

Schnoor, Florian; Ahrens, Volker

Working Paper

SysML als Bindeglied zwischen Wirtschafts- und Ingenieurwissenschaften

Arbeitspapiere der Nordakademie, No. 2017-02

Provided in Cooperation with:

Nordakademie - Hochschule der Wirtschaft, Elmshorn

Suggested Citation: Schnoor, Florian; Ahrens, Volker (2017) : SysML als Bindeglied zwischen Wirtschafts- und Ingenieurwissenschaften, Arbeitspapiere der Nordakademie, No. 2017-02, Nordakademie - Hochschule der Wirtschaft, Elmshorn

This Version is available at:

<https://hdl.handle.net/10419/157362>

Standard-Nutzungsbedingungen:

Die Dokumente auf EconStor dürfen zu eigenen wissenschaftlichen Zwecken und zum Privatgebrauch gespeichert und kopiert werden.

Sie dürfen die Dokumente nicht für öffentliche oder kommerzielle Zwecke vervielfältigen, öffentlich ausstellen, öffentlich zugänglich machen, vertreiben oder anderweitig nutzen.

Sofern die Verfasser die Dokumente unter Open-Content-Lizenzen (insbesondere CC-Lizenzen) zur Verfügung gestellt haben sollten, gelten abweichend von diesen Nutzungsbedingungen die in der dort genannten Lizenz gewährten Nutzungsrechte.

Terms of use:

Documents in EconStor may be saved and copied for your personal and scholarly purposes.

You are not to copy documents for public or commercial purposes, to exhibit the documents publicly, to make them publicly available on the internet, or to distribute or otherwise use the documents in public.

If the documents have been made available under an Open Content Licence (especially Creative Commons Licences), you may exercise further usage rights as specified in the indicated licence.



ARBEITSPAPIERE DER NORDAKADEMIE

ISSN 1860-0360

Nr. 2017-02

**SysML als Bindeglied zwischen
Wirtschafts- und Ingenieurwissenschaften**

Florian Schnoor, Prof. Dr.-Ing. Volker Ahrens

März 2017

Eine elektronische Version dieses Arbeitspapiers ist verfügbar unter:
<https://www.nordakademie.de/die-nordakademie/forschung/publikationen/>



Köllner Chaussee 11
25337 Elmshorn
www.nordakademie.de

SysML als Bindeglied zwischen Wirtschafts- und Ingenieurwissenschaften

Florian Schnoor, Volker Ahrens

NORDAKADEMIE – Hochschule der Wirtschaft, Elmshorn

Die Verknüpfung der einzelnen Disziplinen des Wirtschaftsingenieurwesens wird anhand des Konfigurationsmanagements im Anlagenbau untersucht. Dafür wird vor dem Hintergrund der Modellierung von Systemen im Sinne des Systems Engineering die Methode des Konfigurationsmanagements in die Systems Modeling Language (SysML) überführt und auf ihre Fähigkeiten analysiert. Auf Basis der SysML-Fähigkeiten und der normierten Anforderungen des Konfigurationsmanagements wird ein Lastenheft erarbeitet und kann so ein Stereotyp für SysML definieren. Die Betrachtung zeigt deutlich, wie die SysML ein Bindeglied der verschiedenen Disziplinen darstellen kann.

1 Wirtschaftsingenieurwesen als Verbindung zwischen Betriebswirtschaft und Ingenieurwissenschaften

Mit den immer tiefer greifenden Integrationen von Systemen wird die erfolgreiche Verbindung von technischen Fähigkeiten und ihren wirtschaftlichen Pendanten ein integraler Bestandteil erfolgreicher Projektentwicklungen im Anlagenbau. Moderne Unternehmen bedienen sich in diesem Zusammenhang des Konfigurationsmanagements (KM). Diese Managementmethode dient unter anderem zur Erfüllung verschiedener Qualitätsanforderungen [1]. In diesem Bereich erhebt das Systems Engineering einen zumindest verwandten Anspruch. Die erfolgreiche Kombination dieser Methoden wird nachfolgend betrachtet. Dafür werden zunächst die Einzeldisziplinen für diesen Anwendungsfall charakterisiert. Die hieraus abgeleitete Hypothese wird im Anschluss vor dem Hintergrund der SysML-Fähigkeiten und der Anforderungen des KM geprüft.

1.1 Konfigurationsmanagement als Managementdisziplin

Die DIN ISO 10007 ist Teil der Qualitätsmanagementnormen. Das KM ist dabei ein mögliches Verfahren, um die Anforderungen der ISO 9001 bezüglich der Rückverfolgbarkeit eines Produktes zu erfüllen. Sie beschreibt ferner die Anwendung des KM über den gesamten Lebenszyklus von Produkten. Dabei ist sie lediglich ein Leitfaden und damit nicht zur Zertifizierung von Organisationen vorgesehen [1].

KM beinhaltet die notwendigen Tätigkeiten, um die Lenkung und Leitung der Konfiguration eines betreffenden Produktes über den Lebenszyklus hinweg sicherzustellen. Dafür sind die Konfigurationsangaben in Bezug auf einzelne Konfigurationseinheiten (KE) maßgebend. Die Summe der Konfigurationsangaben zu einer Einheit bildet eine Konfiguration. Wird diese zu einem bestimmten Zeitpunkt fixiert, spricht man von einer Bezugskonfiguration (auch als Baseline bezeichnet). Diese kann im weiteren Verlauf des Produktlebens zu verschiedenen Zwecken herangezogen werden. Änderungen einer Konfiguration bedürfen zum einen einer strukturierten Freigabe durch eine entsprechend befähigte Organisationseinheit (Verfügungsstelle) und zum anderen einer strukturierten Dokumentation mit Hilfe von Änderungsanträgen (ÄA) und einer Buchführung [1].

Das KM hilft damit immer komplexer werdende Produkte erfassen und über verschiedene Lebensphasen steuern zu können.

1.2 Systems Engineering als Disziplin der Ingenieurwissenschaften

Mit den ersten formalisierten Anwendungen des SE in den Bell Laboratories zu Beginn des 20. Jahrhunderts wurden zunächst vor allem technisch orientierte Projekte begleitet [2]. Aus den Ursprüngen in den 1930er Jahren haben sich zunächst militärische Standards (bspw. MIL-STD-499) gebildet. Im weiteren Verlauf der Zeit hat sich zuletzt die ISO 15288 als maßgebende Norm für das SE durchgesetzt [3]. Diese wurde kürzlich aktualisiert und ist als ISO 15288:2015 auf neuestem Stand [4].

Das SE verwendet zur Lösung von Problemen einen generischen Problemlösungszyklus. Dafür werden Systeme hinreichend erfasst, die Differenz zwischen Ist- und Soll-Zustand methodisch beschrieben und mit verschiedenen Lösungsansätzen gelöst. Dabei wird ein besonderes Augenmerk auf die Erfassung und Erfüllung von diversen Randbedingungen gelegt [5]. In diesem Zusammenhang werden neben den oben genannten Anforderungen auch die statische Systemarchitektur und das dynamische Systemverhalten erfasst. Vor allem dieses Verhalten generiert in einer formalen Sprache diverse Möglichkeiten zur Arbeitserleichterung und Qualitätssicherung in großen Projekten [6].

2 SysML als Bindeglied der Disziplinen?

Aus den oben dargestellten Rahmenbedingungen wird die nachfolgende Hypothese abgeleitet: Das Verhalten komplexer Produkte kann mit formalen Sprachen entwickelt und dokumentiert werden. Die Konfigurationserfassung im Sinne des KM ist dabei noch nicht Bestandteil dieser Sprachen.

Methodisch wird die Lösungsfindung hierbei durch ein Lastenheft geführt. Es wird ein Lastenheft für eine etwaige Erweiterung von Standards, die zur formalisierten Beschreibung von Systemen verwendet werden, erarbeitet.

Zur formalen Beschreibung eines Systems im Kontext des SE werden diverse Sprachen verwendet. Diese sind beispielsweise:

- Systems Modeling Language (SysML)
- Petri-Netze
- Specification and Description Language [6]

Die Methoden eignen sich jeweils für unterschiedliche Anwendungszwecke und sind damit im gegebenen Fall auszuwählen [6]. Einige der genannten Sprachen wie beispielsweise die *Specification and Description Language* oder *Petri-Netze* sind jeweils sehr spezifisch geprägt. Sie wurden in verschiedenen Domänen entwickelt und haben dort ein hohes Maß an Integration in die Bedürfnisse der jeweiligen Domäne [6].

Um an dieser Stelle nicht nur eine fachliche Domäne betrachten zu können und einen universellen Ansatz zu verfolgen, werden lediglich die generischen Sprachen betrachtet. Die nachfolgende Betrachtung bezieht sich daher auf die SysML und damit auch die in Teilen enthaltene Unified Modeling Language (UML).

Es werden nun die Eckpunkte der Systembeschreibung in SysML dargestellt (Kapitel 3).¹ Im Anschluss werden die Bedarfe des KM an die Beschreibung einer Konfiguration und einer KE zusammengefasst. Aus diesem Delta zwischen SysML-Fähigkeiten und den KM-Anforderungen, kann dann das Lastenheft abgeleitet werden (Kapitel 4), sodass ein Lösungsansatz ausgearbeitet (Kapitel 5) und bewertet (Kapitel 6) werden kann.

3 Systembeschreibung in SysML

Die formalisierte Beschreibung von Systemen mit der Hilfe von SysML basiert auf der Modellierungssprache UML. In der Entstehung von SysML wurden dabei viele Aspekte und Prinzipien aus der UML übernommen. Für den Anwendungsfall der Systemmodellierung wurden einige neue Aspekte hinzugefügt und einige Bestandteile der UML weggelassen. SysML verfolgt bei der Modellierung von Systemen einige Grundsätze, die für die nachfolgende Lösungssynthese erhalten bleiben müssen und sollen:

- Das Design wird durch die Anforderungen an das System gesteuert.
- SysML setzt auf UML auf und passt die UML nur dann an, wenn es das System erfordert.
- Wenn UML durch SysML erweitert werden muss, werden die vorgesehen Profile dafür genutzt.
- SysML-Pakete sind eine Erweiterungsschicht auf dem UML-Metamodell.
- Der XMI-Austauschstandard der UML wird in SysML übernommen [7].

Die Erweiterung der UML für die Verwendung von SysML wird nachfolgend näher erläutert.

3.1 Diagramm vs. Modell

Die interdisziplinäre Arbeit an einem Systemmodell auf Basis von SysML erfordert, dass die jeweiligen Domänen passende Ansichten des zugrundeliegenden Modells zur Verfügung haben. Dafür werden in SysML verschiedene Diagrammart definiert, wie sie nachfolgend in Abbildung 1 dargestellt werden. Auf einige Diagrammart wird im weiteren Verlauf noch detaillierter eingegangen.

In den Diagrammen werden die Elemente des Modells instanziiert. Das heißt, dass ein Modellelement in einer beliebigen Anzahl von Diagrammen verwendet werden kann und dabei nicht den Bezug zum Quellelement im Modell verliert. Damit werden Änderungen am Systemelement in alle referenzierte Diagramme umgehend eingebracht.

Die Verknüpfung erlaubt es außerdem, die Darstellung der Instanz eines Elements in einem Diagramm anzupassen, ohne das Element selbst zu verändern. Konkret können beispielsweise Darstellungsformen in Form von Bildern, Symbolen oder Textblöcken verwendet werden. Die benötigte äußere Form ergibt sich dabei aus dem jeweiligen Diagramm und dem Anwendungsfall [8]. Je Diagrammart spezifiziert SysML die Art der designierten Systemelemente. Allerdings wird kein Verbot der übrigen Systemelemente gegeben [7].

¹ Die Beschreibung von SysML umfasst dabei nicht den gesamten SysML-Spezifikationsumfang, sondern lediglich die für die weitere Lösungsfindung benötigten Elemente.

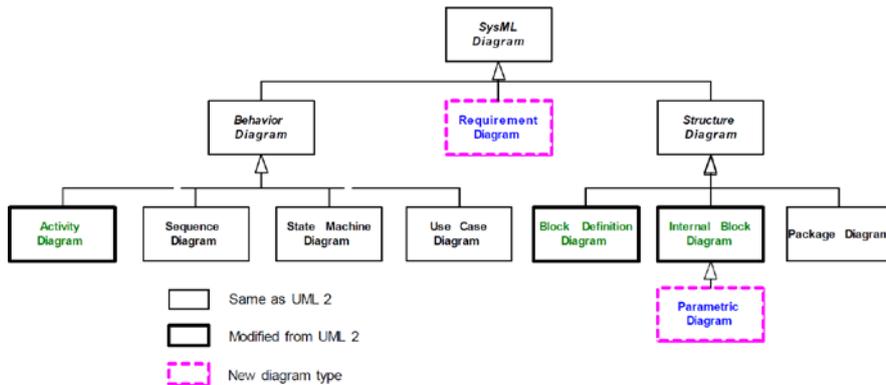


Abbildung 1: Diagrammtypen in SysML 1.4 [7]

Abschließend wird der Zusammenhang von Diagramm und Modell nachfolgend in Abbildung 2 dargestellt. Die Trennung von Modell und Diagramm bedingt im operativen Umgang, dass im Modell durchaus mehr Elemente enthalten sein können, als in der Summe der Sichten enthalten sind [6].

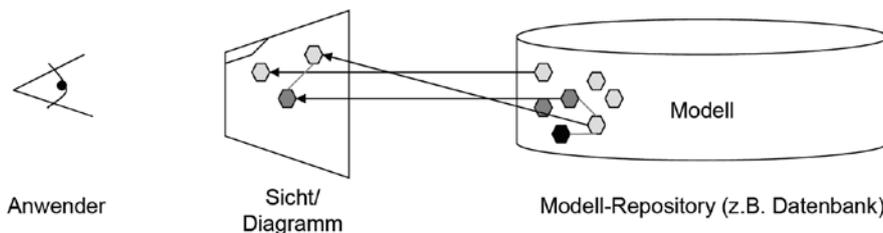


Abbildung 2: Zusammenhang von Diagramm und Modell [6]

3.2 Profilmechanismus

Seitens der Object Management Group (OMG) wird mit dem Profilmechanismus ein Verfahren beschrieben, das die Möglichkeit bietet, die standardisierte UML zu erweitern [6]. Durch dieses Verfahren im UML-Standard werden die Fähigkeiten der UML sehr umfassend erweiterbar. Der Anwendungsbereich der UML ist somit nicht nur der konkret definierte Teil der OMG-Spezifikation, sondern er umfasst darüber hinaus alle weiteren Anwendungsfälle, die sich über Erweiterungen des Anwenders realisieren lassen.

Um Anpassungen zu realisieren, werden Elemente des UML-Metamodells mit Hilfe von *Stereotypen* erweitert. Damit kann einem Element beispielsweise ein weiteres Attribut als Eigenschaft zugewiesen oder seine äußere Form modifiziert werden [9]. Auf dieser Basis lassen sich vielfältige Anpassungen realisieren. Beispielsweise kann eine UML-Erweiterung erzeugt werden, um Petri-Netze abbilden zu können [6].

Mit diesem Verfahren ist auch die SysML als Profil der UML spezifiziert worden [7]. Die Stereotypen innerhalb des SysML-Profiles werden nachfolgend in Abbildung 3 dargestellt.

Mit den Anpassungen der Sprache für einen spezifischen Zweck wird die Standardisierung demnach nicht verlassen. Die OMG ermöglicht es somit, die Sprache zu erweitern und trotzdem

noch standardisierte Werkzeuge zu nutzen [6]. Infolgedessen müssen UML-Werkzeuge mit dem Profilmechanismus arbeiten können.

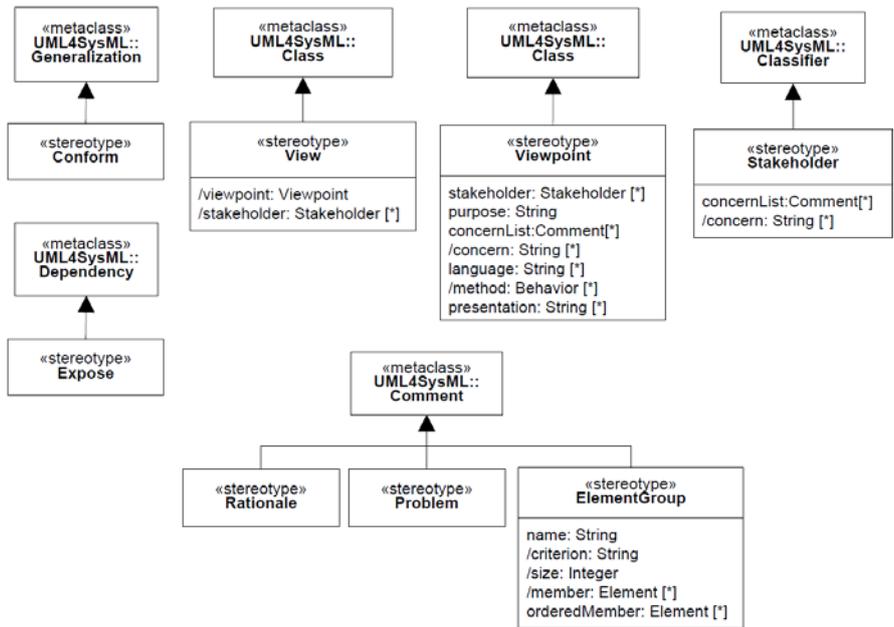


Abbildung 3: Stereotypen in SysML zur Erweiterung von UML [7]

3.3 Views und Viewpoints

Um neben den Anforderungen der System Engineers auch die Informationsbedürfnisse der weiteren Stakeholder zu erfüllen, bietet SysML das Konzept der *Views* und *Viewpoints*. Dieses Konzept ist dabei konsistent zur ISO/IEC/IEEE 42010 [10]. In einem Viewpoint wird beschrieben, welche Informationen des Systemmodells für den Stakeholder relevant sind, der View setzt diese Anforderungen dann um. Je nach Anforderungen werden spezielle Informationen ausgeblendet oder beispielsweise in einer speziellen, äußeren Form dargestellt. Aus diesen Kriterien entsteht dann eine Sicht auf das Modell (View). Der entstehende View zeigt also eine besondere Sicht auf das Systemmodell. Dieses Verhalten lässt sich mit den oben genannten Diagrammen vergleichen. Daher wird der mittlere Teil in Abbildung 2 auch mit „Sicht/Diagramm“ beschrieben.

Im Viewpoint wird außerdem hinterlegt, in welcher Form der View aus dem Modell extrahiert wird. Um die Bedürfnisse der zugeordneten Stakeholder zu erfüllen, ist es zudem möglich, auf mehr als ein Modell zur View-Erzeugung zuzugreifen [7]. Der View selbst ist ein Modellelement. Für den Stakeholder wird, wie zuvor bereits erwähnt, ein Dokument erzeugt, das den View entsprechend abbildet. Dieses erzeugte Dokument ist nicht per se Teil des Modells, sondern wird lediglich durch den View repräsentiert [7].

Während der Viewpoint mit jedem Aufruf einen aktuellen Zustand (unter Anwendung der Anforderungen) zeigt, wird mit dem extrahierten Artefakt ein Schnappschuss erstellt. Klassische Beispiele dieser Artefakte sind PDF-Dokumente oder Foliensätze, die die entsprechenden Daten enthalten [7]. Im Gegensatz zu den oben beschriebenen Diagrammen sind die Inhalte der

Viewpoints nicht fix beschrieben, sondern dienen der konkreten Ausprägung im spezifischen Einzelfall, um damit den Anforderungen eines Stakeholders und seiner spezifischen Informationsaufbereitung zu dienen [6].

Obwohl das generelle Konstrukt beschrieben wird, liegt auch mit der SysML 1.4 die finale Implementierung der View-Erzeugung außerhalb der Spezifikation. In der Vergangenheit haben die verfügbaren Modellierungswerkzeuge ihre jeweils eigenen Umsetzungen gehabt [11].

3.4 Blöcke und Schnittstellen

Die strukturelle Systembeschreibung wird in SysML durch *block*-Elemente erzeugt. Blöcke werden als Erweiterung der UML-*Class* spezifiziert. Dabei werden die Definitionen der UML für SysML erweitert. Allerdings wurden auch einige Aspekte, wie beispielsweise sehr spezielle Beziehungstypen zu weiteren Elementen, entfernt. Diese Aspekte werden ausgeschlossen, um die SysML einfach zu halten [7].

Jeder Block für sich beschreibt eine Sammlung an Eigenschaften des zu spezifizierenden Systems. Damit können neben Hard- und Software auch Verhaltensmuster spezifiziert werden [7]. In der Systemmodellierung werden zuvor definierte Blöcke in den entsprechenden Diagrammen instanziiert (diese heißen dann *properties*). Die Definition der Blöcke erfolgt dabei im *block definition diagram*. Sie werden außerdem in einer gemeinsamen Bibliothek zur übergreifenden Nutzung bereitgestellt [6].

Der Block hat in SysML lediglich ein Attribut. Der boolesche Wert *isEncapsulated* gibt an, ob der Block als Blackbox betrachtet werden soll. Wird dieses Attribut mit WAHR beschrieben, werden weitere Details des Blocks nicht beschrieben [7]. Diese Blackbox kann beispielsweise verwendet werden, wenn die weitere Spezifizierung des Systems nicht relevant ist, da sie an Dritte vergeben wird oder das System *off the shelf* verfügbar ist.

Darüber hinaus wird bereits in der SysML-Spezifikation eine Vielzahl von Stereotypen für die Blöcke angegeben [7]. Da sie an dieser Stelle nicht von weiterer Relevanz sind, wird auf die Details nicht eingegangen.

Das Prinzip der Instanziierung von Blöcken findet analog auch Anwendung für die Schnittstellen in SysML [6]. Die Schnittstellen werden als sogenannte *ports* definiert. Dabei können an diesen Schnittstellen verschiedenste Objekte ausgetauscht werden. Über unterschiedliche Arten von Ports können neben physikalischen Größen auch komplexe und Informationen ausgetauscht werden [7]. Die Verschachtelungen von *Flow Specifications* machen diese komplexen Übertragungen möglich [6]. Für die weitere Unterstützung in der praktischen Anwendung werden in SysML SI-Einheiten implementiert. Dies stellt eine Erweiterung zu UML dar und wurde initial mit der SysML eingeführt [6].

Die Definition der generellen Fähigkeiten erfolgt also über Blockdefinition. Im Anschluss findet dann die spezifische Ausprägung über die Instanziierung der Blöcke statt. Dabei werden dann auch spezifische Größen eingepflegt, die zuvor in der Schnittstellendefinition angelegt wurden.

3.5 Beziehungen

In der SysML können verschiedene Systemelemente zueinander in Beziehung gesetzt werden. Es können sowohl Beziehungen von Elementen gleichen Typs (bspw. zwischen Block-Elementen) als auch Beziehungen zwischen Elementen verschiedenen Typs erzeugt werden. Damit können die „unterschiedlich abstrakten Blickwinkel“ auf das Systemmodell verbunden werden [6]. So lässt sich beispielsweise darstellen, welche Anforderung mit welchem Testfall geprüft

wird oder welche Komponente welche Anforderungen zu erfüllen hat. Für diese Beziehungen definiert SysML den Typ *allocate*. Mit dieser Beziehung wird von einem jeweils abstrakteren auf ein passendes, konkreteres Objekt referenziert [6]. Aus dieser Grundregel lassen sich bereits vielfältige Informationen abbilden.

Weitere Beziehungen sind jeweils innerhalb der einzelnen Elementtypen definiert. So werden beispielsweise für die oben beschriebenen Blöcke verschiedene Formen der *association* definiert [7]. Damit kann dann der strukturelle Aspekt des SE zwischen den Blöcken modelliert werden.

3.6 Dokumente im Umfeld von SysML

Sofern es für das Systemmodell notwendig ist, dass externe Dokumente im Modell **angezogen** werden müssen, steht der Stereotyp *document* für diesen Anwendungsfall zur Verfügung. Über eine *trace*-Beziehung kann ein Dokument dann auf ein beliebiges Modellelement bezogen werden. Über die *properties* können dem Artefakt weitere Eigenschaften zugewiesen werden. Das Stereotyp ist aus der UML2 übernommen [7]. Aus der dortigen Spezifikation geht außerdem hervor, dass ein Artefakt mit unterschiedliche Eigenschaften mehrfach instanziiert werden kann [9].

Kommentiert [AH1]: Allgemeinsprachlich würde das Verb hier nicht passen, wie sieht es fachsprachlich wirklich aus? Oder könnte es auch „herangezogen“ heißen?

Kommentiert [VA2R1]: Fachsprachlich korrekt.

3.7 Erfassen von Konfigurationen in SysML

Innerhalb der SysML-Spezifikation wird ein Beispiel für das Erfassen von Konfigurationen gegeben. Dabei wird ein PKW in einige Komponenten aufgegliedert. Der eindeutigen Fahrzeugidentifikationsnummer werden dabei die Seriennummern der verbauten Komponenten zugeordnet. Um die Konfiguration zu erfassen gibt es die Möglichkeit, einen *context block* zu erzeugen. Innerhalb dieses Blocks werden dann die Merkmale der spezifischen Konfiguration angegeben. Die Merkmalsausprägungen müssen innerhalb der zuvor festgelegten Definitionsbereiche liegen. Im gegebenen Beispiel kann es beispielsweise nur einen Motor geben, der jedoch je nach Konfiguration zwischen vier und acht Zylindern enthalten kann [7].

4 Anforderung an die Beschreibung einer Konfiguration und einer KE

Je nach angewandter Norm werden etwas unterschiedliche Definitionen der Konfiguration bzw. auch der KE angeführt. Für die Betrachtung der Fähigkeiten im Zuge der Systembeschreibung mit SysML werden daher an dieser Stelle die signifikanten Eckpunkte der KM-Anforderungen betrachtet:

- Bestimmen der KE und der gesamten Konfiguration inkl. der Hierarchie
- Eindeutige Merkmale zur Identifizierung und zum Status der KE
- Erfassen der zugehörigen Dokumente von Konfiguration und KE
- Kontinuierliche Verfügbarkeit der aktuellen Konfiguration (Baseline nach *Configuration Management II*)
- Meilensteinbezogene Speicherung von Konfigurationen (Baseline nach [1])
- Freigabeverfahren für Änderungen

Aus den oben ausgearbeiteten Merkmalen der Systemmodellierung in SysML und der zusammengefassten Anforderungen des KM wird nun ein Abgleich durchgeführt. Dafür werden in Tabelle 1 die gelisteten Anforderungen der SysML-Spezifikation gegenübergestellt.

Anforderung	Enthalten in SysML?	Erläuterung
Umfang der Konfiguration	Ja	Keine Vorgabe zur Tiefe der Modellierung. Prinzipiell ist eine hinreichend detaillierte Modellierung möglich.
Hierarchie der Systemelemente	Teilweise	Es werden Beziehungstypen definiert, die hierarchische Strukturen ermöglichen. Die Anwendung auf KE ist zunächst nicht vorgesehen.
Identifizierung	Nein	Eine systematisierte Identifizierung (ID o. ä.) ist nicht vorgesehen. Allerdings verfügen die Werkzeuge in der Praxis in Ansätzen über solche Mechanismen.
Dokumentensteuerung	Teilweise	Dokumente können über den <i>document</i> Stereotyp abgebildet werden. Eine umfangreiche Steuerung ist nicht vorgesehen.
Dokumentengenerierung	Teilweise	Prinzipiell sind Dokumentengenerierungen möglich. Allerdings ist fraglich, ob KM-relevante Inhalte hinreichend abbildbar sind.
CMII-Baselining	Teilweise	Aus den Instanzen der Elemente in freigegebenen Diagrammen kann die aktuelle Konfiguration ermittelt werden. Praktische Anwendung nicht spezifiziert.
DIN ISO 10007-Baselining	Teilweise	Über ein kontextbezogenes Blockdiagramm kann eine Baseline festgehalten werden. Die Inhalte sind ggf. nicht hinreichend im Modell enthalten.
Freigabe- und Änderungsverfahren	Nein	Status und Sperrung von Änderungen sind für die Modellelemente nicht explizit vorgesehen.

Tabelle 1: Abgleich von KM-Anforderungen und der SysML-Spezifikation

Die Versionierung von SysML-Modellen stellt sich praktisch als sehr schwierig dar. Zwar kann ein SysML-Modell (genau wie ein UML-Modell) abgespeichert werden, doch sind damit einige Probleme verknüpft [12].

Die OMG definiert für den Austausch von UML-Modellen die *XML Metadata Interchange (XMI)*. Damit können die Modelle gespeichert und entsprechend adressiert werden [13]. Mit diesem Mechanismus kann allerdings nur ein gesamtes Modell betrachtet werden [12]. Im Hinblick auf den Anlagenbau mit seinen großen Teams und der parallelen Bearbeitung vieler Themengebiete erscheint diese Lösung daher als wenig tragbar. Die Versionierung müsste vielmehr auf Subsystem- oder sogar Blockebene geschehen.

Im Zuge der Versionierung ist auch die Änderung der Systemelemente nicht hinreichend spezifiziert. Definierte Änderungsprozesse, wie sie beispielsweise mit einem ECP gesteuert werden können, werden im Anlagenbau benötigt [14]. Diese sind in SysML allerdings nicht vorhanden.

Für die Erweiterung/Anpassung der SysML werden nachfolgend Lösungen erarbeitet, die neben der Versionierung auch die Änderungsverfahren von SysML-Modellen betrachten.

5 Befähigung der SysML für das Konfigurationsmanagement

Das erarbeitete Delta zwischen normierten SysML-Fähigkeiten und KM-Anforderungen wird nun in mögliche Lösungen überführt.

5.1 Abbildung von Konfigurationseinheiten auf Basis von SysML-Profilen und Viewpoints

Die hinreichende Darstellung von Systemelementen für das KM kann potenziell über den Profilmehanismus realisiert werden. Die Elemente müssen als eindeutige KE bzw. als Konfiguration identifizierbar werden. Um die dafür notwendigen Eigenschaften zu etablieren, wird eine Erweiterung über den Profilmehanismus erzeugt. Das dafür notwendige Stereotyp definiert dann folgende, zusätzliche Eigenschaften:

- ID der KE/Konfiguration
- Liste der übergeordneten ID (0..n)
- Liste der untergeordneten ID (0..n)
- Status (in Erstellung, Freigegeben, in Änderung, Zurückgezogen)

Über einen entsprechenden Viewpoint werden die Informationen im Anschluss für den Stakeholder KM aufbereitet und in einem entsprechenden Dokument (als Artefakt) zur Verfügung gestellt. Nachfolgende Abbildung 3 zeigt eine schematische Darstellung dieser Lösung.

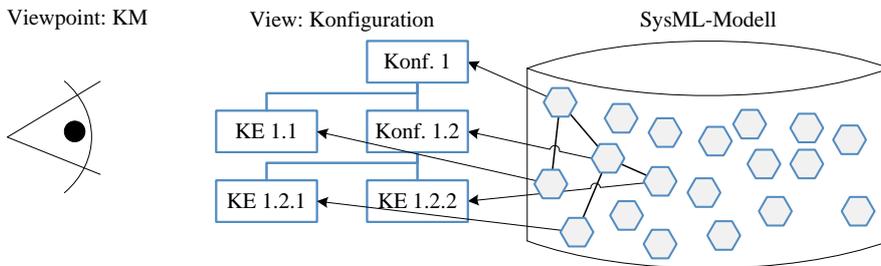


Abbildung 4: KM als Viewpoint eines SysML-Modells (eigene Abbildung in Anlehnung an [6])

Zur Selektion der relevanten Informationen wird als Selektionskriterium das oben genannte Stereotyp angezogen. Alle Elemente, die mit dem Stereotypen erweitert sind, werden selektiert. Inwiefern Elemente nach ihrem Status ausgefiltert werden sollen, hängt vom jeweiligen Anwendungsfall ab. Für diese Lösung werden die Elemente ungeachtet ihres Status in die Selektion aufgenommen.

Wie in Kapitel 3 beschrieben, werden bereits im SysML-Standard verschiedene Beziehungen definiert. So kann die Allokation eine Hierarchie der Elemente, wie sie auch im KM benötigt werden, darstellen. Es können sogar, wie im KM benötigt, mehrere übergeordnete Elemente abgebildet werden, sodass die Fähigkeit der Allokation zunächst hinreichend erscheint.

Allerdings ist zu fraglich, ob die Hierarchie der Modellstruktur auch der benötigten KM-Struktur entspricht. Aus diesem Grund werden im Stereotypen die über- bzw. untergeordneten IDs spezifiziert. Zeigt sich in der praktischen Anwendung, dass beide Hierarchien gleich sind, so kann natürlich auf eine redundante und damit potenziell fehlerträchtige Datenerfassung verzichtet werden. Dies ist einheitlich je Anwendungsfall zu bestimmen. Um an dieser Stelle aber ein Stereotyp zu definieren, das für beide Fälle adäquat ist, werden die IDs aufgenommen.

Nachfolgende Abbildung 5 enthält die abstrakte Syntax des Stereotypen *ConfigurationItem*. Zur besseren Einordnung wird neben dem direkten Metaelement („block“) auch dessen Ursprung in UML („class“) mit aufgenommen.

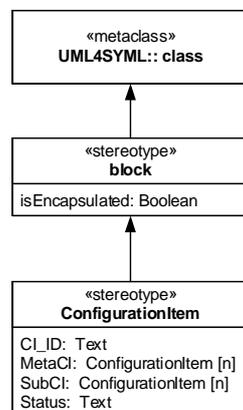


Abbildung 5: Stereotyp zur Abbildung von KE in SysML (eigene Abbildung, Bestandteile aus [7])

Die benötigten Attribute werden im Stereotypen definiert. Dabei wird die ID der KE (*CI_ID*) als Text definiert, um ggf. auch alphanumerische Identifizierer zu ermöglichen. Die Attribute für die über- bzw. untergeordnete KE (*MetaCI* und *SubCI*) werden als Typ des Stereotypen *ConfigurationItem* definiert. Damit können in diesen Attributen Elemente angezogen werden, die mit dem Stereotypen erweitert sind. Entsprechend wird umgekehrt sichergestellt, dass keine Blöcke ohne die KE-Erweiterung aufgenommen werden können. Der Status der KE wird über das Attribut *Status* in Textform festgehalten.

5.2 Ableiten von konfigurationsmanagementrelevanten Dokumenten

Aus der zuvor dargestellten Erweiterung der SysML und den bereits in SysML enthaltenen Fähigkeiten ist in dieser Lösung zu prüfen, ob auch KM-relevante Dokumente abgeleitet werden können.

Bereits berücksichtigt wurden oben die Dokumentationen der Baselines des jeweils betrachteten und modellierten Systems. Nachfolgend werden zunächst potenzielle KM-relevante Dokumente identifiziert und im Anschluss eine adäquate Umsetzung in einem SysML-Modell untersucht.

Die 1 beschreibt eine Reihe von typischen Berichten, die im Rahmen des KM benötigt werden. Nachfolgend werden diese Beispiele kurz aufgeführt:

- Liste der Produktkonfigurationsangaben
- Liste der KE und der Bezugskonfigurationen
- aktueller Änderungsstand²
- Statusbericht der Änderungen
- Status von gelieferten Produkten [1]

In der DIN ISO 10007 wird außerdem die Forderung nach einem KM-Audit definiert. Dabei ist die KE einerseits funktionsbezogen auf die geforderten Leistungsmerkmale und andererseits auf die physische Umsetzung der gestellten Anforderungen zu überprüfen [1]. Zur Durchführung des Audits werden entsprechende Dokumente benötigt, die über den Prüfumfang einen hinreichend detaillierten Aufschluss geben.

Die einzelnen Konfigurationen eines Produktes inklusive ihrer Hierarchien untereinander sind in ihren Eigenschaften zu beschreiben. Neben einer eindeutigen Identifizierung sind sowohl eine Abbildung der enthaltenen Konfigurationen und KE als auch übergeordneten Konfigurationen vorzuhalten. Des Weiteren sind die für die Produktkonfigurationsangaben notwendigen Dokumente wie Stücklisten, Zeichnungen o. Ä. mit ihren jeweiligen Metadaten (Identifikation, Status, Änderungsindex, Freigabedatum, etc.) vorzuhalten. Aus diesen Informationen entsteht ein Konfigurationsidentifikationsdokument (KID).

Die Umsetzung der oben dargestellten Dokumente in einem SysML-Modell kann prinzipiell auf den bereits diskutierten Views und Viewpoints basieren. Je nach weiterer Verwendung und Aufbereitung der Daten aus dem Modell kann es allerdings auch möglich und sinnvoll sein, dass eine Modelltransformation angewandt wird. Das Systemmodell wird dabei zu einem neuen Modell transformiert, das wiederum als Ausgangsbasis für die weitere Verwendung gilt [6].

Mit dem Prinzip der Views lassen sich beispielsweise die KID aus dem Modell ableiten. Sofern der oben spezifizierte Stereotyp *ConfigurationItem* verwendet wird, kann damit eine Vielzahl der Anforderungen abgebildet werden. Weitere Informationen wie beispielsweise die zugehörigen Dokumente werden bereits im SysML-Standard mit den *document* Artefakten erfüllt. Eine detaillierte Beschreibung der Schnittstellen der KE kann außerdem in hohem Maße automatisiert erfolgen. Die *Ports* und *Flows* des zugrundeliegenden Blocks können über das *CI* gegriffen und entsprechend aufbereitet im KID abgebildet werden. Dabei werden dann nicht nur die Menge und Art, sondern bei Bedarf auch die angrenzenden Elemente dargestellt.

Auf Basis des Systemmodells in SysML lässt sich ein typischer Bericht wie der aktuelle Änderungsstand der Gesamtkonfiguration ableiten. Alle enthaltenen Konfigurationen sind über den o. g. Stereotypen definiert und somit systematisch greifbar. Dazu lässt sich der jeweilige Status aggregieren. Allerdings ist der Fortschritt von etwaigen Änderungsanträgen hier nicht abgebildet.

² In Kombination mit der jeweiligen Bezugskonfiguration ergibt sich ein Informationsstand wie ihn das CMII fordert.

Bei den ÄA handelt es sich im KM vorrangig um organisatorische Größen, die der Strukturierung und Koordinierung der Systementwicklung dienen [1]. Daher ist es fraglich, ob sie überhaupt in einem Systemmodell zu führen sind. Vor dem Hintergrund der eindeutigen KE innerhalb des Systemmodells können die ÄA außerhalb des Modells koordiniert werden. Werden die ÄA digital geführt, besteht die Möglichkeit, auf den Daten des Systemmodells aufzubauen. In diesem Fall greift die oben genannte Modelltransformation, indem das Systemmodell automatisiert in eine andere Form (bspw. ein Datenbankformat) transformiert wird und somit der Managementsoftware zur Verfügung steht.

6 Analyse der Hypothese

6.1 Anwendbarkeit des SysML-basierten Konfigurationsmanagements im Anlagenbau

Die Anwendbarkeit der oben spezifizierten SysML-Erweiterung hängt im Anlagenbau ganz entscheidend vom vorhandenen SysML-Modell ab. Wird ein Modell nur auf einer sehr groben Ebene spezifiziert, werden die notwendigen KE nicht hinreichend erfasst. Damit ist die Fähigkeit des oben erarbeiteten Stereotyps irrelevant. Wird ein SysML-Modell allerdings entsprechend detailliert ausgeprägt, kann die dargestellte Erweiterung die notwendigen KM-Inhalte hinreichend widerspiegeln. Dann kann auch über die Views eine entsprechende Funktionalität der Baseline-Erzeugung realisiert werden. Weiterführend ist auch die Befähigung der SysML zur zuvor genannten Dokumentengenerierung nur dann relevant, wenn das Modell entsprechend gepflegt und detailliert ist. Die Anwendbarkeit dieser oben spezifizierten Lösung hängt folglich von der Tiefe des im Projektkontext erzeugten SysML-Modells ab.

In einem beispielhaften Großprojekt zwischen der Bundeswehr und dem Fraunhofer-Institut für Experimentelles Software Engineering wurde ein SysML-Modell zur Aufbereitung eines Anforderungsmodells und zur Modellierung des Systems für die zukünftige Kampfschiffklasse *MKS180* genutzt. Dabei werden im Projektteam die Modellierungstätigkeiten klar aufbereitet, um den besonderen Gegebenheiten des Projektes gerecht zu werden. Die entstehenden Modelle dienen der lösungsneutralen Beschreibung des Bedarfes, um im Anschluss durch einen zu definierenden Auftragnehmer realisiert zu werden [15]. Ein solches Modell bietet zu diesem Zeitpunkt bereits Potenziale, ein KM durch den oben spezifizierten Stereotypen zu realisieren. Aus Sicht des Informationsgehaltes sind die hier enthaltenen Daten bereits die Basis der späteren, technisch konkreten Lösungen. Die Struktur des Systems kann dann schon aus den enthaltenen KE mit ihren unterschiedlichen Verbindungen und jeweils speziellen Anforderungen erzeugt werden. Allerdings ist zu erwarten, dass die Tiefe des Modells noch weiter zunimmt, wenn die technische Realisierung beginnt. Die Anwendbarkeit der oben dargestellten Lösung ist folglich nicht nur in der konkreten Produktentstehung, sondern bereits in Vorprojekten zur Anforderungsfindung gegeben.

Diese Eignung der Lösung für die ersten Lebenszyklusabschnitte hat zusätzlich Auswirkungen auf die weiteren Abschnitte. Ist in der initialen Produktentstehung kein hinreichendes Systemmodell erarbeitet worden, ist der Aufwand im Sinne eines Reverse-Engineerings zur Modellbildung im Nachhinein nicht mehr gegeben. Damit erweitert sich diese Abhängigkeit auf zukünftige Refit- oder Upgradeprojekte des betrachteten Systems.

Obwohl ein Großteil der KM-Anforderungen im zuvor ausgearbeiteten Stereotyp enthalten ist, ist der Punkt der Versionierung nicht einbezogen. Im SysML-Modell selbst erscheint eine sol-

che Lösung mit Hilfe des Profilmechanismus nicht möglich. Daher ist im Weiteren zu betrachten, ob im Umfeld der KM-bezogenen Modelle ein hinreichendes Änderungsmanagement etabliert werden kann.

6.2 Konfigurationsmanagementbezogenes Änderungsmanagement für SysML-Modelle

Wie die oben durchgeführte Betrachtung (vgl. Kapitel 4) zeigt, ist ein hinreichendes Änderungsmanagement in der SysML-Spezifikation nicht gegeben. Statt nun SysML selbst zu einer solchen Eigenschaft zu befähigen, wird in dieser Lösung das Problem auf die umgebenden Tools verschoben.

In der Softwareentwicklung werden weitreichend Systeme eingesetzt, die die Versionskontrolle der Entwicklungsergebnisse steuern. Diese *Software Configuration Management Systeme* verfahren dabei nach unterschiedlichen Methoden, um die Versionierung zu regeln [12]. Dabei werden prinzipiell zwei unterschiedliche Mechanismen unterschieden:

- Sperren-Ändern-Entsperren: Während einer Änderung an einem Element, kann kein weiterer Benutzer das Element ändern
- Kopieren-Ändern-Zusammenführen: Mit Unterstützung des Tools wird nach der (potenziell parallelen) Änderung eine ganzheitliche, aktuelle Version erzeugt [16].

Die Versionierung der SysML-Modelle in ihren Bestandteilen stellt eine Herausforderung dar. Eigner et al. schlagen dafür vor, die Versionierung in einem Product-LifecycleManagement-System (PLM-System) zu realisieren. Aus dem PLM-System werden dann die für den einzelnen Anwender benötigten Elemente ausgecheckt, sodass sie im SysML-Modell bearbeitet werden können [12]. Entsprechend der obigen Lösung ist es mit einem passenden Stereotyp möglich, die KE im Systemmodell abzubilden (vgl. Kapitel 4). Die von Eigner et al. vorgeschlagene Lösung ist dann außerdem auf die im KM benötigten Dokumente auszuweiten. Diese können entweder direkt als *document* im Systemmodell hinterlegt oder alternativ im PLM-System vorgehalten werden. Da sich PLM-Systeme ohnehin mit der Bereitstellung von umfangreichen Datenmengen und den jeweils notwendigen Freigabe- und Änderungsprozessen befassen, liegt es nahe, die Dokumente weitreichend im PLM-System zu hinterlegen.

Andere Ansätze schlagen beispielsweise bei der Modellarbeit für die Softwareentwicklung vor, die Versionskontrolle direkt im Autorensystem zu hinterlegen. So wurde ein erweiterter Drei-Wege-Vergleich in Enterprise Architect hinterlegt, um dort eine feingranulare Auflösung bei parallelen Arbeiten zu ermöglichen. Damit ist für den Fall der Softwareentwicklung das verteilte Arbeiten an einem zentralen (Software-) Systemmodell³ größerer Teams möglich [17].

Um nicht nur bei der Produktentstehung, sondern auch im weiteren Verlauf des Lebenszyklus hinreichende Steuerungen zu ermöglichen, ist ein Änderungsmanagement für Systemmodelle im Anlagenbau daher im PLM-System zu hinterlegen. Die notwendigen Prozesse können hier hinterlegt werden. Damit wird das Problem der Änderungssteuerung in einem dafür adäquaten Tool gelöst. Die jeweiligen Tools zur Modellierung des SysML-Modells können damit auf ihre Kernfunktion, das Modellieren von Systemen, konzentriert werden.

³ Die Unterscheidung zwischen reinem Softwaremodell oder einem interdisziplinären Systemmodell ist aus technischer nicht notwendig. Lediglich Inhalte und ggf. die umgebenden Prozesse können abweichen.

Mit der Verwaltung durch das PLM-System erscheint es zunächst redundant, die Informationen zum Status einer einzelnen KE auch im Stereotyp vorzuhalten. Damit wird allerdings dem operativen Arbeiten im Systemmodell ein wichtiger Informationsbaustein zur Verfügung gestellt. Mit dem Einstieg in das betreffende Systemmodell oder dessen Teil erkennt der Systems Engineer alle maßgeblichen, steuernden Informationen (Version, Status, etc.). Während der Arbeit im Autorensystem des Systemmodells fehlen diese Informationen zunächst. Wird den einzelnen KE allerdings ihr Status angehängt, so kann unmittelbar erkannt werden, ob die Daten eine geeignete Arbeitsgrundlage darstellen oder sich potenziell noch verändern können.

6.3 Abschluss

Die zuvor ausgearbeiteten Inhalte legen nahe, dass die eingangs formulierte Hypothese zunächst als wahr erscheint. Am Beispiel von SysML wird gezeigt, dass die Modellierungssprachen durch ihre Fähigkeiten in der Lage sind, die KM-Inhalte hinreichend abzubilden. Die weitere Betrachtung der Anwendbarkeit und des Änderungsmanagements zeigt allerdings, dass trotz theoretischer Fähigkeit eine praktische Anwendung vor großen Hürden steht. Die Abbildung der KM-relevanten Prozesse zur Änderungsverfolgung wird durch das SysML-Modell nicht unterstützt.

Der Grad der Modellbildung und vielfältige technische Hürden in der Abbildung von Inhalten und an den Schnittstellen von verschiedenen Modellen sind folglich wesentliche Merkmale einer praktischen Umsetzung. Die fortschreitende Standardisierung hilft hier weiter, das KM und die Modellierungssprachen des SE ineinander zu integrieren.

Ferner zeigt die hier angewandte Verknüpfung von Ingenieurs- und Managementdisziplinen deutlich, dass beiden Dimensionen einander bedingen und ergänzen. Dabei hilft die Systemtheorie die verschiedenen Ansprüche und Blickwinkel zu verstehen. Das SE definiert als Teil seiner selbst das KM, dem gegenüber versteht sich das KM als durchaus eigenständige Disziplin im Rahmen eines Qualitätsmanagements. Die hier dargestellte Verbindung von KM und SE beschreibt eine Integration durch das Wirtschaftsingenieurwesen beider Disziplinen, wie sie unter anderem Ahrens aufzeigt [18].

Zur Validierung der Ergebnisse bleibt eine praktische Erprobung offen. Die dafür notwendigen Grundlagen haben beide Disziplinen, sodass keine explizite Einschränkung auf eine der beiden gegeben werden kann und darf.

Literaturangaben

- [1] DIN ISO 10007; 2004-12. *Qualitätsmanagement - Leitfaden für Konfigurationsmanagement*
- [2] Buede, D. M.: *The Engineering Design of Systems: Models and Methods*. Wiley 2011
- [3] Kaffenberger, R., Schulze, S.-O. u. Weber, H.: *INCOSE Systems Engineering Handbuch*. München 2012
- [4] ISO/IEC/IEEE 15288; 2015. *Systems and software engineering -- System life cycle processes*
- [5] Habermellner, R.: *Systems Engineering. Grundlagen und Anwendung*. Zürich: Orell Füssli 2012
- [6] Alt, O.: *Modell-basierte Systementwicklung mit SysML*. Carl Hanser Fachbuchverlag 2012
- [7] SysML 1.4; 2015. *OMG Systems Modeling Language*. <http://www.omg.org/spec/SysML/1.4>, abgerufen am: 31.01.2016
- [8] Friedenthal, S., Moore, A. u. Steiner, R.: *A practical guide to SysML. The systems modeling language*. The MK / OMG Press. Amsterdam: Elsevier 2015
- [9] UML 2.5; 2015. *Unified Modeling Language*. <http://www.omg.org/spec/UML/2.5>, abgerufen am: 31.01.2016
- [10] ISO/IEC/IEEE 42010; 2011. *Systems and software engineering -- Architecture description*
- [11] Weilkens, T.: Was ist neu in der SysML 1.4 – View und Viewpoints, 2015. <http://www.oose.de/blogpost/was-ist-neu-in-der-sysml-1-4-view-und-viewpoints/>, abgerufen am: 29.12.2015
- [12] Eigner, M. Prof. Dr.-Ing. u. Gilz, T. Dipl.-Ing.: Ansatz zur integrierten Verwendung von SysML Modellen in PLM zur Beschreibung der funktionalen Produktarchitektur. In: Maurer, M. u. Schulze, S.-O. (Hrsg.): *Tag des Systems Engineering. Der Weg zu den technischen Systemen von morgen*. München: Carl Hanser Fachbuchverlag 2013, S. 293–302
- [13] XMI 2.5.1; 2015. *XML Metadata Interchange (XMI) Specification*. <http://www.omg.org/spec/XMI/2.5.1/>, abgerufen am: 04.01.2015
- [14] Wölfing, C. u. Treue, I.: Komplexitätsmanagement im Anlagenbau. In: Maurer, M. u. Schulze, S.-O. (Hrsg.): *Tag des Systems Engineering. Der Weg zu den technischen Systemen von morgen*. München: Carl Hanser Fachbuchverlag 2013, S. 11–19
- [15] Webel, C., Darting, S., Schmitt, M., Kleinberger, T., Braun, R. u. Weber, J.: Pragmatisches Systems Engineering in einem Großprojekt mit Einschränkungen. In: Muggeo, C. u. Schulze, S.-O. (Hrsg.): *Tag des Systems Engineering*. Ulm 11.-13. November 2015. München: Carl Hanser Fachbuchverlag 2015, S. 323–332
- [16] Collins-Sussman, B., Fitzpatrick, B. W. u. Pilato, C. M.: *Versionskontrolle mit Subversion*. Für Subversion 1.7. 2004
- [17] Alt, O. Dr., Bretz, R. Dr., Graser, F. Dr. u. Wieland, K.: *Neuer Ansatz für die Versionierung von Modellen*. Softwaremodellierung. Elektronik Praxis (2016)

[18] Systemtheorie als wissenschaftliche Grundlage des Wirtschaftsingenieurwesens.
Arbeitspapiere der Nordakademie 2017-01, Ahrens, V. Prof. Dr.-Ing., Elmshorn 2017

Die Autoren

Florian Schnoor (Jahrgang 1991) studierte an der NORDAKADEMIE sowie an der NORDAKADEMIE Graduate School sowohl im Bachelor- als auch im Masterstudiengang Wirtschaftsingenieurwesen. Im Rahmen der Thesis zum M. Sc. hat er dabei einen „Beitrag zur Systematisierung des Konfigurationsmanagements im Anlagenbau auf Basis des Systems Engineering“ geschrieben. Als ein Auszug dieser Thesis ist dieser Artikel entstanden.

Volker Ahrens, 1963 in Bremen geboren, studierte Maschinenbau an der Universität Hannover und promovierte bei Prof. Dr.-Ing. E.h. mult. Dr. sc. h.c. Dr.-Ing. Hans-Peter Wiendahl am Institut für Fabrikanlagen (IFA) der gleichen Universität zum Thema „Dezentrale Produktionsplanung und -steuerung – Systemtheoretische Grundlagen und Anwendungspotentiale“. Anschließend war er 10 Jahre in der Geschäftsleitung mittelständischer Industrieunternehmen tätig, zuletzt als Technischer Geschäftsführer. Seit 2007 ist er Professor an der NORDAKADEMIE, einer privaten, staatlich anerkannten Fachhochschule in Elmshorn. Dort leitet er den Studiengang Wirtschaftsingenieurwesen. Prof. Dr. Ahrens hat die o. g. Thesis von Herrn Schnoor seitens der Hochschule begleitet.