

# Digitalisierung im Engineering

Ein Ansatz für ein Vorgehensmodell zur durchgehenden, arbeitsteiligen Modellierung am Beispiel von AutomationML

Eike Schäffer, Universität Erlangen-Nürnberg, Lars Penczek, Universität Bochum, Andreas Mayr, Jupiter Bakakeu, Universität Erlangen-Nürnberg, Jörg Franke, Universität Erlangen-Nürnberg und Bernd Kuhlenkötter, Universität Bochum

Die Digitalisierung im Engineering verspricht automatisierte Arbeitsabläufe, höhere Geschwindigkeiten und sinkende Kosten bei der Entwicklung von Automatisierungslösungen. Voraussetzung hierfür ist nicht nur die Modularisierung auf Basis einer strukturierten Beschreibungssprache, sondern auch eine einheitliche, aufeinander aufbauende Modellierung, welche einen automatisierbaren Datenaustausch über die Systemgrenzen hinweg ermöglicht. Um eine breite Anwendung zu erzielen, sollte die zugrundeliegende Ontologie auf bestehenden Normen und Standards aufbauen und in Open-Source-Anwendungen zur Verfügung stehen. Für die kollaborative und konsistente Entwicklung einer solchen Ontologie bedarf es eines strukturierten, methodischen Vorgehens sowie einer damit verbundenen Modellierungslandkarte, welche als Orientierung zur standardisierten, arbeitsteiligen Modellierung dient. Ein möglicher Ansatz für das benötigte Vorgehensmodell sowie der zugehörigen Landkarte wird im Rahmen dieses Beitrags vorgestellt und unter Verwendung von AutomationML validiert. Der vorgestellte Ansatz soll eine mögliche Richtung aufzeigen und weitere prozessgesteuerte Modellierungsbestrebungen von Ontologien anregen.

Die Herausforderungen bei der Einführung von roboterbasierten Automatisierungslösungen resultieren zum einen aus deren hohen Komplexität, zum anderen aus den steigenden Anforderungen an die Verfügbarkeit, Qualität und Sicherheit von industriellen Anwendungen. Eine erfolgreiche Realisierung von Roboteranwendungen erfordert bisher ein tiefes Fach- und umfangreiches Erfahrungswissen. Dies stellt insbesondere für kleine und mittlere Unternehmen (KMU) eine große Hürde dar. Aufgrund der großen Engineering-Aufwände und der hohen Kosten für Systemintegratoren können aktuell viele roboterbasierte Automatisierungslösungen nur bedingt wirtschaftlich dargestellt werden. [1]

Der Lösungsansatz des vom Bundesministerium für Wirtschaft und Energie (BMWi) geförderten Forschungsprojekts ROBOTOP besteht darin, bewährte Automatisierungslösungen, sogenannte Best Practices, digital und wiederverwendbar zur Verfügung zu stellen. Mithilfe eines nutzerzentrierten, webbasierten 3D-Konfigurators wird der Anwender, ausgehend von dessen Anforderungen, durch die Konfiguration einer roboterbasierter Automatisierungslösung geführt. Über eine Hintergrundlogik werden sämtliche benötigten technischen Komponenten ausgewählt. Diese können dann von Systemintegratoren oder Endnutzern in einer standardisierten Form, z. B. dem

neutralen, XML-basierten Datenformat AutomationML (AML), heruntergeladen und in andere Softwaretools zur weiteren Verwendung eingebunden werden. Somit können Engineering-Aufwände reduziert und die Kosten für roboterbasierte Automatisierungslösungen gesenkt werden. [2, 3]

Um dieses Ziel zu erreichen, müssen vorhandene Engineering-Daten aufbereitet und modularisiert werden. Hierfür sind zunächst bestehende Automatisierungslösungen, die Best Practices, zu identifizieren und standardisiert zu beschreiben. Voraussetzung hierfür ist eine klare, systematische Modellierung auf Basis einer einheitlichen Semantik. Ontologien bilden die Basis für die Formalisierung der Semantik und lassen sich gemeinhin als eine explizite, formalsprachliche Spezifikation einer Konzeptualisierung (Begriffsbildung) auffassen [4]. Dadurch ist es möglich, komplexe Zusammenhänge in maschinenlesbarer und -interpretierbarer Form zu beschreiben. Erst dann sind eine einfache Strukturierung, Standardisierung und somit Wiederverwendung von Teillösungen und damit eine skalierbare Konfiguration von neuen Automatisierungslösungen realisierbar. Einzelne Firmen oder Abteilungen

## Digitization in Engineering - A Procedure for the Continuous, Work-sharing Modelling Using the Example of AutomationML

Digitization in engineering promises automated workflows, higher speed and lower costs in the development of automation solutions. The prerequisite for this is not only modularization based on a structured description language, but also uniform, interdependent modeling that ensures automated data exchange across system boundaries. In order to achieve a broad application, the underlying ontology should be based on existing norms and standards and be available in open source applications. However, the collaborative and consistent development of such an ontology requires a structured, methodical procedure and an associated modelling map that serves as an orientation for standardized, work-sharing modelling. A possible approach for the required procedure model and the related map will be presented in this article and validated using AutomationML. The presented approach should point out a possible direction and stimulate further process-controlled modelling efforts of ontologies.

### Keywords:

continuous engineering, AutomationML, modularization, standardization, machine interpretable, modelling methods

Eike Schäffer, M. Sc., M. Sc. arbeitet als wissenschaftlicher Mitarbeiter am Lehrstuhl für Fertigungsautomatisierung und Produktionssystematik der Friedrich-Alexander-Universität Erlangen-Nürnberg.

Lars Niklas Penczek, M. Sc. arbeitet als wissenschaftlicher Mitarbeiter am Lehrstuhl für Produktionssysteme der Ruhr-Universität Bochum.

Andreas Mayr, M. Sc., M. Sc. arbeitet als wissenschaftlicher Mitarbeiter am Lehrstuhl für Fertigungsautomatisierung und Produktionssystematik der Friedrich-Alexander-Universität Erlangen-Nürnberg.

Jupiter Bakakeu, M. Sc. arbeitet als wissenschaftlicher Mitarbeiter am Lehrstuhl für Fertigungsautomatisierung und Produktionssystematik der Friedrich-Alexander-Universität Erlangen-Nürnberg.

Prof. Dr.-Ing. Jörg Franke leitet den Lehrstuhl für Fertigungsautomatisierung und Produktionssystematik der Friedrich-Alexander-Universität Erlangen-Nürnberg.

Prof. Dr.-Ing. Bernd Kuhlenkötter leitet den Lehrstuhl für Produktionssysteme der Ruhr-Universität Bochum.

Eike.Schaeffer@faps.fau.de  
www.faps.fau.de

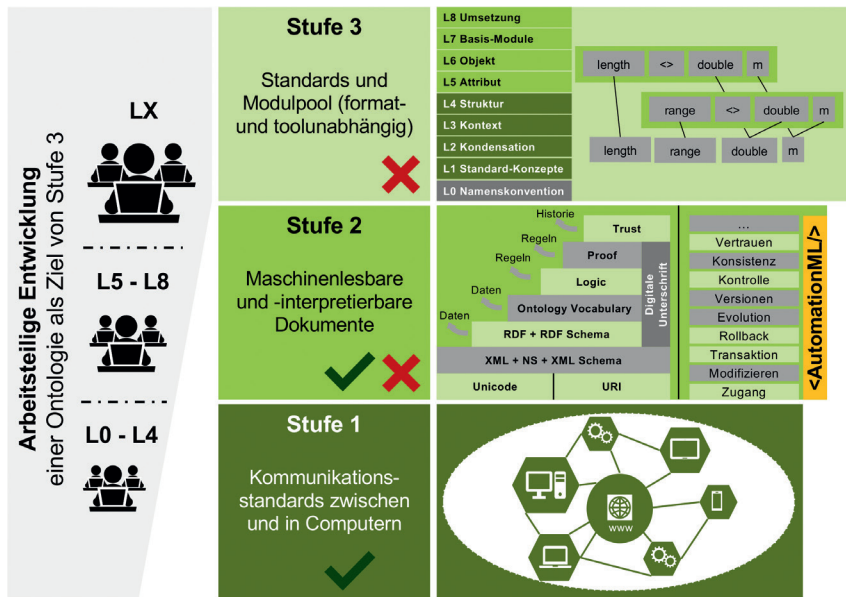


Bild 1: Die drei Stufen hin zur durchgehenden, arbeitsteiligen Modellierung mit Stufe 3 als Handlungsbedarf.

verfügen bereits über Inzellösungen, z. B. auf Basis von Excel-Tools oder PLM-Systemen, doch existiert für die Modellierung bislang keine herstellerübergreifende maschinenlesbare Basis-Ontologie. Dies macht den Austausch über die Systemgrenzen hinweg sehr aufwändig, da zwischen allen Tools Übersetzer geschrieben werden müssen. [5, 6]

### Herausforderungen eines durchgängigen Engineerings

Neben der horizontalen und vertikalen Integration stellt das durchgängige Engineering eine der wesentlichen Charakteristika von Industrie 4.0 dar. Durchgängiges Engineering zeichnet sich dadurch aus, dass das Erzeugnis eines Engineering-Arbeitsschrittes möglichst verlustfrei und mit geringem Aufwand wiederverwendbar ist. Für die weitere Verwendung sollen möglichst wenig redundante Arbeitsschritte notwendig sein. [7]

Voraussetzung hierfür sind die Nutzung eines wertschöpfungsübergreifenden Informationsmodells, die Erstellung von Werkzeugketten, die Verwendung eines gemeinsamen Vorgehensmodells sowie die Verwendung einer einheitlichen Syntax und Semantik [7]. Aufgrund der allgemein hohen Anforderungen in der Anlagenplanung, herstellereinspezifischer Interessen und der bisweilen eingeschränkten Unterstützung von Seiten des Managements wurden solche Ansätze bisher jedoch wenig verfolgt [8].

Der Datenaustausch erfolgt derzeit über diverse Datenformate, wie beispielsweise CAX-Daten zur Beschreibung von Geometrien sowie .pdf-, .docx- oder .pptx-Dateien zur technischen Beschreibung und Visualisierung. Je nach Branche

und Anwendungsfall werden unterschiedliche Datenformate verwendet, wobei gerade die Letztgenannten primär der visuellen Aufbereitung von Inhalten für Menschen dienen. Diese Inhalte können jedoch nur bedingt von Programmen interpretiert werden, wodurch der automatisierte Datenaustausch erschwert wird. Weitere Probleme ergeben sich bei der nicht einheitlich verwendeten Semantik und Struktur innerhalb dieser Dokumente. Die Bezeichnung bzw. Namensgebung von und innerhalb der Dokumente wird häufig der Kreativität des Einzelnen überlassen, was die Lesbarkeit für fremde Anwender erheblich erschwert. Daher müssen Mechanismen für eine einheitliche Modellierung geschaffen werden.

In der Forschung existieren bereits vereinzelt Ansätze, welche zeigen, dass die Modell-zu-Modell-Transformation im Engineering unter Einsatz einer formalen Semantik, die das Wissen auf Basis maschinell interpretierbarer Ontologien abbildet, exemplarisch funktioniert [9]. Anhand von Domänen-Ontologien konnten beispielsweise Modelle semantisch verarbeitet und daraus ein Informationsmodell zur Codegenerierung für Robotersysteme erstellt werden [6]. Die Herausforderung der arbeitsteiligen und einheitlichen Erstellung von Ontologien wurden jedoch kaum betrachtet.

### Standards für maschinenlesbare und -interpretierbare Dokumente

Um Dokumente automatisiert verarbeiten zu können, bedarf es standardisierter, maschinenlesbarer und -interpretierbarer Datenformate wie z. B. die Extensible Markup Language (XML) oder die Engineering-spezifische Erweiterung AML. Hieraus lassen sich dann wiederum die für den Menschen verständlichen Darstellungsformen, z. B. mittels Templates, ableiten. Um die erweiterte Funktionalität von solchen Formaten aufzuzeigen, werden exemplarisch einige wesentliche Standards des World Wide Web Consortiums (W3C) vorgestellt. Die Struktur dieser Elemente lässt sich dem sogenannten Semantic Web Layer Cake [10] entnehmen.

**Namenshierarchien und Tags:** Die Auszeichnungssprache XML ist konform mit der Standard Generalized Markup Language (SGML; ISO 8879:1986) und wird vom W3C empfohlen. Ursprünglich wurde XML als Datenaustauschformat für das Internet entwickelt, wird aber mittlerweile als universelles und software-neutrales Datenformat verwendet. In XML lassen sich unter anderem hierarchische Namensstrukturen anlegen, diese klassifizieren und mit Inhalt versehen [11].

**Beziehungen:** Ein weiterer wichtiger Aspekt bei der Modellierung sind Beziehungen zwischen Elementen. Das Resource Description Framework (RDF) ist ein Datenmodell, welches sich aus sogenannten Tripeln, bestehend aus Subjekt (S), Prädikat (P) und Objekt (O), zusammensetzt. RDF ermöglicht dadurch die explizite Modellierung von Beziehungen, z. B. „KR 500 (S) ist ein (P) Roboter (O)“. Die Menge an Tripeln bildet einen Graphen, worüber RDF auch für den Menschen intuitiv verständlich dargestellt werden kann. Zur Interpretation von in RDF formulierten Aussagen bedarf es eines RDF-Schemas (RDFS), welches ein gemeinsames Vokabular zur Verfügung stellt. Mittels RDFS lassen sich bereits einfache Ontologien formalisieren. [12]

**Logische Bewertbarkeit und Konsistenzprüfung:** Die Web Ontology Language (OWL) ermöglicht die Angabe zusätzlicher Einschränkungen, wie z. B. Kardinalität, Werteinschränkungen oder Eigenschaften wie Transitivität. OWL basiert auf einer Beschreibungslogik und bringt somit Argumentationskraft und logische Bewertbarkeit in semantische Modelle ein. Mittels OWL ist es möglich mehrere zueinander konsistente Dokumente oder Tools abzuleiten, wie z. B. Excel-Arbeitsmappen oder Web-Front-Ends. [13]

### AutomationML als Datenaustauschformat für das Engineering

Um im Engineering-Prozess den Datenaustausch der vielen unterschiedlichen Tools zu vereinheitlichen wurde 2006 das neutrale, XML-basierte Datenformat AML entwickelt [14]. AML speichert technische Informationen nach dem objektorientierten Paradigma und ermöglicht die Modellierung von realen Anlagenkomponenten als Datenobjekte. In den Datenobjekten werden verschiedene Informationen über die Topologie, Geometrie, Kinematik und Logik sowie das Verhalten und die Steuerung von Anlagen berücksichtigt. AML kombiniert bestehende Industriedatenformate, die für die Speicherung und den Austausch verschiedener Aspekte von technischen Informationen ausgelegt sind. Kernstück von AML ist das Datenformat CAEX, das die Datenobjekte sortiert und damit eine hierarchische Struktur aufbaut. Durch diese Struktur hat AML eine inhärent verteilte Dokumentenarchitektur. [8] Die Architektur von AML setzt sich aus den fünf nachfolgenden Bausteinen zusammen:

- *AttributeType*: Definition von allen Attributen
- *RoleClass*: Repräsentation domänenspezifischer Konzepte
- *InterfaceClass*: Abbilden von Schnittstellen zwischen Objekten
- *SystemUnitClass*: Modulkatalog aller Objekte in AML

- *InstanceHierarchy*: Instanziierung von Objekten, welche mittels der anderen Bausteine definiert wurden

Der Vorteil eines einheitlichen Datenmodells liegt u. a. in der systemweiten Portabilität der Informationen und an dem geringen Aufwand zur Einbindung der Informationen aufgrund einer minimalen Anzahl an Konvertierungen [8]. Dabei müssen alle in einem AML-Dokument angelegten Bezeichnungen mit den vorher definierten Komponenten und Attributen übereinstimmen. Vor- und nachgelagerte Engineering-Tools, wie z. B. Datenbanken, CAx-Daten oder Konfigurationsinformationen, sind entsprechend anzupassen. Da dem Anwender in AML die Möglichkeiten zur Strukturierung und Benennung gänzlich offen gehalten sind, ist ein einheitliches Vorgehen bei der Modellierung von besonderer Bedeutung. Der Aufbau der Datenobjekte in AML ist daher immer im ganzheitlichen Kontext festzulegen.

### Die drei Stufen hin zur durchgehenden, arbeitsteiligen Modellierung

Um die Notwendigkeit eines Vorgehensmodells zur arbeitsteiligen Entwicklung einer Ontologie aufzuzeigen und bestehende Basisstandards wie AML einzuordnen, wurde der Weg hin zur durchgehenden, konsistenten Modellierung in drei Stufen unterteilt (siehe Bild 1). Die drei Stufen stellen in abstrahierter Form die historische Entwicklung der technologischen Befähiger dar. Ähnlich einer Treppe können die höheren Stufen erst durch Betreten der unteren Stufen erreicht werden.

**Stufe 1 - Kommunikationsstandards zwischen und in Computern:** Die erste Stufe umfasst die etablierten Kommunikationsstandards zwischen bzw. in Computern. Als Basis für die Datenkommunikation in offenen Systemen hat sich beispielsweise mit der ISO 7498 das OSI-Modell (Open Systems Interconnection) etabliert. Charakteristisch ist dessen hierarchische Strukturierung in mehrere logische Schichten, welche eine arbeitsteilige Entwicklung der jeweiligen Dienste ermöglicht. Das Modell dient bis heute als Grundlage zahlreicher Spezifikationen und hat damit zur Standardisierung sowie arbeitsteiligen Entwicklung von Hardware, Treibern und Protokollen beigetragen.

**Stufe 2 - Maschinenlesbare und -interpretierbare Dokumente:** Die nächste logische Stufe ist die Einführung einer einheitlichen Semantik auf Basis formaler Beschreibungssprachen, um systemneutrale, maschinenlesbare und -interpretierbare Dokumente zu ermöglichen. Zu diesem Zweck wurden bereits viele aufeinander aufbauende



Beschreibungsstandards geschaffen, etwa die des Semantic Web Layer Cakes oder AML (siehe Bild 1, Stufe 2). Die Beschreibungsstandards lassen dem Entwickler jedoch sehr viel Freiheit bei der Modellierung, weshalb ein standardisierter Aufbau von Modellen allein damit nur bedingt möglich bzw. gewährleistet wird.

**Stufe 3 - Standards und Modulpool (format- und toolunabhängig):** Um Modelle arbeitsteilig und konsistent aufzubauen, bedarf es branchenübergreifender Standards sowie eines einheitlichen Vorgehensmodells. Damit muss es möglich sein, alles, von der kleinsten SI-Einheit bis hin zur komplexen Roboterzelle, in einer konsistenten Abfolge zu beschreiben. Das erforderliche Vorgehensmodell unterteilt sich wiederum in mehrere Ebenen bzw. Layer (L0 – LX), die miteinander verknüpft sind. Da die dritte Stufe als solches den Handlungsbedarf darstellt, wird das zugrundeliegende Vorgehensmodell sowie die zugehörige Modellierungslandkarte nachfolgend im Detail erläutert.

Vorgehensmodell für die einheitliche, arbeitsteilige Modellierung

Wie bereits erwähnt lassen die Beschreibungsstandards dem Entwickler sehr viel Freiheit, wodurch bei der Modellierung ein und desselben Szenarios durch verschiedene Personen unterschiedliche Ontologien entstehen. Um die Modellierung zu vereinheitlichen, wird nachfolgend ein Ansatz für ein mögliches, strukturiertes Vorgehensmodell auf Basis einer eigens erstellten Modellierungslandkarte (Bild 2) beschrieben. Mit Ausnahme der Namenskonvention und des Kontexts werden immer nur die Elemente aus den darunterliegenden Ebenen bzw. Layern verwendet:

**Festlegen eines konkreten Use Cases:** Ausgehend von einem konkreten Use Case werden die Anforderungen abgeleitet und festgelegt,

welche Elemente durchgängig beschrieben werden sollen.

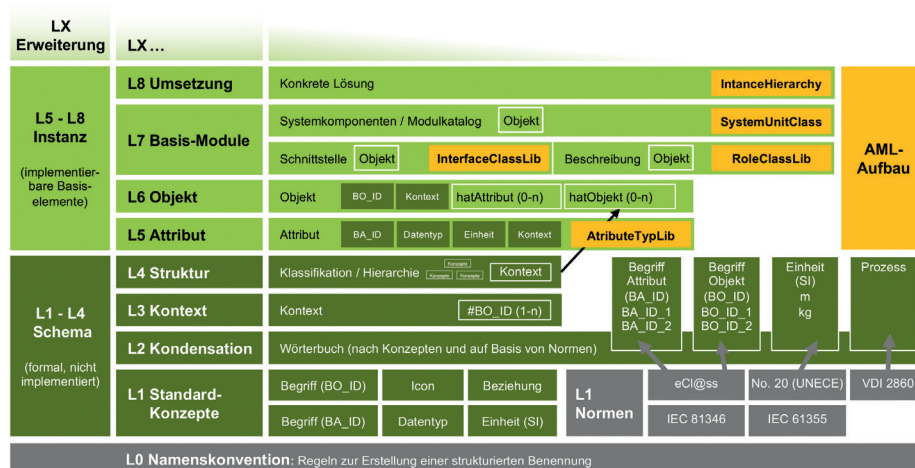
**Festlegung einer Namenskonvention (L0):** Die Grundlage aller Modellierungsebenen stellt eine einheitliche Namenskonvention dar. Diese legt unter anderem fest, in welcher Sprache (z. B. deutsch, englisch), welches ID-Schema, in welcher Schreibweise (z. B. groß, klein), mit welchem Zeichensatz (z. B. ASCII) und unter Berücksichtigung welcher weiterer Regeln Begriffe angelegt werden dürfen.

**Sammeln von Standard-Konzepten und Normen (L1):** Um personen- bzw. unternehmensneutrale Begrifflichkeiten festzulegen, werden Modellierungsstandards gesammelt. Hierin steckt das kondensierte Wissen von Domänenexperten und -branchen. Standards wie z. B. No. 20 (UNECE), VDI 2860 oder IEC 81346 dienen als Referenz und zum Nachweis einer standardkonformen Modellierung. Der Verweis auf diese sollte in der nächst höheren Ebene im Wörterbuch hinterlegt werden, um die Qualität der Modelle auch im Nachgang nachvollziehen zu können. Generell wird die Verwendung des eCI@ss-Standards empfohlen.

**Verdichten und Aufbereiten der Standard-Konzepte und Normen (L2):** Die für den Modellierungsprozess möglichen Attribute, Begriffe für Objekte, Datentypen und SI-Einheiten werden in Form maschinenlesbarer Wörterbüchern kondensiert. In einzelnen Sub-Wörterbücher können dann bestimmte Kategorien bzw. Konzepte gesammelt werden. Übersetzungen von globalen Standards in Unternehmensstandards, fachdisziplin-spezifische Namensgebungen oder verschiedenen Sprachen können hier ebenso hinterlegt werden.

**Anlegen des Kontextes (L3):** Ein Kontext ermöglicht es, Attribute, Objekte oder Klassifikationen später je nach Benutzerrolle zu filtern. Insbesondere die Klassifikation ist sehr stark vom Kontext bzw. der jeweiligen Sichtweise abhängig. So kann die Klassifikation von ein und denselben Objekten aus Sicht des Marketings, der Entwicklung oder der Instandhaltung sehr unterschiedlich sein, da für jede Benutzerrolle unterschiedliche Informationen wichtiger bzw. überhaupt relevant sind. Um das Dilemma zu vermeiden, vorab alle möglichen Kontexte bzw. Nutzerrollen kennen zu müssen, kann der Kontext mittels Hashtags dynamisch erweitert werden. So wird im Zweifelsfall einfach diejenige Klassifikation ausgewählt, welche hinsichtlich der Hashtags die höchste Übereinstimmung hat.

Bild 2: Modellierungslandkarte (Stufe 3) als Orientierung für die standardisierte, arbeitsteilige Modellierung von Ontologien.



**Bestimmung von Klassifikationen zur Strukturierung (L4):** Um Struktur zu schaffen, können Attribute oder andere höhere Konzepte wie z. B. Objekte klassifiziert werden, z. B. in Ober- und Unterbegriffe. Hierbei dürfen nur Begriffe aus den Wörterbüchern verwendet werden. Da Klassifikationen kontextabhängig sind, ist der Kontext in Form von Hashtags aus dem Objekt-pool, z. B. #Maschinenbau, #Robotik, #Grobplanung, mit anzugeben. Der Kontext kann durch eine beliebige Anzahl an Hashtags konkretisiert werden und dient später der Auswahl geeigneter Klassifikationen sowie der Bewertung von deren Relevanz für eine bestimmte Aufgabe. Das Vorgehen ermöglicht es, beliebig viele Klassifikationen anzulegen bzw. zu verwenden. Hierdurch ist es unter anderem möglich, Front-Ends für verschiedenste Benutzerrollen automatisiert auszuleiten, wobei jeder Benutzerrolle nur die relevantesten Informationen angezeigt werden.

**Anlegen von Attributen (L5):** Attribute setzen sich aus der Konstellation „Begriffsattribut (z. B. Gewicht) + Datentyp (z. B. double) + SI-Einheit (z. B. kg)“ zusammen. Die jeweiligen Elemente sind dem darunterliegenden Wörterbuch zu entnehmen. Sollten Elemente fehlen müssen diese von Ebene L1 ausgehend neu angelegt werden. Diese Vorgehensweise stellt sicher, dass für gleiche Attribute wie z. B. Gewicht bei jedem höheren Element exakt dieselbe Begrifflichkeit sowie der gleiche Datentyp und die gleiche SI-Einheit verwendet werden. Die Attribute können damit eindeutig anhand einer ID interpretiert und müssen nicht unbenannt werden, z. B. von „G“ in „Gew“. Somit müssen sich Modellierer auf einer höheren Ebene keine Gedanken über das Anlegen von standardisierten Attributen bzw. die Wahl der Begrifflichkeiten machen.

**Anlegen von Objekten (L6):** Durch die Aggregation von Attributen (0-n) und Objekten (0-n) werden neue Objekte angelegt. Um Objekte eindeutig zuzuordnen zu können, wird diesen analog zu den Klassifikationen und Attributen ein Kontext zugeordnet. Objekte verfügen über verschiedene Attribute, etwa Gewicht oder Preis, sowie gegebenenfalls auch über andere Objekte. Die Begrifflichkeiten und Attribute dürfen wie zuvor nur den darunterliegenden Elementen entnommen werden.

**Überführung in Basis-Module (L7):** Für spezielle Anwendungsfälle wie z. B. Schnittstellen oder (Roboter-)Komponenten werden Objekte zu konkreteren Modulen bzw. Objekten aggregiert. Hierbei dürfen wieder nur Objekte und Attribute aus den darunterliegenden Ebenen verwendet werden.

**Umsetzung (L8):** Im vorerst letzten Layer werden die bisher abstrakten Lösungen, z. B. Schemata mit abstrakten Robotern, in Form von konkreten Instanzen, z. B. einer realen Roboterzelle inklusive der Seriennummer der Einzelteile, implementiert.

Eine kontinuierliche anwendungsbezogene Weiterentwicklung ist durch zusätzliche Use Cases möglich. Dabei werden die Elemente des neuen Use Cases zunächst gemäß eines Top-Down-Ansatzes anhand von Basis-Modulen erstellt. Falls Basiselemente fehlen, ist das Modell im Sinne eines Bottom-Up-Ansatzes von Ebene L1 ausgehend zu erweitern.

Einmal erstellte Ontologien können gespeichert und in Form von Bibliotheken in verschiedene Modellierungstools importiert werden. Die vorgestellte Vorgehensweise und damit einhergehende Modellierungslandkarte kann also dazu dienen, unternehmensübergreifende Open-Source-Ontologien zu erstellen, welche den Aufbau von semantisch konsistenten Modellen bzw. Bibliotheken ermöglichen.

Validierung mittels AutomationML-Implementierung

Das hier beschriebene Vorgehensmodell für eine standardisierte, arbeitsteilige Modellierung von Ontologien wurde innerhalb des BMWi-Projekts ROBOTOP beim Anlegen einer einheitlichen Beschreibung von Roboterzellen im Datenformat AML validiert. Hierbei diente das in Bild 2 beschriebene Landkartenmodell als Orientierung.

Nach dem Festlegen der Namenskonvention (L0) wurden zunächst die Begrifflichkeiten durch den eCl@ss Standard definiert (L1). Diese Begriffe wurden anschließend in einem Wörterbuch zusammengefasst (L2), welches als Grundlage für alle weiteren Benennungen der Objekte und Attribute im Modell dient. Es folgte das Anlegen von Kontexten (L3) und Klassifikationen (L4).

Die Attribute wurden im Rahmen der Attribute-TypeLib definiert (L5), woraus diese anschließend per Drag-and-Drop bei allen Objekten verwendet werden können. Des Weiteren wurden den Attributen Datentypen, Einheiten und ein Kontext zugeordnet. Die Einheiten wurden in der Ebene L1 gemäß der No. 20 (UNECE) [15] definiert. Die Kontexte wurden bereits in Ebene L3 festgelegt und ermöglichen es, einen semantischen Zusammenhang zwischen den betreffenden Objekten und Attributen herzustellen.

Nach den Attributen wurden auch die Objekte festgelegt (L6). Zu den Objekten zählen alle Kom-

#### Literatur

- [1] Eisele, R.: Konzeption und Wirtschaftlichkeit rechnerintegrierter Planungssysteme, Zugl.: Erlangen-Nürnberg, Univ., Diss., 1990. München 1990.
- [2] Schäffer, E.; Bartelt, M.; Pownuk, T.; Schulz, J.-P.; Kuhlentkötter, B.; Franke, J.: Configurators as the basis for the transfer of knowledge and standardized communication in the context of robotics. In: Procedia CIRP 72 (2018), S. 310–15.
- [3] Schäffer, E.; Pownuk, T.; Walberer, J.; Fischer, A.; Schulz, J.-P.; Kleinschnitz, M.; Bartelt, M.; Kuhlentkötter, B.; Franke, J.: System architecture and conception of a standardized robot configurator based on microservices. In: Schuppstahl, T.; Tracht, K.; Franke, J. (Hrsg.): Tagungsband des 3. Kongresses Montage Handhabung Industrieroboter. Berlin, Heidelberg 2018.
- [4] Gruber, T. R.: A translation approach to portable ontology specifications. In: Knowledge Acquisition 5 (1993) 2, S. 199–220.
- [5] Schäffer, E.; Leibinger, H.; Stamm, A.; Brossog, M.; Franke, J.: Configuration based process and knowledge management by structuring the software landscape of global operating industrial enterprises with Microservices. In: Procedia Manufacturing 24 (2018), S. 86–93.
- [6] Hua, Y.; Mende, M.; Hein, B.: Modulare und wandlungsfähige Robotersysteme. Modellbasierte Softwareentwicklung basierend auf AutomationML und ontologischer Semantik. In: Industrie 4.0 Management 2017 (2017) 33, S. 33–37.

ponente, die auf der ROBTOP-Plattform genutzt werden, und alle Prozesse, die eine Roboterzelle ausführen können. Jedem dieser Objekte können mehrere Attribute, Schnittstellen und Rollen zugewiesen werden.

Die Objekte werden allesamt in der SystemUnitClassLib angelegt, welche quasi einen Modulkatalog aller Objekte in AML repräsentiert (L7). Die übergeordnete Struktur der SystemUnitClassLib, InstanceHierarchy und RoleClassLib wurde entsprechend der Prozess-Produkt-Ressourcen-Methode (PPR-Methode) aufgebaut. Diese Methode wird in der DIN EN 62714-1 [16] beschrieben und ist insbesondere für die Strukturierung komplexer anlagenplanungstechnischer Daten geeignet.

Neben der strukturellen Einordnung der Objekte sind noch die bereits erwähnten Schnittstellen und Rollen zu definieren. Eine Rollenklasse repräsentiert ein domänenspezifisches Konzept und wird in der RolleClassLib definiert. Hierbei wurden auf die in AML vordefinierte AutomationMLBaseRoleClassLib zurückgegriffen. Darüber hinaus wurden weitere spezifische Rollen angelegt.

Um nun die Schnittstellen zwischen den Objekten abbilden zu können, wurden diese in der InterfaceClass definiert. Diese Schnittstellen beschreiben abstrahierte Verbindungen, wie z. B. eine Kommunikationsschnittstelle oder eine Adapterplatte, und wurden nach dem Konzept von Pimpler und Eppinger [17] angelegt. Dieses Konzept beschreibt ein definiertes Schema, bestehend aus den vier Untergruppen „Information“, „Physics“, „Energy“ und „Material“.

Nachdem die Definition der Attribute, Objekte und Schnittstellen in AML abgeschlossen ist, müssen die Objekte noch zu einer Best-Practice-Lösung zusammengefügt werden (L8). Dazu wurden die in der SystemUnitClass definierten Objekte innerhalb der InstanceHierarchy per Drag-and-Drop implementiert. Die Grundstruktur ist dabei an den Aufbau der SystemUnitClass angelehnt und ebenfalls an der PPR-Methode orientiert.

## Zusammenfassung

Um ein durchgängiges und arbeitsteiliges Engineering zu ermöglichen, ist neben einer strukturierten Beschreibungssprache auch eine einheitliche, aufeinander aufbauende Modellierung nötig. Voraussetzung hierfür ist eine einheitliche Semantik auf Basis einer klar definierten Ontologie. Erst dann sind eine einfache Wiederverwen-

dung und damit eine skalierbare Konfiguration von neuen Automatisierungslösungen möglich. Derzeit existiert jedoch keine herstellerübergreifende, maschinenlesbare Basis-Ontologie um Modelle über die Systemgrenzen hinweg verfügbar zu machen.

Aus diesem Grund wurde im Rahmen dieses Beitrags eine Vorgehensweise zur durchgängigen und arbeitsteiligen Entwicklung einer Ontologie beschrieben. Zur Beschreibung dieses Vorgehens wurde eine Modellierungslandkarte vorgestellt, welche anhand von mehreren Ebenen bzw. Layern das Vorgehen einer arbeitsteiligen Entwicklung von konsistenten Ontologien ermöglicht. Nach dem Festlegen des Use Cases werden zunächst eine einheitliche Namenskonvention definiert (L0) sowie relevante Modellierungsstandards gesammelt (L1). Auf Basis der zusammengetragenen Standard-Konzepte und Normen wird anschließend ein Wörterbuch kondensiert (L2). Um eine kontextabhängige Klassifizierung der im Wörterbuch enthaltenen Begriffe zu ermöglichen, werden Tags erstellt (L3) und die Begriffe in eine hierarchische Struktur überführt (L4). Im Anschluss werden Attribute (L5) und Objekte (L6) angelegt. Diese können daraufhin zu Basis-Modulen aggregiert werden (L7), welche wiederum in konkrete Lösungen übergehen können (L8).

Die Validierung des Vorgehensmodells erfolgte anhand der Modellierung verschiedener Roboterzellen auf Basis von AML. Wie zuvor beschrieben wurde bei der Benennung und Strukturierung des Modells von vorhandenen Standards Gebrauch gemacht. Durch die Verwendung der Modellierungslandkarte konnte innerhalb kürzester Zeit eine fundierte, konsistente Ontologie erstellt werden. Das beschriebene Vorgehensmodell stellt somit ein herstellernunabhängiges Vorgehen zur arbeitsteiligen Entwicklung von Ontologien dar und hat durchaus das Potenzial, künftig in ein Standardvorgehen für die Modellierung im Engineering überzugehen.

Schlüsselwörter:

Durchgängiges Engineering, AutomationML, Modularisierung, Standardisierung, maschineninterpretierbar, Modellierungsmethoden

*Das Forschungs- und Entwicklungsprojekt ROBTOP wird vom Bundesministerium für Wirtschaft und Energie (BMWi) gefördert. Es ist Teil des Technologieprogramms „PAICE Digitale Technologien für die Wirtschaft“ und wird vom DLR-Projektträger „Informationstechnologien / Elektromobilität“ aus Köln geleitet.*

- [7] VDI/VDE-Gesellschaft Mess- und Automatisierungstechnik: Statusreport: Durchgängiges Engineering in Industrie 4.0-Wertschöpfungsketten 2016.
- [8] Drath, R.: Datenaustausch in der Anlagenplanung mit AutomationML. Integration von CAEX, PLCopen XML und COLLADA. Berlin Heidelberg 2010.
- [9] Biffi, S.; Sabou, M.: Semantic Web Technologies for Intelligent Engineering Applications. Cham 2016.
- [10] Bratt, S.: Semantic Web, and Other Technologies to Watch. URL: [https://www.w3.org/2007/Talks/0130-sb-w3cTechSemWeb/#\(24\)](https://www.w3.org/2007/Talks/0130-sb-w3cTechSemWeb/#(24)). Abrufdatum 22.11.2018.
- [11] Bray, T.; Paoli, J.; C. M., S.-M.; Maler, E.; Yergeau, F.; Cowan, J.: Extensible Markup Language (XML) - Second Edition. Recommendation 16.08.2006, 2. Auflage 2006.
- [12] Berners-Lee, T.; Hendler, J.; Lassila, O.: The Semantic Web. In: Scientific American (2001) 284, S. 34-43.
- [13] W3C OWL Working Group: OWL 2 Web Ontology Language. Document Overview (Second Edition) 2012.
- [14] AutomationML e. V.: Standardized data exchange in the engineering process of production systems. URL: <https://www.automationml.org/o.red/uploads/dateien/1544706233-automationml.pdf>. Abrufdatum 22.11.2018.
- [15] United Nations Economic Commission for Europe (UNECE): Recommendation No. 20. Codes for Units of Measure used in International Trade 2009.
- [16] Deutsche Institut für Normung e. V. (DIN): Datenaustauschformat für Planungsdaten industrieller Automatisierungssysteme - Automation markup language. Teil 1: Architektur und allgemeine Festlegungen DIN EN 62714-2:2015-11.
- [17] Pimpler, T. U.; Eppinger, S. D.: Integration analysis of product decompositions. In: American Society of Mechanical Engineers, Design Engineering Division (Publication) DE 68 (1994), S. 343-51.