

Bild 1: Die Rückführungen im Standardisierungsregelkreis sind gestört.

Bild: ABB AG Forschungszentrum Deutschland

# Warum ein Engineering-Weltmodell bisher nicht gelang

## Serie AutomationML Teil 10: Austausch von Daten zwischen unterschiedlichen Datenmodellen im heterogenen Werkzeugumfeld

Es klingt plausibel: Für ein neutrales Datenaustauschformat für das Engineering benötigt man eine standardisierte Syntax und Semantik. Es müsste ein 'Super-Datenmodell' entwickelt werden, das die wichtigsten Datenmodelle des Engineering vereint und eine eindeutige und verlustlose Abbildung aller Daten ermöglicht.

Die Idee des 'Super-Datenmodells' ist die treibende Kraft aller entsprechenden Standardisierungsaktivitäten, z.B. NE100 [1], STEP [2], ISO15926 [3] u.v.m. Dazu müssen sie stets folgende Schritte durchlaufen:

- a) Experten sammeln für beteiligte Gewerke bzw. Werkzeuge sinnvolle Datenmodelle und Konzepte.
- b) Experten einigen sich auf die gemeinsam benötigten Grundkonzepte.
- c) Experten einigen sich auf ein neutrales, semantisches Datenmodell, das alle Grundkonzepte berücksichtigt, z.B. mit UML.
- d) Experten einigen sich auf eine neutrale Syntax, z.B. basierend auf XML.
- e) Alle gesammelten Konzepte, das semantische Datenmodell und die Umsetzung in einer gemeinsamen Syntax,

werden dokumentiert.

f) Die Industrie und Akademia schafft die Voraussetzungen dafür, dass genügend Experten zur Verfügung stehen, die den Standard beurteilen und implementieren können.

g) Anwender und Werkzeughersteller implementieren und nutzen den Standardentwurf in ihren Werkzeugen.

h) Erfahrungen aus der praktischen Nutzung des Standards fließen in (a) zurück und tragen zur Reifung bei.

Bis heute ist dies nicht gelungen. Bild 1 erklärt den Konflikt: Standardisierung reift nur durch Rückfluss von Erfahrungen aus der praktischen Anwendung. Werkzeughersteller warten jedoch aus Kostengründen typischerweise erst auf einen gereiften Standard, bevor sie ihre Software mit Ex- und Importern austatten. Die Rückführungen der prakti-

sehen Erfahrungen in die Standardisierung (h) sowie die notwendigen Implementierungen der Hersteller (g) sind gestört, die Standardisierung befindet sich in einem Deadlock.

### Die Idee: Ein gemischtes Datenmodell

Ein Weltmodell des Engineering ist offensichtlich nicht in einem Schritt erreichbar. AutomationML [4] verfolgt daher einen anderen Ansatz. Anstatt auf eine Standardisierung zu warten, lässt es die Speicherung proprietärer Semantiken explizit zu [5]. Beim Export aus einem Engineeringwerkzeug werden die Daten syntaktisch neutralisiert, die Semantik bleibt jedoch proprietär ('privat'). Ein Signal würde beispielsweise alle Eigenschaften inklusive seines Namens aus dem Ur-

```

<AdditionalInformation>
  <WriterHeader>
    <WriterName>ToolX AutomationML Exporter</WriterName>
    <ToolWriterID>ToolXToAutomationML123</ToolWriterID>
    <WriterVendor>ToolX Vendor</WriterVendor>
    <WriterVendorURL>http://www.ToolX-Vendor.org</WriterVendorURL>
    <WriterVersion>0.1</WriterVersion>
    <WriterRelease>123 prealpha</WriterRelease>
    <LastWritingDateTime>2011-05-25T09:30:47</LastWritingDateTime>
    <WriterProjectTitle>eCarproduction</WriterProjectTitle>
    <WriterProjectID>eCarproduction_LinePLC.prj</WriterProjectID>
  </WriterHeader>
</AdditionalInformation>
    
```

Bild: ABB AG Forschungszentrum Deutschland

Bild 2: AutomationML-Beispiel-Etikett

sprungswerkzeug behalten. Auch wenn das Zielwerkzeug diese Daten zunächst nicht interpretieren kann (gerade zu Beginn), stellt dies bereits einen erheblichen Gewinn dar: Erstmalig werden proprietäre Datenmodelle aus ihrem Werkzeug herausgelöst, syntaktisch neutralisiert und untersuchbar. Dies erleichtert die beiden Schritte (a) und (b) der beschriebenen Standardisierungskette erheblich. Ziel-Engineering-Werkzeuge können die Datei öffnen, durchsuchen und darstellen. Funktionen wie Navigation, Änderungsberechnungen und Versions-Management sind unabhängig vom Datenmodell und lassen sich generisch entwickeln und wiederverwenden. Es fehlen nur noch die Transformationsroutinen, die für die proprietären Datenmodelle des Quellwerkzeuges zugeschnitten sind. Doch woher kommen sie und wie kann ein Importeur erkennen, welche davon aufgerufen werden müssen?

**Die Idee: Identifikation des Quellwerkzeuges**

AutomationML führt dazu erstmalig einen standardisierten Etikettiermechanismus ein. Die CAEX-Datei bekommt ein Etikett (siehe Bild 2), welches während des Exports in die AutomationML-Datei eingebettet wird und Informationen über das Quellwerkzeug enthält. Ein Importeur kann dieses Etikett auswerten und private Transformationsroutinen anstoßen. Dieses Konzept wird von der frei verfügbaren Software AutomationML Engine [6] unterstützt. Diese Vorgehensweise bedeutet keineswegs, dass jedes Werkzeug seine Daten dann auch gleich in seinem eigenen proprietären Datenfor-

mat exportieren könnte. Bereits die syntaktische Neutralisierung bringt entscheidende Vorteile:

- Der Hauptaufwand eines Importeurs liegt in Operationen wie Versionsmanagement oder Änderungsberechnungen. Diese Operationen sind unabhängig vom Datenmodell. Derartige Funktionalität müsste für jedes proprietäre Datenformat erneut entwickelt werden. Basierend auf einem neutralen Datenformat genügt eine einzige Implementierung. Ein Großteil des Importeurs kann daher – unabhängig vom jeweiligen Quellwerkzeug – wiederverwendet werden.
- Das Etikettierkonzept erlaubt die automatische Absendererkennung. Proprietäre Datenformate wie z.B. Excel-Sheets etc. verfügen über keine standardisierte Methodik dafür.

**Und dann?**

Die quellwerkzeugspezifischen Subroutinen für die Interpretation und den Import müssen erst entwickelt werden. Die Erkennung des Quellwerkzeuges ist der Schlüssel. Der

‘Trick’ besteht darin, dass der Importeur ‘unbekannte private Datenmodelle’ erkennen und in einer separaten CAEX-Bibliothek speichern kann. Mit anderen Worten: Der Importeur weiß, was er nicht weiß. Eine Bibliothek unbekannter Datentypen bzw. Klassen enthält alle Datenmodell-Kandidaten, sozusagen ein Lastenheft für die Softwareentwicklung. Anhand von Häufigkeitsuntersuchungen lassen sie die Kandidaten sogar priorisieren. Private Transformationsroutinen können damit ohne Warten auf einen Standard in lokaler Verantwortung entwickelt werden. Das geht schneller als jede Standardisierung. Das Ziel ist zunächst der gelebte praktische Einsatz und die Reifung unter realen Bedingungen. Nach der Entwicklung erster Transformationsroutinen werden aus ‘unbekannten’ somit ‘bekannte’ Datenmodelle. Sie lassen sich für jeden weiteren Datenaustausch mit demselben Quellwerkzeug wiederverwenden. Das beschriebene Szenario ist unmittelbar einführbar. Zwei Werkzeughersteller oder -nutzer können beliebig Export-Import-Partnerschaften eingehen. Ein dritter Teilnehmer kann sich (auch später) einbinden. Dieses Szenario stellt zugleich Reifegrad 1 eines evolutionären Reifegradmodells dar: (siehe Bild 3). Dieser ist ein erster Meilenstein in der Evolution eines Standardisierungsprozesses: Er erleichtert die Schritte (a) und (b) und belebt die Rückführung von Erfahrungen (h) aus der praktischen Anwendung. Der Deadlock aus Bild 1 wird durchbrochen. Dieser Reifegrad ist für Werkzeuge geeignet

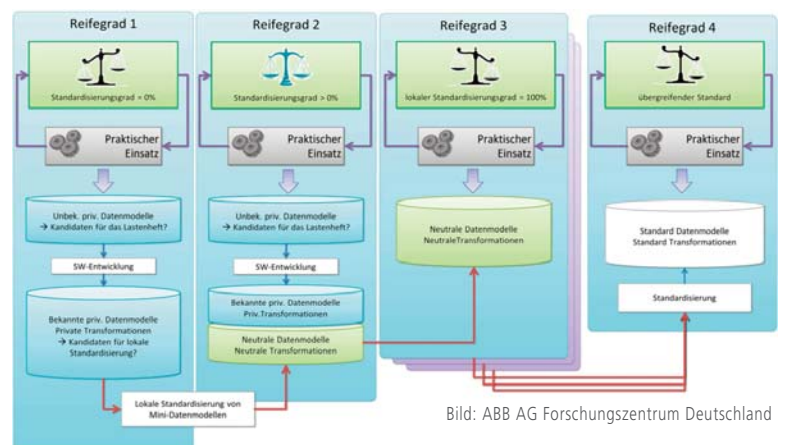


Bild: ABB AG Forschungszentrum Deutschland

Bild 3: Reifegradmodell

und möglicherweise sogar ausreichend, die nur wenige Datenaustauschpartner besitzen. Die Entwicklung privater Transformationsroutinen benötigt deutlich weniger Aufwand als eine Standardisierung. Reifegrad 1 ist weiterhin ideal für Werkzeuge, die lediglich ihre eigenen Daten archivieren bzw. extern manipulieren und anschließend wieder einlesen möchten. Mit einer wachsenden Zahl von Teilnehmern wird eine Bibliothek aus Transformationsroutinen entstehen. Dies ist ein geeigneter Indikator für den Start einer Konsolidierung des Entwicklungsaufwandes und somit ein geeigneter Übergangspunkt zu Reifegrad 2, denn ähnliche Datenmodelle sind Kandidaten für eine erste Standardisierung. Reifegrad 2 entsteht aus der Konsolidierung der Standardisierungskandidaten aus Reifegrad 1. Ähnliche Datenmodelle führen zu Mehrfachentwicklung von ähnlichen Transformationsroutinen und erhöhen unnötig die Entwicklungskosten. Dies fördert die Standardisierungsbereitschaft und führt zu ersten neutralen 'Mini'-Datenmodellen. Für jedes neutralisierte 'Mini'-Datenmodell kann in jedem Importer die Zahl  $n$  der privaten Transformationsroutinen für  $n$  Quellwerkzeuge auf eine einzige Transformationsroutine reduziert werden. Auf diese Weise entsteht ein Pool an neutralen Datenmodellen, während der Pool privater Transformationsroutinen abnimmt. Der Evolutionsprozess hin zu Reifegrad 2 wird somit aus praktischen Projektbedürfnissen getrieben. Dies stellt einen natürlichen und evolutionären Prozess dar. Unter Verwendung erster neutraler Daten-Teilmodelle erzeugt ein Exporter in Reifegrad 2 ein

gemischt standardisiert/privates CAEX-Datenmodell. Reifegrad 3 wird erreicht, wenn alle benötigten Daten-Teilmodelle durch neutrale Datenmodelle abgebildet werden. Er bildet das Gegenstück zu Reifegrad 1, da hierbei der Anteil an privaten Datenmodellen 0% beträgt. Seine Reife erlangt er durch Erfahrungen im praktischen Einsatz. Reifegrad 3 ist nicht an internationale Standards gebunden. Es könnte ebenso ein Standard eines einzelnen Projektes oder eines Konsortiums sein. Dieser Reifegrad kommt der eigentlichen Idee eines neutralen Datenaustauschformates bereits nahe: Private Transformationsroutinen sind nicht mehr notwendig – das Datenmodell ist aus Erfahrungen der Reifegrade 1-2 gewachsen und gereift. Reifegrad 4 wird erreicht, wenn mehrere 'Reifegrad 3 Standards' zu einem übergeordneten Standard kombiniert werden. Die Datenmodelle aus Reifegrad 3 bilden aufgrund praktischer Erfahrungen eine belastbare Basis für die Schritte (a) und (b) eines übergreifenden Standardisierungsvorhabens, z.B. für einen internationalen Standard. Dieser Reifegrad ist als langfristiges Ziel zu verstehen und ein evolutionäres Ergebnis des Durchlaufens der vorigen Reifegrade.

### Diskussion und Zusammenfassung

Standardisierungsgremien und Softwarehersteller haben frühzeitig den Wert eines gemeinsamen Datenmodells erkannt, verfolgen jedoch mehrheitlich Reifegrad 3 bzw. 4. Für die Praxis macht es jedoch keinen Sinn, sofort mit Reifegrad 3 oder 4 zu beginnen. Eine Evolution beginnt stets mit dem ersten Schritt (Reifegrad 1). Mit dem

vorgestellten Konzept wird semantische Vielfalt beherrschbar. Die Mischung privater und neutraler Daten eliminiert den Bedarf nach einem umfassenden und gereiften Engineering-Weltmodell: Der Schwerpunkt verlagert sich stattdessen auf kleine und flexible 'Mini'-Datenmodelle. Diese 'Mikro-Standardisierung' zwischen zwei Werkzeugen ist ein agiler und pragmatischer Weg mit iterativen Standardisierungssprints. Der praktische Einsatz steht in jedem Reifegrad im Vordergrund, die Standardisierung folgt der Nutzung, nicht umgekehrt. Erstmals lässt sich Standardisierungsbedarf inklusive Priorisierung automatisch ableiten. Mit diesem Ansatz versucht AutomationML nicht, die bestehende Heterogenität von Engineering-Werkzeugen und Datenmodellen durch Vereinheitlichung zu überwinden; Das Konzept verfolgt vielmehr das Anliegen, den Datenaustausch in einem heterogenen Werkzeugumfeld auch ohne bzw. nur mit teilweiser Standardisierung durchführen zu können. Anwender können Datenaustauschlösungen deutlich leichter und schneller als bisher entwickeln. Gerade für das Umsetzen von Werkzeugketten mit nur wenigen Datenaustauschpartnern ist der Entwicklungsaufwand deutlich geringer als das Standardisieren eines Weltmodells. Das Warten auf einen Standard ist nicht mehr nötig. Zugleich ebnet diese Vorgehensweise den Weg zu einer evolutionären Standardisierung, entkoppelt jedoch dessen Zeitbedarf vom Projektdruck in der Industrie. Erste Verbände (z.B. VDA und VDMA) haben dies erkannt und verfolgen statt breiter Standardisierung die zügige Entwicklung leichtgewichtiger Schnittstellen für begrenzte Bereiche. ■

### Literatur

- [1] Namur Empfehlung 100: Merkmalleisten zur Erstellung von PLT-Gerätespezifikationen. 2003.
- [2] ISO10303: Automation systems and integration - Product data representation and exchange, formerly known as STEP, Standard for the Exchange of Product model data.
- [3] ISO15926. Industrial automation systems and integration - Integration of life-cycle data for process plants including oil and gas production facilities.
- [4] IEC62714-1 CD norm draft AutomationML Architecture, www.iec.ch, 2011.
- [5] Drath R., Barth M.: Wie der Umgang mit unterschiedlichen Datenmodellen beim Datenaustausch im heterogenen Werkzeugumfeld gelingt. In: Tagungsband Automation 2013, Baden-Baden, VDI-Berichte 2209, ISBN 978-3-18-092209-6, 2013.
- [6] www.automationml.org

[www.automationml.org](http://www.automationml.org)



*Autor: Dr.-Ing. Rainer Drath,  
Senior Principal Scientist,  
ABB AG Forschungszentrum  
Deutschland*