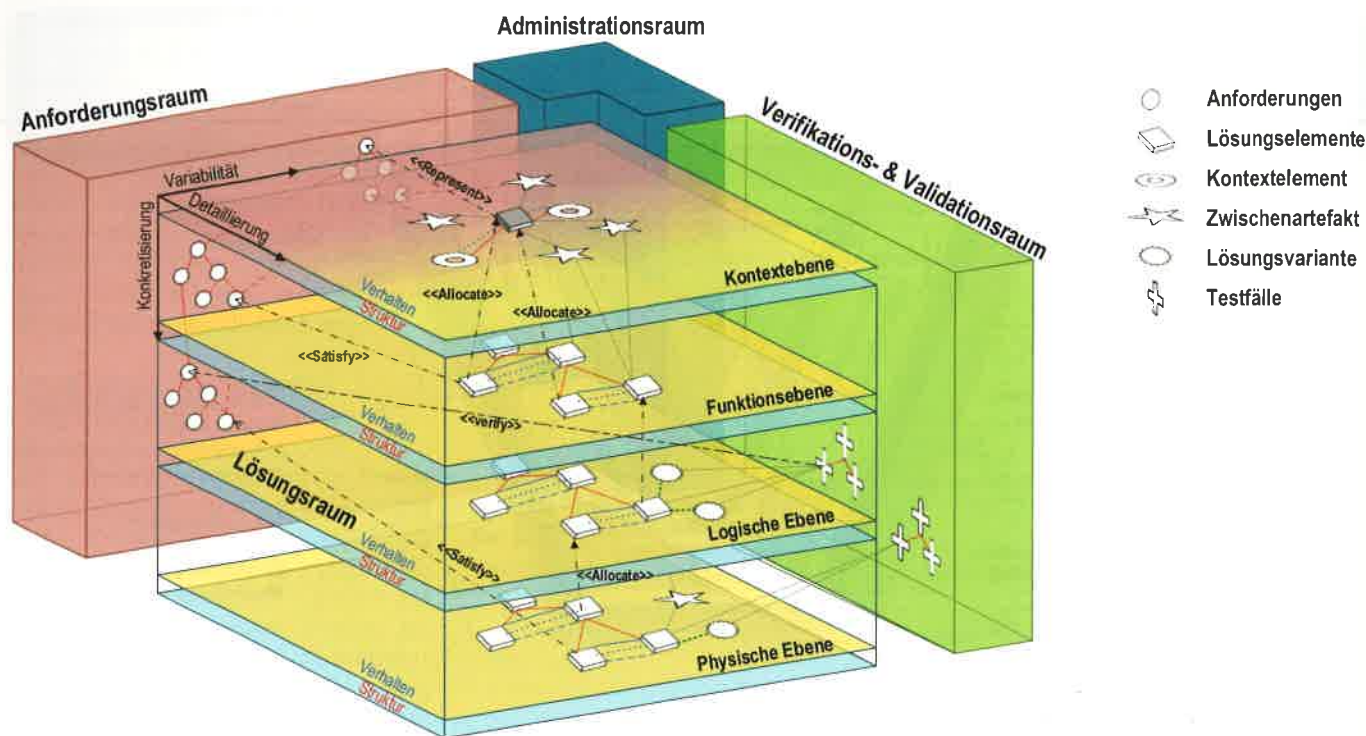


Bild 4

Das Kaiserslauterner Systemkonkretisierungsmodell als Methodenbaustein der ^{VPE}SystemDevelopment Methodology

5 Kaiserslauterner Systemkonkretisierungsmodell

Basierend auf den Vorarbeiten zum mecPro²-Modellrahmenwerk [17, 18, 19] und dem Systemkonkretisierungsmodell (SKM) nach Pfenning [20] wurden beide Ansätze aufgrund ihrer hohen Schnittmenge an gemeinsamen Aspekten zu einem Produktmodell zusammengefügt. Das sogenannte Kaiserslauterner Systemkonkretisierungsmodell (KSKM) beschreibt ein Produktmodell zur systematischen Entwicklung cybertronischer Systeme (Bild 4) und unterstützt das MVPE-Modell als methodische Komponente der ^{VPE}SystemDevelopment Methodology. Die Achse der Konkretisierung liegt dabei parallel zur prozessualen Abfolge im MVPE-Modell.

Die grundlegende Struktur des Kaiserslauterner Systemkonkretisierungsmodells basiert auf dem SKM nach Pfenning [20] sowie dem Münchener Produktkonkretisierungsmodell nach Ponn und Lindemann [21] und wurde um die wesentlichen Konzepte des mecPro²-Modellrahmenwerks [17, 18, 19] ergänzt. Im Allgemeinen lässt sich das KSKM in die folgenden vier Räume unterteilen:

- Anforderungsraum,
- Lösungsraum,
- Verifikations- und Validierungsraum,

- Administrationsraum.

Der Anforderungsraum enthält natürlichsprachliche Kunden- und Systemanforderungen, die in Anforderungsmodelle einer höheren Formalität übersetzt werden. Diese Anforderungen beziehen sich auf Elemente der verschiedenen Konkretisierungsebenen des Lösungsraumes, deren Erfüllung durch die im Verifikations- und Validierungsraum stattfindende Simulationen und Tests der Lösungen überprüft wird.

Im Entwicklungsprozess erfolgt der Übergang vom Anforderungsraum in den Lösungsraum über die Kontextebene. Auf dieser werden die Kontexte des zu entwickelnden Systems – im Sinne von Systemgrenzen, externen Elementen, mit welchen das System im jeweiligen Kontext interagieren soll, sowie durch das erwartete und von außen wahrnehmbare Verhalten des Systems – definiert. Darüber hinaus kann die Kontextebene bereits eine Basisarchitektur des zu entwickelnden Systems beinhalten. Auf der Funktionsebene werden die internen Funktionen sowie die Zusammenhänge zwischen diesen im Sinne von Stoff-, Energie- und Signalflüssen identifiziert. Die Logische Ebene befasst sich mit der logischen, lösungskonkreten Architektur des Systems, durch welche die Funktionsstruktur umgesetzt

werden soll. Die Definition dieser Architektur erfolgt in einem zweistufigen Prozess, in dem zuerst einzelne Lösungsvarianten identifiziert, analysiert und bewertet werden, gefolgt von der Analyse und Bewertung der möglichen alternativen logischen Lösungsarchitekturen. Das Ergebnis der Entwicklungsarbeiten auf der logischen Ebene ist eine vollständige Systemarchitektur, auf deren Basis die disziplinspezifische Entwicklung auf der Physikalischen Ebene anknüpfen kann.

Der Administrationsraum enthält die Metadaten der Modellelemente (wie beispielsweise eine eindeutige ID oder deren Version), welche für die Verwaltung der Elemente über den gesamten Systemlebenszyklus benötigt werden.

6 5-Ebenen-IT-Architektur-Konzept

Für eine interdisziplinäre Entwicklung heutiger Systeme ist die Integration und Durchgängigkeit der Systemdaten und Informationen, die während des gesamten Lebenszyklus entstehen, ebenso wichtig wie die Werkzeuge, in denen die Daten erstellt, verarbeitet oder verwaltet werden. Während das MVPE-Modell und das KSKM beschreiben, wann welche Daten und Informationen benötigt und wie sie erstellt werden sollen, fehlt

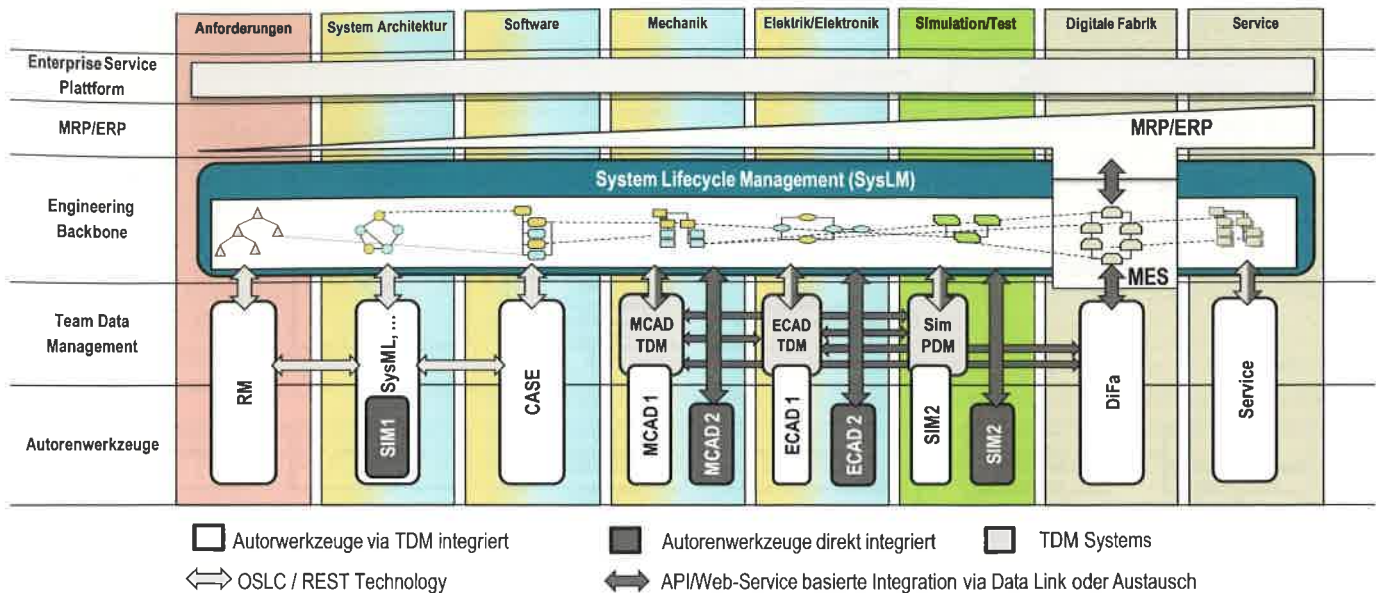


Bild 5
Das 5-Ebenen-IT-Architekturkonzept als Werkzeugbaustein der ^{VPE}System-Development-Methodology

jedoch ein Konzept, um die Erstellung, Verarbeitung und Verwaltung dieser Systemdaten und -informationen zu erleichtern. Wie bereits erwähnt, beschreibt die ^{VPE}System-Development-Methodology einen Makroansatz für die Entwicklung heutiger komplexer Systeme mit Fokus auf die konzeptionelle Systemgestaltung und -entwicklung, jedoch unter Berücksichtigung aller Phasen des Systemlebenszyklus. Folglich kann die Unterstützung von Prozess und Methode nicht durch ein einziges IT-Werkzeug bewältigt werden, sondern es wird ein ganzheitliches und lebenszyklusübergreifendes IT-Architekturkonzept benötigt, welches alle IT-Lösungen integriert und Durchgängigkeit gewährleistet.

Um die Anforderungen an eine geeignete Tool-Implementierung für den gesamten Lebenszyklus zu erfüllen, beinhaltet die ^{VPE}System-Development-Methodology ein fünfstufiges IT-Architekturkonzept [22, 23, 24], welches auf dem Vier-Stufen-Modell des VDA [25] beruht und es in Bezug auf die Entwicklung von Systemen, die im Kontext von IoT und Industrial Internet agieren, erweitert. Wie in Bild 5 zu sehen, bildet sich das IT-Architekturkonzept aus den folgenden fünf Ebenen [22]:

1. Autorentsysteme für RE (Requirements Engineering), MBSE, CASE, MCAD, ECAD, CASE, CAP, CAM, sowie Berechnungs- und Simulationssysteme.
2. Team Data Management (TDM), eine Verwaltungsebene, die einem Autorentsystem direkt zugeordnet

ist. Das Ziel eines TDM-Systems ist die Verwaltung der resultierenden Daten und Informationen in einem vom Autorentsystem festgelegten, meist nativen Datenformat.

3. Engineering Backbone, die zentrale Ebene für die Verwaltung der Systemdaten und -informationen. Er umfasst die interdisziplinäre Produktstruktur mit allen zugehörigen Dokumenten in einem neutralen Datenformat und ist die Grundlage für das Änderungs- und Konfigurationsmanagement. Er bildet den wesentlichen Bestandteil jedes SysLM-Konzepts und kann optional durch Application-Lifecycle-Management- und Service-Lifecycle-Management auf der TDM-Ebene ergänzt werden.
4. MRP/ERP, Verwaltungsebene zur Koordination von Ressourcen, Informationen und Prozessen innerhalb einer Organisation (ERP-System) sowie der Fertigung (MRP-System).
5. Enterprise Service Plattform, Softwarelösungen zur Erfassung und Analyse von System- und Service-daten während der Betriebsphase eines Systems. Die ESP spielt eine wichtige Rolle bei IoS-Strategien, z. B. predicted maintenance.“

7 Zusammenhänge zwischen den Bestandteilen der ^{VPE}System-Development-Methodology

Um die Entwicklung heutiger komplexer Systeme entlang der ^{VPE}System-

Development-Methodology zu ermöglichen, muss eine gleichzeitige und gemeinsame Nutzung aller Bestandteile der Methodologie gewährleistet werden. Während bei der Betrachtung des 5-Ebenen-IT-Architekturkonzepts schnell ersichtlich wird, dass es sowohl das MVPE-Modell als auch das KSKM hinsichtlich der Wahl der benötigten IT-Werkzeuge unterstützt, könnte bei den genannten Modellen der Gedanke aufkommen, sich bei der Entwicklung eines Systems für eines der beiden Modelle zu entscheiden. Jedoch sind das MVPE-Modell und das KSKM komplementär zueinander und ergänzen sich im Sinne der Definition einer Methodologie nach Martin [5] und Estefan [6], da das Kaiserslauterner Systemkonkretisierungsmodell eine auf den im MVPE-Modell dargestellten Prozess abgestimmte methodische Unterstützung darstellt.

Durch die Gegenüberstellung beider Modelle wird ersichtlich, dass das KSKM – trotz seines Fokus auf den frühen Phasen der Systementwicklung – das gesamte MVPE-Modell abdeckt. Wie auch in den Bild 3, 4 und 5 durch eine farbliche Unterlegung gekennzeichnet, decken Anforderungs- (rot) sowie Lösungsraum (gelb/blau) den gesamten linken „Flügel“ des „V“ entlang des KFLP-Ansatzes ab. Der Verifikations- und Validierungsraum dient wie bereits erwähnt der Überprüfung, ob das System richtig beziehungsweise das richtige System gebaut wurde. Da die Überprüfung auch auf Teilsysteme und Komponenten angewendet werden

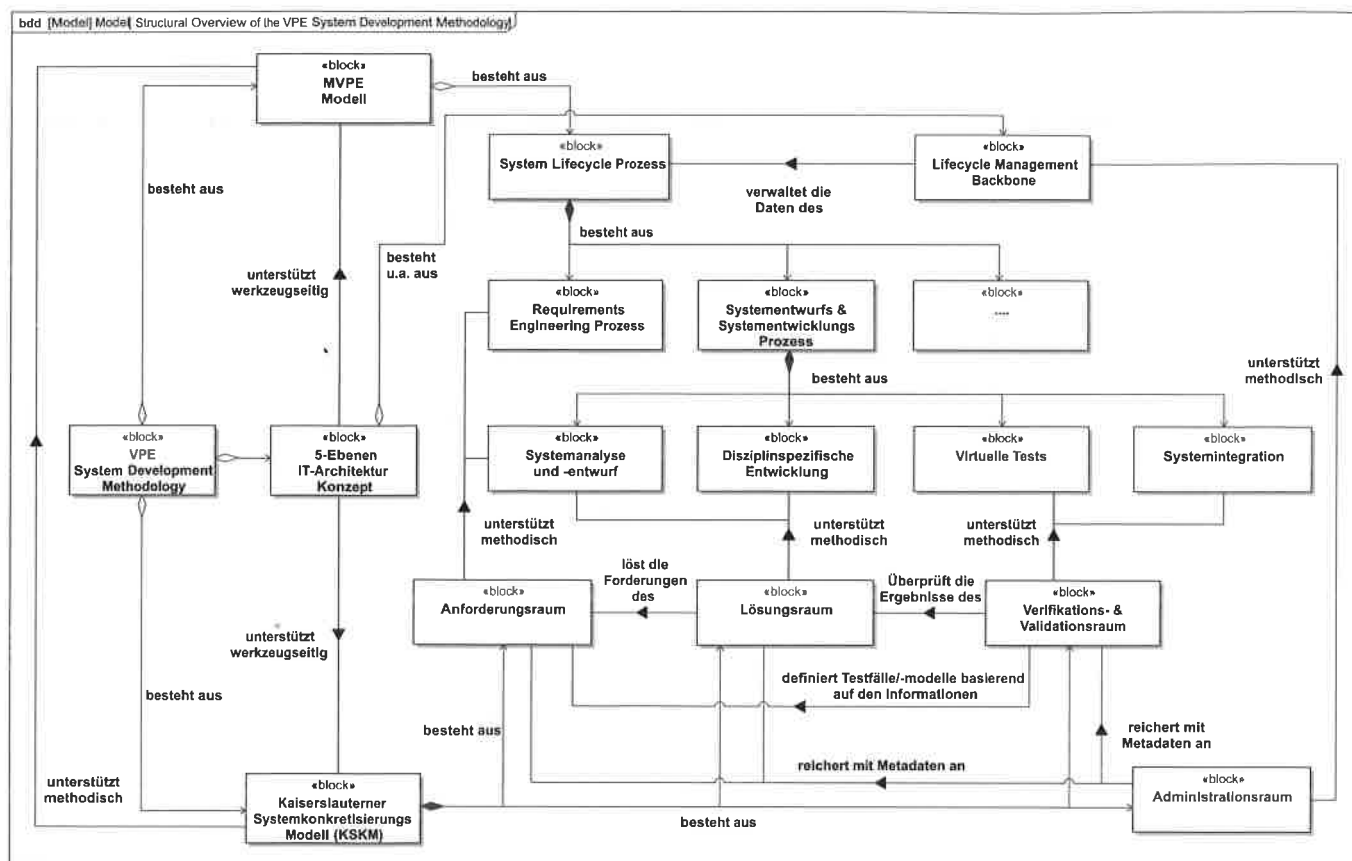


Bild 6
Aufbau der VPE SystemDevelopment Methodology und Darstellung der Abhängigkeiten ihrer Elemente

kann, betrachtet der Verifikations- und Validierungsraum (grün) sowohl den rechten „Flügel“ des „V“ zur Systemintegration als auch den Raum der virtuellen Tests zwischen den beiden „Flügeln“. Der Administrationsraum ist das Bindeglied zwischen der Datenerstellung und der Datenverwaltung. Folglich deckt der Administrationsraum (blau) das gesamte „V“ ab und stellt die Verwaltbarkeit der entstehenden Daten im SysLM Backbone sicher.

Bild 6 liefert eine detaillierte Beschreibung der Bestandteile der ^{VPE}System-Development_{Methodology} und deren Zusammenhänge. Dabei beschreiben die Beziehungen mit einer Raute die hierarchische Aufgliederung der Methodologie und ihrer Unterelemente, und die Beziehungen mit einem geschlossenen schwarzen Pfeil die Abhängigkeiten zwischen diesen Elementen.

8 Zusammenfassung und Ausblick

Der zweiteilige Beitrag – Interdisziplinäre Konstruktionsmethoden und -prozesse zur Entwicklung cybertronischer Produkte – liefert eine Diskussionsgrundlage, um die Komplexität

des PEP heutiger Systeme im Kontext von IoT und Industrial Internet zu beherrschen. Der erste Beitrag identifizierte die bestehende und zukünftige Problematik bei der Entwicklung interdisziplinärer Systeme. Zudem liefert er sowohl einen umfassenden Überblick über bestehende interdisziplinäre und disziplinspezifische Entwicklungsansätze als auch eine Zusammenfassung der methodischen Forschungsaktivitäten des Lehrstuhls für Virtuelle Produktentwicklung der TU Kaiserslautern. Aufbauend auf den im ersten Beitrag genannten Ansätzen und erzielten Ergebnissen zeigt der zweite Beitrag einen ganzheitlichen Ansatz für die Entwicklung heutiger Systeme in Form einer Methodologie, welche neben dem Entwicklungsprozess auch dessen methodische und werkzeugseitige Unterstützung inkludiert.

Weiterführende Arbeiten am Lehrstuhl für Virtuelle Produktentwicklung umfassen die Erweiterung des vorgestellten Ansatzes um Aspekte zur Untersetzung der Entwicklung im Dienstleistungsdomain (Bild 3), sodass Services und unterstützende Infrastrukturen synchron mit dem cybertronischen Produktanteil als ein zu-

sammenhängendes Produkt-Service-System konzipiert und umgesetzt werden können.

Literatur

- [1] Porter, M.; Heppelmann, J.: Wie smarte Produkte den Wettbewerb verändern. In: Harvard-Business-Manager – das Wissen der Besten. Jhrg. 36, Heft Nr. 12 (2014), S. 34–60. – ISSN: 0945–6570
- [2] Lee, E. A.: Cyber Physical Systems: Design Challenges. Technical Report No. UCB/EECS-2008–8, University of California, Berkeley, 2008
- [3] Broy, M.: Cyber-Physical Systems – Innovation durch softwareintensive eingebettete Systeme. Springer-Verlag, Berlin, 2010
- [4] Eigner, M.; Koch, W.; Muggeo, C.: Modellbasierter Entwicklungsprozess cybertronischer Systeme. Springer, Berlin, 2017
- [5] Martin, J.N.: Systems Engineering Guidebook – A process for Developing Systems and Products. CRC Press, Boca Raton, Florida, 1996
- [6] Estefan, J.: Survey of Model-Based Systems Engineering (MBSE) Methodologies. INCOSE MBSE Initiative, 2008
- [7] Verein Deutscher Ingenieure: VDI 2206: Entwicklungsmethodik für mechatronische Systeme. Beuth, Berlin, 2004

- [8] Eigner, M.; Giltz, T.; Zafirov, R.: Proposal for Functional Product Description as Part of PLM Solution in Interdisciplinary Product Development. In: Proceedings of the Design 2012 / 12th International Design Conference, Dubrovnik, Croatia, May 21–24, 2012, pages 1667–1676. The Design Society, Glasgow, 2012
- [9] Eigner, M.; Dickopf, T.; Apostolov, H.; Schäfer, P.; Faißt, K.G.; Keßler, A.: System Lifecycle Management – Initial Approach for a Sustainable Product Development Process Based on Methods of Model Based Systems Engineering. In: Fukuda et al. (Eds.), 11th IFIP WG 5.1 International Conference – PLM 2014, Springer, Heidelberg, 2014
- [10] Eigner, M.; Dickopf, T.; Apostolov, H.: The evolution of the V-model: From VDI 2206 to a System Engineering based Approach for Developing Cybertronic Systems. In: Proceedings of the PLM17 / IFIP 14th International Conference on Product Lifecycle Management, Seville, Spain, July 9–12, 2017, IFIP, Laxenburg, Austria, 2017
- [11] Ströer, F.; Faißt, K.-G.; Eickhoff, T.; Apostolov, H.; Sivasothy, P.; Seewig, J.; Eigner, M.: Big Data in verfügbarkeitsorientierten Produkt-Service-Systemen am Beispiel einer Landmaschine. In: Schulze, S.-O.; Tschirner, C.; Kaffenberger, R.; Ackva, S. (Eds.), Tag des Systems Engineering, Paderborn, November 08–10, 2017, pages 285–294, Hanser Verlag, München, 2017
- [12] Mert, G.; Herder, C.F.; Menck, N.; Aurich, J.C.: Innovative Services for Customized, Availability-oriented Business Models for the Capital Goods Industry. In: Proceedings of the 8th CIRP IPSS Conference on Product-Service Systems across Life Cycle, pages 501–506, Bergamo, Italy, June 20–21, 2016
- [13] Roubanov, D.: Durchgehende IT gestützte Methode für eine effiziente Zusammenarbeit zwischen der interdisziplinären Produktentwicklung und Montagetageplanung. Ph.D. Thesis, Schriftenreihe VPE, Vol. 16, University of Kaiserslautern, 2016
- [14] Cadet, M.; Sinnwell, C.; Fischer, J.; Stephan, N.: Kernelemente für die Zusammenarbeit von CTP-Entwicklung und CTPS-Planung. In: Eigner, M.; Koch, W.; Muggeo, C. (Eds.), Modellbasierter Entwicklungsprozess cybertronischer Systeme – Der PLM-unterstützte Referenzentwicklungsprozess für Produkte und Produktionssysteme. Springer Vieweg, Berlin, 2017
- [15] Sinnwell, C.; Fischer, J.; Meissner, H.; Aurich, J.C.: Modellbasierter Referenzentwicklungsprozess für CTPS in frühen Entwicklungsphasen. In: Eigner, M.; Koch, W.; Muggeo, C. (Eds.), Modellbasierter Entwicklungsprozess cybertronischer Systeme – Der PLM-unterstützte Referenzentwicklungsprozess für Produkte und Produktionssysteme. Springer Vieweg, Berlin, 2017
- [16] Zafirov, R.: Model-based system engineering methods for integrated product design, process planning, and production system design. Ph.D. Thesis, Schriftenreihe VPE, Vol. 21, University of Kaiserslautern, 2017
- [17] Eigner, M.; Dickopf, T.; Schulte, T.; Schneider, M.: mecPro² – Entwurf einer Beschreibungssystematik zur Entwicklung cybertronischer Systeme mit SysML. In: Schulze, S.-O.; Muggeo, C. (Eds.), Tag des Systems Engineering 2015, pages 163–172, Hanser, München, 2015
- [18] Eigner, M.; Dickopf, T.; Huwig, C.: An Interdisciplinary Model-Based Design Approach for Developing Cybertronic Systems. In: Proceedings of the Design 2016 / 4th International Design Conference, Dubrovnik, Croatia, May 16–19, 2016, pages 1647–1656, The Design Society, Glasgow, 2016
- [19] Eigner, M.; Dickopf, T.; Schneider, M.; Schulte, T.: mecPro² – A holistic concept for the model-based development of cybertronic systems. In: Proceedings of the ICED17 / 21st International Conference on Engineering Design, Vancouver, Canada, August 21–25, 2017, pages 379–388, The Design Society, Glasgow, 2017
- [20] Pfenning, M.: Durchgängiges Engineering durch die Integration von PLM und MBSE. Ph.D. Thesis, Schriftenreihe VPE, Vol. 20, University of Kaiserslautern, 2017
- [21] Ponn, J.; Lindemann, U.: Konzeptentwicklung und Gestaltung technischer Produkte – Systematisch von Anforderungen zu Konzepten und Gestaltungs-ingen. Springer-Verlag, Berlin, Heidelberg, 2011
- [22] Eigner, M.: Das Industrial Internet – Engineering Prozesse und IT-Lösungen. In: Ulrich Sandler (Ed.), Industrie 4.0 grenzenlos. Springer Vieweg, Berlin Heidelberg, 2016
- [23] Schulte, T.; Müller, P.; Eigner, M.: mecPro² – Zusammenfassung und Ausblick. In: Eigner, M.; Koch, W.; Muggeo, C. (Eds.), Modellbasierter Entwicklungsprozess cybertronischer Systeme – Der PLM-unterstützte Referenzentwicklungsprozess für Produkte und Produktionssysteme. Springer Vieweg, Berlin, 2017
- [24] Bitzer, M.; Eigner, M.; Faißt, K.-G.; Muggeo, C.; Eickhoff, T.: Framework of the Evolution in Virtual Product Modeling and Model Management towards Digitized Engineering. In: Proceedings of the ICED17 / 21st International Conference on Engineering Design, Vancouver, Canada, August 21–25, 2017, pages 379–388, The Design Society, Glasgow, 2017
- [25] Eigner, M.; Roubanov, D.; Zafirov, R.: Modellbasierte virtuelle Produktentwicklung. Springer, Berlin, 2014

Jörg Frochte Maschinelles Lernen – Grundlagen und Algorithmen in Python

August 2018, Broschur, 400 S.
EURO 38,00

Carl Hanser Verlag

ISBN 978-3-446-45291-6

Maschinelles Lernen ist ein interdisziplinäres Fach, das die Bereiche Informatik, Mathematik und das jeweilige Anwendungsgebiet zusammenführt. In diesem Buch werden alle drei Teilgebiete gleichermaßen berücksichtigt:

- Es wird demonstriert, wie man die Algorithmen des

maschinellen Lernens verwendet und der Hintergrund geliefert, um zu verstehen, wie und warum diese Algorithmen funktionieren.

- Ebenfalls enthalten ist ein kompakter Kickstart zur Verwendung von Python 3 und seinem Ökosystem im Umfeld des maschinellen Lernens.
- Die Algorithmen werden zum besseren Verständnis und praktischen Einsatz anschaulich mittels NumPy und SciPy umgesetzt.
- Für die Support Vector Machines und das Deep

Learning wird auf scikit-learn bzw. Keras zurückgegriffen.

- Es werden verschiedene Methoden des überwachten, unüberwachten und bestärkenden Lernens besprochen, unter anderem Random Forest, DBSCAN und Q-Learning.

Vorausgesetzt werden Kenntnisse in objektorientierter Programmierung und Basiswissen der Hochschulmathematik. Die nötige Mathematik wird eingebettet im Buch präsentiert und die Theorie direkt

in Python-Code umgesetzt.

Das Buch ist ideal für Studierende der Informatik, Mechatronik, Elektrotechnik und der angewandten Statistik/Data Science sowie für Ingenieure und Informatiker in der Praxis.

Prof. Dr. Jörg Frochte lehrt und forscht seit 2010 an der Hochschule Bochum. Als Professor für Angewandte Informatik und Mathematik hält er hier unter anderem Vorlesungen in Mathematik, Simulation und Modellbildung sowie maschinellem Lernen.