

# **Durchgehende Modellerstellung zwischen Simulations-, Visualisierungs- und Konstruktionswerkzeugen für die gesamtheitliche Planung von Produktions- und Intralogistiksystemen**

## ***Continuous Modelling between Simulation, Visualization and Construction Tools for the Holistic Planning of Production and Intralogistics Systems***

David Weigert, Paul Aurich, Tobias Reggelin,  
Otto-von-Guericke-Universität Magdeburg, Magdeburg (Germany),  
david.weigert@ovgu.de, paul.aurich@ovgu.de, tobias.reggelin@ovgu.de

**Abstract:** The automated and semi-automated generation of material flow simulation models is discussed and developed for decades. With AutomationML (AML) a new, open, object-oriented, XML-based storage and exchange format is given, which in addition will allow an exchange between visualization, construction and simulation tools. The application of AML within the different tools intends to promote a consistent design within the product and production creation process of production and intralogistics systems. The focus of the ongoing implementation phase is in the data exchange from the visualization tool for construction and simulation tool. First prototype implementations outline the benefits but also the limits of the automatic model generation via AML. Up to now simple conveyor systems from straight, curved conveyor belts, rotary tables and single stations from the visualization tool Tarakos – taraVR can automatically be modelled in the simulation tool Siemens – Plant Simulation and construction tool Autodesk – AutoCAD. This eliminates the use of expensive and optional interface libraries.

## **1 Einleitung und Problemstellung**

Aufgrund der immer kürzer werdenden Produktlebenszyklen nimmt die Vielseitigkeit, Schnelligkeit und Flexibilität der Produktentstehung stetig zu. Diese Entwicklungen haben direkten Einfluss auf die mitwirkenden Logistik- und Produktionsprozesse (Schenk 2014). Die zunehmende Digitalisierung und Automatisierung beschleunigt die Entwicklung der Durchgängigkeit verfügbarer Datenmodelle für die digitale Produktion und Logistik. Die Verwendung digitaler Werkzeuge aus den Bereichen Simulation, Visualisierung und Konstruktion erhöhen die Qualität der Planung, steigern die Effizienz und verkürzen die Produktentwicklung und

Markteinführung (Schenk 2014; Daft 2016; Klepper 1996; Lüder und Schmidt 2015). Diese Vorteile können nur dann vollständig ausgenutzt werden, wenn es ermöglicht wird, alle relevanten und bislang isolierten digitalen Methoden und Werkzeuge in ein integriertes Planungssystem zu überführen (Schreiber und Zimmermann 2011). Derzeit eingesetzte Werkzeuge decken nur bestimmte Funktionsbereiche innerhalb des Product Lifecycle Management (PLM) ab. Diese Konzentration auf einzelne Anwendungsbereiche des PLM ermöglicht eine hohe Spezialisierung der Werkzeuge. Hierdurch werden begrenzte Lösungsräume durch eine fehlende Schnittstellenintegration geschaffen. Ein Austausch von Planungsdaten in einer heterogenen Systemumgebung ist demnach nur begrenzt möglich (Faltinski et al. 2012; Rawolle et al. 2002). Im praktischen Einsatz besteht daher kein neutraler und IT-gestützter Ansatz für die durchgängige digitale Planung und Steuerung von Intralogistiksystemen und Produktionsbereichen. Es fehlt ein neutrales Austauschformat für die durchgängige Verfügbarkeit und gegenseitige Verwendbarkeit von Simulationsdaten, geometrischen Konstruktionspunkten und Visualisierungselementen. Im weiteren Verlauf werden das Konzept und Werkzeug zur durchgehenden Modellerstellung am Beispiel der Datenaustauschrichtung Visualisierung zu Simulation und Visualisierung zu Konstruktion näher erläutert.

## 2 Motivation und Zielstellung

Eine durchgehende Modellerstellung zwischen Simulations-, Visualisierungs-, und Konstruktionswerkzeugen für die gesamtheitliche Planung von Produktions- und Intralogistiksystemen beschreibt die Verknüpfung der drei wichtigsten digitalen Werkzeuge. Sowohl das Expertenwissen im Umgang mit den Werkzeugen als auch die inhaltliche Komplexität der einzelnen Werkzeuge sind als sehr hoch einzustufen. Durch separate Querschnittsfunktionen und proprietäre Softwarelösungen der einzelnen Tools ist es schwierig, einen gemeinsamen Übergabepunkt zu definieren. Das vorgestellte Konzept und Werkzeug beschreibt die Entwicklung einer anwendungsorientierten und quelloffenen Middleware. Das Ziel ist die Entwicklung eines durchgehenden, digitalen Austauschsystems (Abb. 1).

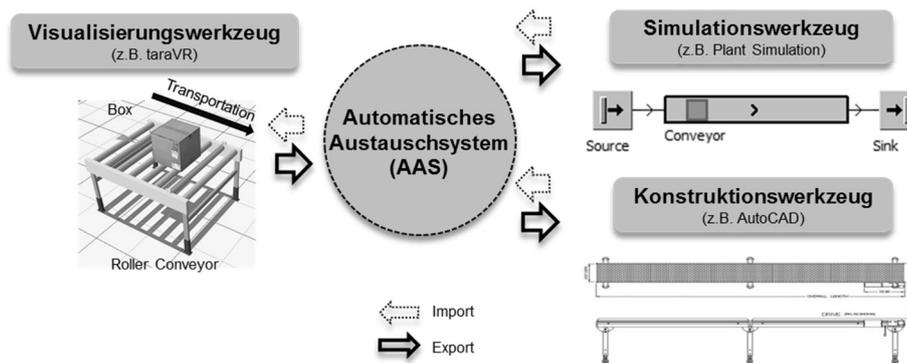


Abbildung 1: Konzeptdarstellung des durchgängigen Austauschsystems

Hierbei steht die verlustfreie und beschleunigte Konvertierung und Modellierung innerhalb der verschiedenen Werkzeuge im Fokus. Folgende Vorteile ergeben sich aus dem Einsatz von einem automatischen Austauschsystem:

- Bestehende Simulations-, Visualisierungs- und Konstruktionswerkzeuge bleiben im Unternehmen, eine kostenintensive Neuinvestition wird verhindert.
- Produktivität und Kostenreduktion können durch den Einsatz der Entwicklung erreicht werden, da die individuellen Vorteile der Werkzeuge kombiniert werden.

Die Visualisierung, Modellierung und Simulation von realen intralogistischen Systemen werden beschleunigt, da der überwiegend manuelle und kostenintensive Aufwand bei einer erneuten Modellerstellung reduziert wird. Weiterführend werden das Konzept und Werkzeug für die durchgehende und ganzheitliche Modellerstellung am Beispiel von dem Simulationswerkzeug Plant Simulation - Tecnomatix, und dem Konstruktionswerkzeug AutoCAD - Autodesk aufgezeigt. Das Visualisierungswerkzeug taraVR - Tarakos dient hierbei als Ausgangslage für die getätigten Untersuchungen.

### **3 Stand der Wissenschaft und Technik**

Die Standardisierung von Systemen, Prozessen und deren Komponenten ist ein wesentliches Element in der Beherrschung von Komplexität sowie Kontrolle und Gestaltung von zukünftigen, digitalen Herausforderungen (Drath 2010; Eigner und Stelzer 2009). Lag der bisherige Fokus der kombinierten Planungsphasen auf einer Verkürzung im Time-to-Market von Produkten durch die Integration von Produkt-, Prozess- und Produktionssystemplanung, besteht der heutige Anspruch darin, frühzeitig belastbare Daten aus der Produktentstehung für die integrierte Fabrikssystementwicklung und -gestaltung zu nutzen (Schenk 2014; Schenk und Schumann 2008). Die Idee von kombinierten Planungsphasen ist bereits seit Jahren durch moderne und leistungsfähige Werkzeuge beschrieben (Dangelmaier 2013). Es existiert hierbei eine Vielzahl an Spezialfällen. Diese Spezialfälle bestehen aus individuell ausgearbeiteten Lösungen und sind für konkrete Problemstellungen ausreichend. Es fehlt der gesamtheitliche Planungsansatz, um eine Integration aller beteiligten Prozesse voranzutreiben. Erste Ansätze zur Generierung von layout-basierten Modellen finden sich bei Lorenz und Schulze (1995). Splanemann et al. (1995) beschreiben einen Ansatz, um sowohl strukturelle als auch Daten aus einem Produktionsplanungs- und Steuerungssystem (PPS) zu nutzen. Eine erste Klassifizierung von automatischen Modellgenerierungsansätzen stellt Eckardt (2002) bereit. Der Autor unterscheidet zwischen parametrischen, strukturellen und hybriden Ansätzen. Eine zusätzliche Möglichkeit zur Klassifizierung von Ansätzen der Modellgenerierung bilden die Klassifikationsmöglichkeiten von Modellgenerierungsansätzen nach Straßburger et al. (2010). Bestrebungen zur durchgehenden digitalen Planung und Steuerung werden mit der Enterprise Application Integration (EAI) und den serviceorientierten Architekturen (SOA) realisiert (Aier 2006; Bieberstein 2008; Kaib 2004). Die EAI stellt integrierte Geschäftsprozesse entlang der Wertschöpfungskette dar. Unternehmensanwendungen verschiedener Generationen und Systemarchitekturen können hierbei über ein gemeinsames Netzwerk interagieren. Die SOA beschreibt eine Methode, die vorhandene IT-Komponenten abkapselt und koordiniert. Hierdurch werden bestehende Dienste konsolidiert und zu einem höheren Dienst zusammengefasst.

Die Ziele von EAI als auch SOA sind langfristig die Senkung der Kosten bei der Entwicklung von Produktionsprozessen und die Steigerung der Flexibilität der Geschäftsprozesse. Grund für die geringe Akzeptanz und Weiterentwicklung der Methoden sind hohe Anforderungen an Datensicherheit und Kontinuität der Werkzeugentwicklung sowie unsichere Systeme und Produktentwicklungsprozesse (Fay 2006; Raupricht et al. 2002). Das Zusammenwirken verschiedener digitaler Planungswerkzeuge innerhalb des Produktlebenszyklus wird oftmals auch unter dem Begriff „Digitale Fabrik“ zusammengefasst (VDI 2008; Wenzel et al. 2003). Auch hier findet sich eine Verknüpfung von unterschiedlichen Planungswerkzeugen. Jedoch fehlt der Einsatz von durchgehenden Planungstools.

### 3.1 Speicher- und Austauschformat

Die teil- und vollautomatisierte Modellgenerierung kann auch als datengetriebene Modellgenerierung bezeichnet werden (Bergmann 2014). Hierbei ist die Herkunft und Art der Daten und Informationen von höchster Bedeutung. Die zurzeit bekanntesten Standards zur automatischen Modellgenerierung sind die Datenformate Simulation Data Exchange Format (SDX) (Sly und Moorthy 2001) sowie Core Manufacturing Simulation Data (CMSD) (Bergmann et al. 2010; Lee 2015). Das hierarchisch strukturierte SDX-Format dient ausschließlich der Bereitstellung von Layoutinformationen. Mit dem quelloffenen, XML-basierten CMSD-Format können sowohl layout- als auch prozessbezogene Informationen übermittelt werden. Die Problemstellung einer Implementierung von umfangreichen Steuerungs- und Routingstrategien und das komplexe Systemverhalten sind jedoch auch mit diesen Datenformaten nicht vollständig lösbar (Bergmann 2014; Bergmann et al. 2010). Durch AML ist ein quelloffenes, objektorientiertes, XML-basiertes Speicher- und Austauschformat entwickelt worden. AML besitzt die technischen Voraussetzungen für die Modellierung von Produktions-, Intralogistik- und Robotersysteme. Grundlegend arbeitet AML mit einer Verknüpfung von Rollenprofilen (Hoernicke et al. 2016; Hundt et al. 2009; Lüder und Schmidt 2015). Bisher können mit AML Topologie, Geometrie, Kinematik und das Verhalten von Systemkomponenten beschrieben werden. Die hierarchische Abbildung der Topologie eines Planungsgegenstandes erfolgt mittels Computer Aided Engineering Exchange (CAEX). Geometrie und Kinematik können durch Collada-Dateien einzelnen Systemkomponenten zugeordnet werden. AML ist anpassbar und flexibel gestaltet, es bietet die Möglichkeit weitere XML-Formate einzubinden (Hundt et al. 2009). Darüber hinaus besitzt das AML-Format eine inhärente verteilte Datenstruktur. Die Informationen werden statt in einem monolithischen XML-Dokument als Einzeldokumente gespeichert. Die Wiederverwendbarkeit einzelner Systemkomponenten und der Aufbau von Element-Bibliotheken werden somit erleichtert (Lüder und Schmidt 2015).

## 4 Automatisches Austauschsystem

Das automatische Austauschsystem (AAS) bildet die Basis für den gemeinsamen Daten- und Informationsaustausch. Das AAS wird durch ein System von Abbildungsregeln und einem dazugehörigen Benutzerinterface definiert. Innerhalb des Austauschsystems wird das zu übertragende Modellelement mit bestehenden oder neuen Rollenprofilen aus der hierfür entwickelten AutomationML-Rollenbibliothek belegt.

Neben der Übertragung von Modellelementen ist es von Bedeutung, deren Attribute wie Lage, Transportrichtung, Geschwindigkeit und weitere Zustandsbeschreibungen zu übernehmen. Des Weiteren müssen die Beziehungen zwischen den Elementen übertragen werden. Ziel ist es, eine unabhängige Modellerstellung über geschlossene Softwarepakete hinweg zu realisieren. Basierend auf der Klassifikationsmöglichkeit von Modellgenerierungsansätzen nach Straßburger et al. (2010) kann die dargestellte Entwicklung folgendermaßen eingeordnet werden:

- Einsatzfall: planungsbegleitend (taktisch)
- Fokussiertes Gewerke: Fördertechnik
- Grad der Automatisierung: teilautomatisch
- Ansatz: direkter generischer Aufbau
- Unterstützung bei Modellerstellung
- Schnittstelle: text- und XML-basiert

Bevor Modelle automatisch generiert werden können, ist es notwendig die Bausteine und Elemente der unterschiedlichen Werkzeuge aufeinander zu mappen. Das Datenmapping beschreibt den Prozess der Abbildung von Datenelementen zwischen unterschiedlichen Datenmodellen (Bogdan et al. 2008). Um dieses Datenmapping zu ermöglichen, wird ein sogenannter Regelinterpreter entworfen. Aus den jeweiligen Visualisierungs-, Konstruktions- und Simulationswerkzeugen (VKS-Werkzeugen) werden die Bibliotheken mit sämtlichen Bausteinen und deren Parametern exportiert und gemappt. Auf diese Art und Weise werden Regeln für die Übertragung definiert.

#### 4.1 Automatisierte Modellgenerierung Simulationswerkzeug

Für das Simulationswerkzeug wird Plant Simulation der Firma Tecnomatix ausgewählt. Plant Simulation verfügt über eine Vielzahl von Lizenzarten und kostenpflichtigen Bibliotheken. Kostenintensive Lizenzarten wie „Professional“ in Kombination mit dem „Interface Package“ werden für die Entwicklung einer gemeinsamen Datenschnittstelle vermieden, da sie das Hauptziel einer quelloffenen und kostenneutralen Kommunikationsschnittstelle nicht erfüllen. Die Lizenzart „Standard“ bildet die Grundlage für die Schnittstelle. Ein Vorteil ist die Aufwärtskompatibilität der einfachen Datenstruktur (.txt-Datei). Ein Nachteil ist die aufwendigere Datenaufbereitung im AAS vor dem Im- und Export in das Simulationswerkzeug. Die Zeichenfolgen müssen einer vorgegebenen Form entsprechen, um exakt codiert und dechiffriert werden zu können. Die ermittelte Minimalanforderung für die geforderten Parameter aus dem Simulationswerkzeug für die Kommunikation mit dem Austauschsystem beinhalten:

- Objektinformationen: Name, Objektart, Geometrie
- Layoutinformationen: Lage und Verbindung zu anderen Elementen
- Materialflussparameter: Zeitverbrauch, Routing

Ein Direktimport von AML-Dateien in das Simulationswerkzeug wird nicht unterstützt. Für das Simulationswerkzeug wird daher eine Schnittstelle entwickelt (s. Alg. 1). Dieser steuert die Identifizierung der Objekte aus der Visualisierung und die Implementierung in das Simulationswerkzeug. Die Grundlage bildet eine Textdatei aus dem vorhergehenden Datenmapping.

**Algorithmus 1:** Pseudocode für Modellgenerierung I – Simulationswerkzeug

---

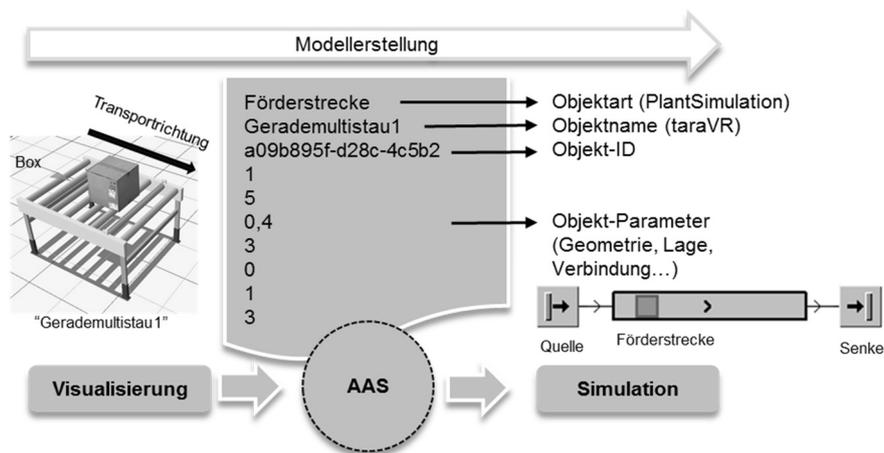
```

open textfile
while not textfile.end
read bibliothek, objecttyp, x-pos., y-pos., name
create object from bibliothek.objecttyp at x-pos.,y-pos.
  if next textfile.line # trennzeichen
  do
  read parameter and parameter.value
  case parameter of
  case "input-id"
  if object.exists(input-id)
  connet object with input-object
  end if
  case "output-id"
  if object.exists(output-id)
  connet object with output-object
  end if
  else
  set object.parameter = parameter.value
  end case
  until next textfile.line # trennzeichen
  end if
endwhile
close textfile

```

---

Das AAS für die Kopplung zwischen Simulation und Visualisierung arbeitet nach einer hierarchischen Verwaltung des Zugriffs auf eine gemeinsame Ressource als Master/Slave-System. Die Masterdatei, ein geschlossenes Simulationsprojekt, enthält alle benötigten Methoden zur Modellerstellung, zum Modellexport, zum Bibliotheks-export sowie zum Import der Bausteine. Die Datei kann selbst nicht geändert werden, lediglich das davon erstellte Abbild.



**Abbildung 2:** Beispiel der Modellerstellung – Visualisierung zur Simulation

Für jeden bisher gemappten Baustein wird ein neuer Quellcodeabschnitt erzeugt. Dies führt langfristig dazu, dass selbst individuelle Bausteinkästen der Endanwender ausgelesen und dynamisch angesprochen werden können. Die Abbildung 2 verdeutlicht die praktische Umsetzung und ergänzt den Pseudocode für die Modellgenerierung im Simulationswerkzeug. Nach der Generierung einer Textdatei aus den Visualisierungsdaten wird das Objekt im Master des Simulationswerkzeuges erzeugt. Hierzu ist es notwendig im vorherigen Schritt die bestehenden Bibliotheken aus dem Simulationswerkzeug auszulesen. Dies ist notwendig um das Objekt der Visualisierung als korrektes Objekt in der Simulation abzubilden. Dies passiert ebenfalls durch das AAS. Das AAS ist als ganzheitliche Middleware zu verstehen. Neben der automatisierten Modellerzeugung in das Simulationswerkzeug wird darin auch die Umsetzung in das Konstruktionswerkzeug verarbeitet.

## 4.2 Automatisierte Modellgenerierung durch das Konstruktionswerkzeug

AutoCAD der Firma Autodesk besitzt deutlich mehr Modifikationsmöglichkeiten im Bereich der Schnittstellenerstellung. Durch den Import von anderen CAD-Tools u. a. mit den Formaten aus 3D Studio, Autodesk Inventor und SolidWorks ist eine breite Basis für den Import vorhanden. Für die Entwicklung zum AAS ist die Vielzahl jedoch nicht von Bedeutung.

Die Grundlage der Entwicklung bildet das neutrale Austauschformat AML. Hierfür werden eigenständige Befehle für Import und Export entwickelt. Das Ziel der Durchgängigkeit zum Visualisierungswerkzeug wird dadurch sichergestellt, dass die Schnittstelle im Format \*.net programmiert und verwendet wird.

Die ermittelte Minimalanforderung für die geforderten Parameter aus dem Konstruktionswerkzeug für die Kommunikation mit dem Austauschsystem beinhaltet:

- Objektinformationen: Name, Geometrie
- Layoutinformationen: Lage (Verbindungen sind nicht notwendig)
- Detailinformation zur Geometrie werden nicht benötigt, können jedoch optional in einer AML-Datei durch das Collada-Format angefügt werden.

Ein Direktimport von AML-Dateien wird durch die Autodesk Softwaresuite nicht unterstützt. Eine eigenständige Schnittstellenumgebung wird daher programmiert.

### *Algorithmus 2: Pseudocode für Modellgenerierung I – Konstruktionswerkzeug*

---

```

define fcn = rootnode.childnodes and scn = FCN.childnodes
open xmlfile
while rootnode has fcn
  read SCN.objecttyp, SCN.x-pos, SCN.y-pos
  create object from blocktable.objecttyp at x-pos., y-pos.
  if object is dynamic
    for i = 1 to SCN.parameter.count
      read SCN.parameter(i) and parameter(i).value
      set object.parameter(i)=parameter(i).value
    next
  end if
end while
close xmlfile

```

---

Innerhalb des Werkzeuges werden nicht nur statische Blöcke angesprochen. Die Blöcke sind vom Nutzer vordefinierte, zweidimensionale Körper. Sie werden in einer Blocktabelle gespeichert. Hiermit wird das Ziel verfolgt, eine dynamisierte Anpassung der Körper innerhalb vom Konstruktionswerkzeug zu erhalten. Eine statische 1:1-Beziehung von Objekten aus unterschiedlichen Werkzeugen entfällt (s. Alg. 2). Um Blöcke dynamisch erstellen und parametrisieren zu können, wird die Schnittstelle in C#.NET programmiert und in eine dll-Datei (Dynamic Link Library) gekapselt. Die gekapselte Programmumgebung wird als AAS.dll geführt und im Konstruktionswerkzeug aufgerufen. Am Beispiel (Abb. 3) wird die praktische Anwendung der Modellerstellung zwischen Visualisierungs- und Konstruktionswerkzeug deutlich. Der XML-Code beschreibt den in der Visualisierung abgebildete Rollenförderer. Der Code wird über die in AutoCAD aufgerufene dll-Datei ausgelesen und im Zielwerkzeug generiert. Innerhalb der Konstruktionsumgebung wird dieser Rollenförderer als dynamischer Block interpretiert. Bei Bedarf kann eine Anpassung des 2D-Layouts vorgenommen werden.

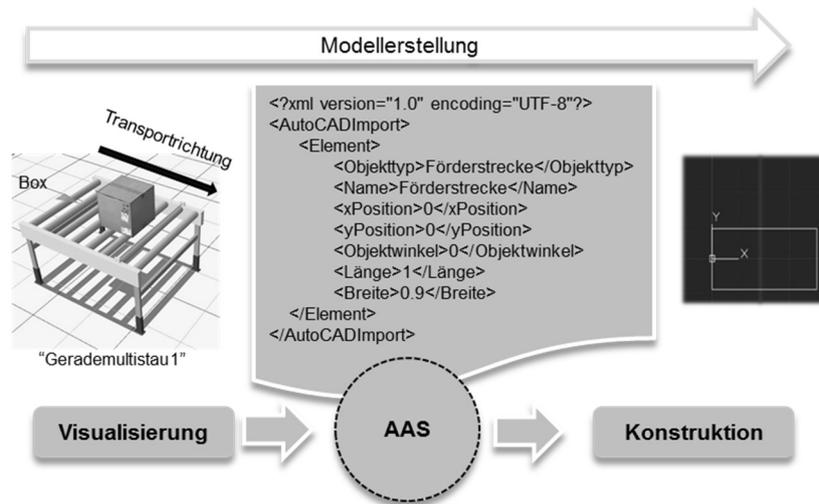


Abbildung 3: Beispiel der Modellerstellung – Visualisierung zur Konstruktion

## 5 Zusammenfassung und Ausblick

Das beschriebene Konzept und Werkzeug stellt einen umfassenden Ansatz für die automatische und quelloffene Modellerstellung und Datenübertragung zwischen unterschiedlichen Werkzeugen dar. Damit ist eine Möglichkeit geschaffen, individuelle AML-Bibliotheken für Simulation, Visualisierung und Konstruktion einer wiederholten Nutzung zuzuführen. Erste Ergebnisse zeigen, dass die Umsetzung erfolgreich möglich ist. Ebenso kommen hierbei keine proprietären Softwarelösungen zum Einsatz. Mit der Möglichkeit, die Modellerstellung innerhalb der unterschiedlichen Werkzeuge zu vereinen, kann es in Zukunft möglich sein, genaue und schnelle Modelle aus einer Hand zu produzieren. Dadurch wird eine kostspielige, fehleranfällige und zeitintensive mehrmalige Modellerstellung schon jetzt vermieden. Der erreichte Entwicklungsstand bildet die Grundlage für die Übertragung der

Modelle in das entsprechende Zielwerkzeug, beschrieben durch Simulation und Konstruktion. Weitere Schritte sind die Detaillierung und Standardisierung des automatisierten Austauschsystems. Die Zusammenarbeit mit Endanwendern aus der Praxis macht deutlich, welche weiteren Anforderungen an Funktionsumfang und Wirkungsweise das AAS zukünftig aufweisen muss. Hierbei sollen auch die Frage nach der Implementierung und Übermittlung der Arbeitspläne und die damit einhergehende Steuerung und das Routing der Elemente im Simulationswerkzeug erforscht werden. Die Rückführung der Informationen und Modelle aus dem Simulations- und Konstruktionswerkzeug in das Visualisierungswerkzeug werden zum aktuellen Zeitpunkt manuell im AAS vorgenommen. Ziel ist es, die Daten aus der Konstruktion durch ein erneutes Datenmapping im AAS zu verarbeiten. Für die Simulation ist die Einbindung des durch die Simulationsläufe erhaltenen Ereignisprotokolls angedacht. Die beschriebenen und geplanten Umfänge und Vorgehensweisen gilt es zukünftig zu vereinfachen und dynamischer zu gestalten. Gleichzeitig ist eine deutliche Steigerung der Benutzerfreundlichkeit im Umgang mit dem Tool eingeplant.

## Literatur

- Aier, S. (Hrsg.): Enterprise application integration: Serviceorientierung und nachhaltige Architekturen. Berlin: GITO-Verlag 2006.
- Bogdan, A.; Chiticariu, L.; Miller, R.J.; Tan, W.-C.: Muse: Mapping Understanding and deSign by Example. In: IEEE: IEEE 24th International Conference on Data Engineering, Cancun, Mexico. New York: IEEE 2008, S. 10-19.
- Bergmann, S.: Automatische Generierung adaptiver Modelle zur Simulation von Produktionssystemen. Ilmenau: Univ.-Verl. 2014.
- Bergmann, S.; Fiedler, A.; Straßburger, S.: Generierung und Integration von Simulationsmodellen unter Verwendung des Core Manufacturing Simulation Data (CMSD) Information Model. In: Zülch, G.; Stock, P. (Hrsg.): Integrationsaspekte der Simulation: Technik, Organisation und Personal. Karlsruhe: KIT Scientific Publishing 2010, S. 461-468.
- Bieberstein, N.: Executing SOA: A practical guide for the service-oriented architect. Upper Saddle River, NJ: IBM Press/Pearson plc 2008.
- Daft, R.L.: Organization theory & design. Boston, MA: Cengage Learning 2016.
- Dangelmaier, W.: Fertigungsplanung: Planung von Aufbau und Ablauf der Fertigung; Grundlagen, Algorithmen und Beispiele. Berlin [u. a.]: Springer 2013.
- Drath, R. (Hrsg.): Datenaustausch in der Anlagenplanung mit AutomationML: Integration von CAEX, PLCopen XML und COLLADA. Berlin: Springer 2010.
- Eckardt, F.: Ein Beitrag zu Theorie und Praxis datengetriebener Modellgeneratoren zur Simulation von Produktionssystemen. Aachen: Shaker 2002.
- Eigner, M.; Stelzer, R.: Product Lifecycle Management: Ein Leitfaden für Product Development und Life Cycle Management. Berlin, Heidelberg: Springer 2009.
- Faltinski, S.; Niggemann, O.; Moriz, N.; Mankowski, A.: AutomationML: From data exchange to system planning and simulation. In: IEEE: Industrial Technology (ICIT), Athen. New York: IEEE 2012 S. 378-383.
- Fay, A.: Reduzierung der Engineering-Kosten für Automatisierungssysteme. Industrie Management (22) (2006) 22, S. 29-32.
- Hoernicke, M.; Messinger, C.; Arroyo, E.; Fay, A.: Topologiemodelle in AutomationML. Atp-Edition: automatisierungstechnische Praxis; Organ der GMA (VDI-VDE-Gesellschaft Meß- und Automatisierungstechnik) und der

- NAMUR (Interessengemeinschaft Automatisierungstechnik der Prozessindustrie) 58 (2016) 5/2, S. 28-41.
- Hundt, L.; Lüder, A.; Barth, H.: Anforderungen an das Engineering durch die Verwendung von mechatronischen Einheiten und AutomationML. SPS/IPC/DRIVES 2009, S. 341-349.
- Kaib, M.: Enterprise Application Integration: Grundlagen, Integrationsprodukte, Anwendungsbeispiele. Wiesbaden: Dt. Univ.-Verl. 2004.
- Klepper, S.: Entry, exit, growth and innovation over the product life cycle. Estados Unidos: American Economic Review 1996.
- Lee, Y.-T.T.: A Journey in Standard Development: The Core Manufacturing Simulation Data (CMSD) Information Model. Journal of research of the National Institute of Standards and Technology 120 (2015), S. 270-279.
- Lorenz, P.; Schulze, T.: Layout based model generation. In: Lilegdon, W.R. (Hrsg.): Proceedings of the 27th Conference on Winter Simulation, Arlington, VA, USA. New York: IEEE 1995, S. 728-735.
- Lüder, A.; Schmidt, N.: AutomationML in a Nutshell. In: Vogel-Heuser, Birgit; Bauernhansl, Thomas; Hompel, Michael ten (Hrsg.): Handbuch Industrie 4.0: Produktion, Automatisierung und Logistik. Wiesbaden: Springer Fachmedien Wiesbaden 2015, S. 1-46.
- Raupricht, G.; Haus, C.; Ahrens, W.: PLT-CAE-Integration in gewerkeübergreifendes Engineering und PlantMaintenance. atp – Automatisierungstechnische Praxis 44 (2002) 2, S. 50-62.
- Rawolle, J.; Ade, J.; Schumann, M.: XML als Integrationstechnologie bei Informationsanbietern im Internet. Wirtschaftsinformatik 44 (2002) 1, S. 19-28.
- Schenk, M.: Fabrikplanung und Fabrikbetrieb: Methoden für die wandlungsfähige: Berlin, Heidelberg: Springer 2014.
- Schenk, M.; Schumann, M.: Interoperable Testumgebung für verteilte domänenübergreifende Anwendungen. In: Scholz-Reiter, B. (Hrsg.): Technologiegetriebene Veränderungen der Arbeitswelt. Berlin: GITO-Verl. 2008, S. 155-169.
- Schreiber, W.; Zimmermann, P.: Virtuelle Techniken im industriellen Umfeld: Das AVILUS-Projekt - Technologien und Anwendungen. Berlin, Heidelberg: Springer 2011.
- Sly, D.; Moorthy, S.: Simulation data exchange (SDX) implementation and use. In: Peters, B.A. (Hrsg.): Proceedings of the 2001 Winter Simulation Conference, Arlington, VA, (USA). New York: IEEE 2001, S. 1473-1477.
- Splanemann, R.; Roth, M.; Soravia, S.: Einsatz der Materialflußsimulation zur Planung, Analyse und Optimierung von verfahrenstechnischen Produktionsanlagen. Chemie Ingenieur Technik 67 (1995) 9, S. 1107-1108.
- Straßburger, S.; Bergmann, S.; Müller-Sommer, H.: Modellgenerierung im Kontext der Digitalen Fabrik - Stand der Technik und Herausforderungen. In: Zülch, G.; Stock, P. (Hrsg.): Integrationsaspekte der Simulation: Technik, Organisation und Personal. Karlsruhe: KIT Scientific Publishing, 2010 S. 37-44.
- VDI 4499 Blatt 1: Digitale Fabrik: Grundlagen. Berlin: Beuth 2008.
- Wenzel, S.; Hellmann, A.; Jessen, U.: e-Services - a part of the "Digital Factory". In: Bley, H. (Hrsg.): Proceedings of the 36th CIRP International Seminar on Manufacturing Systems, Saarbrücken. Saarbrücken: Univ. des Saarlandes 2003, S. 199-203.