

## **Durchgängige modellbasierte Entwicklung mit Methoden des Digital Engineerings**

Wie Unternehmen den Herausforderungen der Digitalisierung begegnen können

Dr.-Ing. Christian Henke, Christopher Lankeit, Jan Michael  
Fraunhofer-Institut für Entwurfstechnik Mechatronik (IEM)  
Zukunftsmeile 1, 33102 Paderborn



Dr.-Ing. Christian Henke  
Abteilungsleiter Regelungstechnik



Dipl.-Wirt.-Ing. Christopher Lankeit  
Gruppenleiter



M.Sc. Jan Michael  
Gruppenleiter

Das Fraunhofer-Institut für Entwurfstechnik Mechatronik IEM bietet Unternehmen am Technologiestandort Paderborn Expertise für intelligente Mechatronik im Kontext Industrie 4.0. Mit der zukunftsorientierten Stoßrichtung »Advanced Systems Engineering für intelligente mechatronische Systeme« wird das Ziel verfolgt, intelligente Produkte und Produktionssysteme, aber auch intelligente Dienstleistungen zu entwickeln.

Das umfangreiche Know-how des Fraunhofer IEM im Bereich von Methoden und Werkzeugen für die Entwicklung intelligenter mechatronischer Systeme wird durch die interdisziplinäre Zusammenarbeit mit dem Heinz Nixdorf Institut und der Universität Paderborn gestärkt.

Als Teil der Fraunhofer-Gesellschaft, Europas größter Organisation für anwendungsorientierte Forschung, sind wir weltweit vernetzt und fördern die internationale Zusammenarbeit. Das mittlerweile 96 Mitarbeiterinnen und Mitarbeiter starke Fraunhofer IEM, ist das erste selbstständige und außeruniversitäre Forschungsinstitut in der Innovationsregion Ostwestfalen-Lippe.

Dieser Beitrag zeigt die exemplarische Anwendung einer durchgängigen modellbasierten Entwicklung mit Methoden und Werkzeugen des Digital Engineerings. Im Rahmen der Komplexitätssteigerung und stärkeren Vernetzung von technischen Systemen im Kontext von Industrie 4.0 werden diese leistungsfähigeren Methoden notwendig. Auf diese Weise wird eine effiziente Entwicklung ermöglicht.

## I. INDUSTRIE IM WANDEL

Im Zeitalter von Industrie 4.0 und der Digitalisierung steigt die Komplexität und die Vernetzung von Systemen und Maschinen stark an. Dies hat erheblichen Einfluss auf den Maschinen- und Anlagenbau. Aufgrund von Zeit- und Kostendruck, sind geeignete Entwicklungsmethoden und Prozesse zwingend erforderlich. Die derzeit auf den Markt vorhandenen Methoden sind allerdings für eine multidisziplinäre Zusammenarbeit unzureichend. Insbesondere muss ein durchgängiges Anforderungsmanagement Anwendung finden, sodass eine Rückverfolgbarkeit vom virtuellen Prototypen bis hin zu integrierten Tests gewährleistet ist.

Neben dem disziplinübergreifenden Ansatz wird das frühzeitige Testen durch virtuelle Prototypen stets wichtiger. Hersteller sind durch Zeitdruck gezwungen, Subsysteme schon vor der Systemintegration zu testen. Die virtuelle Inbetriebnahme dient hier bereits als modernes Verfahren für die zeit- und kosteneffizienten Prüfung und Inbetriebnahme. Es stehen dabei diverse Werkzeuge zur Verfügung, welche sich für die Anwendung im Kontext eingebetteter Industriesteuerungen eignen. Darüber hinaus bieten diese Tools in der Regel proprietäre Software und Laufzeitumgebungen an. Allerdings ist eine ganzheitliche Integration dieser Werkzeuge und Prozesse bislang noch nicht erfolgt, sodass es keine off-the-shelf Lösungen gibt, die eine frühzeitige und individuelle Einbindung der Anforderungen in den Entwicklungsprozess ermöglichen.

Damit ergeben sich drei entscheidende Punkte für die virtuelle Inbetriebnahme:

- Konsistenz über den gesamten Prozess
- Entwurf und Validierung von Simulationsmodellen (als Teil des virtuellen Prototyps), welche das physikalische Verhalten des technischen Systems abbilden
- Rückverfolgbarkeit der definierten Anforderungen bereits in frühen Konzeptphasen bis zur Integrationsphase

Ein ganzheitlicher modellbasierter System-Engineering-Ansatz kombiniert diese beiden entscheidenden Punkte und reduziert die Markteinführungs- und Entwicklungskosten bei gleichzeitiger Maximierung der Qualität durch deutlich ausgereifere Produkte, welche die definierten Anforderungen erfüllen. Der allgemeine Ansatz des MBSE ist in Abschnitt 2 dargestellt. Aufbauend darauf wird die Entwicklung exemplarisch an Auszügen des Systemmodells gezeigt, was die Rückverfolgbarkeit der definierten Anforderungen ermöglicht. Die Einbeziehung

von Simulationsmodellen, welche aufbauend auf dem Systemmodell erstellt werden, werden in Kapitel 4 erläutert. In Kapitel 5 werden die virtuelle Inbetriebnahme und die Prüf-/Testabläufe behandelt.

## II. MODELLBILDUNG

In den Anfängen wurde Systems Engineering lediglich für die Entwicklung hochkomplexer Produkte praktiziert. Hoch-innovative Technologien mit sehr engen Fehlertoleranzen unter komplizierten Einsatzbedingungen wurden hierbei fokussiert.

Im Rahmen der Digitalisierung werden jedoch Systeme miteinander vernetzt, mit zusätzlichen Funktionen ausgestattet, was eine interdisziplinäre Entwicklung unumgänglich macht. Diese neuen Herausforderungen führten letztlich dazu, dass Systems-Engineering zunehmende Akzeptanz erfährt, besonders auch das Model-Based Systems Engineering (MBSE). Vor diesem Hintergrund wurden Methoden entwickelt, welche eine Entwicklung komplexer Produkte ermöglichen [1].

Das MBSE zielt auf einen geführten/geleiteten Prozess ab, welcher die Entwicklung von Systemen unterstützt. Dabei muss die Konsistenz von den Anforderungen bis hin zum aktuellen System gewährleistet sein. Die steigende Bedeutung des Einsatzes von Modellen im Rahmen des Systems Engineering hat dazu geführt, dass sich beispielsweise das International Council on Systems Engineering (INCOSE) gebildet hat [2], das als Organisation Wissenschaft und Bildung im Bereich des Systems Engineering in Industrie, Forschung und Lehre unterstützt und weiterentwickelt. Der Einsatz von adäquaten Modellen stellt eine gute Lösung für einige der genannten Herausforderungen dar. Dieses Vorgehen und einige Hintergründe finden sich in der Literatur [3],

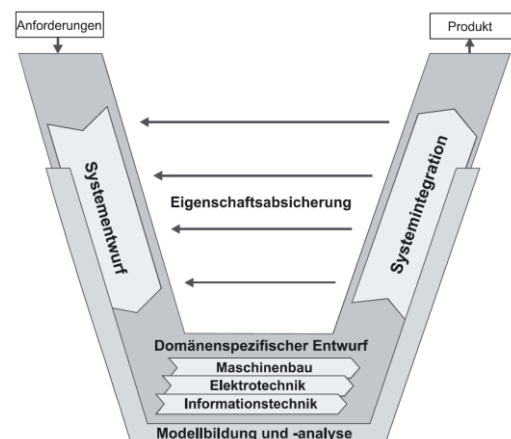


Abb. 1: V-Modell für die Entwicklung mechatronischer Systeme [8]

[4], [5]. Wissenschaftliche Ansätze zu einer durchgängigen Nutzung der Anforderungen und eine entwicklungs-begleitende Modellbildung im Bereich des heutigen Systems Engineering sind in [6], [7] dargestellt. Um hier eine kurze Beschreibung der modellbasierten Entwicklung zu zeigen, wird das V-Modell (Abb. 1) zur Verdeutlichung herangezogen [8].

Ausgehend von den Anforderungen können grundlegend drei verschiedene Phasen der Entwicklung intelligenter technischer Systeme aufgezeigt werden. Die frühe Phase der Entwicklung, die Konzipierungsphase, wird ergänzt von der domänenspezifischen Phase. Nachdem dann in den einzelnen Domänen spezifische Lösungen auf Basis der Anforderungen erarbeitet wurden, werden diese in der Systemintegrationsphase abgeprüft und führen letztlich zum Produkt. Als Unterstützung in der Entwurfsphase dienen sogenannte Partialmodelle [9], welche auf Basis von SysML [10] verschiedene Elemente beinhalten. In der sog. CONSENS Notation (Abb. 2) bilden neun kohärente Partialmodelle die Basis der Entwicklung und führen zum sogenannten Zielsystem.

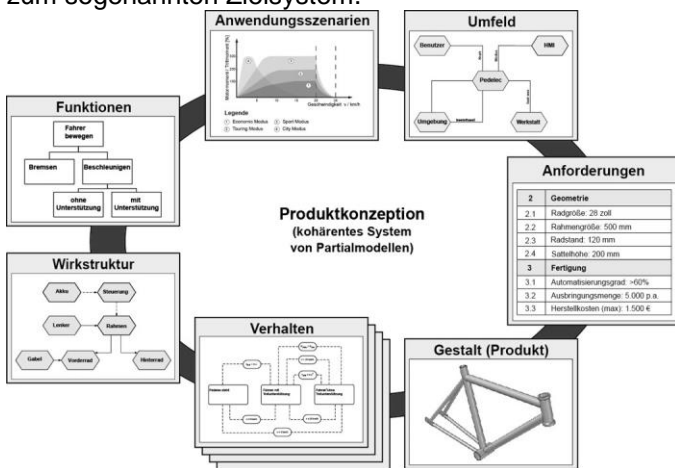


Abb. 2: CONSENS-Partialmodelle [9]

Insbesondere die Anforderungen haben einen starken Einfluss auf die späteren Entwicklungsphasen. Dabei können diese jedoch sehr umfangreich und somit herausfordernd werden. Hierbei gilt es, übergeordnete Ziele für die Entwicklung zu verfeinern und die Anforderungen in verschiedene Level einzugliedern, welche jeweils unterschiedliche Detaillierungsstufen haben. Dieses Vorgehen wird in [4] verdeutlicht, indem vier Level unterschiedlicher Anforderungen definiert und eingeteilt werden. Hierbei wird die Konsistenz erhöht, wenn geeignete Werkzeuge diesen Prozess unterstützen. Die genannten vier Level von Anforderungen sind Ziele, funktionsorientierte Anforderungen, domänenübergreifende Anforderungen und domänenspezifische Anforderungen.

Viele Klassifizierungen von Anforderungen wurden bereits von diversen Autoren beleuchtet und entwickelt,

wobei allerdings die bekannteste, der aus Volere [11] entspricht. Robertson und Robertson unterteilen die Anforderungen hierbei in Kategorien, welche jeweils durch die Beschreibungen der Anforderungen charakterisiert sind, bspw. Sicherheitsanforderungen oder Leistungsanforderungen. Der folgende Abschnitt zeigt die zuvor beschriebene Aufteilung der Anforderungen.

Die Level geben eine Orientierung darüber, welcher Detaillierungsgrad für eine Anforderung sinnvoll für die Nutzung in den verschiedenen Phasen der Entwicklung ist. Innerhalb der einzelnen Level können dann wiederum die bekannten Klassifizierungen (bspw. Volere) Anwendung finden.

Im Normalfall beginnt der Entwicklungsprozess auf einem abstrakten Level und wird stetig ausdetailliert. Einhergehend mit dieser Interpretation verschiedener Level von Anforderungen innerhalb des Entwicklungsprozesses können Methoden angewandt werden, die einen Übergang zwischen den Leveln ermöglichen. Funktionsorientierte Anforderungen können hierbei auf eine Funktionshierarchie bezogen werden. Weiterhin ermöglicht eine Wirkstruktur die Zuordnung von domänenübergreifenden Anforderungen zu den jeweiligen Schnittstellen zwischen einzelnen Systemelementen. In der domänenspezifischen Phase werden dynamische Modelle erstellt, sodass hier gleichermaßen domänenspezifische Anforderungen zugeordnet werden können. Es besteht ebenfalls die Möglichkeit, diese Modelle mit Anforderungen eines abstrakteren Levels zu verknüpfen [5]. Die entstandenen Modelle könne auch dazu genutzt werden, Testscenarien abzutesten, was besonders bei Systemen sinnvoll ist, die real nur schwer oder nicht zu testen sind.

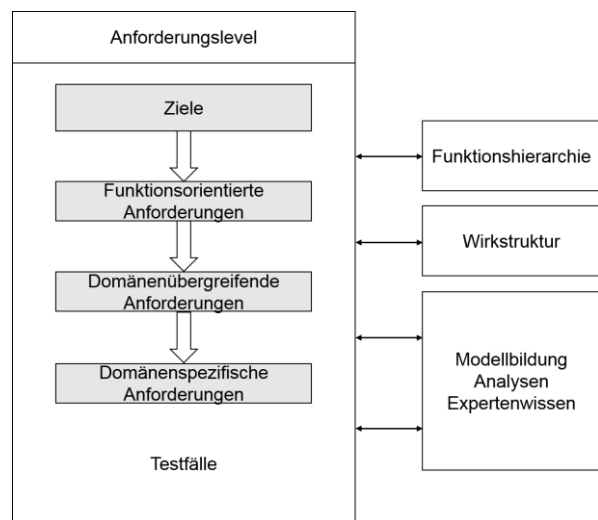


Abb. 3: Anforderungslevel in der Entwicklung

### III. SYSTEMMODEL

In diesem Abschnitt wird das sogenannte Systemmodell beschrieben. Fokussiert werden hierbei die Anforderungen und die Systemarchitektur. Im Fokus stehen hier das Umfeldmodell und die Use-Cases, siehe Abb. 4.

Die Basis für eine effiziente und kundenorientierte Entwicklung intelligenter technischer Systeme stellt das Anforderungsmanagement dar, welches für diese Systeme unerlässlich ist. Es müssen sämtliche Anforderungen erfasst werden, wie z.B. Genauigkeit von Positions- und Geschwindigkeitsreglern, Bandbreite von Aktoren und Sensoren oder mechanische Belastbarkeiten der Komponenten. Dies geschieht sukzessive über die Detaillierung der Ziele und Funktionen. Von großer Bedeutung sind dabei die externen und internen Schnittstellen, wie z.B. Kommunikations-schnittstellen, und mechanische Verbindungen, die in den domänenübergreifenden Anforderungen spezifiziert werden.

Ein weiterer Punkt bzgl. der Konsistenz ist eine nachhaltige und strukturierte Dokumentation. Sämtliche Verlinkungen und Informationen müssen zugänglich gemacht werden und implementiert werden. Eine Möglichkeit, ein durchgängiges Anforderungsmanagement toolunterstützt umzusetzen ist das SysML Requirements Diagram [6]. Weiterhin wird bspw. durch DOORS der Anforderungsprozess unterstützt. Um die Austauschbarkeit allerdings einheitlich und nachhaltig zu gestalten, bietet sich das Requirements Interchange Format (ReqIF) an, welches

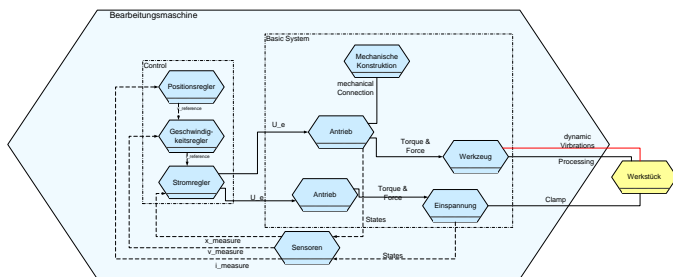


Abb. 4: Systemarchitektur einer Bearbeitungsmaschine

einen standardisierten Austauschprozess der Anforderungen garantiert.

Nachdem die Anforderungen an das System erfasst und beschrieben sind, wird die Systemarchitektur definiert. Die Systemelemente innerhalb der Systemgrenze erfüllen jeweils die erwähnten Funktionen, welche wiederum die entsprechenden Anforderungen erfüllen. Dies erfolgt mit Unterstützung der üblichen Systems Engineering Methoden. Letztlich müssen die Komponenten des Zielsystems sorgfältig ausgewählt werden. Der Markt bietet eine Vielzahl verschiedener

Lösungen, welche zunächst analysiert werden müssen. Hierbei kommt die Simulation zur Anwendung, welche den Aufwand zur Vorabschätzung deutlich reduziert

### IV. SIMULATIONSMODELLE

Für die Entwicklung der Simulationsmodelle ist zunächst, die zuvor beschriebene Systemarchitektur zu betrachten. Diese Architektur wird verwendet, um die Architektur des Simulationsmodells zu definieren. Dies ermöglicht eine klare und durchgängige Entwicklung von Simulationsmodellen und den darin enthaltenen Submodellen, die sehr einfach den zugehörigen Komponenten zugewiesen werden können, siehe Abb. 5. So ist bereits vor Erstellung der Simulationsmodelle ersichtlich, welche Komponenten abzubilden sind und vor

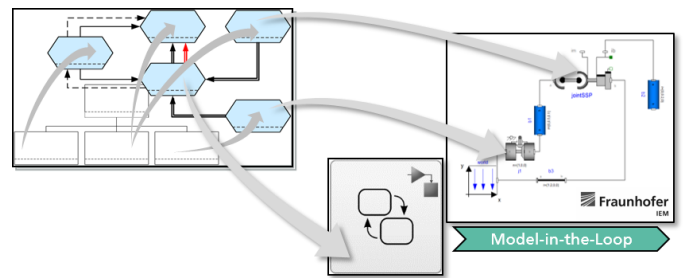


Abb. 5: Systematisches Vorgehen bei der Entwicklung

allein, welche Schnittstellen zwischen den Komponenten abgebildet werden müssen. Die Wechselwirkungen einzelner Systemelemente werden ebenfalls durch das Systemmodell ersichtlich und sind folgend in die Erstellung der Simulationsmodelle umzusetzen.

Betrachtet man die Systemarchitektur, gibt es grundsätzlich zwei elementare Subsysteme des mechatronischen Systems. Die Steuerung/Regelung, die das Verhalten der physikalischen Komponenten des Systems regelt und das eigentliche physikalische System. Auf Seiten der Steuerung werden Algorithmen aus dem Bereich der Regelungstechnik implementiert, die das selbstoptimierende und adaptive Verhalten von intelligenten technischen Systemen repräsentieren.

Der zweite Teil des Systems, welche ebenfalls in Form eines Simulationsmodells aufgebaut wird, ist die physikalische Hardware. Für diesen Zweck werden Mehrkörpersimulationstools angewendet, welche die mechanischen Komponenten des Systems und vor allem deren Wechselwirkungen untereinander abbilden. Hier können Kräfte, Momente und viele weitere Effekte zwischen einzelnen Bauteilen abgebildet werden und in Simulationsstudien analysiert werden.

Nach der Entwicklung der Simulationsmodelle werden diese miteinander in Form einer Systemsimulation zur Analyse genutzt. Es wird zunächst rein virtuell das Verhalten untersucht und optimiert. Sowohl das Steuerungsverhalten als auch die mechanische Konstruktion werden hierbei ausdetailliert.

## V. VIRTUELLE INBETRIEBNAHME

Bei der virtuellen Inbetriebnahme wird die Steuerung des Systems getestet und in Betrieb genommen. Dazu werden die zuvor erstellten Modelle genutzt. Die Steuerung wird in zwei Szenarien getestet:

- Model-in-the-Loop (MiL): Steuerung und Strecke werden miteinander gekoppelt und zunächst virtuell getestet
- Hardware-in-the-Loop (HiL): Das Steuerungsprogramm auf der Steuerungshardware läuft in Echtzeit gegen das physikalische Simulationsmodell

Die virtuelle Inbetriebnahme bietet in beiden Szenarien die folgenden Vorteile:

- Vorverlagerung der Prüfung von Teilsystemen vor dem Zusammenbau, um die Inbetriebnahmezeiten zu senken
- Modellbasierte Optimierung von Anlagenfunktionalitäten und Benutzerfreundlichkeit
- Schulung des Bedien- und Wartungspersonals
- Qualitätssteigerung durch Fehlerinjektion und Wiederverwendung von Testfällen zur Maximierung der Zuverlässigkeit

In Abb. 6 wird die Systemarchitektur für die virtuelle Inbetriebnahme (am Beispiel von ISG virtuos) gezeigt.

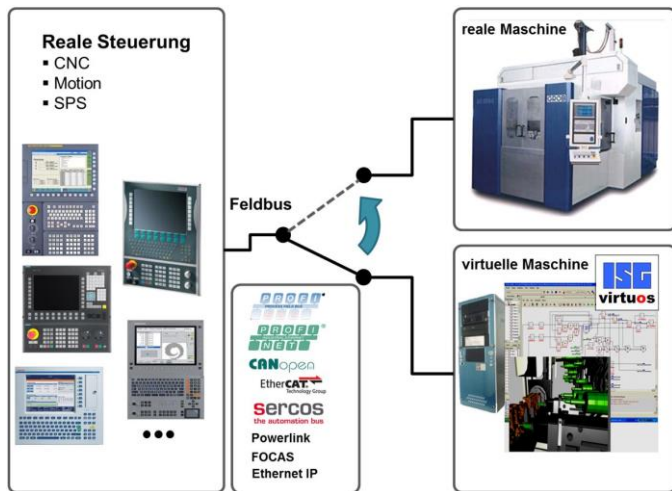


Abb. 6: Architektur für die virtuelle Inbetriebnahme [13]

Um eine effiziente virtuelle Inbetriebnahme zu realisieren, müssen Testfälle aus den Anforderungen abgeleitet werden. Diese werden zur Überprüfung in Checklisten übertragen. Die Vorbereitung dieser Checklisten erfolgt entsprechend dem House of Quality (HoQ) aus dem Quality Function Deployment (QFD). Zunächst werden Kundenwünsche und Bedürfnisse präzise formuliert und dadurch Kategorien (z. B. Abteilungen wie Produktion, Montage, Instandhaltung)

oder Anforderungsbereiche (z. B. Ergonomie, Ökonomie, Umwelt) generiert. Danach wird die Bedeutung der Anforderungen und Bedürfnisse definiert. Der nächste Schritt umfasst eine Bewertung der Erreichung der Kundenpräferenzen. Dabei werden die notwendigen technischen und funktionalen Merkmale definiert und eine Anpassung der Merkmale zur Erfüllung der Anforderungen durchgeführt. Dann werden Beziehungen zwischen Anforderungen und Interdependenzen zwischen Merkmalen geklärt. In einem weiteren Schritt werden die Zielwerte für jedes Produktmerkmal ermittelt. Abschließend wird das Konzept kritisch überprüft, um mögliche Fehler zu finden und Schwachstellen zu identifizieren.

Vor Beginn der virtuellen Inbetriebnahme ist eine Sortierung der Funktionen erforderlich. Die Merkmale mit identifizierten Fehlern und Schwächen werden anhand ihrer Relevanz in der HoQ aufgelistet. Die verbleibenden Merkmale werden ebenfalls nach ihrer Relevanz sortiert und der ersten Gruppe von Merkmalen hinzugefügt. Darüber hinaus werden auch die Fälle von früheren Projekten, vom individuellen Know-how sowie vom Kunden-Feedback hinzugefügt. Bei der virtuellen Inbetriebnahme wird ein besonderer Wert auf die Prüfung der Merkmale gelegt, die im ersten Teil der Liste aufgeführt sind. Für die Testautomatisierung können ergänzend Makros programmiert werden.

Die benötigte Zeit für die Inbetriebnahme kann durch den aufgezeigten Ansatz im Sinne des MBSE und der virtuellen Inbetriebnahme deutlich reduziert werden. Die Projektdauer und die Arbeitszeiten werden aufgrund der gesteigerten Effizienz minimiert. Dazu wird eine konsequente Unterstützung von virtuellen Werkzeugen empfohlen, welche die Durchführung von Test- und Prüfenszenarien bereits in frühen Entwicklungsphasen erlauben.

## VI. ZUSAMMENFASSUNG

In diesem Beitrag ist ein modellbasierter Ansatz des Systems Engineerings für die Entwicklung intelligenter technischer Systeme gezeigt. Der Ansatz fokussiert die virtuelle Inbetriebnahme im Maschinen- und Anlagenbau. Grundlage des entwickelten Ansatzes ist das Systemmodell, das im Entwicklungsprozess um Verhaltensmodelle erweitert wird, die das physikalische Verhalten des Systems darstellen und Simulationen im Rahmen der virtuellen Inbetriebnahme ermöglichen. Aufbauend auf dieser Basis gilt es, die Durchgängigkeit in geeigneten Softwarewerkzeugen abzubilden, um Medienbrüche zu vermeiden. Ein weiterer Meilenstein stellt die Testautomatisierung dar, welche die Effizienz der Entwicklung weiter erhöht.

## VII. LITERATUR

- [1] J. A. Estefan, "Survey of model-based systems engineering (MBSE) Methodologies," INCOSE MBSE Initiative, 2008.
- [2] S. Friedenthal, R. Griego, M. Sampson, "INCOSE model based systems engineering (MBSE) initiative," INCOSE 2007, San Diego, USA, 2007.

- [3] C. Lankeit, M. Lochbichler, A. Trächtler, „Challenges in Requirements Engineering for Mechatronic Systems – Problem Analysis and First Approach“, *Jurnal Teknologi*, 76(4), 2015.
- [4] C. Lankeit, M. Lochbichler, F. Oestersötebier, A. Trächtler, M. Landwehr, „From Goals to Systems via Technical Requirments and Physical Models – Design of an Intelligent Cash Handling System“, In proceedings of the IEEE International Symposium on Systems Engineering, September 28<sup>th</sup> -30<sup>th</sup> 2015, Rome, 2015.
- [5] C. Lankeit, V. Just, A. Trächtler, “Consistency Analysis for Requirements, Functions, and System Elements”, In proceedings of the IEEE Systems Conference, April 2016, Orlando, 2016
- [6] E. Brusa, D. Ferretto, A. Cala, “Integration of heterogeneous functional–vs–physical simulation within the industrial system design activity”, In proceedings of the IEEE International Symposium on Systems Engineering, September 28<sup>th</sup> -30<sup>th</sup> 2015, Rome, 2015.
- [7] R. Barbedienne, Y. B. Messaoud, J. Choley, O. Penas, A. Ouslimani, A. Riviere, “SAMOS for Spatial Architecture based on Multiphysics and Organisation of Systems in conceptual design”, In proceedings of the IEEE International Symposium on Systems Engineering, September 28<sup>th</sup> -30<sup>th</sup> 2015, Rome, 2015.
- [8] VDI Guideline, VDI2206: Design methodology for mechatronic systems, 2004.
- [9] F. Oestersötebier, V. Just, A. Trächtler, F. Bauer, S. Dziwok, “Model-based design of mechatronic systems by means of semantic web ontologies and reusable solution elements,” Proceedings of the ASME 2012 11th Biennial Conference on Engineering Systems Design and Analysis ESDA2012, Nantes, France, 2012.
- [10] T. Weilkens, *Systems Engineering with SysML/UML*, 2008.
- [11] S. Robertson, J. Robertson, “Mastering the Requirements Process: Getting Requirements Right”, Addison Wesley, 3rd revised edition, 2012
- [12] Claude E. Shannon, “Communication in the presence of noise”, Institute of Radio Engineers, 1949
- [13] ISG Webseite: <http://www.isg-stuttgart.de/de/portal.html>