

Herstellerunabhängiger Austausch von Entwurfsdaten für Steuerungssysteme mittels AutomationML

von Prof. Dr.-Ing. Arndt Lüder, Nicole Schmidt

Der verlustfreie Austausch von Entwurfsdaten zwischen verschiedenen Entwurfsprozessphasen, Entwurfswerkzeugen und zwischen Experten verschiedener Entwurfsdisziplinen hat sich zu einem wichtigen Mittel bei der Effizienzerhöhung und Qualitätssicherung im Entwurf von Produktionssystemen entwickelt. Dabei gilt es, zwei generelle Anwendungsprobleme zu lösen: die Auswahl eines geeigneten Datenaustauschformats sowie der Nachweis der praktischen Anwendbarkeit dieses Formats. Beide Probleme sollen in diesem Beitrag betrachtet und für AutomationML als Datenaustauschformat für Entwurfsdaten zum Steuerungsentwurf innerhalb des Entwurfs von Produktionssystemen beantwortet werden.

Einleitung

In Zeiten, in denen neue Produkte wie Autos, Handys, Waschmaschinen etc. stetig komplexer werden, dabei aber schneller produziert werden sollen, müssen die Produktionssysteme diesem Wandel folgen und die damit an sie gestellten Anforderungen erfüllen [1]. Diese Anforderungen gelten jedoch nicht nur für die Nutzungszeit eines Produktionssystems sondern für dessen gesamten Lebenszyklus einschließlich des Zeitraums, in dem das Produktionssystem entworfen und erstellt wird. In diesem Zeitraum arbeiten Experten unterschiedlicher Disziplinen zusammen, um ein Produktionssystem zu entwickeln und zu erschaffen, [20]:

1. das in der Lage ist, die geplanten, komplexen Produkte zu fertigen (Unternehmen produzieren für gewöhnlich nicht nur ein und dasselbe baugleiche Produkt. Sie haben Produktportfolios, bei dem die unterschiedlichen Produkte sich in ihren Features unterscheiden – aus dieser Bandbreite kann der Verbraucher dann in der Regel das passende Produkt auswählen.) und
2. das in stetig kürzer werdender Zeit erstellt oder verändert werden soll. (Die Forderung, die Produkteinführungszeit neuer Produkte zu verkürzen, d. h. die Zeit, die notwendig ist, sie auf den Markt zu bringen, ist allgegenwärtig. Der Lebenszyklus von Produkten, insbesondere die Zeitdauer, in der Produkte gewinnbringend verkauft werden können, verkürzt sich stetig. Außerdem bringt es dem Unternehmen Marktvorteile, wenn es ein neues Produkt mit neuen Features als erstes auf den Markt bringt.)

Um der ersten Anforderung gerecht zu werden, müssen flexiblere und anpassungsfähigere Produktionssysteme entworfen werden. Dies erhöht massiv die Komplexität dieser Produktionssysteme. Um mit dieser Komplexität umgehen zu können, wurden neue Entwurfsmethoden für Produktionssysteme entwickelt, die funktionalen [25, 26], mechatronischen [8] oder strukturellen Grundprinzipien [24] folgen.

Ein gebräuchlicher und viel versprechender Ansatz ist hierbei das mechatronische Denken, das einen objekt- bzw. systemorientierten Ansatz umsetzt [21, 22]. Objektorientierung ist in diesem Fall nicht im Sinne der objektorientierten Programmierung zu verstehen. Hier kommt eher die Idee des objektorientierten Design zum Tragen mit einem weiter gefassten Verständnis von

Systemorientierung. Ein Produktionssystem wird unter Verwendung von Objekten modelliert – sogenannten mechatronischen Einheiten, z. B. Robotern, Förderbändern, Motoren etc. Die während des Entwurfsprozesses entwickelten, zu den Objekten gehörenden Informationen werden diesen als eine Art digitaler Schatten (auch mechatronisches Informationsobjekt genannt) zugewiesen/zusortiert, z. B. der notwendige Arbeitsraum eines Roboters, die Abmaße eines Förderbands oder der Motortyp. Ein solches mechatronisches Informationsobjekt ist schematisch in der nachstehenden Abbildung dargestellt. Diese zeigt die Informationen, die notwendig sind, um ein Objekt im Produktionssystem umfassend zu beschreiben.

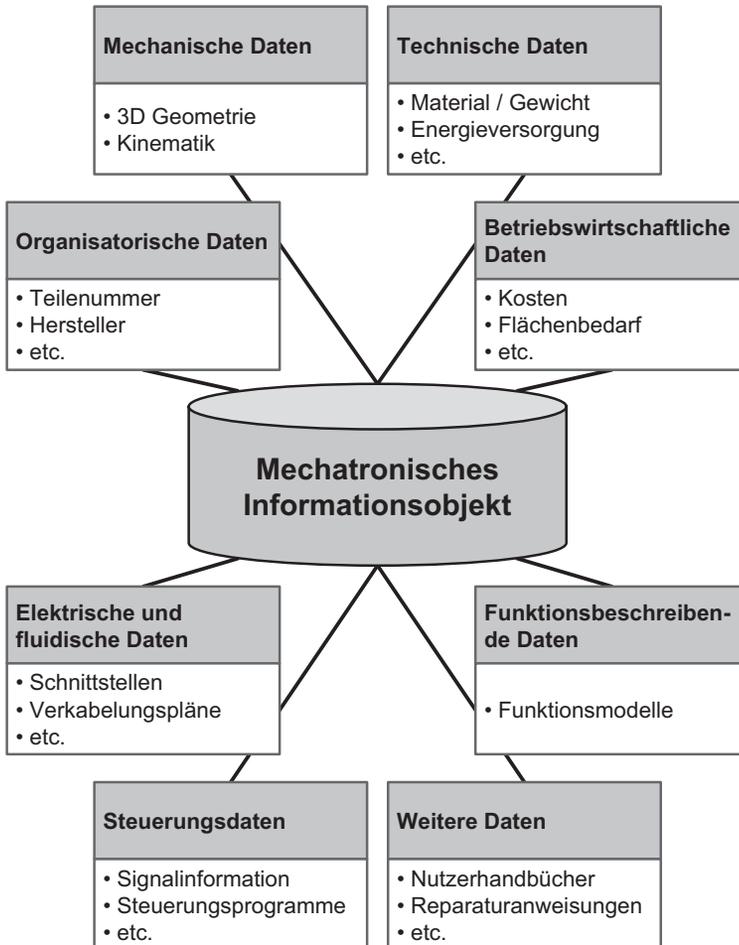


Bild 1 Informationsmenge eines mechatronischen Informationsobjekts (nach [16])

Wird dieser Ansatz konsequent verfolgt, ist es möglich, die Daten mit Fortschreiten im Entwurfsprozess immer weiter anzureichern. In frühen Phasen des Prozesses werden die primären Objekte, die für das angedachte Produktionssystem notwendig sind, grob beschrieben zusammengeführt. Mit Fortschreiten des Prozesses ist es die Aufgabe der unterschiedlichen Experten, die grob beschriebenen Objekte auszudetaillieren, so dass Kaufteile für die Anlage

bestellt werden können und das Produktionssystem letztendlich aufgebaut und in Betrieb genommen werden kann.

Auf der einen Seite ermöglicht das mechatronische Denken also eine konsistente Modellierung und Datenhaltung, was die Experten unterstützt, indem das gesamte Wissen zu einem Objekt genau an diesem gespeichert wird. Auf der anderen Seite ermöglicht der Ansatz das Speichern bzw. das Zusammenführen dieser Objekte in Bibliotheken und damit die Wiederverwendung dieser in anderen Projekten. Das kann den Entwurfsprozess von Produktionssystemen beträchtlich verkürzen, wie es die VDI-Richtlinie 3695 [28] postuliert.

Mit Blick auf die zweite Anforderung kann festgestellt werden, dass der Entwurfsprozess von Produktionssystemen so kurz wie möglich gestaltet werden muss, da dieser maßgeblich die Produkteinführungszeit beeinflusst. Je schneller ein Produktionssystem also geplant und aufgebaut werden kann, desto früher kann ein Unternehmen sein Produkt auf den Markt bringen und damit an den Kunden.

Da der Entwurfsprozess von Produktionssystemen ein komplexes Zusammenspiel aus diversen Aktivitäten bzw. Designentscheidungen ist, welche von Experten unterschiedlicher Disziplinen ausgeführt bzw. getroffen werden, die jeweils eine andere Sicht auf die Objekte haben, gibt es folglich mehrere Ansätze, um diesen Prozess zu verkürzen bzw. zu optimieren. Bleibt zu entscheiden, welcher Ansatz es wert ist, konsequent verfolgt zu werden und letztendlich Geld dafür aufzuwenden, diesen umzusetzen.

2005 führte die Automatisierungsinitiative Deutscher Automobilhersteller (AIDA) eine Studie durch, in der sie den Entwurfsprozess von Produktionssystemen beginnend vom ersten Schritt, dem Auftrag, ein spezifisches Produktionssystem zu entwerfen, bis zum letzten, der Inbetriebnahme, analysierten. Die wesentliche Erkenntnis dieser Studie war, dass die Hälfte der im Entwurfsprozess anfallenden Kosten durch das eigentliche Entwerfen des Produktionssystems anfielen [23]. Angetrieben, den Ursachen dieses erheblichen Kostenfaktors auf den Grund zu gehen, startete Daimler ein Projekt, das zum Ziel hatte, genau diese Kosten zu halbieren. Ein Ergebnis des Projekts war, dass erhebliche, nicht gerechtfertigte Aufwendungen für den Datenaustausch zwischen Softwarewerkzeugen betrieben wurden. Daraufhin wurde der Datenaustausch zwischen verschiedenen Entwurfsprozessphasen, Entwurfswerkzeugen und zwischen Experten verschiedener Entwurfsdisziplinen besonders häufig betrachtet [2, 3]. Ziel war dabei eine horizontale Integration, wie sie in [4] gefordert wird.

Eine nähere Betrachtung des Datenaustauschs innerhalb des Entwurfsprozesses von Produktionssystemen enthüllt eine heterogene Softwarewerkzeuglandschaft. Bild 2 beinhaltet eine Auswahl der oft in diesem Kontext verwendeten Werkzeuge. In Abhängigkeit von der Entwurfsprozessphase und der Disziplin (CAD, CAE, Steuerungsprogrammierung etc.) sowie der Vorliebe der einzelnen Experten werden unterschiedliche Werkzeuge verwendet. In einigen Unternehmen sind auch die zu verwendenden Werkzeuge vorgeschrieben. In jedem Fall sind die unterschiedlichsten Ausprägungen von Werkzeugketten möglich: Das ist die Anzahl an Werkzeugen, die notwendig sind, um ein Produktionssystem zu entwerfen und diese Werkzeuge müssen folglich Daten austauschen. Exemplarisch kann hier die Interaktion des Steuerungsprogrammierers mit anderen beteiligten Ingenieuren genannt werden. Der Steuerungsprogrammierer erhält vom Mechanikkonstrukteur die Liste der zu verbauenden Automatisierungsgeräte (Sensoren, Aktoren und Steuerungsgeräte), der Elektroingenieur übermittelt die Beschreibung der Verkabelung dieser Geräte und der Prozessingenieur gibt vor, wie die Geräte für den Produktionsprozess anzusteuern sind. Diese Informationen braucht der Steuerungsprogrammierer, um die Projektkonfiguration, die Variablendeklaration zu erzeugen und das Verhalten zu programmieren.

Application	Subset	Tools (Examples)
CAD		<ul style="list-style-type: none"> • CATIA v4, v5 • Autocad • UGS • SolidWorks • PTS ProEngineer • MicroStation • Blender • 3d Max • Maya
Simulation	Material Flow Simulation	<ul style="list-style-type: none"> • Simple++/e-MPlant • Witness • Quest • ALB (Automatic Line Builder) Delmia
	Robot Simulation	<ul style="list-style-type: none"> • Cosimir • IGrip D5 • Catia v5 Robotics • Robocad
	Process Simulation	<ul style="list-style-type: none"> • FEM Ansys
	Electrical Simulation	<ul style="list-style-type: none"> • PSPICE • Electronics Workbench • Multisim
Office	Text Processing	<ul style="list-style-type: none"> • MS Word • OpenOffice
	Spreadsheet Analysis	<ul style="list-style-type: none"> • MS Excel • OpenOffice
	Presentation	<ul style="list-style-type: none"> • MS Powerpoint • OpenOffice
	Databases	<ul style="list-style-type: none"> • Access • Oracle • MS Sql
	Communication	<ul style="list-style-type: none"> • Email
Project Management		<ul style="list-style-type: none"> • MS Project • MindManager
Product Data Management (PDM)		<ul style="list-style-type: none"> • UGS TeamCenter • Dassault Smartteam • Dassault Enovia
Product Lifecycle Management (PLM)		<ul style="list-style-type: none"> • E-MPlanner • Delmia E5 DPE
Enterprise Resource Planning (ERP)		<ul style="list-style-type: none"> • SAP R3 • Oracle Peoplesoft • MS Navision

Application	Subset	Tools (Examples)
Reporting		<ul style="list-style-type: none"> • Cognos • Crystal Reports • Eclipse BIRT
Visualization	Mockup	<ul style="list-style-type: none"> • e-MEngineer • Many others
	Plant visualization	<ul style="list-style-type: none"> • Jviz • OpenGT • OpenFlight
	HMI	<ul style="list-style-type: none"> • WinCC / WinCC Flexible • intouch
Control Programming	PLC	<ul style="list-style-type: none"> • STEP 7 • RSLogix • RSLinx • CoDeSys • iMap
	Robot Control	<ul style="list-style-type: none"> • ABB Robotstudio • KUKA SIM • Dürr 3D Onsite
CAE		<ul style="list-style-type: none"> • Ruplan • ePlan • Eagle • Target 3001
Process configuration		<ul style="list-style-type: none"> • 3D Onsite • Robotstudio • Robscan Design/Control • Bos 6000
Facility Management		<ul style="list-style-type: none"> • Bentley Microstation • Speedycon • Triplan • Autocad Architectural
Computerized Maintenance Management System (CMMS)		<ul style="list-style-type: none"> • Maximo • Datastream 7i • API Pro
Authoring		<ul style="list-style-type: none"> • Adobe Acrobat • Illustrator • Wiki • Excel • Sharepoint • MacroMedia
Functional Engineering		<ul style="list-style-type: none"> • AutomationDesigner • Cosmos • AutomationFramework

Bild 2 Auswahl der oft im Kontext des Produktionssystementwurfs verwendeten Werkzeuge (nach [13])

Viele der Softwarewerkzeuge können allerdings nicht in angemessener Art und Weise ihre Daten als Datei austauschen. Viele der dazu nutzbaren Datenformate sind nicht kompatibel, d. h. die Dateien von einem Werkzeug sind mit einem anderen, nachgeschalteten Werkzeug weder offenbar noch importierbar. Die einzige Möglichkeit, die dann übrig bleibt, ist, die Informationen als PDF oder in Papierform auszutauschen. Allerdings bedeutet die Nutzung der sogenannten ‚Papierschnittstelle‘ einen enormen Aufwand für den Experten, da die Informationen aus der Datei händisch in sein Werkzeug einzutragen sind. Dieses Vorgehen ist fehleranfällig, zeitintensiv und damit kostenintensiv. Es sollte folglich bis zu einem Minimum reduziert bzw. gänzlich vermieden werden. Durch einen konsistenten und verlustfreien Datenaustausch kann der Entwurfsprozess von Produktionssystemen verkürzt und die Datenqualität

erhöht werden, d. h. Fehler beim manuellen Übertragen von Daten können vermieden werden, die unter Umständen erst bei der Inbetriebnahme vor Ort entdeckt werden.

Ein möglicher und üblicher Ansatz, um der ‚Papierschnittstelle‘ in einer heterogenen Werkzeuglandschaft zu begegnen, ist die Erstellung von bilateralen Schnittstellen zwischen einzelnen Werkzeugen. Das ermöglicht einen schnellen Austausch von Daten, jedoch nur zwischen diesen beiden Werkzeugen. Doch wie bereits oben erläutert, sind viele Werkzeuge in dem Entwurfsprozess involviert, was beim konsequentem Verfolgen dieses Ansatzes zu vielen Konvertern führen würde, die aktualisiert werden müssten, sobald sich die Werkzeugversion ändert (siehe Bild 3 links). Aus diesem Grund stellt sich der Datenaustausch mit offenen Datenformaten als der geeignetere Ansatz dar (siehe Bild 3 rechts).

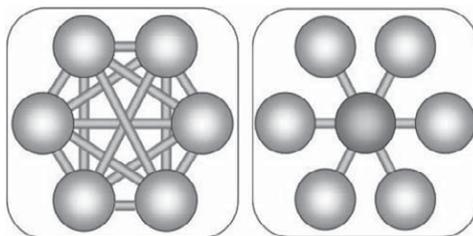


Bild 3 Schematische Darstellung der Anzahl an Schnittstellen in Werkzeugketten bei (links) bilateralem Datenaustausch und (rechts) Nutzung eines offenen Datenaustauschformats

Folglich wurden in den letzten Jahren eine Vielzahl von Datenaustauschformaten entwickelt, die das gesamte Spektrum von allgemeinen, viele Entwurfsdisziplinen und Informationsmengen umfassenden Formaten wie STEP, AutomationML oder XMI bis hin zu speziellen, für konkrete Informationsmengen optimierte Formaten wie JT, X3D oder XDDML abdecken [5, 6]. Als technologische Basis ist XML dabei stark verbreitet, da es neben der Menschenlesbarkeit auch eine automatische Verarbeitung, Erweiterbarkeit und Kombination mit anderen Formaten unterstützt. In der VDI-Richtlinie 3690 wird dazu ein praktisch anwendbarer Prozess für die Auswahl bzw. Entwicklung eines XML-basierten Datenaustauschformats für einen Anwendungsfall beschrieben [6]. Dieser Prozess soll an dieser Stelle auf die Informationen angewendet werden, die im Entwurfsprozess von Produktionssystemen zur Erstellung von Steuerungsprogrammen ausgetauscht werden müssen.

Das Datenaustauschformat AutomationML wurde in den letzten Jahren als ein Kandidat entwickelt, der zur Verbesserung von Entwurfsprozessen für Produktionssysteme genutzt werden kann. Es postuliert, alle relevanten Informationsmengen, die im Entwurfsprozess von Produktionssystemen zwischen Entwurfswerkzeugen auszutauschen sind (siehe Bild 1), abbilden zu können. Auch wenn dieser theoretischen Möglichkeit schwer widersprochen werden kann, ist bisher kein praktischer Anwendungsfall bekannt, der dieses Statement aus praktischer Sicht validiert.

Dieser Beitrag soll den beiden genannten Problemen, der Auswahl eines geeigneten Datenaustauschformats sowie der Präsentation einer praktischen Anwendung, nachgehen. Dazu werden in diesem Beitrag der allgemeine Prozess zur Auswahl bzw. Entwicklung eines Datenaustauschformats rekapituliert, der Entwurfsprozess für Produktionssysteme und die in ihm auszutauschenden Entwurfsdaten dargestellt, ein Labormodell zur praktischen Anwendung vorgestellt und die Modellierung der identifizierten Informationsmengen mit AutomationML beschrieben.