

Syntax, Semantik und Informationsmodelle

DR. MARKUS RICKERT, FORTISS, LANDESFORSCHUNGSINSTITUT DES FREISTAATS BAYERN

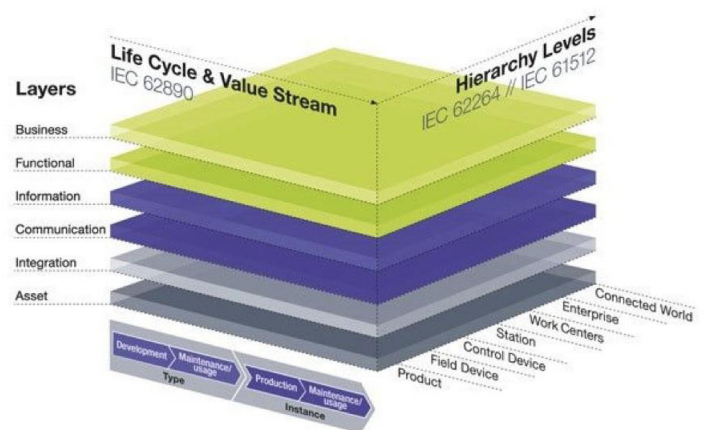
Dem Informationsmanagement [1] und durchgehenden Zugriff auf alle für einen Prozess relevanten Informationen kommt im Zeitalter von Industrie 4.0 eine entscheidende Rolle zu. Hierbei sind Informationen aus unterschiedlichen Unternehmensbereichen und Technologie-domänen relevant, von den Geschäftsmodellen bis hin zu den aktuellen Sensordaten. Sämtliche dieser Informationen müssen dabei in digitaler Form vorhanden und für alle relevanten Systemkomponenten zugänglich sein, um einen entsprechenden Mehrwert generieren zu können. Zusätzlich muss eine ausreichende Datenqualität sichergestellt sein. Grundlegende Ansätze für eine Ablage von Daten reichen von einfachen Datenbanksystemen bis hin zum Data Warehouse [2]. Im Zeitalter echtzeitfähiger Prozesse und Analysen kommt dabei Systemen, die diese Art von Daten direkt im Speicher verfügbar halten, eine große Bedeutung zu [3]. Die Integration von verschiedenen Informationen erfordert dabei Modelle, die über eine reine Datenablage hinausgehen und semantische Verbindungen zwischen diesen Informationen erlauben.

Produktlebenszyklusmanagement (PLM) ist ein systematisches Konzept für die Verwaltung und Entwicklung von Produkten und produktrelevanten Informationen. Dies beinhaltet sowohl den Produktprozess selbst, von der Produktentwicklung bis hin zu Weiterentwicklung und Marketing, als auch den Kundenprozess inklusive Vertrieb, Beschaffung, Herstellung, Auslieferung, Service und Wartung. Alle dabei entstehenden Informationen sollten digital verfügbar sein, unabhängig von Zeit, Ort oder Dateneigentum [4].

Im Kontext PLM beziehen sich Produktdaten auf folgende Informationen, die in drei grobe Kategorien eingeteilt werden können. In den Definitionsdaten sind die physikalischen und funktionalen Produkteigenschaften festgelegt. Die Lebenszyklusdaten beinhalten die aktuelle Phase des Produkts oder des Bestell- bzw. Auslieferungsprozesses. In den Metadaten sind darüber-liegende Informationen wie der aktuelle Aufenthaltsort des Produkts oder der Ablageort seiner Daten hinterlegt [4]. Als Grundlage zur Identifikation, Kodierung oder Benennung von Produkten, Produktelementen, Modulen, Komponenten, Materialien, Dokumenten oder anderen Dingen werden abstrakte Objekte verwendet. Diese Objekte können innerhalb von Klassen logisch strukturiert werden, z. B. in Grafik- oder CAD-Daten [4].

Die Zusammenhänge zwischen den verschiedenen Informationseinheiten im Unternehmen werden in Informationsmodellen beschrieben. Das Produktinformationsmodell beschreibt die für das Produkt relevanten Information und deren Zusammenhänge auf einer konzeptuellen Ebene, wie den

Aufbau eines Produkts aus einzelnen Teilmodulen oder deren Inkompatibilität. Das zugrundeliegende Produktmodell enthält Informationen zu einem spezifischen Produkt, das in verschiedenen Varianten vorliegen kann [4].



Referenzarchitekturmodell Industrie 4.0 (RAMI 4.0)

Quelle: Plattform Industrie 4.0

Das Referenzarchitekturmodell Industrie 4.0 (RAMI 4.0) stellt ein Gerüst zur Abbildung wesentlicher Aspekte im Kontext von Industrie 4.0 dar. Es beinhaltet drei verschiedene Sichtweisen, um komplexe Zusammenhänge der Industrieautomation überschaubar zu strukturieren [5].

Auf der ersten Achse wird eine Hierarchie der Fabrik gemäß IEC 62264 (Enterprise-Control System Integration) und IEC 61512 (Batch Control) von Produkt zu vernetzter Welt definiert. Hierbei wird im Kontext von Industrie 4.0 das Produkt nicht länger als isoliert, sondern als Teil des Netzwerks aus smarter Fabrik, vernetzter Welt und smarter Produkte angesehen. Die Kommunikation ist nun sehr flexibel und dynamisch zwischen allen Systemen und Maschinen und nicht länger strikt hierarchisch zwischen den klassischen Ebenen der Automatisierungspyramide. Funktionen sind damit nicht länger an einzelne Hardwarekomponenten gebunden, sondern können beliebig in diesem Netzwerk aus Hardware- und Softwarekomponenten verteilt werden [5].

Der Produktlebenszyklus auf Basis von IEC 62890 (Lifecycle-Management von Systemen und Produkten der Mess-, Steuer- und Regelungstechnik der Industrie) wird auf der zweiten Achse dargestellt. Dies beinhaltet sowohl die Entwicklungsphase eines Produkttypen inklusive Konstruktion, Simulation,

Prototyp-entwicklung, Benutzerhandbuch etc. als auch die Produktionsphase mit der eigentlichen Produktinstanz, seiner Seriennummer, den Daten, seiner Wartung bis hin zum abschließenden Recycling [5].

Die dritte Achse beschreibt die IT-Architektur. Die unterste Schicht stellen Gegenstände der realen Welt wie beispielsweise Aktuatoren, Sensoren, Bauteile oder Dokumente dar. Darüber liegt eine integrierende Schnittstelle für die entsprechende Darstellung dieser Gegenstände in der IT-Infrastruktur inklusive Mensch-Maschine-Schnittstellen. Die Kommunikationsschicht dient der vereinheitlichten Verbindung zur darüber liegenden Informationsschicht. Die Informationsschicht hat das Ziel, höherwertige Daten zu gewinnen, dabei werden unterschiedliche Daten integriert und Verarbeitungsregeln formal beschrieben. In der nächsten, funktionalen Schicht werden Funktionen formal beschrieben und horizontal integriert zur Absicherung der Integrität von Prozessinformationen und -zuständen. Die oberste Geschäftsschicht bildet Geschäftsmodelle ab und dient der Modellierung von Systemregeln auf Basis von Diensten aus der Funktionsschicht [5].

Im Kontext von Industrie 4.0 wird zudem das Konzept einer Verwaltungsschale definiert (angelsächsisch: Asset Administration Shell, AAS). Ein Gegenstand wird erst mit dieser zu einer Industrie 4.0-Komponente und in die Kommunikation mit anderen I4.0-Komponenten über eine eindeutige ID eingebunden. Gegenstände beinhalten dabei nicht nur Maschinen, sondern z. B. auch Produkte, CAD-Daten, Handbücher, Verträge oder Bestellungen. Verwaltungsschalen können sich dabei auf Maschinen und deren zugehörigen Steuerungen befinden oder z. B. bei Dokumenten über Cloud-Systeme abgebildet werden. Eine I4.0-Komponente kann mehrere Gegenstände abdecken, z. B. innerhalb eines Antriebssystems. Zudem können mehrere I4.0-Komponenten einander logisch zugeordnet werden, etwa alle I4.0-Komponenten innerhalb einer Maschine [5,13,14].

Ein Verwaltungsschale besteht hierbei aus einem Header mit allen Angaben zur Identifikation von sich selbst und den darin enthaltenen Gegenständen. Im zugehörigen Body sind die konkreten Informationen über die entsprechenden Gegenstände in verschiedenen Teilmodellen abgelegt, z. B. für „Auftragsmanagement“ mit Elementen wie Auftrags- und Positionsnummer oder „Achsteuerung“ inkl. der aktueller Position und Geschwindigkeit. Teilmodelle sind dabei auf Basis existierender Standards wie z. B. ecl@ss für Produktklassen oder IEC 61987 zur Definition von Datenstrukturen und -elementen formuliert, bei denen die Umsetzung einzelner Aspekte in die vier Merkmal-Klassen Basis, Pflicht, Optional und Frei aufgeteilt ist. Die beiden Klassen Basis und Pflicht dienen dabei der Gewährleistung einer Interoperabilität unter unterschiedlichen Herstellern [13,14].

Ein offener Standard für den Austausch von Konstruktionsdaten, die im Kontext der Entwicklung von Produktionsanlagen entstehen, ist AutomationML. Grundlage hierfür ist ein XML-basiertes Daten-format, das den verlustfreien Austausch dieser Daten zwischen den Programmen einer Werkzeugkette ermöglicht. Abzudecken sind hierbei die fünf Phasen Analyse, grundlegende Planung, detaillierte Planung, Systemintegration und Inbetriebnahme. Dies bedingt die Abdeckung der Topolo-

gie des Produktionssystems von Betriebsebene bis hin zu Geräten und mechanischen Bauteilen, mechanischen Daten wie Geometrien, Kinematiken oder Materialeigenschaften, sowie elektrischer, pneumatischer und hydraulischer Daten wie elektronischen Zeichnungen oder Rohrleitungsplänen [6].

AutomationML baut auf einer Reihe bestehender XML-basierter Datenformate wie z. B. CAEX für topologische Beschreibungen, COLLADA für Geometrie- und Kinematikbeschreibungen oder PLCopen XML für steuerrelevante Logik auf. CAEX dient auf oberster Ebene der objektorientierten Beschreibung von semantischen Rollen einzelner Objekte in einer Klassenhierarchie, z. B. ein mechanisches Bauteil, ein physikalisches Gerät oder ein logisches Netzwerk. Schnittstellenklassen bieten eine weitere Art der Modellierung an, um abstrakte Verbindungen zu anderen Elementen oder Informationen über Modellgrenzen hinweg auszudrücken. Systemeinklassensystemklassen als dritte Modellierungsvariante stellen wiederverwendbare Systemkomponenten dar, die z. B. als Vorlage zur Strukturierung domänenspezifischer Modelle verwendet werden können [6].

Die flexible Kommunikation in verteilten Systemarchitekturen ist im Kontext von Industrie 4.0 von entscheidender Bedeutung. Zentral ist hierbei die offene OPC Unified Architecture (OPC UA). Im Gegensatz zum klassischen OPC hat OPC UA neben dem reinen Transport von Daten auch die Modellierung von Informationen zum Ziel. Um eine höherwertige semantische Darstellung von Daten zu ermöglichen, die über reine Namensgebung und Angabe von Maßeinheiten hinausgeht, können weitere semantische Informationen wie z. B. die mit den Daten verbundene Gerätekategorie dargestellt werden. Da eine Modellierung von Informationen ohne abgestimmte Strukturen zu inkompatiblen und herstellerspezifischen Lösungen führen würde, können entsprechend abgestimmte Spezifikationen für diese Modelle definiert werden. Diese können im Anschluss von einzelnen Herstellern spezifisch für ihre Geräte erweitert werden [7].

Als Grundlage hierfür dienen u. a. Prinzipien der objektorientierten Modellierung mit Klassenhierarchien und Vererbung. Informationen zu den Typen einzelner Daten sind jederzeit abrufbar. Verknüpfungen von verschiedenen Klassenhierarchien über einzelne Knoten sind möglich, um gleiche Informationen auf unterschiedliche Arten zu repräsentieren. Klassenhierarchien und Verknüpfungsarten können dabei jederzeit erweitert werden, ebenso sind keine Einschränkungen bezüglich der Art der Modellierung vorgegeben. Das Modell wird dabei stets auf dem OPC UA Server vorgehalten.

Eine Vielzahl an relevanten Spezifikationen im Kontext von Industrie 4.0 existieren bereits oder befinden sich derzeit in Entwicklung. Das OPC-UA-Informationsmodell für AutomationML ermöglicht den Austausch von AutomationML-Modellen über OPC UA. Auf der anderen Seite kann die OPC-UA-Systemkonfiguration dadurch als AutomationML-Modell hinterlegt werden [8]. Der VDMA entwickelt derzeit eine Reihe von sogenannten OPC UA Companion Specifications für Bereiche wie Robotik, industrielle Bildverarbeitung, oder elektrische Antriebe.

Einen umfassenden Ansatz zur strukturierten Ablage von Wissen stellt das Semantic Web dar. Hinter diesem steht der Anspruch, das World Wide Web von einem textbasierten und nur für Menschen lesbaren Web zu einer strukturierten und für

Computer verständlichen Variante weiterzuentwickeln. Relevante Inhalte in Bezug auf eine Wissensfrage müssen in diesem Ansatz nicht per Sprachverarbeitung aus mehreren unstrukturierten Texten extrahiert, semantisch aufbereitet und verknüpft werden. Stattdessen sind Daten unterschiedlicher Domänen semantisch modelliert und logisch miteinander verknüpft [9].

Grundlage hierfür ist eine graphbasierte Modellierung aus Knoten und Kanten, die im Resource Description Framework (RDF) standardisiert ist. Einzelne Elemente werden hierbei über einen Universal Resource Identifier (URI) eindeutig identifiziert und in Form von Tripeln aus Subjekt, Prädikat und Objekt miteinander verknüpft. Hiermit kann beispielsweise folgendes modelliert werden: Bauteil X hat Seriennummer Y. Diese werden in speziell dafür angepassten Datenbankstrukturen abgelegt. Das RDF-Schema (RDFS) stellt eine Erweiterung zur Umsetzung einfacher Ontologien in Form von Klassen- und Beziehungshierarchien dar. Das Bauteil X kann damit zur Klasse „Zylinder“ gehören. Ähnlich zur Abfragesprache SQL für relationale Datenbanken gibt es auch bei Graphdatenbanken eine Reihe von graphbasierten Abfragesprachen, z. B. die SPARQL Protocol and RDF Query Language (SPARQL) [9].

Eine weitere Ergänzung dieses Modellierungskonzepts stellt die Web Ontology Language (OWL) dar. Durch ein größeres Vokabular können gegenüber RDFS weitaus komplexere Beziehungen beschrieben werden. Auch Zusammenhänge zwischen unterschiedlichen aufgebauten Ontologien lassen sich damit modellieren. Beispielsweise kann hinterlegt werden, dass der Begriff „Bauteil“ in Ontologie A dem Begriff „Komponente“ in Ontologie B entspricht. Darüber hinaus unterstützt OWL verschiedene Arten von logischer Inferenz, die durch entsprechende Inferenzmodule bereitgestellt werden. Mit diesen können u. a. kausale, zeitliche und wahrscheinlichkeitstheoretische Zusammenhänge erfasst werden [9].

Ontologien, die auf Basis von OWL modelliert wurden, existieren für verschiedene Domänen. Auch der CAEX-Teil von AutomationML ist in einer OWL Ontologie verfügbar, um einen Austausch mit anderen Beschreibungen auf Basis des Semantic Web zu ermöglichen [10]. Im Kontext interaktiver Simulations- und Visualisierungssysteme ist beispielsweise die ARVIDA-Referenzarchitektur entstanden [11]. Dessen Vokabulare umfassen u. a. die Aspekte Metrologie, Unsicherheiten, Messvorgänge, mathematische Grundbegriffe, Services und räumliche Beziehungen. Eine Darstellung von Objekten eines Prozesses bis hin zu einzelnen Punkten, Linien und Flächen der zugehörigen CAD-Darstellung in Form einer entsprechenden Ontologie in Verbindung mit semantischen Modellen von Prozessschritten und Arbeitszellen kann der direkten Generierung von Roboterprogrammen dienen [12].

Literatur

[1] Krcmar, H., „Informationsmanagement“ (6. Ausg.), Springer, 2015

[2] Inmon, W., „Building the Data Warehouse“ (4. Ausg.), Wiley, 2005

[3] Rabin, S., „The Real-Time Enterprise, the Real-Time Supply Chain“, *Information Systems Management*, 20(2), 58 – 62, 2003

[4] Saaksvuori, A., Immonen, A., „Product Lifecycle Management“ (3. Ausg.). Springer, 2008

[5] Adolphs, P., et al., „Referenzarchitekturmodell Industrie 4.0 (RAMI 4.0)“, Verein Deutscher Ingenieure e.V., Zentralverband Elektrotechnik und Elektronikindustrie e.V., 2015

[6] Schmidt, N., Lüder, A., „AutomationML in a Nutshell“, AutomationML Consortium, 2015

[7] Mahnke, W., Leitner, S.-H., Damm, M., „OPC Unified Architecture“, Springer, 2009

[8] AutomationML e.V., OPC Foundation 2016, „OPC UA Information Model for AutomationML“, Spezifikation

[9] Shadbolt, N., Hall, W., Berners-Lee, T., „The Semantic Web Revisited“, *IEEE Intelligent Systems*, 21(3), 91 – 101, 2006

[10] Kovalenko, O., Grangel-Gonzalez, I. (23. März 2016 aufgerufen). „AutomationML Ontology“, w3id.org/i40/aml

[11] Schreiber, W., Zühl, K., Zimmermann, P. (Hrsg.), „Web-basierte Anwendungen Virtueller Techniken: Das ARVIDA-Projekt – Dienste-basierte Software-Architektur und Anwendungsszenarien für die Industrie“, Springer, 2017

[12] Rickert, M., Perzylo, A., „Industrieroboter für KMU: Flexible und intuitive Prozessbeschreibung“, *Industrie Management*, 32(2), 46 – 49, 2016

[13] Adolphs, P., et al., „Struktur der Verwaltungsschale: Fortentwicklung des Referenzmodells für die Industrie 4.0-Komponente“, Bundesministerium für Wirtschaft und Energie, 2018

[14] Bedenbender, H., et al., „Verwaltungsschale in der Praxis“, Bundesministerium für Wirtschaft und Energie, 2019