

# Methoden zur teilautomatischen Generierung von Emulationsmodellen

## *Methods for a Semiautomatic Generation of Emulation Models*

Torben Meyer, Volkswagen AG, Wolfsburg (Germany),  
torben.meyer@volkswagen.de

Steffen Straßburger, Technische Universität Ilmenau, Ilmenau (Germany),  
steffen.strassburger@tu-ilmenau.de

**Abstract:** Emulation models as the basis for virtual commissioning of control software and hardware are traditionally built manually and require a high amount of time and effort for building them. This paper addresses this issue by discussing different options for a semi-automatic generation of emulation models. We discuss different scenarios (from automatic emulation model generation to model transformation approaches) to and the associated benefits and requirements. Based on a comparison of the different scenarios, we discuss the economic efficiency and the applicability of all scenarios. Finally we present a prototypical implementation of a method for transforming a simulation model originally used for planning purposes into an emulation model used for virtual commissioning.

## 1 Einführung und Motivation

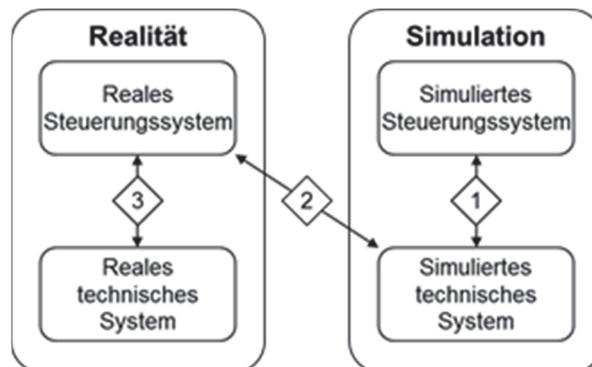
Die virtuelle Inbetriebnahme (VIBN) hat sich in den vergangenen Jahren als Methode zur Verkürzung der Inbetriebnahmedauer von Maschinen und Anlagen und zur Verbesserung des Softwaretests von Produktions- und Logistikeitständen sowie speicherprogrammierbaren Steuerungen etabliert (Salinas und Klemm 2010).

Methodisch basiert die VIBN auf einer Emulation des technischen Systems (Kövári 2011), welche erfahrungsgemäß mit großen Aufwänden verbunden ist. Das emulierte technische System wird in der VIBN typischerweise mit der realen Steuerung verbunden. Für die Erstellung der Emulationsmodelle bei der VIBN eines Produktions- oder Logistikeitstands werden häufig ereignisdiskrete Simulatoren aus dem Bereich der Produktion und Logistik eingesetzt (Gutenschwager et al. 2000).

In der Literatur sind vielfältige Nutzungsmöglichkeiten von Simulationsmodellen beschrieben (VDI 3633 Blatt 1). Diese lassen sich meist relativ exakt einzelnen Phasen des Lebenszyklus von Produktions- oder Logistiksystemen (z.B. Planung, Inbetriebnahme, Betrieb) zuordnen. Die Benutzung eines Simulationsmodells über mehrere Lebenszyklusphasen hinweg (z.B. die Nachnutzung eines Modells der

Planungsphase in der Inbetriebnahme- oder Betriebsphase) wird gegenwärtig aber nur in ausgewählten Beispielprojekten erfolgreich umgesetzt (Mayer und Spieckermann 2010). Gründe hierfür liegen im Falle der VIBN häufig in der Tatsache, dass eine Simulation zur Inbetriebnahme (Emulation) aus Sicht des Auftraggebers nicht notwendig respektive bekannt ist oder als zu aufwandsintensiv eingeschätzt wird.

Das Prinzip der Emulation für die Inbetriebnahme von Steuerungssystemen ist in Abbildung 1 dargestellt. Auf der einen Seite ist bei der klassischen Simulation das simulierte Steuerungssystem mit dem simulierten technischen System verbunden (1). Auf der anderen Seite ist in der Realität das reale Steuerungssystem mit dem realen technischen System verbunden (3). Die Verbindung zwischen dem realen Steuerungssystem und dem simulierten technischen System (2) wird dann als Emulation bezeichnet. Dieses Szenario bildet die Basis der meisten VIBN-Ansätze. Zwischen den Steuerungssystemen und den technischen Systemen kommen Kommunikationsprotokolle zum Einsatz, deren konkrete Ausprägung nicht zuletzt von dem Anwendungsfall und den verwendeten Komponenten abhängt.



**Abbildung 1:** Prinzip der Simulation und Emulation (in Anlehnung an Follert und Trautmann 2006)

In diesem Aufsatz werden mehrere methodische Ansätze zur effizienteren Emulationsmodellerstellung diskutiert. Diese beruhen entweder auf einer durchgängigen Nutzung eines Teils des Simulationsmodells, einer teilautomatischen Transformation des Simulations- in ein Emulationsmodell oder auf einer teilautomatischen datengetriebenen Modellgenerierung.

## 2 Stand der Technik: Gegenwärtige Ansätze zur Modellgenerierung

Methoden zur automatischen oder teilautomatischen Generierung von Simulationsmodellen mit ereignisdiskreten Materialflusssimulatoren sind in der Literatur etabliert (Straßburger et al. 2010). Jörg und Wacker (2008) zeigten beispielsweise eine „generische Simulationslösung für Fördertechnik“. Ebenda wurde auf der Grundlage von Layoutdaten aus Planungssystemen der Digitalen Fabrik ein Simulationsmodell mit Fördertechnikelementen automatisiert erstellt. Auf

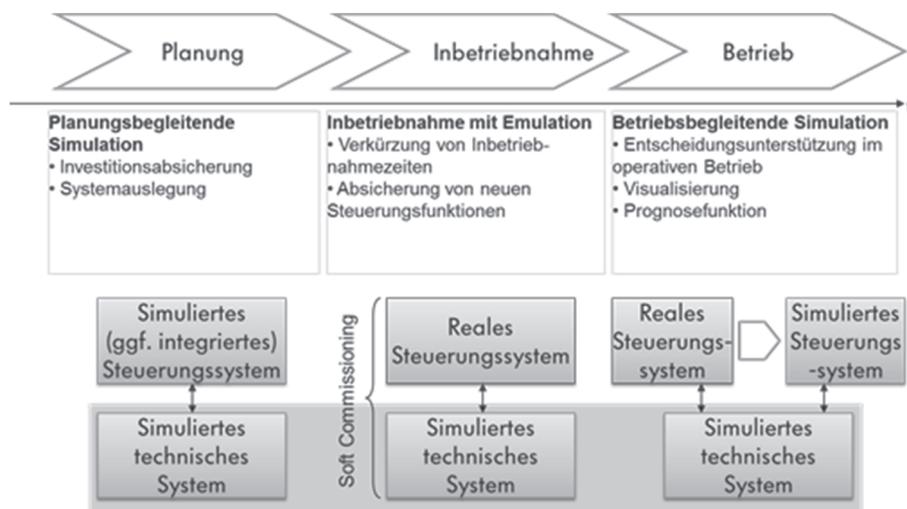
Eingangsdaten aus der betrieblichen Datenerfassung sowie der Produktionsplanung und -steuerung setzt Selke auf (Selke 2005).

Auch im Kontext der VIBN auf Feldebene (Anlagensimulation) sind Methoden zur (teil)-automatischen Modellgenerierung bekannt. Eine Möglichkeit zur automatisierten Generierung eines Simulationsmodells haben Barth und Fay (2010) vorgestellt. Aus der Maschinenbaubranche stellen Eger und Croon (2011) eine Möglichkeit zur Erstellung eines Modells zur virtuellen Inbetriebnahme „auf Knopfdruck“ dar. Als Eingangsdaten werden dabei vor allem die 3D-Geometrien (d.h. CAD-Daten) verwendet.

Unsere Literaturstudien haben gezeigt, dass bisher kaum Methoden zur automatischen oder teilautomatischen Generierung von Emulationsmodellen zur virtuellen Inbetriebnahme mit ereignis-diskreten Materialflusssimulatoren im Bereich von Produktion und Logistik verwendet werden. Diese Lücke soll in diesem Aufsatz mit einer Diskussion verschiedener Varianten adressiert werden.

### 3 Modellüberführung von der Simulation zur Emulation

Die Modelle der Materialflusssimulation lassen sich in verschiedenen Phasen verwenden. Neben der Planungsphase (Simulation), zählt dazu auch die Inbetriebnahmephase (Emulation) sowie die Betriebsphase (betriebsparallele Simulation, vgl. VDI 3633 Blatt 1). Eine Überführung des für die Planung erstellten Modells in die Inbetriebnahmephase ist (entsprechender Bedarf vorausgesetzt) aus wirtschaftlichen Gründen erstrebenswert, weil mehrfacher Modellierungsaufwand vermieden werden kann.



**Abbildung 2:** Simulationsmodelle im Kontext des Lebenszyklus von Produktions- und Logistiksystemen

Wie in Abbildung 1 dargestellt, verwenden sowohl das Simulations- als auch das Emulationsmodell ein simuliertes technisches System. Bei der Emulation wird

jedoch anstelle des simulierten Steuerungssystems eine Verbindung zum realen Steuerungssystem verwendet. Aus diesem Grund sind bei der Überführung von der Simulation zur Emulation entsprechende Anpassungen am Modell vorzunehmen. Dies betrifft die Kommunikation zum externen Steuerungssystem (d.h. die Implementierung der informationstechnischen Schnittstellenprotokolle) sowie vor allem die Übersetzung der externen Steuerungsinformationen in die korrekten Reaktionen der Modelllogik und die Übersetzung der Informationen vom Modell zurück zur realen Steuerung (z.B. Sensorsignale und Rückmeldungen).

Das simulierte Steuerungssystem ist somit für das Emulationsmodell nicht erforderlich. Aus der Sichtweise der Emulation ist es wünschenswert, dass bereits während der Erstellung des Simulationsmodells eine Trennung zwischen simuliertem technischem System und simuliertem Steuerungssystem an den Stellen vorgenommen wird, an denen auch in der Realität zwischen realem Steuerungssystem und realem technischem System entsprechende Schnittstellen existieren (s. Abb. 2). Dies vereinfacht die Erstellung des Emulationsmodells, da die Trennung bereits an den richtigen Stellen durchgeführt wurde. Bisher ist eine solche modulare Trennung zwischen Steuerung und technischem System im Simulationsmodell jedoch nicht durchgehend anzutreffen.

Darüber hinaus lässt sich ein durchgängiges simuliertes technisches System, welches für die Planungs- oder Inbetriebnahmesimulation erstellt wurde, auch für die Betreibersimulation weiternutzen. In diesem Fall wird das simulierte technische System zuerst vom realen technischen System vorbelegt. Dies kann entweder auf der Basis der betrieblichen Datenerfassung erfolgen oder indem die Steuerungsbefehle zwischen realer Steuerung und realem technischem System auch von dem simulierten technischen System interpretiert werden. Zu einem definierten Zeitpunkt, beispielsweise wenn eine operative Entscheidungsunterstützung notwendig geworden ist, wird die Verbindung zur realen Steuerung getrennt und durch die simulierte Steuerung ersetzt. Das so skizzierte Verfahren entspricht der aus der Literatur bekannten „Online-Simulation“ (Davis 1998).

#### **4 Methoden zur teilautomatischen Generierung von Emulationsmodellen**

Grundsätzlich lassen sich vier verschiedene Möglichkeiten zur Erstellung bzw. Überführung eines Emulationsmodells unterscheiden (vgl. Abb. 3): Entweder wird ein Modell in der Planung und Inbetriebnahme durchgängig und ohne dedizierte Transformation genutzt (Variante 1), das Simulationsmodell wird teilautomatisch in ein Emulationsmodell überführt (Variante 2) oder das Emulationsmodell wird ohne Nutzung eines gegebenenfalls vorhandenen Simulationsmodells erstellt (Variante 3). Als vierte Variante wird eine Generierung aus Quellsystemen diskutiert. Darüber hinaus ist ebenfalls eine Kombination der Varianten denkbar.

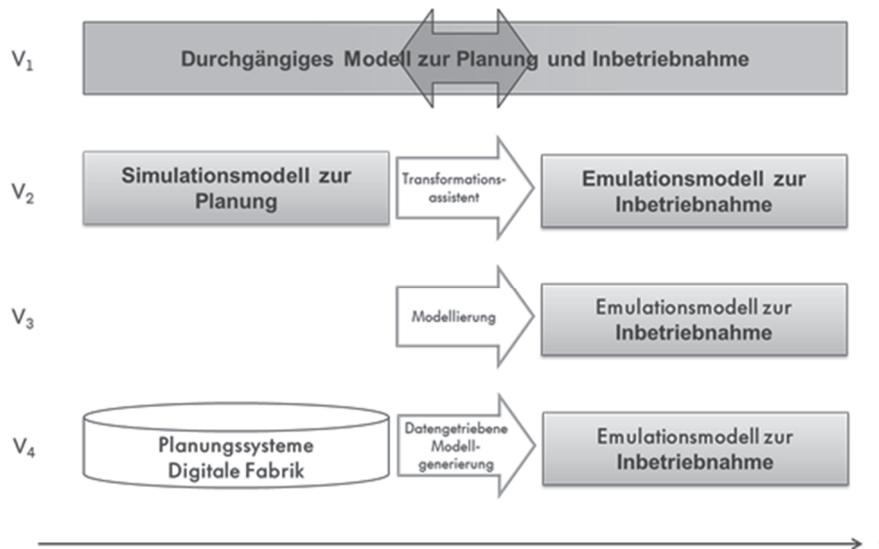
Die Idee der *Variante 1* ist die durchgängige Nutzung des Modells sowohl für die Planung als auch für die Inbetriebnahme einer Anlage. Die Grundlage dafür bildet die Tatsache, dass in beiden Nutzungsmöglichkeiten das simulierte technische System verwendet wird (vgl. Abb. 2).

Dies bedingt allerdings die Deaktivierung der in der Planungssimulation vorhandenen simulierten Steuerung und stattdessen die Anbindung an die reale

Steuerung. Eine Ergänzung des Modells um die Möglichkeit zur Kommunikation mit der realen Steuerung ist zu realisieren.

Eine Voraussetzung zur Unterstützung einer durchgängigen Modellierung des technischen Systems ist, dass die notwendige Trennung von Steuerung und technischem System bereits während der Erstellung des Simulationsmodells berücksichtigt wird, also die Schnittstellen zwischen simulierten technischem System und simulierter Steuerung vergleichbar zu den Schnittstellen im realen System sind. Dies ist allerdings nur möglich, wenn die Schnittstellen bereits während der Planung bekannt sind.

Die Vorteile der Variante 1 sind somit, dass das Modell des simulierten technischen Systems übernommen werden kann und auf diesem Weg keine Anpassung für das Emulationsmodell notwendig ist. Nachteilig ist allerdings, dass diese Modellierungsweise nur einsetzbar ist, wenn frühzeitig die Schnittstellen des realen Systems bekannt sind oder geeigneten Schnittstellen- oder Architekturstandards gefolgt wird.



**Abbildung 3:** Varianten zur Erstellung eines Emulationsmodells

Variante 2 setzt voraus, dass ein Simulationsmodell aus der Planungsphase existiert, das als Emulationsmodell weiterverwendet werden soll. Der methodische Ansatz ist die Transformation von Simulationsobjekten in Objekte, die die Emulation unterstützen. Dies wird dadurch erreicht, dass jeder Objektklasse in der Simulation mindestens eine äquivalente Klasse von Emulationsobjekten gegenübersteht. Auf diese Weise lassen sich beispielsweise mit einem Transformationsassistenten Modelle überführen.

Bei der Transformation wird das Simulationsobjekt durch eine Instanz der assoziierten Klasse für die Emulation ersetzt. Darüber hinaus ist auch die

Konfiguration, also der Inhalt der Objektattribute, zu übernehmen. Schließlich ist auch noch die Materialflussverknüpfung auf das neue Objekt zu übertragen.

Voraussetzung für die Durchführung einer derartigen Transformation ist allerdings die Implementierung von Objektklassen für die Emulation, welche inhaltlich die Simulationsobjekte ergänzen können und darüber hinaus auch die notwendigen Voraussetzungen zur Kommunikation mit der realen Steuerung erfüllen. Eine weitere Prämisse ist, dass das Simulationsmodell in großem Maße auf den Einsatz von instanziierten Objekten von einer überschaubaren Anzahl an Klassen setzt. Die Vorteile dieses Vorgehens sind in der Möglichkeit einer weitgehend automatischen Transformation des Modells zu sehen.

Sollten die ersten beiden Möglichkeiten aus technischen (Überführung zu aufwendig) oder organisatorischen Gründen (kein Simulationsmodell vorhanden) nicht in Betracht kommen, so ist ein Aufbau des Emulationsmodells von Null an ggf. notwendig (*Variante 3*). Dies erfolgt über eine unter Umständen aufwendige Modellierung des gesamten Emulationsmodells, so dass bei dieser Variante auch nicht von einer (teil-)automatischen Generierung des Modells gesprochen werden kann.

Auf der einen Seite ist der Vorteil der Modellierung ohne eine vorhandene Basis (Simulationsmodell aus der Planungsphase oder eine anderes System als Datenquelle), dass dieser Ansatz immer anwendbar also universell einsetzbar ist. Auf der anderen Seite wird diese Art der Flexibilität in der Modellierung auch durch einen hohen Aufwand in der Modellierungsphase erkauft, da das Wissen, welches in den anderen Varianten übernommen werden konnte in dieser Variante ggf. erneut aufgebaut werden muss. Variante 3 wird somit in den meisten Anwendungsfällen der Emulation als am aufwendigsten einzustufen sein.

Die *vierte Variante* basiert auf einer „datengetriebene(n) Modellgenerierung“ (Eckardt 2002). In dem Bereich der Planungssimulationen ist die Möglichkeit der (teil-)automatischen Modellgenerierung aus verfügbaren innerbetrieblichen Datenquellen in der Literatur diskutiert worden (Straßburger et al. 2010). Die Kernidee der Variante 4 ist die Übertragung der Methode der datengetriebenen Modellgenerierung auf die Inbetriebnahmephase eines Produktions- und Logistiksystems, um an einem möglichst weitgehend generierten Emulationsmodell die Erprobung und Testung von Steuerungssoftware durchzuführen.

Die Voraussetzung für die Generierung eines Emulationsmodells ist das Vorhandensein einer ausreichenden Datenbasis. Während für die Generierung eines Modells bei einer Planungssimulation für eine Produktionsmaschine beispielsweise schon die Taktzeit, die Verfügbarkeit, die Materialflussverkettung, etc. ausreichend sind, werden für Generierung von Emulationsmodellen detaillierte Informationen, wie z. B. das Signalverhalten für die externe Steuerung benötigt. Für emulierte Komponenten lassen sich diese Informationen zur Ansteuerung beispielsweise der Komponentendokumentation entnehmen.

Als Datenquelle kommen grundsätzlich Planungssysteme, etwa aus dem Bereich der Digitalen Fabrik (z.B. E-CAD-Systeme für die Elektrotechnikplanung), in Frage. Darüber hinaus ist außerdem zu berücksichtigen, dass die relevanten Daten frühzeitig zur Verfügung stehen. Die Informationen aus den Planungssystemen bilden die Grundlage für die Instanziierung von Simulationsobjekten aus einer Simulations- bzw. Emulationsobjektbibliothek, welche eine Voraussetzung für die

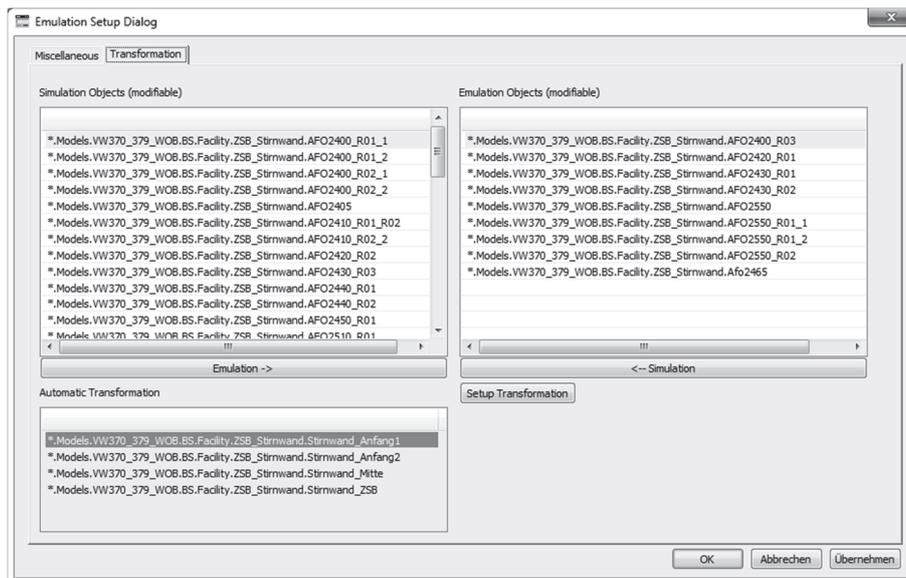
Durchführung einer derartigen datengetriebenen Modellgenerierung ist und für eine spezielle Reihe von gleichartigen Anwendungsfällen erschaffen worden ist.

Damit ist die Durchführung von Variante 4 besonders wirtschaftlich, wenn eine größere Anzahl von Emulationsmodellen mit einer vorliegenden Bibliothek generiert wird. Im Gegensatz zu den vorherigen Varianten 1 und 2 wird in dieser Variante allerdings das ggf. vorhandene Simulationsmodell aus der Planungsphase nicht genutzt, so dass es zu Unterschieden zwischen geplanter und realisierter Anlage kommen kann.

Die Varianten 1, 2 und 3 konnten in der betrieblichen Praxis (zum Teil prototypisch) umgesetzt werden. Aufgrund unzureichender Datenquellen steht die Evaluierung von Variante 4 noch aus.

## 5 Prototypische Umsetzung

Die Umsetzbarkeit der oben vorgestellten Variante 2 wurden im Rahmen einer prototypischen Implementierung evaluiert. Zum Einsatz gekommen ist dabei der Simulator Plant Simulation 10.0 von Siemens PLM mit dem Automotive Bausteinkasten des VDA (Verband der Automobilindustrie). Der Automotive Bausteinkasten ist bei vielen deutschen Automobilherstellern als Objektbibliothek zum standardisierten und effizienten Simulationsmodelllaufbau im Einsatz (Mayer und Pöge 2010).



**Abbildung 4:** Screenshot des Assistenten für die Transformation

Im Kern des Prototyps steht ein Assistent (vgl. Abb. 4), welcher die Transformation durchführt. Das gesamte Simulationsmodell wie in einer Baumstruktur von der Wurzelebene an durchsucht und sämtliche Simulationsobjekte, die für eine Transformation in Betracht kommen, werden registriert. Dafür ist eine

Konfiguration erforderlich, die festlegt welche Simulationsklasse mit welchen Emulationsklassen assoziiert ist. Darüber hinaus sind auch die zu übertragenden Attribute zu konfigurieren.

Im Assistenten können die erkannten Objekte ausgewählt werden und transformiert werden. Die einzelnen Simulationsobjekte werden dann eins zu eins in das assoziierte „Emulationsobjekt“ transformiert. Dabei wird zunächst jeweiliges Emulationsobjekt erzeugt und entsprechend benannt. Die Materialflussverknüpfungen werden vom alten auf das neue Objekt übertragen. Anschließend erfolgt die Übertragung der Werte von (vorher konfigurierten) Attributen. Schließlich kann das alte Objekt entfernt werden.

Die Objektklasse Querverschiebewagen (QVW) aus dem VDA Automotive Bausteinkasten hat bereits eine einfache Steuerungslogik für die Simulation vorimplementiert, die ankommende Paletten (o.ä.) abholt, mit einem QVW transportiert und an konfigurierten Ausgängen abgibt. Auf dieser Grundlage wurde ein Baustein für die Emulation erstellt. An allen Stellen, wo der QVW Fahrweisungen (nächster Aufnahme- oder Absetzpunkt) anfragt bzw. Statusrückmeldungen (Beginn und Ende von Lade- bzw. Endlagevorgängen) liefert, werden Meldungen an die externe, reale Steuerung abgesetzt.

Die Emulationslogik ist in diesem Fall schnittstellenübergreifend implementiert und abstrahiert von der konkret genutzten informationstechnischen Basis (wie z.B. OPC oder Profibus). Über standardisierte Funktionsaufrufe können so genannte Transportaufträge angelegt, verändert und gelöscht werden. Diese Transportaufträge werden dann durch den QVW abgearbeitet. Für spezifische Projekte sind in diesem Fall also nur noch die Anpassungen für die externen Schnittstellen durchzuführen.

Darüber hinaus wurde der VDA-Baustein für einen Kanbanpuffer ebenfalls angepasst. Der Simulationsbaustein reagiert auf eingehende Abrufe aus dem (sortenreinen) Puffer und löst ggf. Abrufe in vorgelagerten Kanban-Bausteinen aus. Ebendiese Funktionen werden in einem zweiten Emulationsbaustein auch der Emulation zur Verfügung gestellt.

Erfahrungsgemäß lassen sich projektspezifische Anpassungen in der Emulation, also die Einführung von weiteren Objekten, nicht vermeiden. Aus diesem Grund wurde eine Vorlage für Emulationsbausteine entwickelt, die bereits die wichtigsten Anforderungen abdeckt. Dieser Baustein enthält eine lokale Signalliste (Signalbezeichnung, aktueller Wert, Methode für den Aufruf bei einer Wertänderung sowie den Initialisierungswert), eine Methode zur Kommunikation mit externen Steuerungen und für die Emulation angepasste Initialisierungs- und Reset-Methoden.

## **6 Zusammenfassung und Bewertung**

In einer vorherigen Publikation wurde bereits der Aufbau von Objektbibliotheken für die Emulation bei vorhandenen Simulationsobjektbibliotheken diskutiert (Meyer et al. 2012). Dieser Aufsatz fokussierte nun auf die Automatisierung des Übergangs von einem Simulations- zu einem Emulationsmodell. Hierzu wurden vier verschiedene Varianten diskutiert, die entweder auf bestehenden Simulationsmodellen aufsetzten, eine datengetriebene Modellgenerierung nutzen oder eine vollständige Neumodellierung einsetzen.

Im Abschnitt Stand der Technik wurde bereits verdeutlicht, dass zwischen automatischen und teilautomatischen Ansätzen zur Simulationsmodellgenerierung unterschieden werden muss. Die Ansätze dieses Aufsatzes sind eindeutig in den Bereich der teilautomatischen Generierung einzuordnen, da eine Anpassung von Simulationsobjekten/-bausteinen an individuelles Verhalten, welches außerhalb von den definierten Freiheitsgraden der Simulationsobjektbibliotheken liegt, nie ausgeschlossen werden kann, sondern sogar berücksichtigt und zugelassen werden muss.

Generell ist sowohl bei automatischen als auch teilautomatischen Ansätzen der Modellgenerierung zu beachten, dass diese auf entsprechenden Bibliotheken aufsetzen. Erst der Aufbau dieser Bibliotheken ermöglicht die effiziente Generierung von Modellen, verursacht aber seinerseits beachtenswerte Aufwände durch die erstmalige Erstellung und die regelmäßige Pflege. Folglich muss den Initialaufwänden eine ausreichende Anzahl an Emulationsprojekten, die die Modellgenerierung nutzen, gegenüberstehen, um die Wirtschaftlichkeit dieses Ansatzes zu gewährleisten.

Insgesamt lässt sich festhalten, dass die vorgestellten Methoden einen Beitrag zur Wirtschaftlichkeit der Emulation leisten. Die Auswahl der konkreten Methoden ist dabei projekt- und kontextabhängig

## Literatur

- Barth, M.; Fay, A.: Automatische Generierung von objektorientierten Simulationsmodellen für die Unterstützung des PLS-Test-Engineerings. In: Jumar, U.; Schnieder, E.; Diedrich, C. (Hrsg.): 11. Fachtagung EKA mit Tutorium, Magdeburg, 25.-27. Mai 2010, S. 235-244.
- Davis, W. J.: On-Line Simulation: Need and Evolving Research Requirements. In: J. Banks, J. (Hrsg.) Handbook of Simulation. John Wiley & Sons Inc. 1998, S. 465-516.
- Eckardt, F.: Ein Beitrag zu Theorie und Praxis datengetriebener Modellgeneratoren zur Simulation von Produktionssystemen. Dissertation. Aachen: Shaker 2002.
- Eger, U.; Croon, N.: Generierung virtuelle Maschinen zur virtuellen Inbetriebnahme auf Knopfdruck. In: Bender, K. (Hrsg.): SPS/IPC/Drives, Nürnberg, 22.-24. November 2011, S.205-213.
- Follert, G.; Trautmann, A.: Emulation intralogistischer Systeme. In: Wenzel, S. (Hrsg.): 12. ASIM Fachtagung, Kassel, 26.-27. September 2006, S. 521-530.
- Gutenschwager, K.; Fauth, K.-A.; Spieckermann, S.; Voß, S.: Qualitätssicherung lagerlogistischer Steuerungssoftware durch Simulation. Informatik-Spektrum 23 (2000) 1, S. 26-37.
- Jörg, T.; Wacker, R.: Generische Simulationslösung Fördertechnik. In: Rabe, M. (Hrsg.): Advances in Simulation For Production and Logistics Applications, Berlin, 1.-2. Oktober 2008, S. 11-20.
- Kövari, L.: Konzeption und Realisierung eines neuen Systems zur produktbegleitenden virtuellen Inbetriebnahme komplexer Förderanlagen. Dissertation. Karlsruhe: KIT Scientific Publishing 2011.
- Mayer, G.; Pöge, C.: Auf dem Weg zum Standard - Von der Idee zur Umsetzung des VDA Automotive Bausteinkastens. In: Zülch, G.; Stock, P. (Hrsg.):

- Integrationsaspekte der Simulation: Technik, Organisation und Personal. Karlsruhe: KIT Scientific Publishing 2010, S. 29-36.
- Mayer, G.; Spieckermann, S.: Life-cycle of simulation models: requirements and case studies in the automotive industry. *Journal of Simulation* 4 (2010) 4, S. 255-259.
- Meyer, T.; Straßburger, S.: Using protocol state machines to support simulation-based emulation projects. In: Geril, P.; Klumpp, M. (Hrsg.): *Proceedings of the 2012 European Simulation and Modelling Conference*, Essen, 22.-24. Oktober 2012, S. 234-238.
- Meyer, T.; Pöge, C.; Mayer, G.: Integration of Emulation Functionality into an established Simulation Object Library. In: Laroque, C.; Himmelspach, J.; Pasupathy, R.; Rose, O.; Uhrmacher, A. M. (Hrsg.): *Proceedings of the 2012 Winter Simulation Conference*, Berlin, 11.-14. Dezember 2012, ohne Seitenzahl.
- Salinas, L.; Klemm, P.: Virtuelle Inbetriebnahme und Test von Leitsystemen. In: Bender, K. Schumacher, W. (Hrsg.): *SPS/IPC/Drives 2010*, Nürnberg, 23.-25. November 2010, S.435-443.
- Selke, C.: *Entwicklung von Methoden zur automatischen Simulationsmodellgenerierung*. Dissertation. Technische Universität München 2005.
- 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 3633, Blatt 1, *Simulation von Logistik-, Materialfluss- und Produktionssystemen*. Berlin: Beuth 2007.