

# Entwicklung eines generischen Modells für die standardisierte Beschreibung von Ressourcen in der Luftfahrtproduktion

Lasse Beers\*, Maximilian Weigand, Alexander Fay  
Institut für Automatisierungstechnik  
Helmut-Schmidt-Universität / Universität der Bundeswehr  
Hamburg  
Hamburg, Deutschland  
\*lasse.beers@hsu-hh.de

Alain Chahine  
Airbus Operations GmbH  
Hamburg, Deutschland

**Kurzfassung** – Um in frühen Phasen des Engineerings Entscheidungen über die Konfiguration von Produktionssystemen zu treffen, sind Modelle von verfügbaren Produktionsressourcen notwendig. Um diesem Bedarf gerecht zu werden, wurde ein SysML-Modell entwickelt, mit dessen Hilfe eine einheitliche Beschreibung von Komponenten, Schnittstellen und Fähigkeiten von Produktionsressourcen möglich ist. Die Granularität des Modells ist relativ grob, da zwar eine gewisse Dekomposition der Ressourcen in einzelne Komponenten möglich sein soll, jedoch keine physikalische Repräsentation vorgesehen ist. So kann es effizient und automatisiert ausgewertet werden, ohne detailreiche und rechenintensive Simulationen durchzuführen. Zukünftig soll die Kompatibilität von unterschiedlichen Ressourcen basierend auf deren Schnittstellen oder die Durchführbarkeit von Prozessschritten basierend auf den Fähigkeiten von Ressourcen untersucht werden können. So können sowohl die Durchführbarkeit eines gesamten Produktionsprozesses untersucht werden als auch die möglichen Durchführungsvarianten anhand von KPIs miteinander verglichen werden.

**Stichworte** – Flugzeugproduktion, SysML, Automatisierung, Informationsmodell, Ressourcenmodellierung

## I. DIGITALE MODELLIERUNG VON RESSOURCEN IN DER FLUGZUGPRODUKTION

Passagierflugzeuge sind Systeme, deren hohe Komplexität sich aus ihrer physischen Größe und der Vielfalt an technischen Disziplinen ergibt, die in der Entwicklung beteiligt sind. Hinzu kommt eine große Variabilität, die durch den Endkunden gefordert wird. Dieser hat zum Beispiel je nach Einsatzzweck genaue Anforderungen an das Layout der Kabine.

Auch Aspekte, die zur Zeit der Produktion gesetzt sind, wie die Art des Antriebs, können in der Entwurfsphase noch offen sein. Ein elektrischer Antrieb hätte beispielsweise große Auswirkungen auf die Klimaanlage, da diese bei Einsatz von konventionellen Strahltriebwerken über den Verdichter des Triebwerks versorgt wird. Da gerade Verkehrsflugzeuge über lange Zeiträume entwickelt werden, können sich weitere Anforderungen durch externe Einflüsse, wie gesetzliche Beschränkungen, ändern. Außerdem können einzelne

technische Problemstellungen durch den Einsatz von Neuentwicklungen gelöst werden.

Mit Hilfe von Digitalisierungstechniken können hierzu benötigte Modelle von Produkten mit der gewünschten Detaillierungstiefe und Variabilität erstellt werden. Jedoch muss ebenfalls die Komplexität der benötigten Produktionsumgebung beachtet werden. Auch diese soll geplant werden, basierend auf der digitalen Beschreibung der herzustellenden Produkte, der Produktionsressourcen und der dafür benötigten Prozesse [1]. Bestehende Technologien ermöglichen die Modellierung der Geometrie oder des kinematischen Verhaltens von Produktionsressourcen. Solche Modelle werden oft von Produktionstechnikerherstellern, auch in digitaler Form, veröffentlicht. In einer Planungssoftware kann dann eine Produktionsumgebung aus einzelnen Ressourcenmodellen zusammengestellt werden. Jedoch ergibt sich dadurch oft ein hoher Zeit- und Kostenaufwand, der in frühen Engineeringphasen vermieden werden sollte, da die Modelle in diesen Phasen häufigen Änderungen unterliegen. In dieser Phase werden Modelle der Produktionsressourcen benötigt, die von Geometrie und Kinematik abstrahieren. Modelbasierte Ansätze ermöglichen eine schnellere und effektivere Reaktion der Planer auf die zahlreichen Anforderungsänderungen, die während des Entwicklungsprozesses auftreten [2]. Aber nicht nur die Reaktion auf Änderungen, sondern auch die Identifikation und Bewertung von Realisierungsalternativen sind für die System Engineers nicht leicht zu finden und erfordern oft eine mühsame manuelle Definition der Implementierungselemente [3].

Daher gewinnt die Modellierung und Verwaltung von Produktionsressourcen für die Prozess- und Ressourceneinsatzplanung für Fertigungsunternehmen immer mehr an Bedeutung [4].

Die Nutzung der Modelle ist dabei nicht allein auf die „Erstmodellierung“ der Systeme beschränkt, sondern kann im weiteren Produktlebenszyklus auch für die Identifikation von rekonfigurierbaren Fertigungsressourcen genutzt werden, die in der Luftfahrtproduktion immer wichtiger werden [5]. Um den Aufwand bei der Erstellung solcher Modelle zu reduzieren sowie eine einheitliche Struktur bei der Erstellung zu

gewährleisten, stellt dieser Beitrag ein generisches Modell zur standardisierten Beschreibung von Fertigungsressourcen vor.

Der nachfolgende Beitrag ist wie folgt gegliedert: In Kapitel II werden verwandte Forschungsansätze sowie verschiedene Beschreibungsmittel vorgestellt. Anschließend wird in Kapitel III das generische Modell vorgestellt, welches in Kapitel IV für eine beispielhafte Modellierung genutzt wird. Abschließend wird in Kapitel V ein Fazit gezogen und ein Ausblick in zukünftige Weiterentwicklungspotenziale gegeben.

## II. STAND DER TECHNIK

Bei der Auswahl der gewählten Modellierungssprache sind in der Literatur wesentliche Unterschiede im Detaillierungsgrad der Modellierung sowie bei den beschriebenen Anwendungsfällen zu erkennen. Als die vielversprechendsten und zugleich meistgenutzten Modellierungssprachen wurden die Web Ontology Language (OWL), die Automation Markup Language (AML) sowie die System Modelling Language (SysML) identifiziert. Die folgenden Unterkapitel stellen eine Auswahl bestehender Ansätze vor und diskutieren die Eignung für den beschriebenen Anwendungsfall.

### A. Web Ontology Language (OWL)

Die OWL ist eine durch das World Wide Web Consortium (W3C) standardisierte, computerinterpretierbare Sprache für Semantic-Web-Anwendungen, welche es ermöglicht, komplexes Wissen als Graphen zu beschreiben [6]. Die Daten des Graphen werden in Form von Individuen und Beziehungen zwischen diesen festgehalten. Ebenfalls kann die Semantik des Graphen explizit definiert werden, indem Individuen und Beziehungen klassifiziert werden sowie definiert wird, welche Beziehungen zwischen bestimmten Klassen von Individuen bestehen können. Diese Metainformationen werden auch Ontologie genannt [7].

Im Forschungsgebiet Semantic-Web werden neben Beschreibungssprachen auch Sprachen zur Abfrage von Wissen aus OWL-Graphen (wie z.B. SPARQL [8]) oder Sprachen zur Definition von Regeln, durch die auf zusätzliches Wissen geschlussfolgert werden kann, entwickelt (z.B. SWRL [9]).

Ontologien und Wissensgraphen werden auch in der aktuellen Forschung in den Ingenieurwissenschaften, im Speziellen in der Automatisierungstechnik, angewandt.

Zum Beispiel können Ontologien genutzt werden, um Begriffe, Attribute und Beziehungen aus dem Bereich der Automatisierungstechnik und der Robotik zu beschreiben [10]. Auch Know Rob 2.0, ein Wissensverarbeitungssystem für autonome Serviceroboter, basiert auf Ontologien [11]. In der Robotik ist oft auch die Beschreibung der Umgebung, in welcher der Roboter betrieben wird, von großer Bedeutung. Die von Lemaignan et al. definierte Ontologie kann für die Beschreibung der Umwelt und der Interaktion von Menschen und Maschine genutzt werden [12]. Auch ein Beitrag von Bruno et al. befasst sich mit der Mensch-Maschine-Interaktion im speziellen Anwendungsfeld der Service-Roboter für die Pflege von hilfsbedürftigen Menschen [13].

Hildebrandt et al. stellen einen Ansatz vor, mit dessen Hilfe einzelne Aspekte cyber-physikalische Systeme (CPS), wie Ressourcenstruktur, Fähigkeiten oder Skills, mit einzelnen Ontologien beschrieben werden können, welche

mithilfe einer übergeordneten Ontologie miteinander in Verbindung gesetzt werden [14].

In weiteren Beiträgen werden die Aspekte Fähigkeiten und Skills genauer betrachtet, ebenfalls basierend auf Ontologien [15, 16]. Auch Haage et al. stellen eine Reihe von Ontologien vor, die zusammen die Implementierung und Roboterfähigkeiten ermöglichen und bei der Rekonfiguration dieser Systeme helfen [17]. Diab et al. entwickeln eine Robotik-Ontologie mit dem Ziel, Wissen aus der Domäne der Aufgaben- und Pfadplanung zu inferieren [18].

### B. System Modelling Language (SysML)

Die SysML ist eine universelle grafische Modellierungssprache zur Spezifikation, Analyse, Gestaltung und Überprüfung komplexer Systeme. Sie stellt eine Teilmenge der UML 2 sowie eine Erweiterung um zusätzliche Funktionen und Diagrammtypen dar, sodass die SysML auch im Engineering komplexer physischer Systeme eingesetzt werden kann [19].

Mykoniatis et al. stellen einen modellbasierten Forschungsansatz für eine integrierte konzeptionelle Entwurfsbewertung mechatronischer Systeme unter Verwendung von SysML vor. Es werden verschiedene Ressourcenkomponenten beschrieben, sowie eine Methode vorgestellt mit deren Hilfe verschiedene Kombinationen hinsichtlich definierter Kriterien bewertet werden. Die Modellierung beschränkt sich auf die Darstellung von Komponenten und berücksichtigt weder Komponentenschnittstellen noch Ressourcenfähigkeiten [20].

Tsuji et al. kritisieren, dass aufgrund von fehlenden standardisierten Modellen die Wiederverwendbarkeit von Systemarchitekturen für Roboter bei der Nutzung von Middleware-Plattformen wie ROS gering ist. Daher entwickeln die Autoren ein SysML-Modell für die Systementwicklung von Robotern, welches eine hohe Wiederverwendbarkeit besitzt. Der Fokus der Modellierung liegt auf der Beschreibung von Prozessen und dem Datenaustausch der einzelnen Komponenten [21].

Huckaby et al. beschreiben die Modellierung von Roboterfähigkeiten in einer Assembly Domain [22]. Der Ansatz beschreibt jedoch keine Darstellung der physischen Struktur der entsprechenden Ressourcen.

### C. Automation Markup Language (AML)

Die AutomationML ist eine XML-basierte, objektorientierte Modellierungssprache. Abbildbar sind die Objekttopologie inklusive Hierarchien, Eigenschaften und Beziehungen, Geometrien und Kinematiken und die Modellierung von diskretem Verhalten. AML greift auf standardisierte und etablierte Datenformate und Normen zurück [23].

Hoang et al. nutzen die AML zur Implementierung eines maschineninterpretierbaren Fähigkeitsmodells. Das Modell soll durch den Abgleich von Produktanforderungen und Ressourcenfähigkeiten Produktionsplanungs- und Rekonfigurationsprozesse erleichtern [24].

Ferreira et al. stellen einen AML-basierten PPR-Ansatz vor, der ausführbare Ressourcen-Fähigkeiten beschreibt. Die Beschreibung der Fähigkeiten wird in einem Plug & Produce Szenario eingesetzt und mit Montageanforderungen verglichen. Durch den Aufbau von Bibliotheken wird der Aufwand bei der Erstellung der Modelle reduziert [25].

#### D. Auswahl des Beschreibungsmittels

Eine Modellierung basierend auf OWL zeichnet sich vor allem durch einen hohen Formalisierungsgrad aus. Ein hoher Freiheitsgrad in der Modellierung von Metamodell und Modellinhalten ermöglicht die Abbildung beliebigen Wissens, setzt aber ein großes Expertenwissen für den Aufbau, die Erweiterung und die Pflege der Modelle voraus. Eine grafische Repräsentation ist nicht vorgesehen, dies schränkt die Anwendbarkeit und Lesbarkeit für Laien ein. Daher wird von der Nutzung von OWL abgesehen.

Die AutomationML ist ein objektorientiertes, maschinenlesbares Datenformat, welches für Software-Werkzeug-übergreifenden Datenaustausch geeignet ist. Es können zwar domänenspezifische Bibliotheken aufgebaut und genutzt werden, allerdings ist die Modellierung auf Ressourcen beschränkt. Eine ebenfalls entscheidende Anforderung an das Beschreibungsmittel ist die Modellierung von Anforderungen sowie die Analyse, ob eine Ressource für einen bestimmten Prozess geeignet ist.

Die SysML ist ebenfalls eine objektorientierte Modellierungssprache, welche eine diagrammbasierte, grafische Repräsentation beinhaltet. Dies erleichtert die Anwendbarkeit und Lesbarkeit speziell für Laien. Profile ermöglichen eine beliebige Erweiterung der Modelle. Viewpoints können genutzt werden, um die modellierten Ressourcen domänenspezifisch zu filtern und darzustellen. Das Modellieren und Überprüfen von Anforderungen ist eine Kernkompetenz von SysML und gewährleistet die richtige Auswahl der Ressourcen. Tools, die SysML implementieren, wie z.B. Cameo Systems Modeler, stellen zusätzliche Funktionen, beispielsweise zur Simulation oder Validierung der Modelle, bereit. Die geringere Ausdrucksmächtigkeit im Vergleich zu OWL ist ein Nachteil, jedoch für den Aufbau der Ressourcenmodelle nicht problematisch, da die darzustellenden Ressourcen physikalische Systeme sind, auf die sich die SysML beschränkt.

### III. BESCHREIBUNG DES GENERISCHEN MODELLS

Die Analyse der Literatur zeigt, dass SysML eine geeignete Modellierungssprache für die Erstellung eines generischen Modells für die Darstellung von Produktionsressourcen ist. Die Struktur des Modells sowie die Darstellungsart der Inhalte wird durch die objektorientierte Darstellung von SysML kaum eingeschränkt. So ist sichergestellt, dass alle Inhalte, die für die anfangs genannten Anwendungsfälle des Modells notwendig sind, dargestellt werden können.

Neben diesem Aspekt hat jedoch auch die Akzeptanz des Modells durch die nutzenden Planer eine hohe Priorität. Diese soll durch eine Orientierung der Struktur sowie der Inhalte des Modells an vorhandenen und etablierten Standards und Normen sichergestellt werden. Durch eine ähnliche Struktur können Daten aus bestehenden Datenquellen in das Modell übertragen oder aus dem Modell in einer Form ausgegeben werden, die für andere Anwendungen lesbar ist. Die referenzierten Normen und Standards stammen aus den Bereichen Automatisierungstechnik und Fertigungstechnik. Der aus den Standards resultierende Aufbau des Modells sowie die Erweiterung um neue Aspekte und Verknüpfungen werden in den folgenden Unterkapiteln beschrieben.

#### A. Grundstruktur des Modells nach PPR

Obwohl der Fokus des Modells auf der Darstellung von Ressourcen liegt, wird als oberste Ebene eine weitere Abstraktion eingeführt. Das PPR-Modell (Produkte, Prozesse und Ressourcen) ist eine gängige Darstellung in der Produktionstechnik und wird auch vom Netzwerk Plattform Industrie 4.0 empfohlen [26], wo es zusätzlich um Capabilities erweitert wird. Das in SysML übertragene Modell wird in ABBILDUNG 1 dargestellt und bildet die Basis für das hier vorgestellte Ressourcenmodell.

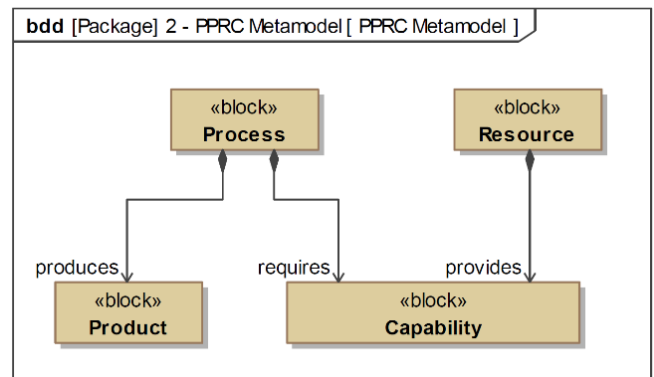


ABBILDUNG 1: IN SYSML ÜBERTRAGENES PPR-MODELL, ERWEITERT UM CAPABILITIES [26].

Über den Block Capability entsteht die Verbindung zu den Blöcken Process und Product. Diese Verbindung ist zunächst nicht erforderlich, soll aber später genutzt werden, um automatisiert abzufragen, welche Ressourcen für die Durchführung eines Prozesses und damit die Herstellung eines Produktes geeignet sind.

#### B. Klassifizierung von Ressourcen und technische Spezifikation

Als Grundgerüst für die Struktur des Modells wird der OPC-UA-Standard herangezogen. Dieser besteht aus mehreren Bestandteilen, sogenannten Specifications, die unter anderem ein Informationsmodell definieren, mit dem industrielle Kommunikationsarchitekturen dargestellt werden können [27]. Das grundlegende Meta-Modell wird sehr abstrakt beschrieben, z.B. durch die Definition von Datentypen [28], während erweiternde Specifications praxisnahe Meta-Modelle für die Darstellung konkreter Hard- und Software bereitstellen [29]. Zusätzlich besteht für industrielle Vereinigungen durch sogenannte Companion-Specifications die Möglichkeit, OPC UA domänenspezifisch zu erweitern [27], z.B. im Bereich der Robotik [30].

Im hier dargestellten Ressourcenmodell wird auf die Specification für Devices [29] und auf die Companion-Specification für Robotics [30] zurückgegriffen. Beide hängen miteinander zusammen und werden für das Ressourcenmodell in SysML übertragen. ABBILDUNG 2 zeigt einen Ausschnitt aus dem Ressourcenmodell, das die Robotics-Companion-Specification abbildet.

Im Bereich der industriellen Kommunikation ist OPC UA ein anerkannter Standard, sodass durch die Orientierung an diesem die Akzeptanz des Ressourcenmodells unter den später nutzenden Ingenieurinnen und Ingenieuren gesichert werden kann. Ebenfalls kann die Beschreibung der Kommunikationsinfrastruktur später automatisiert aus dem SysML-Modell in Form eines OPC-UA-Informationsmodells

ausgegeben und für die Implementierung der realen Infrastruktur verwendet werden.

Nachteilig ist, dass der OPC-UA-Standard auf die Darstellung von Netzwerkarchitekturen fokussiert und weniger die physische Struktur von Ressourcen berücksichtigt wird. Das OPC-UA-Informationsmodell ist aber grundsätzlich erweiterbar. Der Standard stellt daher das Grundgerüst für die Modellierung der physischen Strukturen dar, muss jedoch um einige Aspekte erweitert werden.

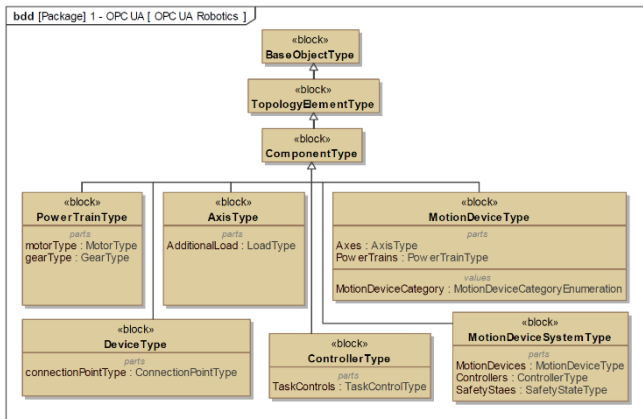


ABBILDUNG 2: NACHBILDUNG DER OPC UA COMPANION SPECIFICATION FOR ROBOTICS IN SYSLM.

Dafür wurde ein Interface definiert, welches von physischen Komponenten (wie z.B. technischen Ressourcen) implementiert werden kann. Ein Interface ist hier im Sinne der objektorientierten Programmierung als eine abstrakte Superklasse zu verstehen, die von spezifischeren Klassen implementiert werden kann. Die physische Komponente erbt vom Interface IPhysicalComponentType typische Eigenschaften, wie z.B. die Masse (mass) oder die Staubschutzklasse (IP rating), was in ABBILDUNG 3 dargestellt wird.

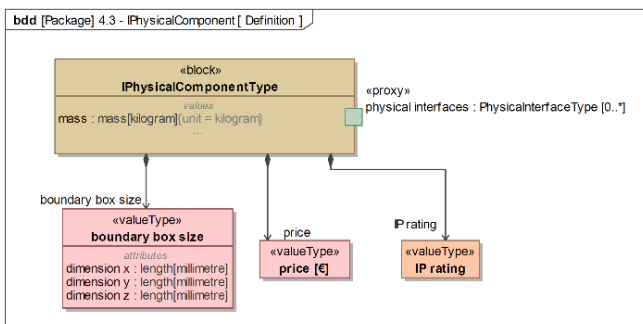


ABBILDUNG 3: INTERFACE IPHYSICALCOMPONENTTYPE ZUR VERERBUNG VON EIGENSCHAFTEN PHYSIKALISCHER KOMPONENTEN.

### C. Schnittstellen

Neben der Beschreibung der physischen Komponenten einer Ressource muss auch die Kompatibilität dieser mit anderen Ressourcen darstellbar sein. Dies wird mit Hilfe von Schnittstellen umgesetzt, welche zunächst in mechanische Schnittstellen zur Kraftübertragung, elektrische Schnittstellen zur Stromversorgung und Kommunikationsschnittstellen untergliedert werden. Weitere Schnittstellen, z.B. zur Beschreibung von Versorgung mit Hydraulikflüssigkeit, Kühlmittel oder Druckluft sind aktuell kein Bestandteil und sollen in der weiteren Umsetzung berücksichtigt werden.

Da gerade zum aktuellen Zeitpunkt Anwendungen im Bereich der Robotik im Fokus stehen, wird zunächst die ISO 9409 zur Beschreibung einer Unterklasse von mechanischen Schnittstellen referenziert. Diese Norm beschreibt Abmaße und Bezeichnungen typischer mechanischer Schnittstellen für die Installation und Austauschbarkeit von Endeffektoren in der Robotik. Durch eine spezielle Codierung hilft die Norm, passende Endeffektoren zu identifizieren. [31, 32]

Für elektrische Steckverbindungen existieren einige Normen, die in das Modell einfließen. Beispiele sind RJ-45 oder IEC 19/20 für Steckverbindungen. Diese dienen der Beschreibung von elektrischen Schnittstellen zur Stromversorgung bzw. kabelgebundenen Kommunikation. Auch für kabellose Kommunikation existieren Standards, wie zum Beispiel der Standard IEEE 802.11.

Kommunikationsprotokolle sind meist unabhängig von der physischen Umsetzung der Verbindung und können daher Kommunikationsschnittstellen allgemein zugeordnet werden. Hier wird ein Bezug zum Standard OPC UA DI, zu sogenannten ConnectionPoints, hergestellt. Eine Auflistung konkreter, gängiger Kommunikationsprotokolle der Automatisierungstechnik ist nicht in OPC UA vorhanden. Hier wird auf das OSI-Modell [33] zurückgegriffen, mit dem nach einem Layer-Prinzip verschiedene Kommunikationsprotokolle beschrieben werden können, auch industrielle Protokolle wie PROFINET [34].

### D. Fähigkeiten

Neben der Beschreibung der physischen Struktur und Schnittstellen ist die Beschreibung der Ressourcen-Fähigkeiten, in ABBILDUNG 1 als Capability beschrieben, ein essenzieller Bestandteil, um geeignete Ressourcen oder Konfigurationen zu identifizieren. Die Untergliederung der Klasse ist in ABBILDUNG 4 dargestellt. Im Fokus des Modells steht die Beschreibung von Produktionsressourcen, sodass auf etablierten Normen dieser Domäne zurückgegriffen wurde, um eine eindeutige Taxonomie zu definieren.

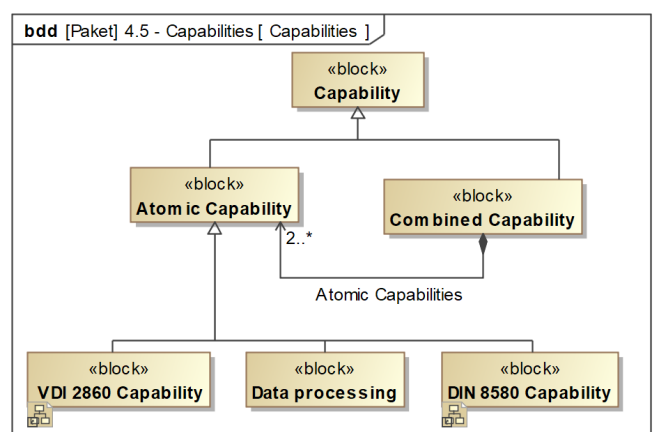


ABBILDUNG 4: UNTERGLIEDERUNG DER KLASSE CAPABILITY.

Die DIN 8580 [35] definiert und erläutert Grundbegriffe im Gesamtbereich der Fertigungsverfahren. Sie bildet die Grundlage für den Aufbau eines Ordnungssystem durch die Untergliederung in Haupt- und Untergruppen. Die VDI 2860 [36] definiert und unterteilt das Handhaben von Material und Produkten. Sie gliedert Prozesse des Handhabens in fünf Hauptgruppen, welche wiederum in Teilfunktionen unterteilt werden. Ziel dabei ist es, die Planung und Auslegung solcher Prozesse zu erleichtern.

Zunächst ist zwischen einer Atomic Capability und einer Combined Capability zu unterscheiden. Eine Ressourcen-Komponente, welche genau über eine atomare Fähigkeit verfügt, besitzt eine Atomic Capability, während eine Komponente mit mindestens zwei Capabilities ein Combined Capability besitzt.

Dem gegenüber steht ein Prozess, welcher aus mehreren Prozessschritten besteht. Jeder einzelne Schritt benötigt stets genau eine Capability für seine Umsetzung. Ein Prozessschritt, welcher mehr als eine Capability benötigt, wird aufgeteilt und als parallellaufende Prozessschritte modelliert. Neben der Beschreibung der Capabilities sind auch die entsprechenden Prozessparameter zu beschreiben. Diese können zum einen den Normen entnommen, zum anderen durch Experten-Interviews und durch Durchsicht von entsprechenden Standard Operation Instructions (SOI) identifiziert werden. Nur wenn die Capability und der entsprechende Parametersatz die Prozessanforderungen erfüllen kann, ist die Ressource zur Prozessumsetzung geeignet.

Eine detailliertere Beschreibung des Prozessmodells wird in einer zukünftigen Veröffentlichung vorgestellt.

#### IV. BEISPIELHAFT E UMSETZUNG

Zur Veranschaulichung des Modells wird eine Ressource beispielhaft modelliert. Dabei handelt es sich um einen Industrieroboter, exemplarisch einen UR10 des Herstellers Universal Robots. Das definierende SysML-Block Definition Diagramm (Block Definition Diagram, bdd) wird in ABBILDUNG 5 dargestellt. Um eine Ressource in der Ressourcenbibliothek anzulegen, wird diese als Unterklasse einer kategorisierenden Klasse (hier: *IndustryRobot*) angelegt. Eine weitere Systematisierung ist denkbar, z.B. die Definition einer Klasse für Roboter einer Produktlinie. Die hier definierte Klasse UR10 überschreibt die Standardwerte verschiedener Eigenschaften, die sie von *IndustryRobot* oder abstrakteren Klassen erbt. Eine Angabe zur Staubschutzklasse (*IP rating*) wird z.B. von der Klasse *IPPhysicalComponentType* geerbt und hier mit IP54 überschrieben. Auch komplexere Angaben können überschrieben werden, z.B. die Ressourcenfähigkeiten (*provides*). In diesem Fall wird dem Roboter die Fähigkeit Führen (nach der Definition der VDI Richtlinie 2860) mit einer Genauigkeit von 100 µm zugeordnet.

Auf die gleiche Weise ist auch die Modellierung anderer Komponenten wie Tools, Sensoren oder Controllern möglich.

Die in der Ressourcenbibliothek angelegten Komponenten können für die Modellierung eines übergeordneten Systems in einem weiteren bdd genutzt werden. In diesem Modellierungsbeispiel wird, wie in ABBILDUNG 6 zu sehen, ein Robotersystem, bestehend aus einem UR10, einem 2D Laser Scanner und einen Greifer-Tool modelliert.

Während der UR10 als MotionDevice feststeht, stehen je zwei Sensoren und Tools als mögliche Komponenten zur Verfügung. Außerdem wurde eine maximale Masse des Gesamtsystems von 33 kg als Requirement für den Anwendungsfall definiert. Das Parametric Diagramm ermöglicht durch Constraint Blocks die Modellierung von mathematischen und logischen Zusammenhängen wie hier die Berechnung der Gesamtmasse. Die verschiedenen Systemvarianten können anschließend in einer Instance Table implementiert und mit Hilfe des Cameo Simulation Tool-Kit ausgewertet werden.

Der Instance Table aus ABBILDUNG 6 ist zu entnehmen, dass die Möglichkeit 3 aufgrund ihrer Gesamtmasse von 34.05 kg nicht für den Einsatzzweck in Frage kommt.

#### V. FAZIT UND AUSBLICK

Das in Kapitel IV gezeigte Beispiel stellt dar, wie Ressourcen modelliert werden können.

Auch die Modellierung verschiedener Systemvarianten konnte gezeigt werden, wenn auch nur an einem einfachen Anwendungsbeispiel. Die Anwendbarkeit des Modells bei der Modellierung komplexerer industrieller Systeme sowie der Überprüfung komplexerer Anforderungen wird zukünftig überprüft.

Weiterhin wurden automatisierte Auswertungen des Modells bisher nicht behandelt und nur prototypisch implementiert. Geplant sind Auswertungen, welche Ressourcen, basierend auf ihren Schnittstellen, miteinander kompatibel sind und welche Prozesse sie, basierend auf ihren Fähigkeiten, durchführen können. Eine Kombination beider Auswertungen ist ebenfalls sinnvoll, da industrielle Ressourcen oft zusammengesetzt sind (z.B. ein Roboterarm mit einem Greifer) und Prozesse oft nur durch kombinierte Fähigkeiten (hier: *Bewegen* und *Greifen*) realisierbar werden. Um beliebig

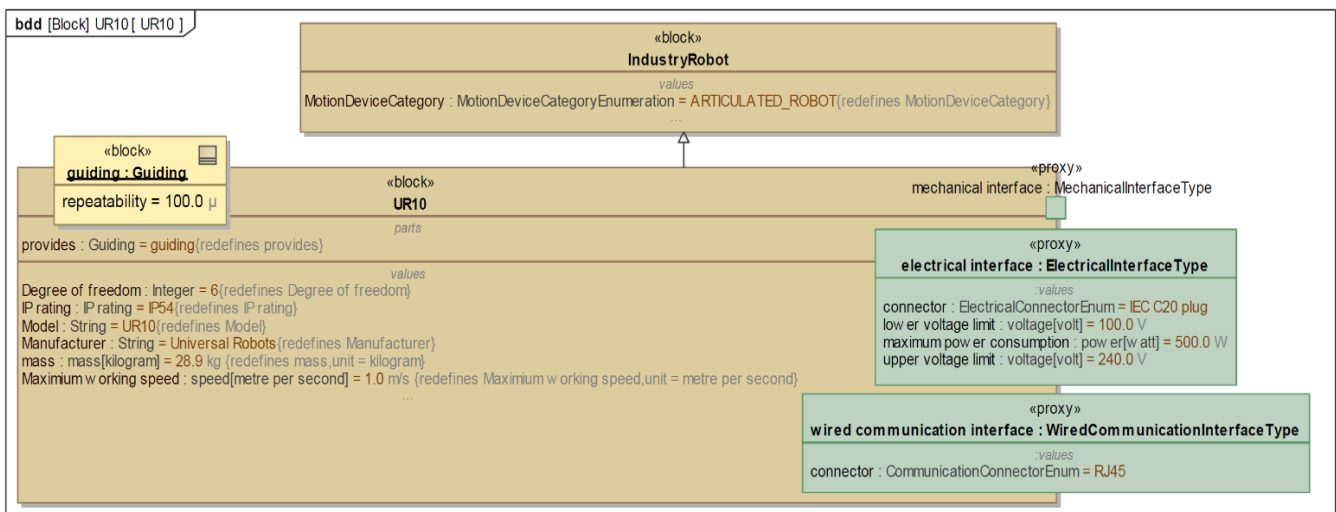
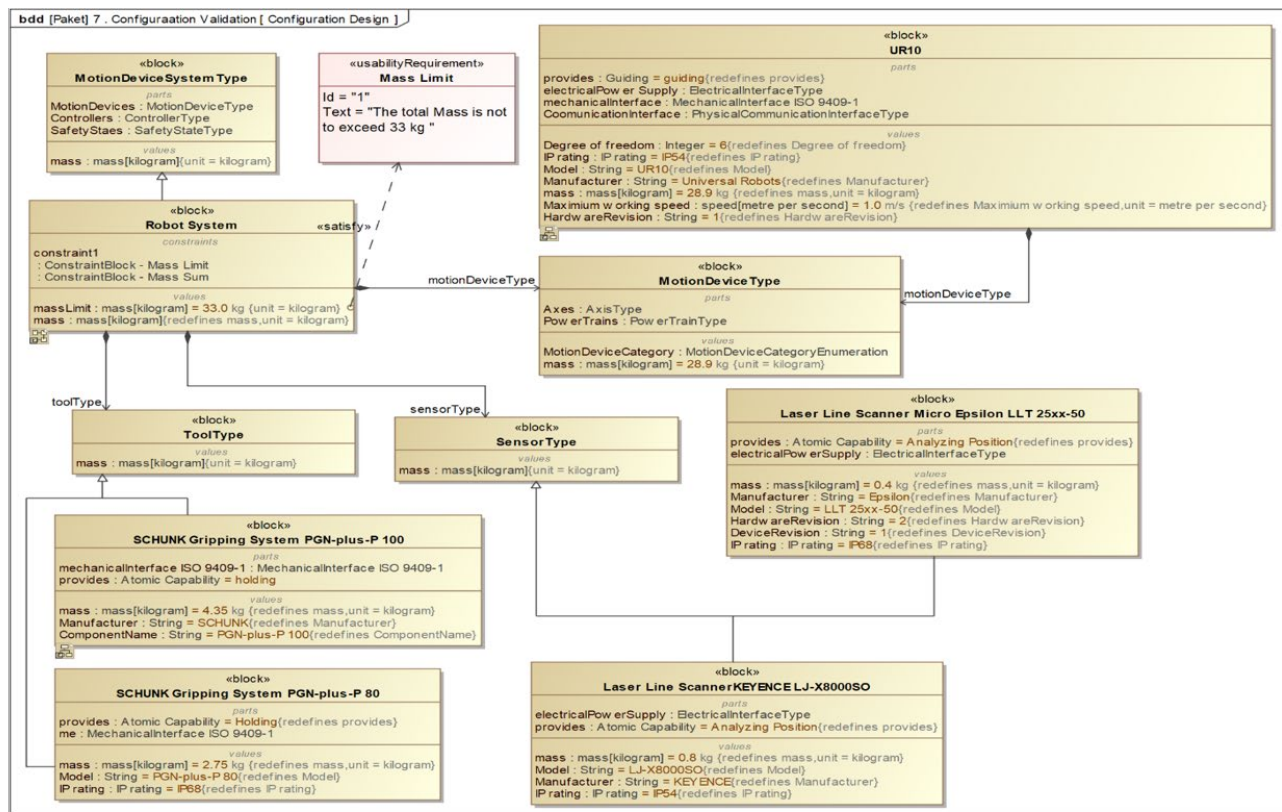


ABBILDUNG 5: SYSML BLOCK DEFINITION DIAGRAM ZUR SPEZIFIKATION DER RESSOURCE UR10 (ZU BESSEREN LESBARKEIT WERDEN EINIGE EIGENSCHAFTEN UND SCHNITTSTELLEN VERKÜRZT DARGESTELLT).



#	Name	massLimit mass[kilogram]	mass mass[kilogram]	toolType : ToolType	sensorType : SensorType	motionDeviceType : MotionDeviceType
1	robot System	33 kg	32.75 kg	schunk gripping system pgn-plus-p 100 : SCHUNK Gripping System PGN-plus-P 100	4.35 kg	laser Line Scanner Micro Epsilon LLT 25xx-50.4 kg
2	robot System1	33 kg	32.05 kg	schunk gripping system pgn-plus-p 80 : SCHUNK Gripping System PGN-plus-P 80	2.75 kg	laser Line Scanner Micro Epsilon LLT 25xx-50.4 kg
3	robot System2	33 kg	34.05 kg	schunk gripping system pgn-plus-p 1001 : SCHUNK Gripping System PGN-plus-P 100	4.35 kg	laser Line Scanner 23 : Laser Line Scanner 20.8 kg
4	robot System3	33 kg	32.45 kg	schunk gripping system pgn-plus-p 801 : SCHUNK Gripping System PGN-plus-P 80	2.75 kg	laser Line Scanner 24 : Laser Line Scanner 20.8 kg

ABBILDUNG 6. MODELLIERUNG VERSCHIEDENER RESSOURCEN-KOMPONENTEN KOMBINATION (OBEN) SOWIE DIE AUSWERTUNG DER GESAMTMASSE INNERHALB EINER INSTANCE TABLE (UNTEN)

komplexe, logische Abfragen an Informationsmodelle zu stellen, ist SPARQL eine geeignete Abfragesprache. Eine Methode, die den Zugriff auf Informationen aus einer beliebiger Software in Form eines Graphen ermöglicht [37], wurde anhand des Ressourcenmodells umgesetzt. In ersten Versuchen wurden einfache Abfragen gestellt. Da SPARQL komplexe logische Abfragen zulässt, ist zu erwarten, dass die benötigten Informationen über die genannte Schnittstelle abfragbar sind.

Zum aktuellen Zeitpunkt ist der Aufwand für die Modellierung von Ressourcen aufgrund viel manueller Arbeit hoch. Es ist daher notwendig zu untersuchen, ob Informationen über Ressourcen auch aus bestehenden Quellen automatisiert übertragen werden können.

### DANKSAGUNG

Die Autoren bedanken sich für die Förderung bei dtec.bw – Zentrum für Digitalisierungs- und Technologieforschung der Bundeswehr [Projekt iMOD].

### REFERENCES

[1] F. Gehlhoff *et al.*, “Challenges in Automated Commercial Aircraft Production,” *IFAC-PapersOnLine*, vol. 55, no. 2, pp. 354–359, 2022.  
 [2] J. D’Ambrosio and G. Soremekun, “Systems engineering challenges and MBSE opportunities for automotive system design,” in *2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 2017.  
 [3] P. Z. Chadzynski, B. Brown, and P. Willemsen, “Enhancing Automated Trade Studies using MBSE, SysML and PLM,” *INCOSE International Symposium*, vol. 28, no. 1, pp. 1626–1635, 2018.

[4] E. M. Sanfilippo *et al.*, “Modeling Manufacturing Resources: An Ontological Approach,” in *IFIP Advances in Information and Communication Technology, Product Lifecycle Management to Support Industry 4.0*, P. Chiabert, A. Bouras, F. Noël, and J. Rios, Eds., Cham: Springer International Publishing, 2018, pp. 304–313.  
 [5] R. Arista, F. Mas, D. Morales-Palma, and C. Vallellano, “Industrial Resources in the design of Reconfigurable Manufacturing Systems for aerospace: A systematic literature review,” *Computers in Industry*, vol. 142, p. 103719, 2022.  
 [6] W3C OWL Working Group, “OWL 2 Web Ontology Language,” Accessed: Aug. 3 2022. [Online]. Available: <https://www.w3.org/TR/owl2-overview/>  
 [7] R. Neches *et al.*, “Enabling Technology for Knowledge Sharing,” *AI Magazine*, vol. 12, no. 3, pp. 36–56, 1991.  
 [8] E. Prud’hommeaux and A. Seaborne, “SPARQL Query Language for RDF,” W3C Recommendation, W3C, 2008. Accessed: Aug. 3 2022. [Online]. Available: <http://www.w3.org/TR/2008/REC-rdf-sparql-query-20080115/>  
 [9] I. Horrocks, P. F. Patel-Schneider, H. Boley, S. Tabet, B. Groszof, and M. Dean, “SWRL: A Semantic Web Rule Language,” Accessed: Aug. 3 2022. [Online]. Available: <https://www.w3.org/Submission/SWRL/>  
 [10] C. Schlenhoff *et al.*, “An IEEE standard Ontology for Robotics and Automation,” in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2012.  
 [11] M. Beetz, D. Bessler, A. Haidu, M. Pomarlan, A. K. Bozcuoglu, and G. Bartels, “Know Rob 2.0 — A 2nd Generation Knowledge Processing Framework for Cognition-Enabled Robotic Agents,” in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 2018.  
 [12] S. Lemaignan, R. Ros, L. Mösenlechner, R. Alami, and M. Beetz, “ORO, a knowledge management platform for cognitive architectures in robotics,” in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2010.

- [13] B. Bruno *et al.*, "Paving the way for culturally competent robots: A position paper," in *2017 26th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, 2017.
- [14] C. Hildebrandt *et al.*, "Ontology Building for Cyber-Physical Systems: Application in the Manufacturing Domain," *IEEE Trans. Automat. Sci. Eng.*, vol. 17, no. 3, pp. 1266–1282, 2020.
- [15] A. Kocher *et al.*, "Automating the Development of Machine Skills and their Semantic Description," in *2020 25th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, Vienna, Austria, 2020, pp. 1013–1018.
- [16] A. Kocher, C. Hildebrandt, L. M. Da Vieira Silva, and A. Fay, "A Formal Capability and Skill Model for Use in Plug and Produce Scenarios," in *2020 25th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, Vienna, Austria, 2020, pp. 1663–1670.
- [17] M. Haage, J. Malec, A. Nilsson, K. Nilsson, and S. Nowaczyk, "Declarative-knowledge-based reconfiguration of automation systems using a blackboard architecture," in *Eleventh Scandinavian Conference on Artificial Intelligence*: IOS Press, 2011, pp. 163–172.
- [18] M. Diab, A. Akbari, M. Ud Din, and J. Rosell, "PMK-A Knowledge Processing Framework for Autonomous Robotics Perception and Manipulation," *Sensors*, vol. 19, no. 5, p. 1166, 2019.
- [19] Object Management Group (OMG), "OMG Systems Modeling Language (OMG SysML™)," 2019. Accessed: Aug. 3 2022. [Online]. Available: <https://www.omg.org/spec/SysML/1.6/PDF>
- [20] K. Mykoniatas, A. Angelopoulou, A. S. Akbas, and P. A. Hancock, "Multi-method modeling and simulation of a face detection robotic system," in *2016 Annual IEEE Systems Conference (SysCon)*, 2016.
- [21] H. Tsuji *et al.*, "Reusable robot system for display and disposal tasks at convenience stores based on a SysML model and RT Middleware," *Advanced Robotics*, vol. 34, 3-4, pp. 250–264, 2020.
- [22] J. Huckaby and H. I. Christensen, "A case for SysML in robotics," in *2014 IEEE International Conference on Automation Science and Engineering (CASE)*, Taipei, 2014, pp. 333–338.
- [23] Rainer Drath, Ed., *AutomationML: A Practical Guide*. Berlin, Boston: De Gruyter Oldenbourg, 2021.
- [24] X.-L. Hoang and A. Fay, "A Capability Model for the Adaptation of Manufacturing Systems," in *2019 24th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, 2019.
- [25] P. Danny, P. Ferreira, N. Lohse, and M. Guedes, "An AutomationML model for plug-and-produce assembly systems," in *2017 IEEE 15th International Conference on Industrial Informatics (INDIN)*, 2017.
- [26] A. Bayha, J. Bock, B. Boss, C. Diedrich, and S. Malakuti, "Describing Capabilities of Industrie 4.0 Components: Joint White Paper between Plattform Industrie 4.0, VDI GMA 7.20, BaSys 4.2," 2020.
- [27] *OPC 10000-1 OPC Unified Architecture Part 1: Overview and Concepts*, OPC Foundation.
- [28] *OPC 10000-5 OPC Unified Architecture Part 5: Information Model*, OPC Foundation.
- [29] *OPC 10000-100 OPC Unified Architecture Part 100: Devices*, OPC Foundation, London.
- [30] *OPC 40010-1 OPC UA for Robotics Part 1: Vertical Integration: Part 1: Vertical Integration*, OPC Foundation.
- [31] *ISO 9409-1 Manipulating industrial robots - Mechanical interfaces - Part 1: Plates*, International Organization for Standardization, Geneva, Switzerland, London.
- [32] *ISO 9409-2 Manipulating industrial robots - Mechanical interfaces - Part 2: Shafts*, International Organization for Standardization, Geneva, Switzerland, London.
- [33] M. M. Alani, "OSI Model," in *Guide to OSI and TCP/IP Models*: Springer, Cham, 2014, pp. 5–17. Accessed: Aug. 28 2022. [Online]. Available: [https://link.springer.com/chapter/10.1007/978-3-319-05152-9\\_2](https://link.springer.com/chapter/10.1007/978-3-319-05152-9_2)
- [34] M. Popp, *Das PROFINET IO-Buch: Grundlagen und Tipps für den erfolgreichen Einsatz*, 2nd ed. Berlin, Offenbach: VDE Verlag, 2010.
- [35] *DIN 8580: Fertigungsverfahren –Begriffe, Einteilung*, DIN-Normenausschuss Technische Grundlagen (NATG), DIN-Normenausschuss Werkstofftechnologie (NWT), 2020.
- [36] *VDI 2860: Montage- und Handhabungstechnik Handhabungsfunktionen, Handhabungseinrichtungen, Begriffe, Definitionen, Symbole*, Verein Deutscher Ingenieure.
- [37] M. Weigand and A. Fay, "Creating Virtual Knowledge Graphs from Software-Internal Data," *IECON 2022 – 48th Annual Conference of the Industrial Electronics Society*.