

# Interdisziplinäre Design Methodik

Martin Eigner, Thomas Dickopf und Hristo Apostolov

## Einleitung

Statistiken der letzten Jahre bestätigen eine permanente Veränderung des Produktentwicklungsprozesses <sup>0</sup>. 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 Derivate 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 Prozent. 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) <sup>00</sup> oder cybertronischen Systemen (CTS) <sup>0</sup>. 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 aus-

gebildet werden, denken und arbeiten. Eine Übersicht über disziplinenorientierte und mechatronische Vorgehensmodelle in der Produktentwicklung zeigt Abbildung 1.

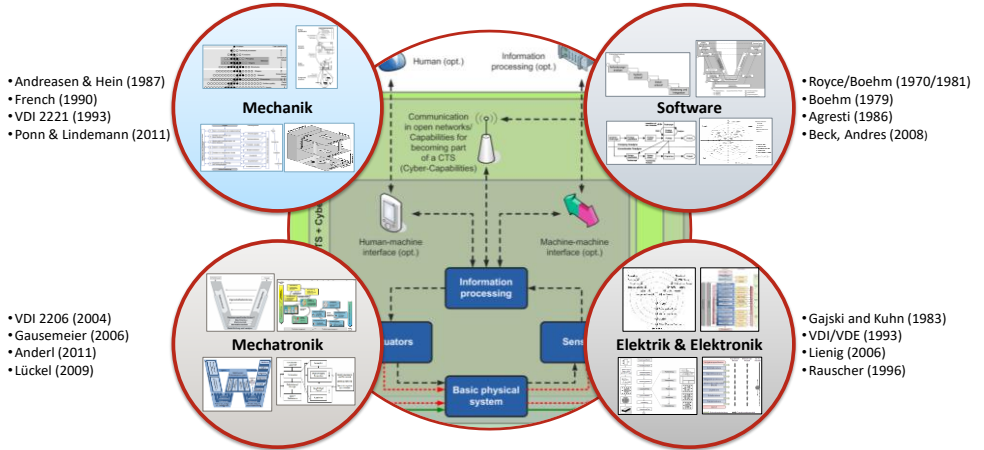


Abbildung 1: Disziplinen-orientierte Vorgehensmodelle in der Produktentwicklung

Nach einem kurzen Überblick über aktuelle disziplinspezifische und interdisziplinäre 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.

## 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 0. 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 0, French 0, Malmquist und

Svensson 0, Ehrlenspiel und Meerkamm 0 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 0. Einen anderen Ansatz stellt das Münchner Konkretisierungsmodell von Ponn/Lindemann 0 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 00 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 0. 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 0.

Im Bereich der **Softwareentwicklung** zeigen Entwicklungsmethodiken ähnlich der Vorgehensweise in der Elektronik häufig andere Muster – etwa neben der Funktions- eine starke Verhaltensorientierung 0. Mit dem Ziel die Softwareentwicklung produktiver und effizienter zu gestalten, wurden unter dem Begriff Software Engineering mehrere phasenorientierte Modelle entwickelt, die den Softwaredesignprozess unterstützen 0, z. B. 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 0. 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) [1] und der Rational Unified Process (RUP) [2]. Designansätze, die auf schnelle Implementierung und Flexibilität während des Designprozesses abzielen und nicht auf spezifischen Designprozeduren basieren, nennt man agile Softwareentwicklungsansätze [3].

Der Begriff **Mechatronik** wurde erstmals 1969 von Ko Kikuchi [4] 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 [5]. Mechatronische Designansätze basieren auf einem allgemeinen Verständnis des Designprozesses [6], auf dem mechanischen Designprozess von Pahl und Beitz [7] oder auf Variationen des V-Modells [8]. Der etablierteste Ansatz ist die Designrichtlinie VDI 2206 [9].

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 [10]. Mehrere Standards definieren den SE-Prozess (IEEE 1220 [11], ANSI / EIA 632 [12]; ISO / IEC 15288 [13]). Weitere Entwicklungsansätze des Systems Engineering, welche in der Industrie und Forschung Beachtung gefunden haben sind der Harmony-SE Ansatz des Softwareanbieters IBM [14], die Object-Oriented Systems Engineering Method (OOSEM) [15], die Vitech Model-based Systems Engineering Methodology [16], die JPL State Analysis [17], die Object-Process Methodology [18], das Zackman Framework [19] oder der Rational Unified Process for Systems Engineering (RUP SE) [20], welcher sich aus dem Rational Unified Process [19] der Softwareentwicklung ableitete.

Während klassische Methoden des Systems Engineerings dokumentenbasiert sind, ermöglicht **Model-Based Systems Engineering** (MBSE) als Weiterführung des Systems Engineerings, ein Entwicklungskonzept, welches auf die Integration von Modelle entlang des Systemlebenszyklus setzt <sup>0</sup>. 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 <sup>0</sup>, wurden hinsichtlich einer Nutzung der SysML angepasst <sup>00</sup> oder nutzen eine Modellierungssprache, welche sich aus der SysML ableitet <sup>00</sup>. 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. Abbildung 1), aber je nach Forschungsrichtung immer noch eine starke Disziplinentorientierung aufweisen <sup>0</sup>. 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 (↪ Digitalisierung der Prozesskette bis hin zum Digital Twin als eine Basis service-orientierter Geschäftsmodelle) sowie

die Gewährleistung einer kontinuierlichen Entwicklung ohne Medienbrüche und Informationsverlust genannt werden.

## **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 aufeinander aufbauende Vorgehensmodelle in den letzten Jahren entstanden, deren Endergebnis im Folgenden vorgestellt wird.

### **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 0* und *Estefan 0*. *Martin 0* 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 hingegen 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 essenzielles Element für einen erfolgreichen Systementwurf sieht *Martin 0* die Umwelt an. Sie umfasst alle externen Faktoren, welche das Handeln der Entwickler beeinflussen. Abbildung 2 verdeutlicht die Zusammenhänge der genannten Bestandteile einer Methodologie mit zusätzlichem Bezug auf die Themen Mensch und Technologie.

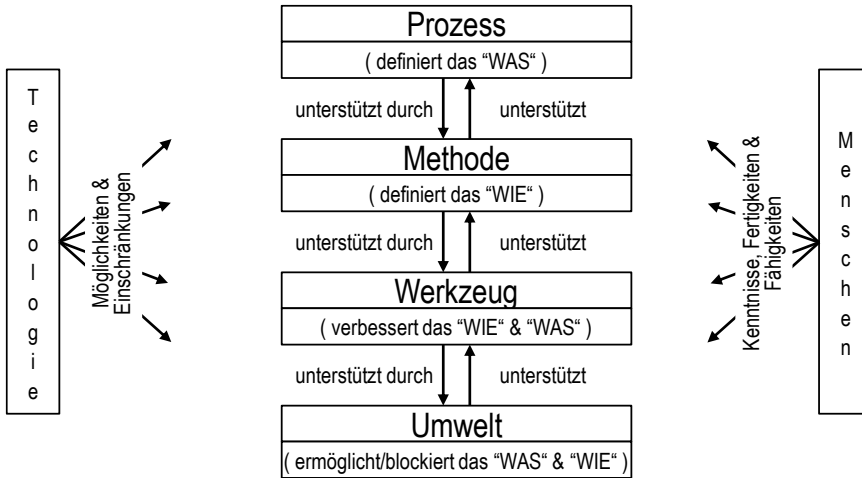


Abbildung 2: Bestandteile einer Methodologie in Bezug auf die Themen Mensch und Technologie (eigene Abb. nach Martin 0)

Estefans Definition einer MBSE-Methodologie 0 basiert auf den von Martin 0 genannten Merkmalen, adaptiert diese jedoch auf die Disziplin des Systems Engineering und den Entwurf eines Systems in einem modellbasierten bzw. modellgetriebenen Kontext. Zudem sollte erwähnt werden, dass jede Methode wiederum ein Prozess sein bzw. einen beinhalten kann. Denn sobald eine Methode aus einer Folge von Abläufen besteht, wird das "WIE" auf der höheren Abstraktionsebene zum „WAS“ der nächstniedrigeren 00.

### **<sup>VPE</sup>SystemDevelopmentMethodology**

Die <sup>VPE</sup>SystemDevelopmentMethodology beschreibt ein ganzheitliches Makro-Vorgehen für die Entwicklung cybertronischer Systeme im Kontext des Internet of Things and Services. Mit Bezug auf die VDI Richtlinie 2206 0 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 <sup>VPE</sup>SystemDevelopmentMethodology als Makro-Methodologie übernimmt und vereint dabei Konzepte verschiedenster interdisziplinärer und disziplinspezifischer Entwicklungsansätze und passt diese an die Anforderungen und

Rahmenbedingungen der Cybertronik an. Angelehnt an den Definitionen einer Methodologie nach *Martin 0* und *Estefan 0* besteht die  $VPE_{SystemDevelopmentMethodology}$  aus den folgenden drei wesentlichen Bestandteilen (Abbildung 3): (1) das für den Kontext des Internet of Things angepasste MVPE-Modell; (2) das neue Kaiserslauterner Systemkonkretisierungsmodell (KSKM); sowie einem (3) 5-Ebenen-IT-Architekturkonzept.

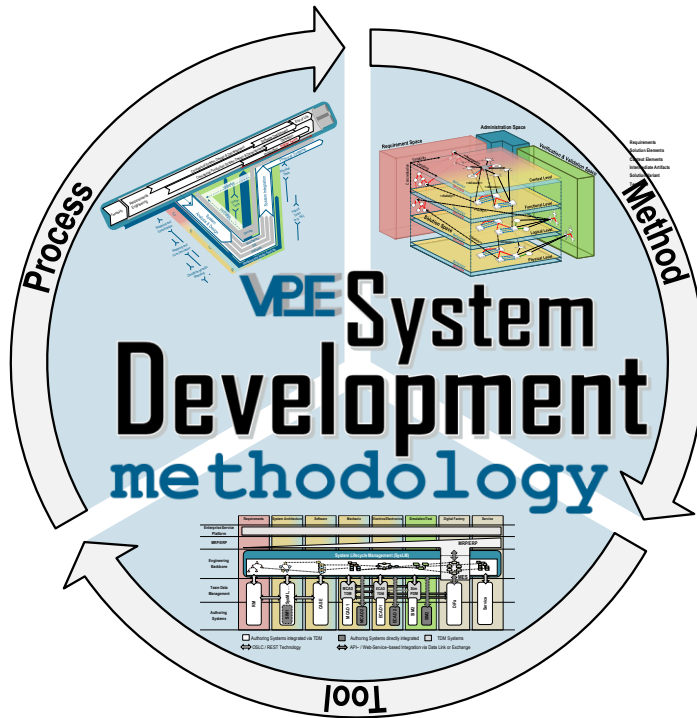


Abbildung 3:  $VPE_{SystemDevelopmentMethodology}$

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).

## MVPE-Modell

Erstmals eingeführt im Jahre 2012 [0], wurde das MVPE-Modell über die Jahre mehrmals angepasst und erweitert [00]. 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 [000] waren jedoch weitere Anpassungen und Erweiterungen notwendig. Diese betreffen im Allgemeinen alle Phasen des Lebenszyklus jedoch im Detail die frühen Phasen der Entwicklung (Abbildung 4). Eine allgemeine Beschreibung des generellen Aufbaus des MVPE-Modells erfolgte bereits im ersten Teil dieser Beitragsserie. Im Folgenden wird lediglich auf die Anpassungen des MVPE-Modells hinsichtlich der Entwicklung cybertronischer Systeme im Kontext von IoT und Industrial Internet eingegangen.

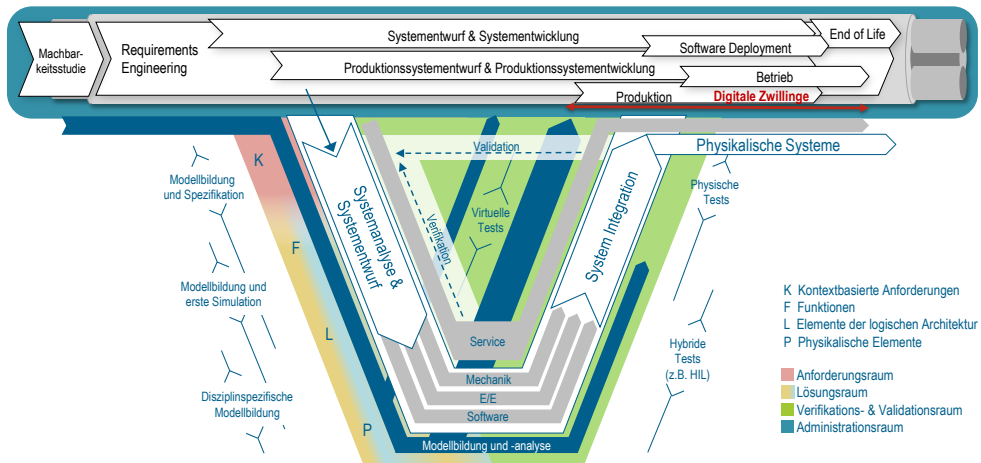


Abbildung 4: Das MVPE-Modell als Prozessbaustein der VPE-SystemDevelopment-Methodology

Roubanov [0], Cadet et al. [0], Sinnwell et al. [0] sowie Zafirov [0] 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 Informati-

onsaustausch 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 (Abbildung 4). 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 0. 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 Hardware-Komponenten angepasst oder ersetzt werden.

In Bezug auf die frühen Phasen des Entwicklungsprozesses (Abbildung 4, 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 viel mehr, dass der Kontext (K), in dem sich das System während seines Betriebs als Teil eines CTS bzw. CPS (z. B. 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 Abbildung 4 anhand seines Phasenpfeiles ersichtlich.

Bezogen auf die <sup>VPE</sup>SystemDevelopmentMethodology 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 mecPro<sup>2</sup> 0 entstanden.

### Kaiserslauterner Systemkonkretisierungsmodell

Basierend auf den Vorarbeiten zum mecPro<sup>2</sup> Modelrahmenwerk 000 und dem Systemkonkretisierungsmodell (SKM) nach Pfenning 0 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 (Abbildung 5) und unterstützt das MVPE-Modell als methodische Komponente der <sup>VPE</sup>SystemDevelopment<sub>Methodology</sub>. Die Achse der Konkretisierung liegt dabei parallel zur prozessualen Abfolge im MVPE-Modell.

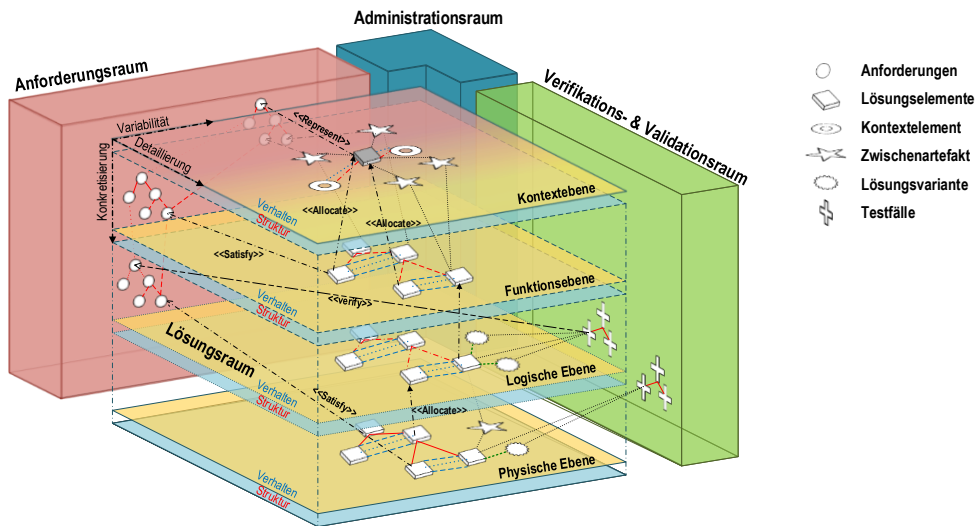


Abbildung 5: Das Kaiserslauterner Systemkonkretisierungsmodell als Methodenbaustein der <sup>VPE</sup>SystemDevelopment<sub>Methodology</sub>

Die grundlegende Struktur des Kaiserslauterner Systemkonkretisierungsmodells basiert auf dem SKM nach Pfenning 0 sowie dem Münchener Produkt-

konkretisierungsmodell nach *Ponn und Lindemann* 0 und wurde um die wesentlichen Konzepte des mecPro<sup>2</sup> Modellrahmenwerks 000 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 entwickelten 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 entwickelten 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.

## 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 jedoch ein Konzept, um die Erstellung, Verarbeitung und Verwaltung dieser Systemdaten und -informationen zu erleichtern. Wie bereits erwähnt, beschreibt die <sup>VPE</sup>SystemDevelopment<sub>Methodology</sub> 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 <sup>VPE</sup>SystemDevelopment<sub>Methodology</sub> ein fünfstufiges IT-Architekturkonzept 000, welches auf dem Vier-Stufen-Modell des VDA 0 beruht und es in Bezug auf die Entwicklung von Systemen, die im Kontext von IoT und Industrial Internet agieren, erweitert. Wie in Abbildung 6 zu sehen, bildet sich das IT-Architekturkonzept aus den folgenden fünf Ebenen 0:

1. Autoren Systeme für RE, MBSE, CASE, MCAD, ECAD, CASE, CAP, CAM, sowie Berechnungs- und Simulationssysteme.
2. Team Data Management (TDM), eine Verwaltungsebene, die einem Autorensystem direkt zugeordnet ist. Das Ziel eines TDM Systems ist die Verwaltung der resultierenden Daten und Informationen in einem vom Autorensystem 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 ei-

nem neutralen Datenformat und ist die Grundlage für das entwicklungstechnische Änderungs- und Konfigurationsmanagement. Er bildet den wesentlichen Bestandteil jedes SysLM-Konzepts und kann optional durch Application-Lifecycle-Management- und Service-Lifecycle-Management-Lösungen 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 Servicedaten während der Betriebsphase eines Systems. Beinhalten unter anderem den Digitalen Zwilling des realen Systems.

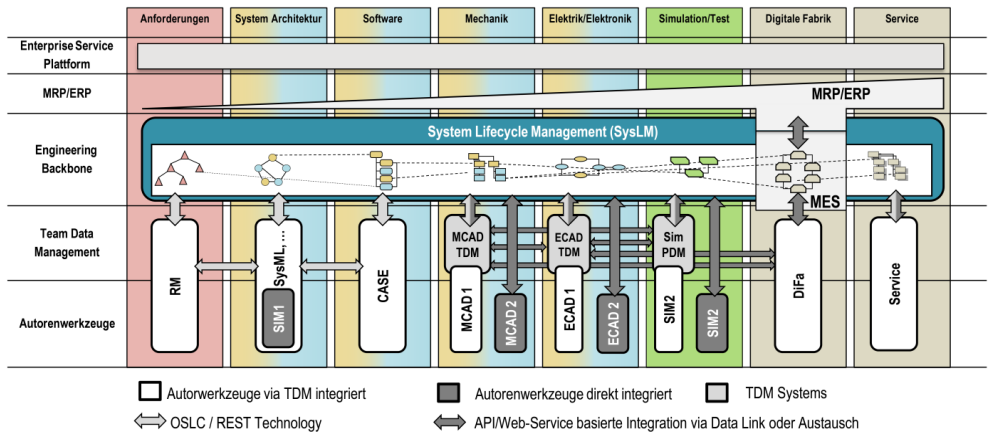


Abbildung 6: Das 5-Ebenen IT-Architekturkonzept als Werkzeugbaustein der *VPE System Development Methodology*

## 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 0* und *Estefan 0*, 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 Abbildung 4, 5 und 6 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 bzw. das richtige System gebaut wurde. Da die Überprüfung auch auf Teilsystemen und Komponenten angewendet werden 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.

Abbildung 7 liefert eine detaillierte Beschreibung der Bestandteile der <sup>VPE</sup>SystemDevelopmentMethodology 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.

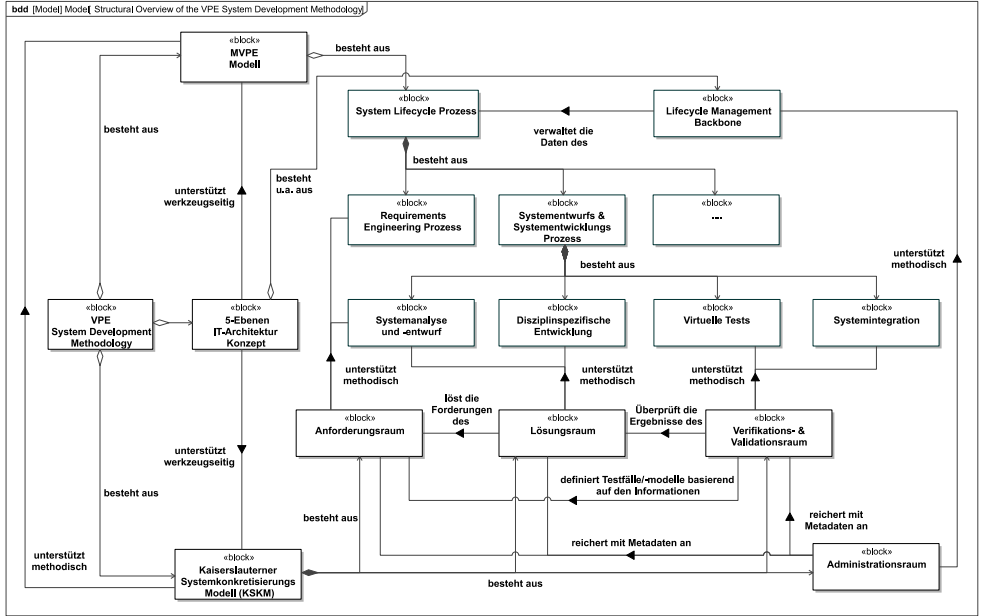


Abbildung 7: Aufbau der <sup>VPE</sup>SystemDevelopmentMethodology und Darstellung der Abhängigkeiten ihrer Elemente

## Zusammenfassung und Ausblick

Der 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. Außerdem lieferte 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 (vgl. Abbildung 4), sodass Services und unterstützende Infrastrukturen synchron mit dem cybertronischen Produktanteil als ein zusammenhängendes Produkt-Service-System konzipiert und umgesetzt werden können.

## Quellenangabe

- Porter, M.; Heppelmann, J.: Wie smarte Produkte den Wettbewerb verändern, in: Harvard-Business-Manager - das Wissen der Besten. Jhrg. 36, HeftNr.. 12 (2014), S. 34-60. – ISSN: 0945-6570.
- Lee, E. A.: Cyber Physical Systems: Design Challenges, Technical Report No. UCB/EECS-2008-8, University of California, Berkeley, 2008
- Broy, M.: Cyber-Physical Systems - Innovation durch softwareintensive eingebettete Systeme, Springer-Verlag, Berlin, 2010
- Eigner, M.; Koch, W.; Muggeo, C.: Modellbasierter Entwicklungsprozess cybertronischer Systeme, Springer, Berlin, 2017
- Ponn, J.; Lindemann, U.: Konzeptentwicklung und Gestaltung technischer Produkte – Systematisch von Anforderungen zu Konzepten und Gestaltlösungen, Springer, Berlin, 2011
- Andreasen, M.M.: Machine Design Methods Based on a Systematic Approach, Ph.D. Thesis, Lund University, 1980
- French, M. J.: Conceptual design for engineers, Springer, London, 1999
- 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
- Ehrlenspiel, K.; Meerkamm, H.: Integrierte Produktentwicklung, Carl Hanser, München, 2013
- Pahl, G.; Beitz, W.: Konstruktionslehre, Springer Vieweg, Berlin, 1977
- VDI 2221 - Methodik zum Entwickeln und Konstruieren technischer Systeme und Produkte, Beuth, Berlin, 1994
- Rauscher, R.: Entwurfsmethodik hochintegrierter anwendungsspezifischer digitaler Systeme, Pro-Universitate-Verlag, Sinzheim, 1996
- Gajski, D.D.; Abdi, S.; Gerstlauer, A.; Schirner, G.: Embedded system design, Springer, Boston, 2009
- Lienig, J.: Layoutsynthese elektronischer Schaltungen - Grundlegende Algorithmen für die Entwurfsautomatisierung. Springer Berlin Heidelberg, 2006
- 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

- Pomberger, G.; Pree, W.: Software-Engineering, Carl Hanser, München, 2004
- 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
- Rumbaugh, J.; Jacobson, I.; Booch, G.: The Unified Modeling Language Reference Manual, Pearson Education, 2010
- Kruchten, P.: The rational Unified Process (Addison-Wesley object technology series). Addison-Wesley, Reading, 1999
- Beck, K.; Andres, C.: Extreme Programming Explained: Embrace Change, 2nd Edition. Addison-Wesley Professional, 2005
- 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
- Isermann, R.: Mechatronische Systeme. Grundlagen, Springer, Heidelberg, 2008
- Gausemeier, J.; Lückel, J.: Entwicklungsumgebungen Mechatronik: Methoden und Werkzeuge zur Entwicklung mechatronischer Systeme, Heinz Nixdorf Institut, Paderborn, 2000
- Bender, K.: Embedded Systems - qualitätsorientierte Entwicklung, Springer, Heidelberg, 2005
- 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
- VDI 2206: Entwicklungsmethodik für mechatronische Systeme – Design methodology for mechatronic systems, Beuth, Berlin, 2004
- INCOSE Systems Engineering Handbook: A Guide for System Life Cycle Processes and Activities, Wiley, New Jersey, 2015
- IEEE 1220-2005 - IEEE Standard for Application and Management of the Systems Engineering Process. IEEE, New York, N.Y., 2011
- ANSI/EIA 632: Processes for Engineering a System, SAE International, USA, 2003
- ISO/IEC/IEEE 15288:2015 Systems and Software Engineering –System Life Cycle Processes, 2015
- 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
- Friedenthal, S.; Moore, A.; Steiner, R.: A Practical Guide to SysML, Morgan Kaufmann, 2012
- Long, J.E.: Systems Engineering (SE) 101 - CORE®: Product & Process Engineering Solutions. Vitech training materials, Vitech Corporation, Vienna, VA, 2000
- Estefan, J.: Survey of Model-Based Systems Engineering (MBSE) Methodologies. INCOSE MBSE Initiative 2008, [http://www.omgSysml.org/MBSE\\_Methodology\\_Survey\\_RevB.pdf](http://www.omgSysml.org/MBSE_Methodology_Survey_RevB.pdf)

- Dori, D.: Model-Based Systems Engineering with OPM and SysML. Springer Science+Business Media, New York, 2016
- Zachman, J.A.: A framework for information systems architecture. In: IBM Systems Journal, 26, pages 276-292, 1987
- Cantor, Murray: Rational Unified Process for Systems Engineering, RUP SE Version 2.0. IBM Rational Software white paper, IBM Corporation, 2003
- Eigner, M.; Dickopf, T.; Schulte, T.; Schneider, M.: mecPro<sup>2</sup> - Entwurf einer Beschreibungssystematik zur Entwicklung cybertronischer Systeme mit SysML. In (Schulze, S.-O.; Muggeo, C. Eds.): Tag des Systems Engineering 2015. Hanser, Munich, 2015
- Gabrisch, S.; Tenbergen, B.: SPES2020 - Zusammenfassendes Deliverable D2.1C: SysML Profil für Enterprise Architect zur Modellierung modellbasierter Anforderungen im SPES - Requirements View. <http://spes2020.informatik.tu-muenchen.de/results/D2.1.C.pdf>
- Iwanek, P.; Kaiser, L.; Dumitrescu, R.; Nyßen, A.: Fachdisziplinübergreifende Systemmodellierung mechatronischer Systeme mit SysML und CONSENS. In: Maurer, M., Schulze, S.-O. (Eds.), Tag des Systems Engineering, Stuttgart, November 6-8, 2013, pages 337-346, Carl Hanser Verlag, München, 2013
- Grundel, M.; Abulawi, J.; Moeser, G.; Weillkiens, T.; Scheithauer, A.; Kleiner, S.; Kramer, C.; Neubert, M.; Kümpel, S.; Albers, A.: FAS4M – No more: “please mind the gap!” In: Maurer, M., Schulze, S.-O. (Eds.), Tag des Systems Engineering, Bremen, November 12-14, 2014, pages 65-74, Carl Hanser Verlag, München, 2014
- Roques, P.: MBSE with the ARCADIA Method and the Capella Tool. ERTS<sup>2</sup> 2016 - 8th European Congress on Embedded Real Time Software and Systems, Toulouse France, January 27-29, 2016
- Gausemeier, J.; Dumitrescu, R.; Steffen, D.: Systems Engineering in industrial practice, Paderborn, 2015
- Martin, J.N.: Systems Engineering Guidebook – A process for Developing Systems and Products. CRC Press, Boca Raton, Florida, 1996
- Eigner, M.; Gilz, 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
- 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.), 11<sup>th</sup> IFIP WG 5.1 International Conference - PLM 2014, Springer, Heidelberg, 2014
- 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 14<sup>th</sup> International Conference on Product Lifecycle Management, Seville, Spain, July 9-12, 2017, IFIP, Laxenburg, Austria, 2017
- 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
- 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
- Roubanov, D.: Durchgehende IT gestützte Methode für eine effiziente Zusammenarbeit zwischen der interdisziplinären Produktentwicklung und Montageplanung. Ph.D. Thesis, Schriftenreihe VPE, Vol. 16, University of Kaiserslautern, 2016
- 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
- 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
- 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
- Eigner, M.; Dickopf, T.; Huwig, C.: An Interdisciplinary Model-Based Design Approach for Developing Cybertronic Systems. In: Proceedings of the Design 2016 / 4<sup>th</sup> International Design Conference, Dubrovnik, Croatia, May 16-19, 2016, pages 1647-1656, The Design Society, Glasgow, 2016
- Eigner, M.; Dickopf, T.; Schneider, M.; Schulte, T.: mecPro<sup>2</sup> - 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
- Pfenning, M.: Durchgängiges Engineering durch die Integration von PLM und MBSE. Ph.D. Thesis, Schriftenreihe VPE, Vol. 20, University of Kaiserslautern, 2017
- Eigner, M.: Das Industrial Internet – Engineering Prozesse und IT-Lösungen. In: Ulrich Sandler (Ed.), Industrie 4.0 grenzenlos. Springer Vieweg, Berlin Heidelberg, 2016

- Schulte, T.; Müller, P.; Eigner, M.: mecPro<sup>2</sup> - Zusammenfassung und Ausblick. In: In: Eigner, M.; Koch, W.; Muggeo, C. (Eds.), Modellbasierter Entwicklungsprozess cybertronischer Systeme – Der PLM-unterstützte Referenzentwicklungsprozess für Produkte und Produktionssysteme. Springer Vieweg, Berlin, 2017
- 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
- Eigner, M.; Roubanov, D.; Zafirov, R.: Modellbasierte virtuelle Produktentwicklung, Springer, Berlin, 2014

## Kontakt

Prof. Dr.-Ing. Martin Eigner\*,  
Dipl.-Ing. Thomas Dickopf,  
Hristo Apostolov, MSc  
Lehrstuhl für Virtuelle Produktentwicklung  
Fachbereich Maschinenbau und Verfahrenstechnik  
Technische Universität Kaiserslautern  
Gottlieb-Daimler-Str. 44  
67663 Kaiserslautern  
[vpe.mv.uni-kl.de](mailto:vpe.mv.uni-kl.de)  
[vpeinfo@mv.uni-kl.de](mailto:vpeinfo@mv.uni-kl.de)

\*ehemals Leiter VPE, jetzt EIGNER Engineering Consult  
Höllhäuserweg 38a  
76534 Baden-Baden  
[eigner.engineeringconsult@gmail.com](mailto:eigner.engineeringconsult@gmail.com)