

Informationsaustausch im interdisziplinären Entwicklungsprozess

Holowenko, Olaf¹; Oehm, Lukas²; Majschak, Jens-Peter²; Ihlenfeldt, Steffen¹

Zusammenfassung

Die Entwicklung moderner produktionstechnischer Systeme, z. B. Verpackungs- oder Werkzeugmaschinen, erfolgt aufgrund derer steigender Komplexität zunehmend in interdisziplinären Projektteams. Diesen gehören Entwickler unterschiedlichster Fachbereiche (Domänen) an, wie beispielsweise Vertriebsmitarbeiter, Konstrukteure, Arbeitswissenschaftler, Informatiker oder Designer. Sie alle generieren im iterativen Entwicklungsprozess Inhalte, die wiederum für andere Entwickler als Arbeitsgrundlage notwendig sind und an diese übergeben werden müssen. Der erforderliche, interdisziplinäre Informationsaustausch stellt heute immer noch ein Kernproblem bei der Maschinenentwicklung dar. In der vorliegenden Arbeit wird die aktuelle Situation bei der Informationsübergabe im Entwicklungsprozess analysiert, der Bedarf nach Assistenzsystemen abgeleitet und eine neue Herangehensweise zur Unterstützung der interdisziplinären Entwicklung als theoretische Basis für die Erstellung der Software „Smarte Werkbank“ vorgestellt.

1 Ausgangssituation

Zur industriellen Herstellung von Gütern kommen produktionstechnische Systeme zum Einsatz, z. B. Verarbeitungsmaschinen zur Konsumgüter-Produktion [1] und Werkzeugmaschinen zur Fertigung mechanischer Maschinenkomponenten mit Hilfe von Werkzeugen [2]. Aktuelle Umfragen in [3], [4] haben ergeben, dass sich die Hersteller dieser Maschinen in überdurchschnittlich vielen Fällen als Innovationsführer auf ihrem Gebiet sehen. Forschung und Entwicklung findet danach zu großen Teilen in Deutschland statt, als „Innovations- und Wachstumstreiber“ werden qualifizierte Mitarbeiter gesehen [5]. Der Schutz von Know-how und die Wiederverwendung funktionierender Lösungen sind unerlässlich, um auch zukünftig konkurrenzfähig zu sein. Dies betrifft nach [3], [4] Unternehmen aller Größen.

Von den befragten Herstellern werden meist individuell zugeschnittene Komplettlösungen entwickelt, die bedarfsgerecht an andere Teilsysteme beim Kunden vor Ort angepasst

¹ TU Dresden, IWM

² Fraunhofer, IVV Dresden

werden [3]. Eine Steigerung der Nachfrage nach **kundenspezifischen Systemen** wird als Top-Trend gesehen [5]. Dies wiederum bedeutet, dass innerhalb der Maschinen vermehrt individualisierte Baugruppen verbaut werden, die in kreativen Arbeitsschritten entstehen.

Eine Grundlage zur Erfüllung dieser Anforderung ist die Möglichkeit zur schnellen Anpassung der Anlagen an veränderliche Randbedingungen sowie zur schnellen Reaktion im Fehlerfall [6]. Um unproduktive Stillstandszeiten zu minimieren muss eine hohe Maschinenqualität erreicht werden [3], [7]. **Kurze Entwicklungszeiten** sind besonders vor dem Hintergrund der hohen Löhne in Deutschland marktstrategisch wichtig [8], auch im Hinblick auf die kontinuierlich erstarkende, internationale Konkurrenz.

Im Zuge der digitalen Revolution ab den 1980er Jahren wurden flächendeckend Computer verfügbar. Diese bieten viele neue Möglichkeiten in nahezu allen Bereichen, z. B. grafische Benutzeroberflächen, Individualisierung der Maschinen, flexibilisierte Produktion, leistungsstarke Simulationswerkzeuge oder fortschrittliche Steuerungshardware. Die verbesserten Mittel führen dazu, dass immer **komplexere Aufgaben** gestellt werden und zu lösen sind, welche immer spezialisierteres Fachwissen erfordern. Deshalb geht der aktuelle Trend dahin, die Aufgaben durch interdisziplinäre, hoch spezialisierte Projektteams lösen zu lassen. Dabei wünschen sich die Entwickler laut Umfragen die Nachnutzung bekannter Werkzeuge und Beschreibungsmittel. Als Herausforderung werden „die voraussichtlich hohen Investitionen und eine häufig noch unklare Wirtschaftlichkeitsrechnung für neue Industrie 4.0-Anwendungen“ [9] gesehen.

Der **Zielkonflikt**, als Innovationsführer *kundenspezifische Lösungen* in *verkürzten Zeiten* bei gleichzeitig *steigender Komplexität* der Aufgabenstellung zu erzeugen, stellt schon seit Langem eine große Herausforderung dar. Besonders die interdisziplinäre Kommunikation und Informationsweitergabe sowie das dafür erforderliche, gemeinsame Verständnis der Inhalte sind dabei bekanntermaßen problematisch [10].

In dieser Arbeit wird zunächst die aktuelle Situation bei der Entwicklung komplexer Maschinen analysiert. Daraus wird der Bedarf nach Assistenzsystemen abgeleitet, verfügbare übergreifende Ansätze bewertet und darauf basierend ein Konzept für eine Software zur Assistenz der Entwicklung vorgestellt.

2 Entwicklung von produktionstechnischen Systemen

2.1 Problemfeld

2.1.1 Unterschiedliche Entwickler

An der Entwicklung produktionstechnischer Systeme sind Mitarbeiter verschiedener Fachgebiete beteiligt. Sie unterscheiden sich durch das eingebrachte Fachwissen sowie ihre Ausbildung und Qualifikation. Damit verbunden sind einerseits untereinander abweichende Zuständigkeiten und Berechtigungen im Entwicklungsprozess. Andererseits resultieren daraus verschiedene Sichtweisen auf den Entwicklungsprozess und auf das Entwicklungsergebnis, dargestellt mit diversen Beschreibungsmitteln. Dazu kommen (schwer messbare) persönliche Kompetenzen oder Erfahrungen der Einzelnen, welche entscheidend für den Erfolg der Entwicklung sind.

Die Eingliederung in feste Domänen fällt dabei zunehmend schwer, da mit der Komplexität der Entwicklungsaufgabe immer neue Fachbereiche dazu kommen und zur Aufgabenlösung nicht nur eine Domäne, sondern vielmehr die individuellen Fähigkeiten des Entwicklers wichtig sind. *Beispiele:*

- Elektromeister und Elektrolehrling gehören einer Domäne an, sind aber im Entwicklungsprozess mit gänzlich unterschiedlichen Aufgaben betraut (Planung versus Ausführen der Vorgaben)
- Steuerungsentwickler und HMI-Designer werden meist als einer Domäne zugehörig betrachtet, lösen allerdings gänzlich verschiedene Aufgaben (Treiberentwicklung versus Oberflächendesign)

Reduzierung auf wenige Domänen mit zu vielen geforderten Kompetenzen

In der industriellen Praxis werden aus strukturellen Gründen die Entwickler in unterschiedlichen Domänen gruppiert und somit ihre Kompetenzen generalisiert, wobei häufig die Domänen Maschinenbau, Elektrotechnik und Informationstechnik benannt werden [11–14]. Diese Sicht ist historisch gewachsen [15], [16] und resultiert daraus, dass bis Mitte der 1950er Jahre Maschinen rein mechanisch gesteuert wurden (*Maschinenbau*). Später folgten erste NC-Maschinen, die auf der festen Verdrahtung diskreter Bauelemente basierten (Relaissteuerungen, hydraulische Steuerungen, *Elektrotechnik*). In den 1970er Jahren folgten dann mikroprozessorbasierte (CNC-)Steuerungen, mit welchen sich die *Informationstechnik* als dritte Domänen etablierte. Andere Fachrichtungen, die Einfluss auf die Entwicklung haben (z. B. Kunde, Designer, Vertrieb, Technologe, Materialwissenschaftler), werden bei der Betrachtung meist nicht direkt als Entwickler benannt. Nachteilig an dieser Sichtweise ist, dass die mit der Computertechnik einhergehenden, neuen Aufgaben (GUI-

Design, Simulation, etc.) auf die drei traditionellen Domänen aufgeteilt werden müssen.

Beispiele:

- Maschinenbau: nicht nur mechanische Konstruktion oder Materialauswahl, sondern auch Prozessmodellierung und Produktdesign
- Informationstechnik: nicht nur Steuerungsentwicklung, sondern auch HMI-Design, Inbetriebnahme der Maschine oder Realisierung der Maschinensicherheit

Daraus resultiert, dass zunehmend Aufgaben an die Entwickler gestellt werden, die sich nicht mit ihren erworbenen Kompetenzen decken. Problematisch daran ist, dass dieses Vorgehen konträr zum Trend hin zu immer spezialisierteren Fachkräften ist. Im Ergebnis besteht die Gefahr, dass eine für die Problemstellung optimale Lösung nicht gefunden wird, weil das erforderliche Fachwissen nicht ausreicht oder in seiner fallspezifischen Gewichtung nicht zum Tragen kommen kann.

2.1.2 Informationsaustausch zwischen den Entwicklern

Anforderungserfassung

Den Ausgangspunkt bei der Entwicklung komplexer Maschinen bildet die ausführliche Beschreibung der zu realisierenden Aufgaben, welche als Anforderungen formuliert und in der Regel in einer Anforderungsliste festgehalten werden. Diese wird initial vom Kunden, teilweise unterstützt vom Vertrieb, befüllt und in Form eines Lastenheftes abgelegt. Möglichkeiten zur Klassifizierung von Anforderungen werden u. a. in [14], [17] diskutiert. Zur Beschreibung werden grafische Darstellungen, Tabellen oder Formulare vorgeschlagen, „Texte sind knapp und klar strukturiert abzufassen“ [18]. Grundlegend für einen positiven Ausgang der Entwicklung ist, dass die Anforderungsliste über den gesamten Entwicklungsprozess hinweg aktualisiert und gepflegt wird [12]. *Beispiele:*

- Inhalte: Maschinenfunktion, Aufstellbedingungen, Serviceanforderungen, Design, Bedienbarkeit
- Richtlinien und Normen: Maschinenrichtlinie, Zulassung von Materialien für den Lebensmittelkontakt z. B. durch die Food and Drug Administration (FDA)³

Eine große Herausforderung dabei ist die Erfassung aller notwendigen Anforderungen. Welche Anforderungen für ein spezielles Projekt relevant sind, ist wiederum extrem

³ US-amerikanische Behörde für Lebensmittel- und Arzneimittelsicherheit

unterschiedlich. Die Entscheidung darüber erfordert fundiertes Expertenwissen im entsprechenden Bereich und spiegelt gleichzeitig das Know-how der entwickelnden Firmen wider.

Umgang mit Anforderungsänderungen

Anforderungsänderungen kommen einerseits zustande, wenn Anforderungen bei der Erstellung der Anforderungsliste nicht benannt und nachträglich ergänzt werden, wodurch sich die Anzahl an Anforderungen ändert. Andererseits treten Änderungen auf, wenn bereits benannte Anforderungen ihren Wert ändern. In beiden Fällen müssen alle bis dahin entstandenen Lösungen auf Einhaltung der veränderten Anforderung hin geprüft werden, wobei alle beteiligten Entwickler eingeschlossen werden müssen. *Beispiele:*

- Erhöhung der Nennausbringung
- Verarbeitung zusätzlicher Produktrezepturen

Anforderungsänderungen sind aufwändig zu handhaben und führen oft zu einer Steigerung der Entwicklungskosten oder -dauer. Dies ist darin begründet, dass Arbeitsschritte mehrfach durchgeführt werden müssen, um zu prüfen, ob die Änderungen die erarbeitete Lösung betreffen und diese Lösung ggf. geändert werden muss. Mehrkosten können entstehen, wenn bereits umgesetzte Lösungen angepasst und Bauteile nicht nur im Modell, sondern auch in bereits gefertigten Baugruppen ersetzt werden müssen.

Unterschiedliche Beschreibungsmittel zum Informationsaustausch

Innerhalb der Entwicklungsschritte werden unter Zuhilfenahme fachspezifischer Vorgehensweisen sowohl die Anforderungen, als auch die vorher von anderen Entwicklern festgelegten Inhalte aufgegriffen und in kreativen Arbeitsschritten in Lösungen umgesetzt. *Beispiele:*

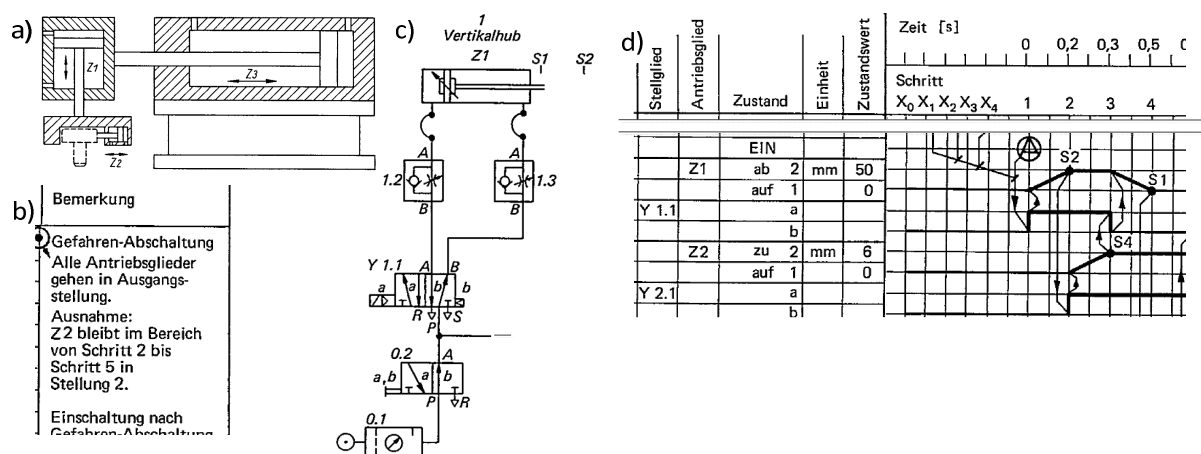


Abbildung 1: Ausschnitt aus der Darstellung der Funktionsfolge einer Zubringeinrichtung in [19]: a) Lageplan, b) Fehlerreaktion, c) Fluid-Schaltplan, d) Funktionsdiagramm

- Sollprozess aus Sicht der Produkte wird während der Steuerungsentwicklung um eine Maschinensicht erweitert
- Kollisionsmöglichkeiten in der Mechanik führen zu Verriegelungen innerhalb der Steuerung
- Verdrahtung der Sensoren oder Aktoren legt fest, welches Signal an welcher Stelle abgegriffen werden muss
- Verhalten von Sensoren und Aktoren beeinflusst beispielsweise, in wie weit Signalfilter eingesetzt werden müssen

Zur Dokumentation der Entwicklungsergebnisse kommen, neben der natürlichen Sprache, die etablierten Beschreibungsmittel der jeweiligen Entwickler zum Einsatz, da diese auf die darzustellenden Hauptinhalte zugeschnitten sind. Beispiele sind Prinzipskizzen, Schaltpläne oder Funktionsdiagramme (Abbildung 1), aber auch Petri-Netze, Flussdiagramme, und viele weitere. Mit diesen Beschreibungsmitteln werden die Entwicklungsergebnisse notiert und zur Dokumentation als Skizzen, in Zeichnungen, in Schaltplänen und in Stücklisten abgelegt. Die entstehenden Dokumente werden zwischen Entwicklern ausgetauscht und dienen ihnen als Arbeitsgrundlage.

Eine große Herausforderung dabei ist, dass die Entwickler die unterschiedlichen Beschreibungsmittel kennen müssen, um die für sie relevanten Inhalte extrahieren zu können. Der Erfolg dieser Informationsübersetzung hängt dabei einerseits stark von den individuellen Erfahrungen und Kenntnissen des *Empfängers* ab, der die Dokumente lesen können muss. Andererseits lassen verschiedene Beschreibungsmittel mehrere Darstellungsmöglichkeiten für identische Inhalte zu, weshalb die Erfahrungen und die Qualifikation des *Senders* eine wichtige Rolle spielt. Als Beispiel sei das Petri-Netz in Abbildung 2 genannt, welches trotz formaler Basis unterschiedlich dargestellt werden kann.

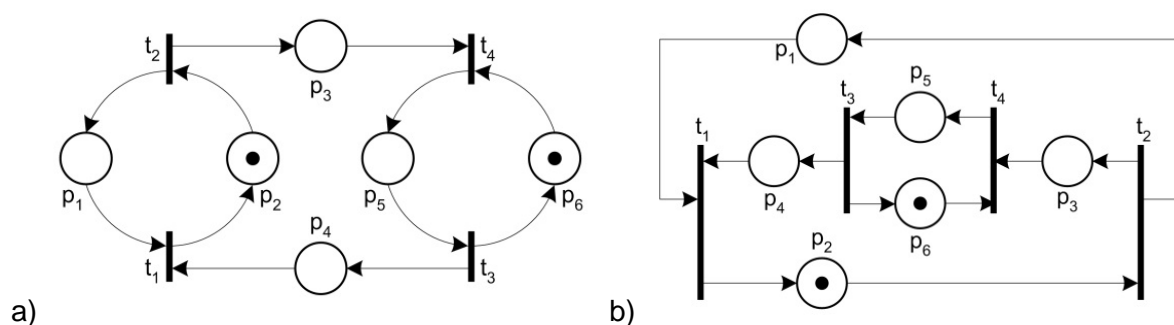


Abbildung 2: Beispiele für zwei identische Petri-Netze (Synchronisationsgraph), nach [20]
a): transparente Darstellung, b): unübersichtliche Darstellung

Mehrfaches, manuelles Eingeben von Inhalten

Wie bereits beschrieben wird die Maschine aus unterschiedlichen Blickrichtungen heraus entwickelt. Dabei gibt es eine Vielzahl von Inhalten, die bereits als Information in einem oder mehreren Dokumenten vorliegen und in weiteren Entwicklungsschritten weiter verwendet werden. *Beispiele:*

- Elektroplan: Fokus der Elektrotechnik liegt auf Energiefluss und Verknüpfungen, Steuerungsentwickler benötigt Anschlussbelegung von Sensoren oder Aktoren
- Prozesssicht: Technologie oder Maschinenbediener schaut aus Produktsicht, Steuerungsentwickler aus Maschinensicht

Die Übergabe der Inhalte erfolgt oft mit universellen, plattformunabhängigen Dateiformaten, welche mit Standard-Programmen gelesen werden können. Durch den Export in solche allgemeinen Dateiformate gehen allerdings Inhalte verloren, die beim Lesen mühsam wieder erfasst und rekonstruiert werden müssen. Dies ist mit Mehraufwand und damit höheren Entwicklungskosten verbunden. Weiterhin ist das manuelle Lesen und Verstehen von Informationen fehleranfällig und, gerade vor dem Hintergrund steigender Komplexität bei sinkenden Entwicklungszeiten, nicht mehr zeitgemäß.

Redundante Entwicklungsinhalte

Eng damit verbunden sind redundante Inhalte in unterschiedlichen Dokumenten. Abbildung 3 zeigt ein Beispiel aus [19], in dem eine Zubringeeinrichtung dargestellt wird. Es sind die unterschiedlichen Sichtweisen der Entwickler zu erkennen. Während (a) die Lage der Baugruppen illustriert, ist in (c) die Verschaltung dieser aus Sicht der Fluidtechniker dargestellt. Die Darstellung (d) zeigt den gewünschten Sollprozess als Funktionsdiagramm.

Weitere *Beispiele:*

- Sensor-Aktor-Belegung in Fluid- und Konstruktionsplan sowie Steuerungssoftware
- Bauteilbezeichnung in Getriebe- und Konstruktionsplan

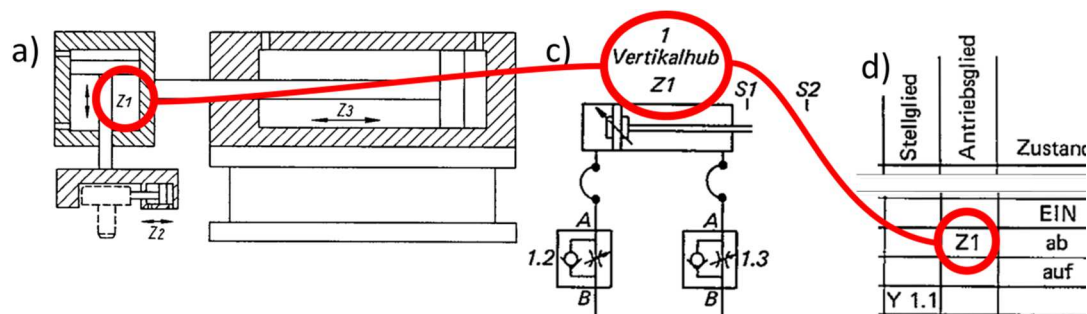


Abbildung 3: Redundante Inhalte, Beispiel aus Abbildung 1, nach [19], Achse Z1 im a) Lageplan, c) Fluid-Schaltplan, d) Funktionsdiagramm

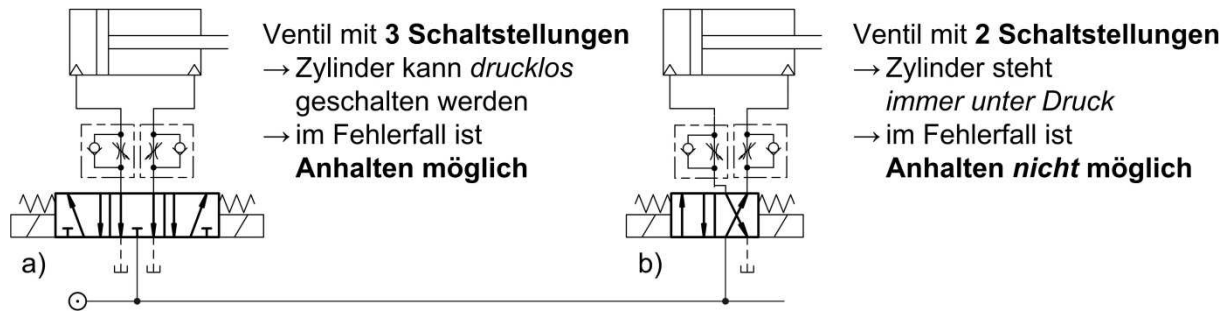


Abbildung 4: Beispiel für Pneumatik-Schaltung, a) drei, b) zwei Schaltstellungen

Redundanzen verursachen dann Probleme, wenn z. B. nach Änderungen in unterschiedlichen Dokumenten gleiche Inhalte verschiedene Werte besitzen. In dem Fall ist unklar, welcher Inhalt korrekt ist. Die Folgen daraus können Fehler an der Maschine oder kostenintensiver Mehraufwand bei der Entwicklung sein.

Fehlende Beschreibungsmöglichkeiten zur Eingabe spezieller Informationen

Wie sich bei der Analyse des Standes der Technik herausgestellt hat gibt es einige Inhalte, die mit klassischen Beschreibungsmitteln nicht explizit oder überhaupt nicht erfasst werden können, die aber dennoch von größtem Interesse bei der Entwicklung sind. *Beispiele:*

- Kollisionsmöglichkeiten an der Mechanik, die während der Konstruktion entstehen und in der Steuerung oder bei der Bewegungsplanung bekannt sein und berücksichtigt werden müssen
- Fehlerszenarien und deren potenzielle Ursachen, die für Fehlermeldungen oder später beim Service der Maschine von Interesse sind (vgl. Abbildung 1, b)
- Verwendung von zwei- gegenüber drei-Wegeventilen in Verbindung mit den damit möglichen Fehlerreaktionen, welche nur implizit aus dem Fluidschaltplan ersichtlich sind (Abbildung 4)

Durch fehlende Beschreibungsmittel können die genannten Inhalte nicht oder in nur ungenügender Form an andere Entwickler übergeben werden. Dies führt zu Mehraufwand, die Informationsbasis der Entwicklung ist unvollständig.

Vernetzung von Entwicklungswerkzeugen

Für die meisten Entwickler existieren unterstützende Softwarewerkzeuge, die von unterschiedlichen Herstellern vertrieben werden und verschiedene Schnittstellen anbieten. Weit verbreitet sind Office-Programme für das Verfassen von Texten, das Erstellen von Präsentationen oder die Tabellenkalkulation. Bei Anwendungen außerhalb des Funktionsumfangs dieser Programme werden spezialisierte Werkzeuge eingesetzt, z. B. für

Requirements Engineering, Berechnung und Simulation, Konstruktion, Steuerungsentwicklung, Elektroplan- oder Fluidplan-Erstellung.

Die Übergabe von Inhalten zwischen verschiedenen Programmen kann auf unterschiedlichen Wegen erfolgen:

- manuelles Übertragen der Inhalte durch den Entwickler
- standardisierte Export-Import-Formate, z. B. DXF, STL, Collada, AutomationML, SysML, PLCopen-XML
- direkte Interaktion der Werkzeuge durch angebotene Schnittstellen, z. B. MS Office DDE API, TwinCAT oder IndraWorks Automation Interface, SolidWorks API

Eine Herausforderung in diesem Bereich ist es einerseits, die unterschiedlichen Werkzeuge miteinander zu verbinden, andererseits verursachen die Schnittstellen selbst oft Probleme. Die *manuelle Übertragung* ist zeitaufwändig und es kann zu Übertragungs- und Interpretationsfehlern kommen. Auch der *Im- oder Export von Dateiformaten* ist oft fehlerbehaftet, wenn er nur unzureichend von den Werkzeugen unterstützt wird. So kommt es vor, dass einzelne Inhalte in unterschiedlichen Programmen verschieden repräsentiert werden [21], wodurch Fehlfunktionen ausgelöst werden können. Weiterhin werden von unterschiedlicher Software teilweise sehr unterschiedliche Dateiformate unterstützt, vgl. [22]. Für eine direkte Interaktion von Werkzeugen über spezielle Schnittstellen müssen diese einerseits in ausreichendem Umfang implementiert sein. Andererseits sind deren Implementierungen nicht standardisiert und teilweise sehr unterschiedlich, was die Anwendung erschwert [23].

Dokumentation des Entwicklungsverlaufes

Unterschiedliche gesetzliche Vorgaben [24–26] verlangen eine ausführliche Dokumentation des Entwicklungsverlaufes. Wichtig für die Rückverfolgbarkeit ist dabei, dass Einträge innerhalb der Dokumentation nicht nachträglich unerkannt geändert werden können. Mit steigender Komplexität der Entwicklungsaufgabe wird auch die Dokumentation der Entwicklung komplexer, was zu einem erhöhten Aufwand führt. *Beispiele:*

- Dokumentation von Entscheidungen
- Änderungen an Baugruppen

In vielen Fällen erfolgt die Dokumentation des Entwicklungsverlaufes manuell. Dies ist mit erheblichem Mehraufwand verbunden und wird oft aus Gründen der Zeitersparnis auf ein Minimum reduziert [27]. Dies führt wiederum dazu, dass u. U. nicht alle Änderungen dokumentiert werden und die Dokumentation damit unvollständig oder fehlerhaft ist.

2.1.3 Fazit

Das dargelegte Problemfeld führt zur Notwendigkeit einer besseren Unterstützung des Entwicklungsprozesses. Dies betrifft vor allem die **Integration aller Entwickler**, die am Entwicklungsprozess beteiligt sind, und die **Unterstützung des Informationsflusses** zwischen den beteiligten Entwicklern. Verschiedene Ansätze fokussieren diesen Bedarf. Diese sollen im folgenden Abschnitt dargestellt und vor dem Problemfeld bewertet werden.

2.2 Lösungsansätze im Stand der Technik

2.2.1 Anforderungen an Assistenzsysteme

Anforderungen an die im Folgenden bewerteten Ansätze sind

- Integration verschiedenster Entwickler, nicht nur der etablierten Domänen Maschinenbau, Elektrotechnik und Informationstechnik
- Realisierung von Anforderungserfassung und Änderungsverfolgung
- Beherrschung von Inkonsistenzen und Redundanzen in den Entwicklungsinhalten
- Bereitstellung von Möglichkeiten zur Notation untypischer Inhalte
- Gewährleistung von Erweiterbarkeit durch herstellerunabhängige Vernetzung von Entwicklerwerkzeugen und Beschreibungsmitteln
- Dokumentation des Entwicklungsverlaufes
- Eignung für Entwicklung von Sondermaschinen auch in KMU

2.2.2 Kommerzielles Umfeld

Mit der *Aucotec Engineering Base* [28], [29] wird das Ziel verfolgt, die inhaltliche, sprachliche und technologische Koordinierung global verteilter Projekte mit unterschiedlichen Domänen zu ermöglichen. Dabei sollen auch redundante und inkonsistente Inhalte beherrscht werden. Zur einfachen Handhabung werden Dokumente und Modelle mit MS Visio erstellt, die Ablage und Verwaltung von Daten aus fachspezifischen Entwicklertools soll möglich sein. Mit dem Programm lassen sich einige Fachprogramme verbinden, z. B. SolidEdge, SAP oder ELCAD. Das System ist konzipiert für KMU und auf die Entwicklung prozesstechnischer Maschinen sowie Elektroplanung ausgerichtet. Andere Fachrichtungen, deren Beschreibungsmittel oder Werkzeuge werden nicht unterstützt.

Mit dem *SAP Engineering Control Center* (SAP EC-TR) wird angestrebt, etablierte Entwicklerwerkzeuge an die Software SAP PLM / SAP ERP anzubinden. Dazu wird eine Integrationsplattform geschaffen, die dies für einige Werkzeuge aus den Bereichen mechanische und elektrische Konstruktion unterstützt. Als Funktionalitäten, die die Entwicklung vereinfachen sollen, stehen z. B. Projektmanagement, Dokumentenmanagement oder das Erzeugen von Stücklisten zur Verfügung [30]. SAP ist für den Einsatz in großen Unternehmen konzipiert, die Umstellung auf SAP lohnt sich für KMU nur in seltenen Fällen [31]. Zentrale Entwicklungszweige, wie z. B. die Steuerungsentwicklung, werden nicht unterstützt.

Die Softwarelösung EPLAN EEC geht zurück auf [32], [33]. Im Kern wird ein Engineering-Ansatz mit „virtuellen mechatronischen Komponenten“ angewandt, mit dem aus unternehmensspezifischen Baukästen Stromlaufpläne oder Steuerungssoftware erstellt

werden kann. Im verwendeten Baukasten werden domänenspezifische Templates und Makros definiert, auf Basis derer ein automatisiertes Generieren von Elektro- oder Fluidplänen, der Mechanikkonstruktion sowie SPS-Programmen möglich ist [34]. Es werden die Fachrichtungen Konstruktion, Elektrotechnik und Steuerungsprogrammierung adressiert, andere Entwickler werden nicht integriert. Die Ergebnisse der Entwicklung werden in den üblichen Beschreibungsmitteln der Domänen generiert und gespeichert [34]. Das System basiert auf einem Baukastensystem vorgefertigter Komponenten, eine manuelle Nacharbeit ist nicht vorgesehen. Dadurch ist der Ansatz für sich ständig ändernde Sondermaschinen und die damit verbundenen Anforderungen wenig geeignet. Es werden nur ausgewählte Entwicklerwerkzeuge unterstützt, was nachteilig bei der Beherrschung der Komplexität ist.

Mit dem *Siemens Automation Designer* steht ein PLM-System zur Verfügung, welches zum Ziel hat, dass die relevanten Informationen die Ebene der „Digitalen Fabrik“ verlassen und die Ebene der Produktionsanlage erreichen. Dabei setzt der Automation Designer auf eine (PLM-typische,) interdisziplinäre Datenbasis [35]. Das Konzept der Software beruht auf Vorlagenbibliotheken, kombiniert mit im Framework definierten Schritten des Engineering und Abhängigkeiten zwischen den Disziplinen [35]. Das System ist gebunden an den Hersteller Siemens und unterstützt nur die traditionellen Disziplinen. Die Beherrschung von Inkonsistenzen oder Redundanzen wird nicht adressiert, auch dieses System fokussiert auf modulare Maschinen und nicht auf Sondermaschinen.

Mit WSCAD Suite steht ein E-CAD-Werkzeug zur Verfügung, welches die übergreifende Zusammenarbeit von Elektrotechnik, Schaltschrankbau, P&ID, Fluidtechnik, Gebäudeautomation und Elektroinstallation fokussiert [36]. Das System basiert auf einer zentralen Datenbank, von der aus das aktuelle Projekt bearbeitet wird. Dadurch sollen Redundanzen im Datenbestand vermieden werden. Über offene Schnittstellen können z. B. PDM-, PLM- oder ERP-Systeme angebunden werden. Aus dem Datenbestand können verschiedene Dokumente wie Elektro- oder Fluidschaltpläne, Materiallisten oder auch E/A-Belegungspläne abgeleitet werden [37]. Damit integriert WSCAD nur wenige ausgewählte Entwickler und bildet den Informationsfluss nur unzureichend ab.

2.2.3 Universitäres Umfeld

Im SFB 614 wurde die Vision verfolgt, eine neue Konstruktionslehre des „Entwurfs der maschinenbaulichen Systeme von morgen“ zu etablieren [38]. Der Fokus lag dabei auf der Konzeption selbstoptimierender Systeme. Im Rahmen des Projektes wurde die Spezifikationstechnik *CONSENS* entwickelt, welche zur Beschreibung von Prinziplösungen fortgeschrittener mechatronischer Systeme dient. Als Beschreibungsmittel kommt die Sprache *MechatronicUML* zum Einsatz. Neben den Entwurfsmethoden wurden auch

spezielle Entwicklungswerkzeuge entwickelt, z. B. Mechatronic Modeller und Embedded Modeller. Aktuell werden die Ideen im BMBF-Spitzencluster „Intelligente Technische Systeme Ostwestfalen-Lippe (it's OWL)“ weiter verfolgt. Ziel dabei ist es, die Methoden und Werkzeuge in die Industrie zu transferieren. In den Ansätzen wird als Lösung eine „umfassende fachdisziplinübergreifende Systembetrachtung“ angestrebt [39].

Der SFB 768 „Zyklusmanagement von Innovationsprozessen“ [40] zielt darauf ab, die Planung von Innovationsprozessen zu verbessern. Dazu werden Methoden entwickelt, mit denen Einflussgrößen auf den Innovationsprozess abschätzbar werden und dieser dadurch besser zu handhaben ist. Es sollen sogenannte *Product-Service Systems* entwickelt werden, welche „Sachleistung aus Mechanik, Elektrik/Elektronik und Software“ und Dienstleistungsanteile zusammenfassen [40]. Im Projekt wird ein Ansatz zum Projekt- und Änderungsmanagement verfolgt, welcher „die Planung, Steuerung und Kontrolle von Zyklen und ihrer Wechselwirkungen“ umfasst [41]. Es fokussiert damit nicht primär auf den Informationsfluss und die Integration vieler Entwickler.

Neuerliche Bestrebungen widmen sich der Entwicklung des Datenformats *AutomationML* (DIN EN 62714). Ziel dabei „ist die möglichst vollständige Beschreibung kompletter Anlagen mitsamt den dort enthaltenen Komponenten über alle Engineering-Disziplinen und -Phasen hinweg“ [42]. Es basiert auf CAEX (IEC 62424) zur Abbildung hierarchischer Objektstrukturen und bildet eine Brücke über andere, standardisierte Formate. Aktuell werden Geometrie und Kinematik über Collada (ISO/PAS 17506) sowie die „Beschreibung des Verhaltens in Bezug auf logische Daten von verschiedenen AutomationML Objekten“ [43] mit PLCopen XML abgebildet. „Der Hauptfokus von AutomationML liegt im Austausch der Daten, nicht in der Implementierungsnähe oder Ablauffähigkeit“, „der herstellerunabhängige Austausch von beliebigen Steuerungsprogrammen ist bislang kein Ziel von AutomationML“ [42]. Damit werden die traditionellen Domänen ohne Steuerungsentwicklung angesprochen. Laut [42] ist AutomationML eine eingetragene Marke der Siemens AG und damit nicht herstellerunabhängig. Da es sich dabei um Austauschformate handelt, muss AutomationML durch geeignete Software unterstützt werden [42], Beispiele sind in [44] aufgelistet.

2.2.4 Fazit

Das Hauptdefizit im aktuellen Stand der Technik ist der momentan nur mangelhaft unterstützte Informationsaustausch zwischen den Entwicklern. Zwar existieren Assistenzsysteme, die sich dem Thema annehmen, diese unterstützen allerdings nur ausgewählte Domänen oder den Informationsaustausch in bestimmten Bereichen. Damit werden die beschriebenen Probleme bei der Entwicklung produktionstechnischer Systemen bisher nur unzureichend gelöst.

3 Die Smarte Werkbank

3.1 Einordnung

Der identifizierte Entwicklungsbedarf wird im Rahmen des Projekts „Smarte Werkbank“ adressiert. Resultierend aus den im Stand der Technik separat ausgeprägten Informationsplattformen und Entwicklungsabläufen sowie den über Schnittstellen realisierten Informationsflüssen wird in diesem Projekt das Paradigma umgekehrt und der Informationsfluss im interdisziplinären Gesamtentwicklungsablauf ins Zentrum gerückt.

Im Projekt wird ein Assistenzsystem entwickelt mit dem Ziel, den Entwicklungsprozess zu verkürzen durch die Einbeziehung der unterschiedlichsten Entwickler und Informationen, die Sicherstellung eines zielgenauen Informationsaustausches und einer bedarfsgerechten Informationsaufbereitung. Zudem sollen Fehlermöglichkeiten reduziert und die Wirtschaftlichkeit der Entwicklung gesteigert werden. Dabei werden keine neuen Beschreibungsmittel, Entwicklerwerkzeuge oder Vorgehensmodelle entstehen. Vielmehr sollen sich bestehende Ansätze mit dem Assistenzsystem kombinieren lassen.

Die „Smarte Werkbank“ besteht dazu im Kern aus einem Modell des Informationsflusses. Dieses wird in einer Plattform umgesetzt, welche den Austausch und die Aufbereitung der Inhalte realisiert. Weiterhin werden Schnittstellen (z. B. „Automation Interface“, standardisierte Beschreibungsmittel, etc.) zu spezialisierten Entwicklerwerkzeugen (z. B. CAD-Software, Steuerungsentwicklungsframeworks, etc.) geschaffen, die deren Integration in ein Gesamtsystem ermöglichen (vgl. Abbildung 5).

Im Folgenden wird das Modell des Informationsflusses vorgestellt. In den anschließenden Kapiteln werden die Grundidee der Modellierung des Informationsaustausches innerhalb des Entwicklungsprozesses und Anwendungsszenarien beschrieben. Abschließend wird aufgezeigt, wie mit Hilfe eines solchen Modells die dargelegten Defizite gelöst werden können.

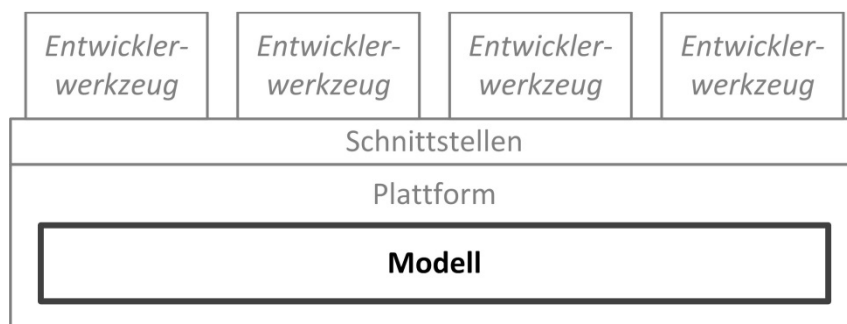


Abbildung 5: Überblick über den Aufbau der Smarten Werkbank

3.2 Modell des Informationsflusses

Das *Modell des Informationsflusses* bildet den Kern des Ansatzes. Es dient dazu, die Informationsflüsse innerhalb des Entwicklungsprozesses zwischen den Entwicklern abzubilden. Die für das Modell erforderlichen Modellelemente ergeben sich aus dem Entwicklungsprozess. Dieser setzt sich üblicherweise aus Entwicklungsschritten zusammen, welche die Lösung einer gestellten Entwicklungsaufgabe kapseln. Einem Entwicklungsschritt sind ein oder mehrere Entwickler zugeordnet, welche in teils kreativen Arbeitsschritten die Lösung erarbeiten. Dabei wird entweder die Aufgabe in kleinere Teilaufgaben untergliedert, oder aber die Lösung als Komplex erarbeitet.

Eingangsgrößen eines jeden Entwicklungsschrittes sind sowohl Anforderungen an das Ergebnis des Schrittes, als auch erweiterte Teillösungen aus vorangegangenen Entwicklungsschritten. Als Ergebnis des Entwicklungsschrittes entstehen einerseits Lösungen für die gestellten Aufgaben, andererseits können auch neue Anforderungen gestellt werden, die in nachfolgenden Entwicklungsschritten berücksichtigt werden müssen. Ändern sich dadurch Eingangsgrößen vorgelagerter Entwicklungsschritte, müssen diese Schritte iterativ erneut bearbeitet werden.

Der Entwicklungsprozess beginnt mit einem Entwicklungsschritt, dessen Aufgabe die Identifizierung, Konkretisierung und Spezifizierung von Anforderungen ist. Diese werden in einer Anforderungsliste gesammelt und dienen als Ausgangspunkt der Entwicklung. Das Ende eines Entwicklungsprozesses ist erreicht, sobald alle gestellten Anforderungen berücksichtigt wurden.

Modellelemente

Aus dieser Betrachtung ergeben sich die Elemente für das Modell innerhalb der Smarten Werkbank, dessen prinzipieller Aufbau in Abbildung 6 dargestellt ist. Einerseits handelt es sich um grundlegende, relevante Inhalte, welche in **Anforderungen** (1) und **Lösungen** (2) unterschieden werden. Während Anforderungen keine konkrete Realisierung, wie beispielsweise Funktionsprinzipien, Baugruppen oder Algorithmen beinhalten, bilden die Lösungen eben diese Umsetzungen ab. Die Änderung von Inhalten erfolgt innerhalb von **Entwicklungsschritten** (3). Diese kapseln in sich geschlossene Entwicklungsaufgaben, welche allerdings auf andere Entwickler ausgelagert werden. Entwicklungsschritte dienen damit der Strukturierung und Organisation des Entwicklungsprozesses und bündeln die für die Bearbeitung der Aufgabe erforderlichen Informationen, wie beispielsweise Aufgabenbeschreibung, Zeitraum oder beteiligte Entwickler. Die eigentliche Überführung der Anforderungen in Lösungen erfolgt in den spezialisierten Entwicklerwerkzeugen, die über

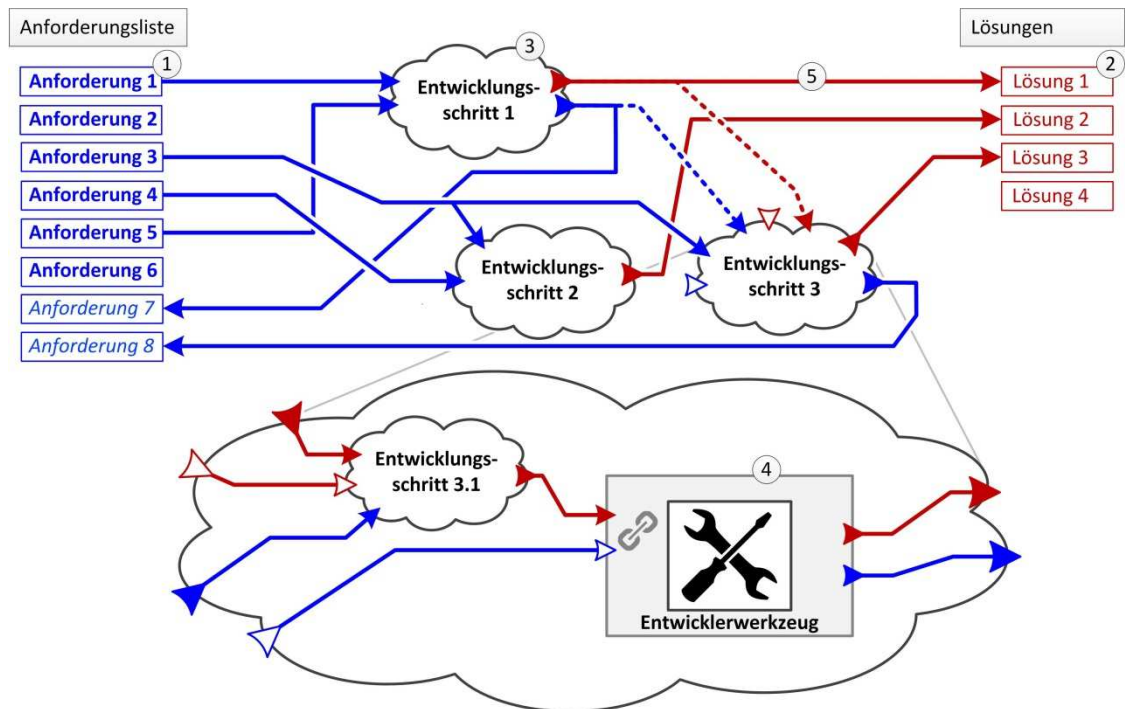


Abbildung 6: Prinzipieller Aufbau des Informationsfluss-Modells

Schnittstellen (4) an das Modell angebunden sind. Anforderungen und Lösungen sind mit den Entwicklungsschritten und Schnittstellen verbunden, wobei diese **Verbindungen** (5) die Relationen zwischen den Elementen und damit den eigentlichen Informationsfluss darstellen.

Eigenschaften des Modells

Mit dem beschriebenen Modell werden die **Zusammenhänge zwischen Anforderungen und Lösungen explizit formuliert**. Dadurch wird eine übergreifende Zuordnung zwischen den Anforderungen, den an den Entwicklungsschritten beteiligten Entwicklern, den eingesetzten Werkzeugen und den Lösungen geschaffen. Es wird gefordert, dass alle verknüpften Anforderungen explizit erfüllt sein müssen, um einen Entwicklungsschritt zu beenden. Der Entwicklungsprozess als Ganzes ist abgeschlossen, wenn *alle* Anforderungen explizit erfüllt wurden.

Durch **Modularisierung der gestellten Aufgabe** in Teilaufgaben (durch die Zuweisung von Entwicklungsschritten) werden die Trennung von Zuständigkeiten des einzelnen, an einem Schritt beteiligten Entwicklers und eine Fokussierung auf kleine Teilaufgaben erreicht.

Der Informationsaustausch zwischen einzelnen Entwicklern wird auf die **notwendigen Inhalte beschränkt**. Es werden nur die Inhalte übergeben, die von anderen Entwicklungsschritten bzw. den daran beteiligten Entwicklern nachgefragt werden. Dadurch wird das Datenaufkommen auf das nötige Minimum beschränkt, was gerade vor dem Hintergrund der steigenden Komplexität relevant ist. Zur Datenübergabe werden die Übergabeformate

verwendet, die vom *Empfänger* verstanden werden. Da nur der *Sender* den Inhalt vollends versteht, kann auch nur dieser die Inhalte extrahieren, die der Empfänger benötigt.

Im Ansatz wird weiterhin die explizite **Berücksichtigung und Verarbeitung bisher nicht explizit beschriebener Inhalte** verfolgt. Gemeint sind damit die in heutigen Systemen nicht berücksichtigten Inhalte, die durch ihr Fehlen Mehraufwand im Entwicklungsprozess verursachen. Als Beispiel sei eine fehlende Beschreibung von Kollisionsszenarien genannt.

Eine weitere Anforderung ist, dass den **Eingangsgrößen** der Entwicklungsschritte **Werte zugewiesen sind**. Eingehende Anforderungen oder Teillösungen müssen damit zwingend mit Werten belegt werden, da sie die Grundlage für die Bearbeitung der Teilaufgabe sind. Sind keine konkreten Werte bekannt, müssen Schätzwerte angenommen werden, welche später revidiert werden können. Diese Revision kommt einer Anforderungsänderung gleich und muss entsprechend behandelt werden.

Durch **Zuordnung von Metadaten zu Modellelementen** wird der Verlauf der Modell-erstellung rückverfolgbar und eine Dokumentation des Entwicklungsprozesses möglich. Als Metadaten sind beispielsweise Angaben zum Ersteller, Erstellungs-/Änderungszeitpunkt, etc. denkbar.

Modellerstellung

Aufgrund der Dynamik des Entwicklungsprozesses kann das Modell initial nicht komplett erstellt werden. Vielmehr soll es mit dem Fortschritt der Entwicklung wachsen und an die Bedürfnisse und Erfordernisse angepasst werden. Daher werden zur Modellerstellung vom Projektleiter Entwicklungsziele als Meilensteine definiert und mit Entwicklungsschritten untersetzt. Diesen werden einerseits Anforderungen oder vorhandene Teillösungen zugewiesen. Andererseits werden verantwortliche Entwickler, welche die Lösung der Aufgabenstellung erarbeiten, geplante Entwicklungszeiten, verfügbare finanzielle Mittel, etc. festgelegt. Die Modellerstellung wird von grafischen Konfigurationswerkzeugen unterstützt.

Verantwortliche Entwickler agieren innerhalb des Entwicklungsschrittes als Teilprojektleiter und haben die Möglichkeit, die Aufgabe weiter in Teilaufgaben zu untergliedern oder aber eigene Werkzeuge einzusetzen, um die Lösung zu erarbeiten. Stellt sich während der Bearbeitung heraus, dass notwendige Vorgaben in Form von Anforderungen oder Teillösungen fehlen, sollen diese aus dem Entwicklungsschritt heraus als notwendige Inhalte definiert werden. Die Herkunft der Inhalte muss wiederum vom übergeordneten Projektleiter festgelegt werden, wodurch eine klare Aufgabentrennung realisiert wird.

3.3 Anwendungsszenarien

Um das theoretisch dargelegte Prinzip der Smarten Werkbank zu verdeutlichen, werden beispielhaft drei mögliche Anwendungsszenarien aus verschiedenen Entwicklungsphasen skizziert.

Entwickler: HMI-Gestaltung durch Steuerungsentwickler

Im Rahmen eines Entwicklungsauftrags soll eine Maschinensteuerung inklusive Human Machine Interface (HMI) entworfen werden. Diese Aufgabe hat der Steuerungsentwickler bei vorangegangenen Aufträgen bearbeitet, ohne selbst tiefgreifendes Wissen über die menschengerechte Gestaltung eines solchen Geräts zu besitzen. Verbessert werden kann die Situation indem Vertreter aus den Fachbereichen Design und Arbeitsgestaltung involviert werden, welche jeweils bereichsspezifische Anforderungen an ein HMI einbringen und in Kooperation mit einem Interfaceprogrammierer ausarbeiten können. Dadurch kann dem Kunden ein in Bedienbarkeit, Funktionalität und Qualität deutlich höherwertiges Produkt erstellt werden, welches die Kundenzufriedenheit und damit auch -bindung stärkt.

Information: Änderung einer Anforderung während der Entwicklung

Während der Entwicklung einer Maschine wird vom Kunden nachträglich eine Erhöhung der Ausbringung gewünscht. Durch den Einsatz der Smarten Werkbank zur Entwicklung dieser Maschine kann die initiale Anforderung angepasst und direkt der Einfluss auf sämtliche Entwicklungsschritte ermittelt werden, welche von dieser Änderung betroffen sind. Dadurch kann beispielsweise direkt erkannt werden, dass der aktuelle, vom Technologen vorgeschlagene Prozess in der Ausbringung begrenzt ist. Der Entwickler der Mechanik stellt fest, dass diese Änderungen eine kritische Erhöhung der Eigenfrequenz der aktuellen Maschine zur Folge hat und Komponenten ersetzt werden müssen. Auch die Qualitätssicherung kann die geforderte Ausbringung nicht mit den vertraglich festgelegten Prüfverfahren garantieren. Das Management kann durch diese Aussagen zeitnah eine Abschätzung der Zusatzentwicklungszeit und -kapazität vornehmen und dem Kunden detailliert die zusätzlich entstehenden Kosten auflisten.

Dokumentation: Fehler bei Maschineninbetriebnahme

Nach der durch die Smarte Werkbank unterstützten Entwicklung und Fertigstellung einer Maschine zum Fördern von zylindrischen Bauteilen stellt ein Mitarbeiter fest, dass der verwendete Sauggreifer keinen Unterdruck zum Transport des Werkstücks aufbauen kann. Daraufhin öffnet dieser Mitarbeiter die 3D-Visualisierung der Maschine, wählt die entsprechende Baugruppe aus und fügt eine Notiz in Form einer textbasierten Beschreibung sowie eines Videos des Fehlers hinzu. Außerdem liest er mittels einer integrierten

Schnittstelle die Sensor- und Maschinendaten aus, aus denen hervorgeht, dass im Vakuumsystem kein Druckanstieg während des Greifens erfolgt. Daraus schlussfolgert er, dass der Sauger Luft zieht. Zu diesem Fehlerbild kann er sich, falls während der Entwicklung bereits angelegt, Lösungsvorschläge direkt anzeigen lassen. Falls noch keine Beschreibung vorliegt, kann direkt eine Frage an alle an der Entwicklung der speziellen Komponente oder Baugruppe beteiligten Entwickler versendet und die Dokumentation der damit zusammenhängenden Entwicklungsschritte angeschaut werden. Dadurch wird eine Lösung und Bewertung des Fehlers beschleunigt und kann zusätzlich für die Erstellung von Fehlerauflistungen und die Aufstellung von Fehlerhäufigkeiten weitergenutzt werden.

3.4 Potenziale des Ansatzes

3.4.1 Umgang mit Entwicklern

Das Konzept der Smarten Werkbank sieht die Einbindung aller am Entwicklungsprozess beteiligten Fachbereiche vor. Damit geht die Möglichkeit einher, die Entwickler nach ihren Kompetenzen feingranularer einzuteilen. Dies kann projektspezifisch differieren, es kann sowohl ein Fachbereich von mehreren Mitarbeitern abgedeckt werden als auch ein Mitarbeiter mehrere Fachbereiche abdecken.

Um den Arbeitsablauf der Entwickler zu erleichtern, ist eine personenspezifische Sicht auf das Modell des Entwicklungsprozesses erforderlich. Dabei werden den Entwicklern nur diejenigen Entwicklungsschritte angezeigt, in denen ihre Expertise benötigt wird und welche dadurch für sie relevant sind.

3.4.2 Informationsverarbeitung

Anforderungserfassung

Zur Anforderungserfassung werden zwei Wege beschritten. Einerseits wird durch die Anbindung externer Werkzeuge die Möglichkeit geschaffen, die benötigten Anforderungen direkt aus Programmen zum Requirements Engineering auszulesen. Diese Werkzeuge können so gleichzeitig auch zur Spezifikation von Testfällen und zur Testdurchführung genutzt werden und die Qualitätssicherung unterstützen. Andererseits können neue Methoden (z. B. Fragenkataloge) in eigenen Werkzeugen integriert und so die Anforderungserfassung vereinfacht werden. Dies bietet die Möglichkeit, schneller eine vollständige Anforderungsliste zu erarbeiten und somit früher in den eigentlichen Entwicklungsprozess einzusteigen.

Änderungsverfolgung

Da innerhalb des Modells die Relationen zwischen Anforderungen, Entwicklungsschritten, Entwicklern und Lösungen definiert sind, können Änderungen in den Anforderungen, wie in Abbildung 7 dargestellt, leicht den beteiligten Entwicklungsschritten, Entwicklern und sich ergebenden Lösungen zugeordnet werden. Da heute Entwickler teils mehrere Projekte parallel bearbeiten, führt eine derartige Filterung zu einer deutlich erhöhten Arbeitseffizienz, da alle irrelevanten Benachrichtigungen für nicht beteiligte Entwickler entfallen.

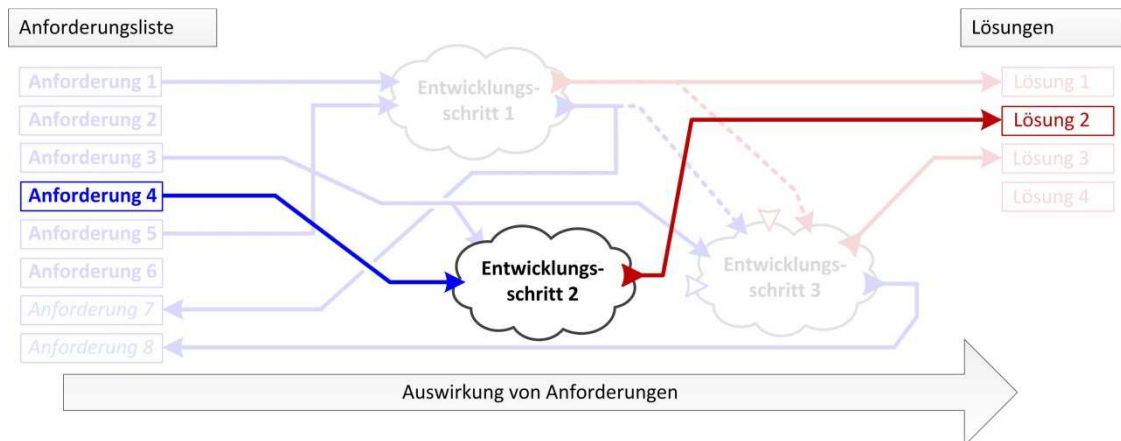


Abbildung 7: Auswirkung von Anforderungen

Werden Anforderungen geändert, zieht dies zumeist eine Steigerung von Projektdauer und -kosten nach sich. Diese sind aktuell jedoch schwierig zu bestimmen, da die Auswirkungen der Änderungen nicht umfassend ermittelt werden können. Mit dem beschriebenen Modell wird dies möglich, sodass den an der Anforderungsänderung beteiligten Personen (zumeist Kunde und Projektleiter) die Konsequenz der Änderung aufgezeigt wird.

Vernetzung von Entwicklungswerkzeugen

Durch die gezielte Anbindung etablierter Entwicklerwerkzeuge an eine gemeinsame, übergreifende Informationsübergabe-Plattform wird es möglich, die Werkzeuge miteinander zu vernetzen. Dadurch wird der Informationsaustausch verbessert und werden Fehler eliminiert. Durch Integration etablierter Werkzeuge werden die Inhalte der Entwickler in bekannten Entwicklungsumgebungen unterstützt und die Akzeptanz des Systems verbessert.

Umgang mit unterschiedlichen Beschreibungsmitteln und Inhalten

Im verfolgten Ansatz muss der Informationssender seine fachspezifischen Inhalte (manuell oder automatisiert) für den Informationsempfänger passend aufbereiten. Dadurch wird einerseits sichergestellt, dass alle erforderlichen Inhalte richtig übergeben werden.

Andererseits muss der Empfänger so nicht mehr ihm unbekannte Beschreibungsmittel lesen, wodurch die Fehleranfälligkeit gesenkt wird.

Manuelle Eingabe von und Redundanzen innerhalb der Inhalte

Durch die über alle Entwickler durchgängige Herangehensweise wird es möglich, Entwicklungsergebnisse aus vorgelagerten Entwicklungsschritten zu übernehmen und Inhalte lösungsorientiert zu ergänzen. Durch die Kenntnis vom und das Wissen über die Herkunft der Inhalte können Redundanzen und die doppelte Eingabe dieser Inhalte vermieden werden.

Fehlende Beschreibungsmöglichkeiten zur Eingabe spezieller Informationen

Mit der Smarten Werkbank werden speziell auch solche Inhalte adressiert, die heute nicht explizit in den etablierten Beschreibungsmitteln vorhanden sind. Dadurch ist eine Organisationsstruktur vorhanden, um prinzipiell beliebige Inhalte zu transportieren.

Explizite Erfüllung aller Anforderungen und Prüfung auf Vollständigkeit

Da die Zusammenhänge zwischen Anforderungen und Lösungen bekannt sind wird es möglich zu prüfen, ob alle gestellten Anforderungen erfüllt sind, alle Lösungen auf einer Anforderung basieren oder Entwicklungsschritte noch offene Eingangsgrößen besitzen, vgl. Abbildung 8. Damit lässt sich überprüfen, ob alle Anforderungen erfüllt und keine überzähligen Lösungen implementiert sind und ob alle Entwicklungsschritte mit aktuellen Daten arbeiten.

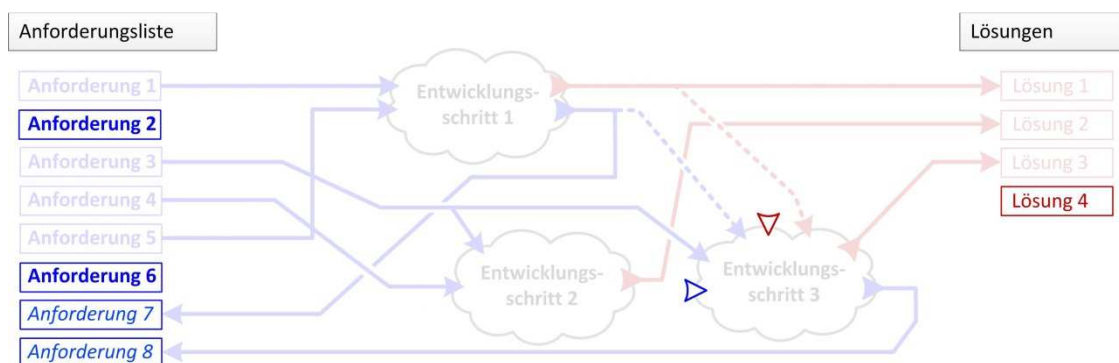


Abbildung 8: Unbearbeitete Anforderungen

Dokumentation des Entwicklungsverlaufes

Durch die Kenntnis der Verknüpfungen sowie des Vorhandenseins von Metadaten zu allen Modellelementen ist es möglich, die Lösungsentstehung zurückzuverfolgen (siehe auch Abbildung 9). Auf diesem Weg kann automatisch ein Protokoll über die Produktentstehung erstellt und in Form einer Maschinendokumentation festgehalten werden.

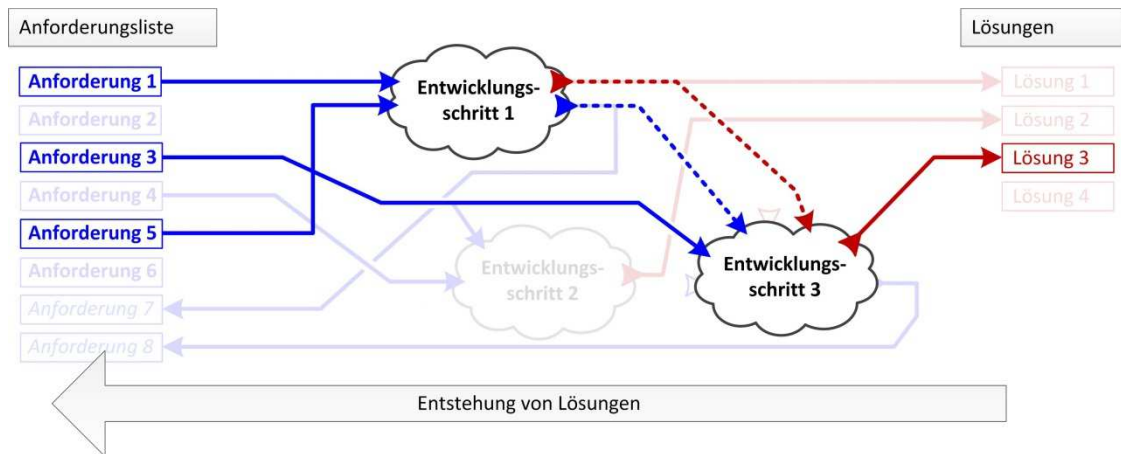


Abbildung 9: Rückverfolgbarkeit der Lösungsentstehung zur Dokumentation

4 Ausblick

In weiterführenden Arbeiten wird die beschriebene Herangehensweise zur Modellierung des Informationsflusses mit grafischen Hilfsmitteln in Mock-ups umgesetzt. Dies ermöglicht die Erprobung des im Rahmen dieser Arbeit beschriebenen, theoretischen Modells am Beispiel ausgewählter produktionstechnischer Systeme und Anwendungsszenarien, wie sie beispielhaft in Kapitel 3.3 skizziert wurden. Neben der Funktionalität zur Modellerstellung werden die Filterung der generierten Inhalte nach verschiedenen Kriterien sowie die Anforderungserfassung im Zentrum stehen. Dies ermöglicht die Prüfung der erarbeiteten Potentiale der Smarten Werkbank hinsichtlich der Senkung von Entwicklungszeit und der Fehlerreduktion. Darüber hinaus wird die Anbindung ausgewählter Entwicklerwerkzeuge entwickelt (z. B. CAD, Steuerungsentwicklung). Abschließend wird ein Framework zur Abarbeitung der modellierten Informationsflüsse prototypisch umgesetzt. Die beschriebenen Funktionen werden daran getestet.

Die Herangehensweise zur Entwicklung produktionstechnischer Systeme wird in den nächsten drei Jahren im Rahmen des Projekts „Smarte Werkbank“ ausgearbeitet und als Assistenzsystem in Form einer prototypischen Software umgesetzt. Der Entwicklungsfortschritt des open-source-Projekts wird unter www.smarte-werkbank.de dokumentiert.

5 Danksagung

Das beschriebene Projekt „Smarte Werkbank“ wird vom Europäischen Sozialfonds für Deutschland (ESF) sowie der Sächsischen AufbauBank (SAB) gefördert. Die Autoren danken dem ESF und der SAB für die großzügige Unterstützung der in der vorliegenden Publikation beschriebenen Arbeit. Des Weiteren danken wir Hrn. Sebastian Carsch (Fraunhofer IVV Dresden) für seine Unterstützung.

6 Literatur

- [1] BLEISCH, GÜNTER; ; GOLDHAHN, HORST ; SCHRICKER, GERHARD ; VOGT, HELMUT (Hrsg.): *Lexikon Verpackungstechnik*. 1. Aufl. Hamburg : Behr's, 2003
- [2] REIMUND NEUGEBAUER (Hrsg.): *Werkzeugmaschinen*. Berlin, Heidelberg : Springer, 2012
- [3] BRODTMANN, THILO ; ACKERMANN, ULRICH ; TRISCHLER, JOSEF ; VDMA ; MCKINSEY&COMPANY (Hrsg.): *Erfolgsmuster und Trends im deutschen Nahrungsmittel- und Verpackungsmaschinenbau - Handlungsansätze für mehr Wachstum und Profitabilität*, 2014
- [4] BRODTMANN, THILO ; ACKERMANN, ULRICH ; TRISCHLER, JOSEF ; OTHERS ; VDMA ; MCKINSEY & COMPANY (Hrsg.): *Zukunftsperspektive deutscher Maschinenbau - Erfolgreich in einem dynamischen Umfeld agieren*, 2014
- [5] DISPAN, JÜRGEN ; SCHWARZ-KOCHER, MARTIN ; IMU INSTITUT STUTTGART (Hrsg.): *Maschinen- und Anlagenbau in Deutschland - Entwicklungstrends und Herausforderungen. Eine Literaturstudie*. Stuttgart, 2014
- [6] HENSEL, THOMAS CHRISTIAN: *Modellbasierter Entwicklungsprozess für Automatisierungslösungen*, TU München, Dissertation, 2011
- [7] VOGEL-HEUSER, BIRGIT (Hrsg.): *Erhöhte Verfügbarkeit und transparente Produktion*. Kassel : kassel university press, 2011
- [8] BRECHER, CHRISTIAN (Hrsg.): *Integrative Produktionstechnik für Hochlohnländer*. Dordrecht, Heidelberg, London, New York : Springer, 2011
- [9] KOCH, VOLKMAR ; KUGE, SIMON ; GEISSBAUER, REINHARD ; SCHRAUF, STEFAN: *Industrie 4.0 - Chancen und Herausforderungen der vierten industriellen Revolution*, 2014
- [10] BULLINGER, HANS-JÖRG (Hrsg.): *Entwicklung und Erprobung innovativer Produkte - Rapid Prototyping*. Berlin, Heidelberg, New York : Springer
- [11] RICHTLINIE VDI 2206:2004: *Entwicklungsmethodik für mechatronische Systeme*. Berlin : Beuth, 2004
- [12] RICHTLINIE VDI 2221:1993: *Methodik zum Entwickeln und Konstruieren technischer Systeme und Produkte*. Berlin : Beuth, 1993
- [13] RICHTLINIE VDI 2222-1:1997: *Konstruktionsmethodik – Methodisches Entwickeln von Lösungsprinzipien*. Berlin : Beuth, 1997

- [14] FELDHUSEN, JÖRG ; GROTE, KARL-HEINRICH (Hrsg.): *Pahl/Beitz Konstruktionslehre: Grundlagen erfolgreicher Produktentwicklung, Methoden und Anwendung*. 8. Aufl. Berlin, Heidelberg : Springer Vieweg, 2013
- [15] PRITSCHOW, GÜNTER (Hrsg.): *Einführung in die Steuerungstechnik*. München, Wien : Hanser, 2006
- [16] HIRSCH-KREINSEN, HARTMUT: Entwicklung einer Basistechnik - NC-Steuerungen von Werkzeugmaschinen in den USA und der Bundesrepublik. In: DÜLL, KLAUS ; LUTZ, BURKART (Hrsg.): *Technikentwicklung und Arbeitsteilung im internationalen Vergleich*. Frankfurt, New York : Campus Verlag, 1989
- [17] EHRENSPIEL, KLAUS: *Integrierte Produktentwicklung - Denkabläufe, Methodeneinsatz, Zusammenarbeit*. München, Wien : Hanser, 2009
- [18] RICHTLINIE VDI 2519-1:2001: *Vorgehensweise bei der Erstellung von Lasten-/Pflichtenheften*. Berlin : Beuth, 2001
- [19] RICHTLINIE VDI 3260:1977: *Funktionsdiagramme von Arbeitsmaschinen und Fertigungsanlagen*. Berlin : Beuth Verlag, 1977. — Zurückgezogen im Dezember 1994
- [20] KÖNIG, RAINER ; QUÄCK, LOTHAR: *Petri-Netze in der Steuerungstechnik*. Berlin : Verlag Technik, 1988
- [21] MCHENRY, KENTON ; KOOPER, ROB ; MARINI, LUIGI ; ONDREJCEK, MICHAEL: The ISDA Tools: Preserving 3D Digital Content. In: *The Preservation of Complex Objects Symposia* (2011)
- [22] WIKIPEDIA: *Liste von CAD-Programmen*. URL https://de.wikipedia.org/wiki/Liste_von_CAD-Programmen. - abgerufen am 2018-01-03
- [23] TAY, B. H. ; ANANDA, A. L.: A Survey of Remote Procedure Calls. In: *SIGOPS Oper. Syst. Rev.* Bd. 24 (1990), Nr. 3, S. 68–79
- [24] NORM DIN EN ISO 12100: *Sicherheit von Maschinen – Allgemeine Gestaltungsleitsätze – Risikobeurteilung und Risikominderung*. Berlin : Beuth, 2011
- [25] NORM DIN EN ISO 13849-1: *Sicherheit von Maschinen – Sicherheitsbezogene Teile von Steuerungen – Teil 1: Allgemeine Gestaltungsleitsätze*. Berlin : Beuth, 2016
- [26] EG-MASCHINENRICHTLINIE 2006/42/EG: *Die neue EG-Maschinenrichtlinie 2006*. Berlin : Beuth, 2006

- [27] UHLIG, REINER: *SPS - Modellbasierter Steuerungsentwurf für die Praxis*. München : Oldenbourg, 2006
- [28] AUCOTEC AG: *Engineering Base - Essentials* : Aucotec AG, Hannover, 2015
- [29] AUCOTEC AG: *3D Integration Engineering Base to Solid Edge*. URL https://www.youtube.com/watch?v=gdhIH0lwWmA&list=PL71zX_9Jn1-ijnzvN8gJS8c0PkHSYIssZ. - abgerufen am 2017-06-02
- [30] SCHULZE, K. ; LISSE, R.: SAP Engineering Control Center with the new CIDEON interfaces. In: *sbn conference 21st/22nd October 2015*, 2015
- [31] HUGFARD, ANDREAS ; HECHT, HEIKO ; WALZ, WOLFGANG ; HENNERMANN, FRANK ; BROSCHE, GERALD ; MEHLICH, SABINE ; BÄTZ, CHRISTIAN: *Business Integration mit SAP-Lösungen*. Berlin, Heidelberg : Springer, 2004
- [32] VDMA: *Baukastenbasiertes Engineering mit Föederal - Ein Leitfaden für Maschinen- und Anlagenbauer* : Frankfurt a.M. : VDMA, 2004
- [33] VDMA (Hrsg.): *Aquimo - Adaptierbares Modellierungswerkzeug und Qualifizierungsprogramm für den Aufbau firmenspezifischer mechatronischer Engineeringprozesse* : Frankfurt a.M. : VDMA, 2010
- [34] EPLAN SOFTWARE & SERVICE GMBH & Co. KG: *EPLAN Engineering Configuration One: Einstieg ins mechatronische Engineering*. URL <https://www.eplan.de/de/loesungen/elektrotechnik/eplan-engineering-configuration-one/>. - abgerufen am 2017-06-08
- [35] SIEMENS AG: *SIMATIC Automation Designer based on Comos increases all potentials of Digital Engineering for both time and cost savings*. Order No: E20001-A300-P230-X-7600. Aufl., 2010
- [36] WSCAD GMBH: *WSCAD Suite - Electrical Engineering*. URL <http://www.wscad.com/suite/>. - abgerufen am 2017-12-25
- [37] WSCAD GMBH: *WSCAD Suite - ElektroHandwerk*. URL <http://www.wscad.com/elektrohandwerk/>. - abgerufen am 2017-12-25
- [38] *DFG GEPRIS - Ergebnisse des SFB 614: Selbstoptimierende Systeme des Maschinenbaus*. URL <http://gepris.dfg.de/gepris/projekt/5485260/ergebnisse>. - abgerufen am 2018-01-12

- [39] GAUSEMEIER, JÜRGEN ; DUMITRESCU, ROMAN ; STEFFEN, DANIEL ; CZAJA, ANJA ; WIEDERKEHR, OLGA ; TSCHIRNER, CHRISTIAN ; HNI PADERBORN ; FRAUNHOFER IPT ; UNITY (Hrsg.): *Systems Engineering in der industriellen Praxis*. Paderborn, 2013
- [40] *Projektseite SFB 768*. URL <https://www.sfb768.tum.de/>. - abgerufen am 2018-01-12
- [41] WOLFGANG BAUER ; NEPOMUK CHUCHOLOWSKI ; FATOS ELEZ ; MAIK MAURER ; UDO LINDEMANN (Hrsg.): *Zyklusorientiertes Modul- und Plattformdenken - Ein Leitfaden für Praktiker*. München, 2014
- [42] DRATH, RAINER: *Datenaustausch in der Anlagenplanung mit AutomationML*. Heidelberg, Dordrecht, London, New York : Springer, 2010
- [43] BIRGIT VOGEL-HEUSER ; THOMAS BAUERNHANSL ; MICHAEL TEN HOMPEL (Hrsg.): *Handbuch Industrie 4.0 Bd. 2 - Automatisierung*. Berlin : Springer, 2017
- [44] *AutomationML tools*. URL <https://www.automationml.org/o.red.c/tools.html>. - abgerufen am 2018-01-16