

Ahrens, Volker; Hieronymus, Martin

Working Paper

Modellbasierte Unterstützung der Implementierung mechatronischer Systeme

Arbeitspapiere der Nordakademie, No. 2013-04

Provided in Cooperation with:

Nordakademie - Hochschule der Wirtschaft, Elmshorn

Suggested Citation: Ahrens, Volker; Hieronymus, Martin (2013) : Modellbasierte Unterstützung der Implementierung mechatronischer Systeme, Arbeitspapiere der Nordakademie, No. 2013-04, Nordakademie - Hochschule der Wirtschaft, Elmshorn

This Version is available at:

<https://hdl.handle.net/10419/87718>

Standard-Nutzungsbedingungen:

Die Dokumente auf EconStor dürfen zu eigenen wissenschaftlichen Zwecken und zum Privatgebrauch gespeichert und kopiert werden.

Sie dürfen die Dokumente nicht für öffentliche oder kommerzielle Zwecke vervielfältigen, öffentlich ausstellen, öffentlich zugänglich machen, vertreiben oder anderweitig nutzen.

Sofern die Verfasser die Dokumente unter Open-Content-Lizenzen (insbesondere CC-Lizenzen) zur Verfügung gestellt haben sollten, gelten abweichend von diesen Nutzungsbedingungen die in der dort genannten Lizenz gewährten Nutzungsrechte.

Terms of use:

Documents in EconStor may be saved and copied for your personal and scholarly purposes.

You are not to copy documents for public or commercial purposes, to exhibit the documents publicly, to make them publicly available on the internet, or to distribute or otherwise use the documents in public.

If the documents have been made available under an Open Content Licence (especially Creative Commons Licences), you may exercise further usage rights as specified in the indicated licence.



ARBEITSPAPIERE DER NORDAKADEMIE

ISSN 1860-0360

Nr. 2013-04

Modellbasierte Unterstützung der Implementierung mechatronischer Systeme

Prof. Dr.-Ing. Volker Ahrens, Martin Hieronymus

Oktober 2013

Eine elektronische Version dieses Arbeitspapiers ist verfügbar unter:
www.nordakademie.de/arbeitspapier.html



Kölnner Chaussee 11
25337 Elmshorn
www.nordakademie.de

Modellbasierte Unterstützung der Implementierung mechatronischer Systeme

Volker Ahrens, Martin Hieronymus

NORDAKADEMIE – Hochschule der Wirtschaft, Elmshorn

Abstract: Die Entwicklung mechatronischer Systeme basiert auf einer zunehmenden Unterstützung durch Modelle. Dies beginnt inzwischen schon bei der Identifikation der Anforderungen und reicht weit in die Implementierung hinein. Um diese Implementierungsphase geht es im vorliegenden Beitrag. Am Beispiel der diskreten Simulation wird untersucht, ob es möglich ist, Modelle, die im Zuge der Planung und Konzeption mechatronischer Systeme entwickelt werden, auch in den Implementierungsphasen weiterzuverwenden. Dass dies prinzipiell möglich ist, wurde im Rahmen von Grundlagenforschungen an der TU Dresden bereits bewiesen. Die hier vorgestellten Untersuchungen bauen darauf auf und konkretisieren sie im Hinblick auf die praktische Anwendung.

1 Anlass

Technische Produkte sind heute ebenso wie die zu ihrer Herstellung erforderlichen Produktionssysteme zumeist eine Kombination aus Mechanik, Elektrotechnik, Elektronik, Datenverarbeitungs-, Informations- und Kommunikationstechnik. Daher bezeichnet man sie mit dem Kofferwort „Mechatronik“¹, in dem Teile der Begriffe „*Mechanical Engineering*“ und „*Electronic Engineering*“ miteinander verschmolzen sind. Aufgrund dieser Verbindung von Disziplinen, die jede für sich bereits komplex sind, hat die Komplexität der Produkte und Produktionssysteme signifikant zugenommen. Eine der Konsequenzen aus diesem Trend ist die Einsicht, dass die Modelle, Methoden und Werkzeuge zur Entwicklung solcher Systeme weiter ausgebaut und konsequenter als bisher eingesetzt werden müssen.

Vor diesem Hintergrund fordert und fördert das einflussreiche International Council on Systems Engineering (INCOSE) die Ablösung einer bisher zu stark dokumentenorientierten Entwicklung durch eine modellbasierte Entwicklung: „Model-based systems engineering (MBSE) is the formalized application of modeling to support system requirements, design, analysis, verification and validation activities beginning in the conceptual design phase and continuing throughout development and later life cycle phases. MBSE is part of a long-term trend toward model-centric approaches adopted by other engineering disciplines, including mechanical, electrical and software. In particular,

¹ Alternativ werden solche Systeme inzwischen auch als „cyber-physical systems“ bezeichnet (vgl. z. B. Ahrens 2012, 30 f.).

MBSE is expected to replace the document-centric approach that has been practiced by systems engineers in the past and to influence the future practice of systems engineering by being fully integrated into the definition of systems engineering processes“ (INCOSE 2007, 15).

In den Kernphasen technischer Entwicklungen werden Modelle und Werkzeuge zur Unterstützung ihrer Anwendung wie z. B. CAD²-Systeme zum Entwurf der Geometrie, Finite-Elemente-Methoden zur Festkörperberechnung oder die Simulation zur Abbildung dynamischer Vorgänge schon heute konsequent angewendet. Im Zuge einer Ausweitung der Modellunterstützung technischer Entwicklungen im Sinne des MBSE sollen Modelle zukünftig auch in den frühen und in den späten Phasen zur Anwendung kommen. Dies betrifft am Anfang von Entwicklungen z. B. die Anforderungsdefinition und am Ende die Implementierung, die vor allem deshalb bisher noch nicht im gleichen Umfang von Modellen unterstützt wird wie die vorausgehende Planung und Konzeption, weil nun bereits die realen Systeme zur Verfügung stehen, die ihre virtuellen Repräsentanten überflüssig machen.

Wenn sowohl die frühen als auch die späten Phasen technischer Entwicklungen zukünftig ebenso konsequent durch Modelle unterstützt werden sollen wie die mittleren Phasen, so erscheint es naheliegend, von den bereits vorhandenen Modellen auszugehen und deren Anwendung auf die frühen und auf die späten Phasen auszudehnen. Dafür sprechen vor allem wirtschaftliche Gründe, denn die zur Unterstützung des Modelleinsatzes verwendeten Werkzeuge wie z. B. CAD-Systeme oder Simulationsprogramme sind in der Regel nicht ganz billig; den Nutzen bereits vorhandener Werkzeuge auszubauen ist also attraktiver als zusätzliche Werkzeuge zu beschaffen. Darüber hinaus verspricht die Ausweitung der Anwendung von bereits im Einsatz befindlichen Modellen eine Vermeidung zusätzlicher Schnittstellen und damit den Verlust von bereits erhobenen und generierten Daten. Schließlich werden vor allem die Werkzeughersteller und -vertreiber ein großes Interesse haben, die Anwender ihrer Produkte von zusätzlichen Nutzenpotenzialen zu überzeugen, um den Absatz zu fördern. Vor diesem Hintergrund wird aus einer wissenschaftlichen und insoweit unabhängigen Perspektive genau zu prüfen sein, welche Argumente inhaltlich bestehen können und welche Argumente möglicherweise in erster Linie dem Marketing dienen.

Immerhin gibt es einen prinzipiellen Anfangsverdacht, der gegen eine Ausweitung der Verwendung bereits genutzter Modelle spricht, nämlich die Erfahrung, dass Modelle stets nur für begrenzte Anwendungsfälle tauglich sind, so dass zur Lösung komplexer Aufgaben eine große Zahl verschiedener Modelle erforderlich ist. Das spricht nicht generell gegen die Möglichkeit einer Ausweitung vorhandener Modelle, begründet aber eine kritische Haltung im Hinblick auf deren Sinnhaftigkeit.

Vor diesem Hintergrund können zwei aufeinander aufbauende Fragestellungen formuliert werden, nämlich (1) die nach der *Möglichkeit* einer Ausweitung der Verwendung von bereits vorhandenen Modellen sowie (2) die nach der *Notwendigkeit* dieses Ansatzes. Zur Beantwortung der erstgenannten Frage sollen die folgenden Untersuchungen einen Beitrag leisten. Die Beantwortung der zweitgenannten Frage ist einem weiteren

² CAD: Computer Aided Design

2 Situationsanalyse

Das Diagramm stellt das V-Modell der Systementwicklung dar, das die Beziehung zwischen Entwicklung und Integration verdeutlicht. Die linke Seite des V steht für die Entwicklung, die rechte Seite für die Integration und die Basis für die Implementierung und Verifikation.

Links (Entwicklung):

- Ziele:** Der Ausgangspunkt der Entwicklung.
- A: Modellierung und Spezifikation:** Die erste Phase der Entwicklung.
- F: Modellierung und erste Simulation:** Die zweite Phase der Entwicklung.
- L: Disziplinspezifische Modellierung und Simulation:** Die dritte Phase der Entwicklung.

Rechts (Integration):

- Virtuelle Tests:** Tests, die während der Entwicklung durchgeführt werden.
- Physische Tests:** Tests, die an physischen Prototypen durchgeführt werden.
- Hybride Tests (z. B. HiL):** Tests, die sowohl virtuelle als auch physische Elemente umfassen.

Unten (Implementierung und Verifikation):

- Dienstleistung:** Die Basis der Implementierung.
- Mechanik:** Die mechanische Implementierung.
- Elektrik/Elektronik:** Die elektrische Implementierung.
- Software:** Die softwaretechnische Implementierung.
- Modellierung und Analyse:** Die abschließende Phase der Implementierung.

Legende:

- A: Anforderung
- F: Funktion
- L: logisches Lösungselement
- P: physisches Element

Mit der Systems Modeling Language (SysML) wurde eine relativ neue Modellierungstechnik entwickelt, von deren Einsatz man sich zu Beginn einer Entwicklung eine systematischere Identifikation und Formulierung der Anforderungen an komplexe Produkte und Produktionssysteme verspricht als von bisher üblichen Dokumenten wie dem Lasten- und Pflichtenheft (vgl. z. B. Alt 2012, 29; Eigner 2013, 91 ff.; Friedenthal/Moore/Steiner 2009; INCOSE 2007). In Abb. 1 ist diese frühe Phase der Entwicklung mit „A“ gekennzeichnet.

In der mit „F“ markierten Phase werden die Funktionen des zu entwickelnden Systems³ definiert. Dazu werden Modelle eingesetzt, die erstmals auch eine Simulation von Funktionen erlauben. Für den hier intendierten Zweck wird das Simulationsprogramm Enterprise Dynamics (ED) der Firma INCONTROL Simulation Solutions aus Utrecht (Niederlande) herangezogen. Alternativen und Ergänzungen dazu sind z. B. Modelica, Matlab/Simulink oder in der Elektronik VHDL⁴. Ergebnis dieser Phase ist eine zunächst noch disziplin- und lösungsneutrale Sicht auf das Gesamtsystem in Form einer ersten Spezifikation (vgl. Eigner 2013, 102).

Daran schließt sich die interdisziplinäre Definition und Beschreibung der logischen Komponenten an, die erforderlich sind, um die Funktionen der Elemente des Gesamtsystems zu realisieren. In Abb. 1 ist diese Phase daher mit „L“ gekennzeichnet. Als Ergebnis dieser Phase liegen das logische und das physikalische Verhalten sowie die Struktur des Gesamtsystems vor (vgl. ebd.).

In der darauf folgenden, in Abb. 1 mit „P“ gekennzeichneten Phase ist eine interdisziplinäre Entwicklung kaum noch möglich. Nun sind vor allem Spezialisten gefordert, Detaillösungen zu erarbeiten. Dazu werden domänenspezifische Modelle, Methoden und Werkzeuge verwendet, so z. B. CAD-Systeme zum Entwurf der Geometrie, Finite-Elemente-Methoden zur Festkörpersimulation usw. Parallel dazu sollen allerdings auch die in der vorausgegangenen Phase erstellten Modelle weiter eingesetzt und zunehmend detailliert werden. Simulationsprogramme wie das hier verwendete ED bieten dazu beispielsweise CAD-Schnittstellen an, um die zunächst programmintern nur grob entworfenen Modellelemente realitätsnäher darzustellen.

In der nächsten Phase beginnt die Implementierung, in der die domänenspezifischen Teillösungen nach und nach zu einem mechatronischen Gesamtsystem zusammengefügt werden. Dominka und Bender (2007, 25 ff.) erklären in diesem Zusammenhang unter Verweis auf eine Studie des Verbands Deutscher Werkzeugmaschinenfabriken (VDW) aus dem Jahr 1997, dass dabei vor allem die Implementierung von Software ein großes Problem darstellt. Darauf würden ca. 15 % der Gesamtprojektdauer entfallen, wobei ein großer Teil dieser Zeit benötigt wird, um Softwarefehler zu identifizieren und zu beheben.

Die Ausweitung der Modellunterstützung im Sinne des MBSE setzt u. a. an diesem Problem an. Die im Zuge der Entwicklung (linker Schenkel des V in Abb. 1) entworfenen Modelle sollen in der Implementierungsphase (rechter Schenkel des V in Abb. 1) quasi als virtuelle Prototypen weiterverwendet werden, um mit der Codierung der Software so früh wie möglich beginnen zu können und dazu die Möglichkeit zu schaffen, diese an der virtuellen, also noch in Form von Modellen vorliegenden Hardware zu testen, bevor die realisierte Hardware dafür zur Verfügung steht. Dies wird als hardware in the loop (HiL) oder Emulation bezeichnet (siehe Abb. 1).

³ Viele Quellen zum Systems Engineering, u. a. die VDI-Richtlinie 2206, bezeichnen den technischen Gegenstand, den es zu entwickeln gilt, als System. Diese Bezeichnung wird hier beibehalten, wenngleich sie missverständlich ist, denn tatsächlich ist der System-Begriff sehr viel allgemeingültiger. Beispielsweise ist auch ein Modell ein System.

⁴ VHDL: Very High Speed Integrated Circuit Hardware Description Language

Dafür stehen spezielle Modelle und entsprechende Werkzeuge zur Verfügung. Die Firma SIEMENS bietet für ihre Automatisierungslösungen beispielsweise das Programm SIMATIC an. Allerdings taucht an dieser Stelle die eingangs aufgeworfene Frage wieder auf, ob nicht auch oder sogar besser solche Modelle und Werkzeuge weiterverwendet werden können und sollen, die bereits in den früheren Planungs- und Konzeptionsphasen eingesetzt wurden und insofern bereits vorhanden sind, so dass ein möglicherweise erneuter Modellierungsaufwand unter Einsatz zusätzlicher Werkzeuge vermieden werden kann.

3 Zielsetzung

Um eine solche Möglichkeit und zunächst nur diese zu prüfen, wird hier das Beispiel der ereignisorientierten (diskreten) Simulation herangezogen. Im Rahmen des grundlagenorientierten Forschungsprojekts OMSIS (On-the-fly-Migration und Sofort-Inbetriebnahme von automatisierten Systemen) wurde an der Technischen Universität Dresden in Kooperation mit zahlreichen Verbundpartnern aus Industrie und Dienstleistung eine Schnittstelle zwischen einer Speicherprogrammierbaren Steuerung (SPS) und dem Simulationsprogramm ED entwickelt (Theiss et al. 2010, 10 f.). Dabei wurde die grundsätzliche Möglichkeit einer auf einer solchen Konfiguration basierenden Emulation anhand eines mit Fischertechnik gebauten Modells einer Montageanlage bereits nachgewiesen. Ob sich diese Lösung allerdings auch in der Praxis bewährt, konnte dort mangels einer realen Montageanlage nicht gezeigt werden.

Um diesen auf die praktische Anwendung zielenden Nachweis im Zuge einer anwendungsnahen Forschung zu erbringen, wurden die Ergebnisse der universitären Grundlagenforschung an der NORDAKADEMIE, einer privaten, staatlich anerkannten Fachhochschule in Elmshorn, aufgegriffen und weiterentwickelt, denn hier steht eine reale Montageanlage für Forschungszwecke zur Verfügung (Abb. 2).



Abb. 2: Reale Anlage an der NORDAKADEMIE

4 Einsatz des Simulationsprogramms

Dem V-Modell folgend, wird ein Simulationsprogramm wie ED zur Entwicklung und Auslegung der Funktionen (in Abb. 1 mit „F“ gekennzeichnet) und der logischen Ele-

mente (in Abb. 1 mit „L“ gekennzeichnet) eingesetzt. Zu diesem Zweck bildet der Simulator nicht nur die Hardware ab, sondern auch die Steuerungslogik. In der darauf folgenden, in Abb. 1 mit „P“ gekennzeichneten Phase beginnt die Realisierung der physikalischen Elemente (Hardware). Parallel dazu wird die Software codiert. Für den Test des Codes sind an sich die physikalischen Elemente erforderlich. Solange diese noch nicht zur Verfügung stehen, können sie durch ihre Modelle (virtuelle Hardware) ersetzt werden. Dazu werden hier die mit dem Simulationsprogramm ED in den vorausgegangenen Planungs- und Konzeptionsphasen bereits erstellten Modelle verwendet. Da die Steuerung aber nun durch die SPS erfolgt, wird die Steuerungslogik des Simulators nun quasi überbrückt. Die Modellelemente folgen also nicht mehr der in ED programmierten Logik, sondern der Logik der SPS, die nun von außen auf die Modellelemente des Simulators zugreift. Der Test erfolgt durch einen Vergleich der ursprünglich innerhalb des Simulators programmierten Verhaltensweise des Simulationsmodells und der Verhaltensweise des Modells, das nun von der externen SPS gesteuert wird. Dabei gilt die Verhaltensweise des Simulationsmodells als Referenz. Stimmt die Verhaltensweise des von der SPS gesteuerten Modells damit überein, so ist der Test erfolgreich.

Abb. 3 zeigt die dazu erforderliche Konfiguration in einer vereinfachten Darstellung, in der zur Wahrung der Übersichtlichkeit auf die Darstellung untergeordneter Geräte wie z. B. eines Systemanschlusses zur Verbindung eines Computers mit dem PROFIBUS (CP 5512) oder einer SIMATIC ET 200 zur Anbindung verteilter Steuerungselektronik an die zentrale SPS verzichtet wurde. Mit Hilfe des Programmiergeräts (PG) wird zunächst die Anlagenkonfiguration modelliert und die entsprechende Steuerung programmiert. Das Programm wird anschließend in

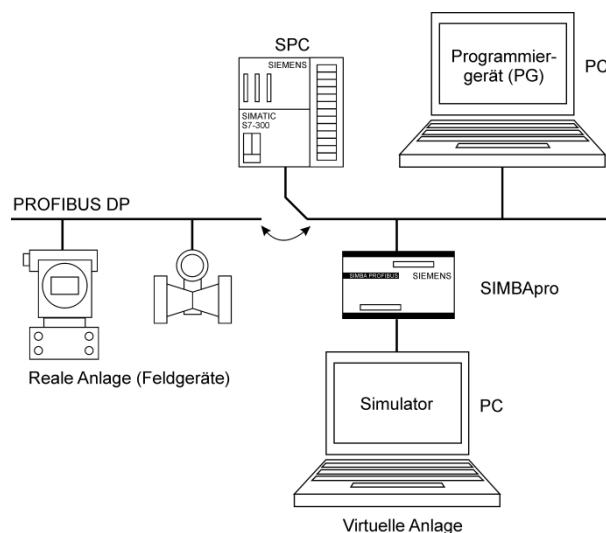


Abb. 3: Anlagenkonfiguration

die SPS übertragen. Solange die reale Anlage (linker Ast des Feldbusses) noch nicht zur Verfügung steht, kann deren Verhalten mit Hilfe der am rechten Ast des Feldbusses angekoppelten virtuellen Anlage nachgebildet werden. Dazu wird die Konfiguration der projektierten Anlage in die Hardwareschnittstelle (SIMBApro) des Simulators überspielt. Diese Schnittstelle verhält sich gegenüber der SPS genauso wie die Ein- und Ausgangssignale der realen Feldgeräte. Auf der anderen Seite dieser Schnittstelle werden die Ein- und Ausgangssignale an den Simulator weitergegeben und dort zur Visualisierung des dynamischen Verhaltens der projektierten Anlage aufgrund des für die SPS entwickelten Steuerungsprogramms verwendet.

Die in Abb. 3 dargestellte Möglichkeit, zwischen der virtuellen und der realen Anlage umzuschalten, betrifft den einfachen Fall, dass beide Anlagen vollständig sind. Tatsächlich sind auch Zwischenschritte möglich, bei denen die Komponenten der realen Anlage nach und nach in Betrieb genommen werden. Dominka und Bender sprechen in diesem Fall von einer hybriden Inbetriebnahme (2007, 25 ff.). Abb. 3 lässt sich dazu gedanklich so verändern, dass die Geräte, die darin mit dem linken Zweig des Feldbusses verbunden sind, in dem Moment, in dem sie real verfügbar sind, an den rechten Zweig, also parallel zur virtuellen Anlage, angekoppelt werden. Zugleich wird das entsprechende Modellelement in der virtuellen Anlage gelöscht, so dass die SPS nun sowohl diese virtuelle Anlage ohne die bereits real vorhandenen Komponenten als auch diese bereits vorhandenen, realen Komponenten steuert, bis schließlich die Anlage komplett realisiert ist, so dass die virtuelle Anlage bzw. das, was bis dahin noch von ihr übrig ist, vollständig abgeschaltet werden kann.

5 Ergebnisse

Die Experimente haben gezeigt, dass die Weiterverwendung eines in den Planungs- und Konzeptionsphasen erstellten Modells zur ereignisorientierten Simulation automatisierter Systeme in der Implementierungsphase nicht nur im Modellbetrieb, sondern auch unter realen Bedingungen *möglich* ist. Abb. 4 zeigt dazu das Simulationsmodell der Montageanlage. Dieses Modell wurde um drei an der TU Dresden entwickelte Modellbausteine erweitert (Theiss et al. 2010, 10 f.).

Der erste Modellbaustein, in Abb. 4 am Logo des PROFIBUS erkennbar, repräsentiert die Hardwareschnittstelle (SIMBApro), während die beiden weiteren Modellbausteine das Simulationsmodell mit den Ein- und Ausgangssignalen der SPS verbinden (in Abb. 4 sind nur Ausgänge erkennbar). Für jedes Signal der SPS muss jeweils einer dieser beiden Modellbausteine in das Simulationsmodell eingefügt und mit der Signalkennung der SPS bezeichnet werden. Mit Hilfe eines Codes, der in der Skriptsprache des Simulators (bei ED heißt diese Sprache 4DScript) formuliert werden muss, wird die Ablauflogik des Simulators durch die von der SPS vorgegebene Logik ersetzt.

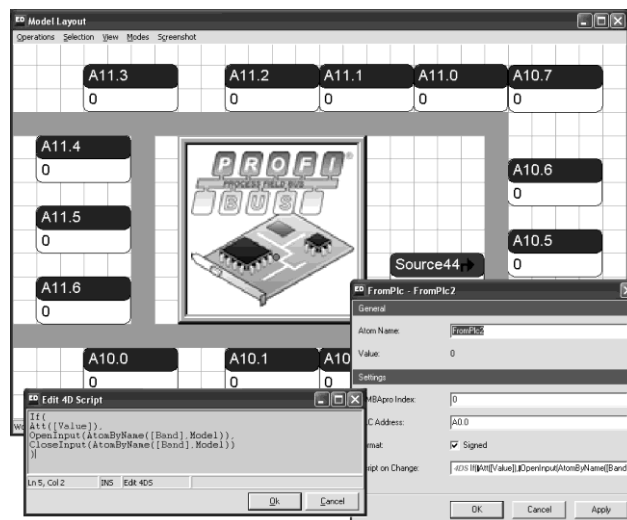


Abb. 4: Ausschnitt des Simulationsmodells

Anhand eines konkreten Beispiels soll verdeutlicht werden, wie dies im Einzelnen umgesetzt wird. Dazu zeigt Abb. 4 unten rechts die grafische Benutzungsschnittstelle eines Modellbausteins, der ein Ausgangssignal der SPS empfängt. In diesem Fall ist dieses Signal mit `A0.1` bezeichnet und entspricht damit der Bezeichnungskonvention speicherprogrammierbarer Steuerungen. Dieses Signal soll im Simulationsmodell dazu führen, dass das Förderband `Band` in Abhängigkeit des Signals `Value` gestartet oder gestoppt wird. Dazu werden die 4DScript-Befehle `OpenInput` bzw. `CloseInput` verwendet. Die Adressierung des Förderbands kann auf verschiedene Weise erfolgen. Hier wurde aus Gründen der Anschaulichkeit eine absolute Adressierung mit Hilfe der Bezeichnung des Förderbands gewählt (`AtomByName`)⁵. Alternativ wäre auch eine Adressierung möglich, und am effizientesten wäre ein Aufruf über einen Pointer auf die Speicheradresse des Atoms.

6 Fazit und Ausblick

Ziel dieses Beitrags ist es nachzuweisen, dass es im Sinne eines modellbasierten Systems Engineering möglich ist, Modelle, die im Zuge der Planung und Konzeption mechatronischer Systeme entworfen werden, um deren dynamisches Verhalten nachzubilden, im Verlauf der Implementierung weiterzuverwenden. Dieser Nachweis ist gelungen. Damit ist allerdings noch nichts darüber ausgesagt, ob und ggf. in welchen Hinsichten dies auch *notwendig* und *sinnvoll* ist. Dies nachzuweisen erfordert aufwändige Gegenüberstellungen mit Alternativen, in dem hier untersuchten Fall z. B. mit dem Simulator SIMIT aus der SIMATIC-Produktfamilie der Firma SIEMENS, sowie das Verfolgen alternativer Entwicklungspfade. Entsprechende Untersuchungen sind noch nicht abgeschlossen.

Der Ansatz eines modellbasierten Systems Engineering reicht allerdings noch über das hinaus, was hier zunächst zu Grunde gelegt wurde. So endet das in Abb. 1 dargestellte Vorgehensmodell mit der Inbetriebnahme einer Anlage. Tatsächlich erscheint es jedoch erstrebenswert, im Verlauf der Entwicklung erstellte Modelle auch in der Anwendungsphase technischer Systeme weiterzuverwenden. Diesbezügliche Nutzenpotenziale liegen z. B. in der Unterstützung von Schulungs- und Trainingsmaßnahmen, in der Vorausberechnung von Prozesswerten oder in der Prozessoptimierung (Kain et al. 2008, 59 ff.). Daher gilt es, über die hier behandelten Fragestellungen hinaus zu untersuchen, ob und ggf. in welcher Form Modelle, Methoden und Werkzeuge der diskreten Simulation auf dem Detaillierungsgrad der Feldebene (Level 1 der Automatisierungspyramide) auch zu diesen Zwecken verwendet werden können.

7 Literaturverzeichnis

- Ahrens, V. 2012: Inflation industrieller Revolutionen. In: Productivity Management. Nr. 5, S. 30 – 31.
Alt, O. 2012: Modellbasierte Systementwicklung mit SysML. München.

⁵ In ED werden Modellelemente als Atome bezeichnet.

- Dominka, S./Bender, K. 2007: Hybride Inbetriebnahme – von der Hardware in the Loop-Simulation zur realen Anlage. In: ZWF – Zeitschrift für wirtschaftlichen Fabrikbetrieb, Jahrg. 102, Nr. 1-2, S. 25 – 31.
- Eigner, M. 2013: Modellbasierte Virtuelle Produktentwicklung auf einer Plattform für System Lifecycle Management. In: Sendler, U. (Hrsg.): Industrie 4.0 – Beherrschung der industriellen Komplexität mit SysLM. Berlin, Heidelberg.
- Friedenthal, S./Moore, A./Steiner, R. 2009: A Practical Guide to SysML. Burlington.
- INCOSE 2007: Systems Engineering Vision 2020. San Diego: International Council on Systems Engineering.
- Kain, S./Heuschmann, C./Schiller, F. 2008: Von der virtuellen Inbetriebnahme zur Betriebsparallelen Simulation – Herausforderungen im Anlagenbetrieb und Nutzenpotentiale der betriebsparallelen Simulation. VDI-Bericht 2032, S. 59.
- Theiss, S./Pitsch, H./Kabitzsch, K. 2010: Omsis – Emulation, Hardware-in-the-Loop. In: Edition Enterprise Dynamics Simulation News. S. 10 – 11.
- VDI 2004: VDI-Richtlinie 2206 – Entwicklungsmethodik für mechatronische Systeme. Düsseldorf.