

Dr. Annerose Braune, Prof. Dr. Martin Wollschlaeger

Von der Planung bis zum Betrieb – ein steiniger Weg

Der Weg zu einer Automatisierungslösung ist begleitet von einer Vielzahl an Informationen, die verschiedenen Quellen und Sichten entstammen. Bis dato werden diese fast ausschließlich händisch zwischen den Phasen des Lifecycle übergeben. Wie kann XML dazu beitragen, zumindest einen Teil der hinderlichen Steine aus dem Weg zu räumen?

Die Entwicklung einer Automatisierungslösung durchläuft typischerweise einen Lebenszyklus mit aufeinander aufbauenden Phasen: Nach Klärung und Festlegung der Anforderungen zwischen Auftragnehmer und Auftraggeber ist die Lösung zu planen und zu projektieren, geeignete Geräte sind auszuwählen und zu bestellen, die erforderlichen Hardware- und Softwarelösungen sind zu implementieren, und schlussendlich werden die Geräte konfiguriert und in Betrieb genommen. Damit nicht genug: Während oder nach einer Betriebsphase sind Geräte gegebenenfalls zu erneuern oder zu ergänzen. Dieser gesamte Vorgang ist hochgradig komplex, da hierbei nicht nur die gewünschten Funktionen, sondern auch nicht funktionale Anforderungen zu erfüllen sind, die sich unter anderem aus den künftigen Umgebungs- und Einbaubedingungen, den vorhandenen Schnittstellen oder aus den Kenntnissen und Erfahrungen der Anwender ergeben. Außerdem erfordert die Applikationserstellung in der Regel ein stark interdisziplinäres Vorgehen, da Spezialisten unterschiedlichster Fachdisziplinen und Gewerke gemeinsam zur Lösung beitragen müssen.

(Bild: Computer & AUTOMATION, Quelle: TU Dresden)

In der Praxis werden heute entlang der Planungsphasen detaillierte Anforderungen an Geräte bestenfalls in Form von Eigenschafts- oder Merkmalleisten spezifiziert. Zur Auswahl und Bestellung geeigneter Geräte sind aber genau diese Anforderungsmerkmale mit den entsprechenden Gerätemerkmalen zu vergleichen, die die Hersteller für ihre Komponenten angeben. Zur Integration der Geräte in ein Automatisierungssystem sowie zur Projektierung und Inbetriebnahme der Geräte sind noch weitere Merkmale von Bedeutung – beispielsweise was die konkrete Auslegung von Kommunikationseigenschaften oder der Konfiguration interner Verarbeitungsfunktionen betrifft. Jede Phase des Lebenszyklus und jede an einem Projekt beteiligte Fachdisziplin benötigt somit eigene Daten beziehungsweise definiert eigene Sichten auf ein Gerät, die sich aufgrund der diversen Sichten und auch aufgrund historischer Entwicklungen in jeweils spezifischen Gerätemodellen und Sprachen widerspiegeln. Unterschiedliche Informationen, unterschiedliche Bezeichnungen für gleiche Informationen und unterschiedliche Strukturierungen sind damit an der Tagesordnung. Ergo fließen die Daten innerhalb des Lifecycle nicht ungestört, sondern unterliegen Brüchen an den Phasenübergängen. Computer-unterstützte Planungs- und Projektierungswerkzeuge (CAE) definieren zwar intern oft durchgängige Datenstrukturen über mehrere Phasen, bieten aber im Allgemeinen ebenfalls keine Durchgängigkeit zu Fremdgewerken oder Geräte-Anbietern.

Aus dieser unbefriedigenden Situation heraus sind Bestrebungen entstanden, die auf eine übergreifende Standardisierung bestimmter Sichten auf ein Gerät zielen. Ein Beispiel hierfür ist die Namur-Empfehlung NE100 zur Vereinheitlichung von Lieferanten-Kunden-Beziehungen. Darüber hinaus existieren verschiedene Ansätze zur Definition von unabhängigen Austauschformaten über den gesamten Lifecycle, von Produktdaten wie zum Beispiel STEP (STandard for the Exchange of Product model data) oder auch zur Definition von Metasprachen zum Zwecke des Datenaustausches. Welchen Einfluss nehmen diese nun konkret auf die einzelnen Phasen des Lifecycle?

Die Planungsphase

Die Werkzeuge für die Planung reichen von verfahrenstechnischer Planung mit

funktionalen Bestandteilen über die Planung von Installationen für Rohrleitungen, Förderbänder, Kessel oder auch die elektrische Installationstechnik bis zur Planung von Bedienoberflächen. Es ist naheliegend, dass die Anforderungen, aber auch die Komplexität der Aufgaben stark von der Branche beziehungsweise dem Einsatzbereich des Systems abhängen. Die einzelnen Prozesse besitzen einen unterschiedlichen Wiederholungsgrad, sind unterschiedlich stark verteilt und spezialisiert, und daher auch unterschiedlich informationstechnisch gekoppelt. Der Datenaustausch zwischen den einzelnen Anwendungen erfolgt damit traditionell in einer 1:1-Relation, das heißt, die Anwendungen kommunizieren direkt miteinander über ein spezifisches Austauschformat.

Mit dem XML-basierten Format CAEX (Computer Aided Engineering eXchange) wird derzeit ein einheitliches Austauschformat eingeführt, das von der RWTH Aachen und dem ABB Forschungszentrum Ladenburg entwickelt und von Firmen wie BASF, Bayer, Wacker, Intergraph oder Innotec genutzt wird. CAEX ist bereits in die Normung eingeflossen und Bestandteil der IEC PAS 62424. Mit diesem Format sollen sich die ausgetauschten Daten in Bibliotheken organisieren und damit in anderen Applikationen leichter wieder verwenden lassen. CAEX stellt dazu eine herstellerneutrale und gewerke-übergreifende Informationsplattform zur Verfügung und vereinigt Objekte, Attribute und Beziehungen aus den spezifischen Engineering-Werkzeugen in einem durch Abstraktion entstandenen Informationsmodell. Wesentliche Bestandteile des CAEX-Konzeptes sind Schnittstellen, Rollen und Units.

CAEX beschreibt eine Anlage auf einer Meta-Ebene. Durch konkrete Bibliotheken – die letztlich die Hersteller von Softwarewerkzeugen definieren und liefern müssen – wird schließlich die konkrete Hierarchie mit den Daten einer Anlage festgelegt. Dies bedeutet aber auch, dass die Abbildung von Geräte-Eigenschaften, wie etwa Parametern auf CAEX, abhängig von der Modellierung des Gerätes in einer Rolle oder Bibliothek ist – auch wenn beide Informationen nachher in XML vorliegen. Alles in allem stellt CAEX somit ein einheitliches, generisches Format dar, welches aber für eine

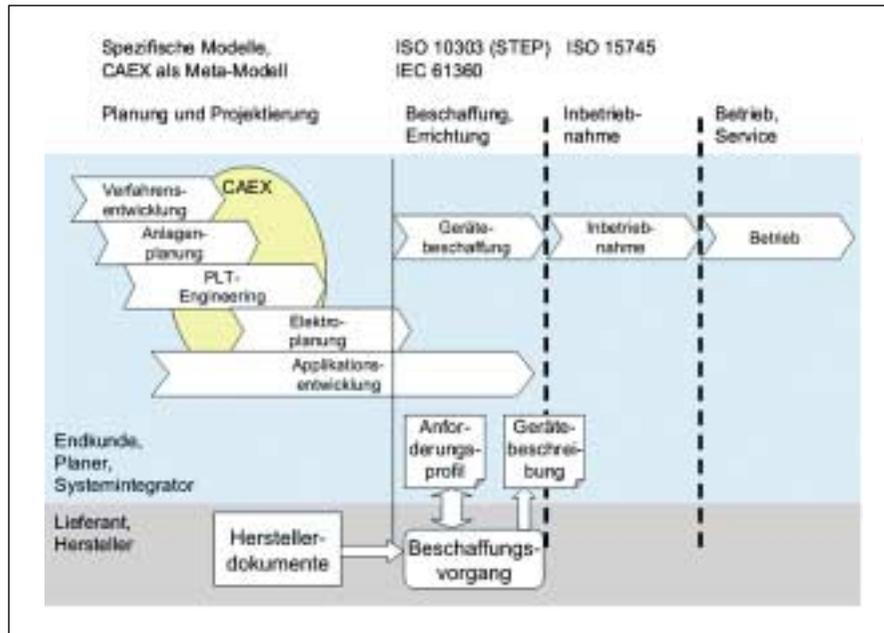


Bild 1. Ausgewählte Aktivitäten und Beschreibungsmodelle in den Phasen des Lifecycle einer Automatisierungslösung. Die nahe liegende Idee eines einheitlichen Datenmodells scheidet an der Vielfalt von Anforderungen, von Sichten, von Anbietern und Werksnormen.



und geometrischen Eigenschaften sowie ihrer Materialien hierarchisch gliedern. So gehört beispielsweise ein Motor neben Steckern, Sicherungen, Relais und Schaltern zu den elektromagnetischen Komponenten und wird weiter klassifiziert nach linearen oder rotierenden, Gleich- oder Wechselstrom-, Schritt- oder Universalmotoren. Ein ausgewählter Motortyp wird beschrieben durch relevante Eigenschaften wie Eingangsspannung, Leistung, Größe und Temperaturbereich. Für die Auswahl und Bestellung eines Gerätes definiert die NE100 folgende Merkmalleisten:

- ▷ Administrative Merkmale (Anfrage, Angebot, Geschäftspartner, Kontaktadressen, ...),
- ▷ Kommerzielle Merkmale (Preise, Lieferzeit, Lieferbedingungen, ...),
- ▷ Gerätemerkmale (technische Beschreibungen, die in CAE-Systemen zur Planung und Projektierung erforderlich sind)
- ▷ und Betriebsmerkmale (Umgebungsbedingungen, ...).

In der Praxis ist das Vorgehen wie folgt: Ein Planer oder Systemintegrator entnimmt aus seinem CAE-System relevante Anforderungen an die zu bestellenden Geräte und fragt damit bei einem Anbieter an. Im Falle von NE 100 kann dies elektronisch per Internet erfolgen. Im besten Fall entsprechen diese Daten hinsichtlich Struktur, Inhalt und Bedeutung genau den Datenblättern eines Anbieters. Für den Austausch dieser Informationen wurde ausgehend von der formalisierten Darstellung in Excel in Zusammenarbeit mit SAP ein XML-Schema definiert, das leider nur den PROLIST-Mitgliedern zur Verfügung steht. PROLIST ist eine Projektgruppe, die in Zusammenarbeit mit Organisationen wie der Namur, dem ZVEI und eClass so genannte „Merkmalleisten“ entwickelt und pflegt. Unter den Mitgliedern finden sich Anwender und Gerätehersteller, insbesondere aus dem Bereich Verfahrenstechnik.

effektive Verwendung einer weiteren Spezialisierung bedarf. Solche Spezialisierungen sind zum Beispiel Rollenbibliotheken, wie sie etwa von Software-Herstellern entwickelt werden, die ihr spezifisches Objektmodell mit CAEX abbilden.

Als Beispiel für die Planung mittels CAEX sei die Planung einer Durchflusssmessstelle genannt. Diese hat bestimmte Eigenschaften wie den Messbereich, die physikalische Einheit oder den Wert selbst. Solche Größen lassen sich als Attribute eines Prozessleittechnik-Objektes „Durchflusssmessstelle“ begreifen und in CAEX entsprechend ablegen. Die Attributwerte, die in CAEX definierbar sind, beinhalten dann die konkreten Werte – etwa die Zah-

lenwerte 0 und 100 für die untere beziehungsweise obere Messbereichsgrenze.

Die Gerätebeschaffung

Auf der Basis der Anforderungsmerkmale der Planung erfolgt dann die Auswahl und Bestellung von Geräten. Die Namur-Empfehlung NE 100 hat das Ziel, Durchgängigkeit zwischen verschiedenen Planungswerkzeugen und Geräteherstellern durch eine einheitliche Definition von Merkmalleisten zu erreichen. Grundlage hierfür sind verschiedene Gerätemodelle – unter anderem festgeschrieben in den Standards IEC 61360 und ISO 13584 –, die eine technische Komponente hinsichtlich ihrer Bauteile, ihrer elektrischen, magnetischen

Ein Anbieter schickt daraufhin in seinem Antwortdokument eine gleichermaßen formalisierte Beschreibung der Eigenschaften des angebotenen Gerätes. Besteht Durchgängigkeit in den Informationsmodellen, so kann der Systemintegrator diese Daten direkt in sein CAE-System einlesen und mit den technischen Anforderungen vergleichen. Üblicherweise befinden sich im Lieferumfang der Geräte detaillierte Dokumentationen sowie Gerätebeschreibungen, die den aus der Planung stammenden Informationsbestand anreichern. Diese werden in den weiteren Phasen des Lifecycle mit verwendet.

Auf das Beispiel der Durchfluss-Messstelle bezogen würden im Beschaffungsbblatt gemäß NE 100 Informationen in die Rubrik „Messgröße analog“ mit weiteren Detail-Angaben wie etwa Bezeichnung, Einheit und Messbereich in ein Excel-Sheet eingetragen (siehe *Bild 2*, S. 38, mittlerer Teil).

Die Anwendungsentwicklung

Für die Definition der Anwendungsfunktionen eines Automatisierungssystems

gibt es verschiedene Wege. Modernen Lösungen gemein ist, dass sie die Definition der Gesamtfunktion mit Hilfe wieder verwendbarer Strukturen erlauben. Ein typischer Vertreter einer solchen Lösung ist die Programmentwicklung für SPS. In diesem Umfeld hat sich der Standard IEC 61131 aus guten Gründen auf breiter Front etabliert; unter anderem deshalb, weil gerade hier der Austausch zwischen verschiedenen Programmierwerkzeugen und zwischen Programmierwerkzeug und Werkzeugen anderer Lebensphasen durch Definition eines XML-basierten Formats adressiert wurde.

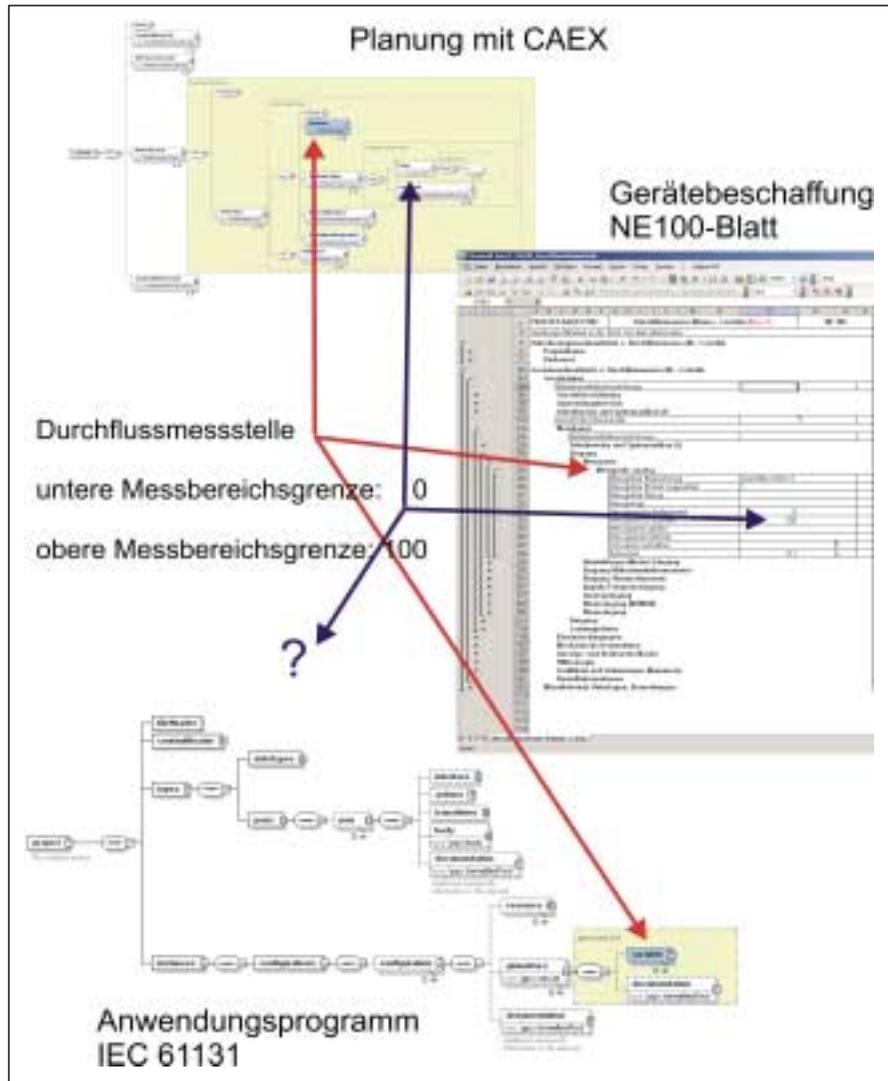
Damit SPS-Programme auf Daten von Feldgeräten zugreifen können, werden diese in die Ein- beziehungsweise Ausgangsdatenbereiche der SPS abgebildet. Dies ist eine typische Funktion der Anwendungsentwicklung, die in IEC 61131 zumeist durch Definition von globalen Variablen erfolgt. Dazu werden Informationen über die Feldgeräte benötigt, damit die Instanzadresse, das Datenformat und gegebenenfalls die Struktur der Daten mit den Sprachmitteln von IEC 61131 abgebildet

werden. Eine Wiederverwendung der Gerätebeschreibungen würde hier zu einer erheblichen Zeitersparnis beitragen. Die Elemente für Eingangs- und Ausgangsdaten der in XML vorliegenden Gerätebeschreibungen müssen deshalb den Sprachelementen des XML-Austauschformats der IEC 61131 zugeordnet werden.

Neben diesen offensichtlichen Abbildungen von Variablen besteht mit der Definition von „Function Blocks“, wie sie etwa für Profibus-PA-Geräte typisch sind, ein erhebliches Potenzial für eine vereinfachte Anwendungsentwicklung. Aus einer XML-basierten Beschreibung eines solchen Funktionsblocks ließe sich relativ leicht eine Definition eines IEC-61131-Function-Blocks ableiten, der seinerseits über Bibliothek-Konzepte wieder verwendbar wäre. Allerdings ist dieses Potenzial bis dato weitgehend ungenutzt!

Die Betriebsphase

Der Inbetriebnahme des Automatisierungssystems schließt sich die Betriebsphase an, in der Informationen aus den Automatisierungsgeräten in überlagerte



Applikationen genutzt werden. In diesem Kontext hat sich OPC zum Standard für die systemneutrale Anbindung von verschiedenen Applikationen entwickelt – vom Bediensystem bis zur MES- und ERP-Lösung. Die Kapselung der kommunikationstechnischen Details durch den OPC-Server ist hier sicherlich ein wesentlicher Vorteil; gleichzeitig jedoch eine Herausforderung, da diese Kapselung durch ein Mapping der von einem Feldgerät freigegebenen Prozessdaten auf OPC-Daten erfordert, und die Freiheitsgrade bei der Definition dieser Server-Namespaces inkompatible Abbildungen geradezu herausfordern. Stammen Feldgerät und OPC-Server vom gleichen Anbieter, existiert oft bereits eine gewisse Durchgängigkeit, sodass sich die vom Feldgerät zur Kommunikation freigegebenen Prozessdaten aus dem Applikationsprojekt extrahieren und rechnergestützt

direkt zur Konfiguration des Servers nutzen lassen. In besten Falle stimmen dann auch Eigenschaften wie Datentypen und Zugriffsrechte überein.

Kommen hingegen Feldgeräte und OPC-Server unterschiedlicher Hersteller zum Einsatz, so ist der Namespace unter Umständen von Hand zu füllen und auch die erforderlichen Eigenschaften sind händisch anzupassen. Werden zum Beispiel Grenzwerte einer Messgröße benötigt, um Ansprechschwellen für Alarmer festzulegen oder das Totband für die Übertragung des Wertes einzustellen, so sind diese von Hand im OPC-Server einzutragen. Die dafür maßgebliche Informationen liegen gemäß *Bild 2* jedoch bereits seit der Gerätebestellung logisch an die Messgröße gebunden vor, gegebenenfalls sogar rechnerlesbar. Während der Applikationsentwicklung gehen derartige Zusammenhänge aber in der Regel verloren und sind bei der

Bild 2. Gleiche Information in unterschiedlichen Dokumenten im Lifecycle: Einzelne Informationen wie eine Messgröße lassen sich in CAEX, NE100 und OPC abbilden, Zusatzinformationen wie etwa der Messbereich können jedoch nicht direkt zugeordnet werden.

OPC-Konfiguration erneut herzustellen. Im Rahmen von OPC UA wird versucht, diese Heterogenität einzuschränken, und durch so genannte „Semantic Layer“ Informationsmodelle einzubinden sowie auf diese Weise Vorzugsbelegungen zu definieren. Die Entwicklungen diesbezüglich sind allerdings noch in Gange.

Zum Thema XML lässt sich abschließend sagen: Ebenso wie bei den Geräteherstellern (siehe „XML in der Automation – Teil 2“, *Computer&AUTOMATION* 2007, H. 09, S. 32ff.) ist auch bei den Systemintegratoren die Hoffnung auf einheitliche, über den gesamten Lifecycle nutzbare Sprachen aus politischen, ökonomischen und technischen Gründen nicht umsetzbar. Realisierbar wäre es jedoch, Beziehungen zwischen den verschiedenen Modellen klar zu definieren, um so eine leichtere Zuordnung relevanter Informationsquellen zu ermöglichen. Dies kann als Erweiterung oder Add-on zu bestehenden Lösungen schrittweise erfolgen. Ein geplantes „Wiki der Automation“ unter Mitwirkung einer breiten Anwendergemeinschaft ist hierfür eine wesentliche Voraussetzung. gh

Artikel-Download:
www.elektroniknet.de/automation



Dr. Annerose Braune

ist Mitarbeiterin am Lehrstuhl für Automatisierungstechnik der TU Dresden, Fakultät Elektrotechnik und Informationstechnik.



Prof. Martin Wollschlaeger

ist Inhaber der Professur Prozesskommunikation an der Fakultät Informatik der TU Dresden. Er leitet den Arbeitskreis „XML in automatisierungstechnischen Anwendungen“.