Ansätze zur Systemsimulation im Automobilbau

Ewald Hessel, Hella KGaA Hueck & Co, Sprecher AK30
Joachim Haase, Fraunhofer IIS/EAS Dresden
Ewald.Hessel@hella.com, joachim.haase@eas.iis.fraunhofer.de

Zusammenfassung

Probleme in Zusammenhang mit der Erstellung und Nutzung von Simulationsmodellen elektrischer Komponenten in einem Fahrzeug und ihre Nutzung zur Untersuchung der Wechselwirkung mit der Fahrzeugumgebung und nichtelektrischen Baugruppen sind Gegenstand des vorliegenden Beitrags. Einen Schwerpunkt bildet dabei die Auseinandersetzung mit Fragen, die sich durch die unterschiedlichen Anforderungen der im Entwicklungsprozess Beteiligten – Fahrzeughersteller, Zulieferer und Halbleiterhersteller – und deren Simulationslösungen ergeben. Der Beitrag will zu weiterer Diskussion darüber anregen wie in diesem Zusammenhang insbesondere der Aufwand für die Modellerstellung reduziert werden kann.

1 Einleitung

Neben der Nutzung von Simulationsverfahren bei der Entwicklung einzelner Fahrzeugkomponenten wird die Simulation zunehmend für die Untersuchung, Validierung und Optimierung der aus den Komponenten zusammengefügten Systeme und Teilsysteme eingesetzt. Eines dieser Teilsysteme ist das Bordnetz, das aus allen elektrischen Komponenten eines Fahrzeugs gebildet wird.

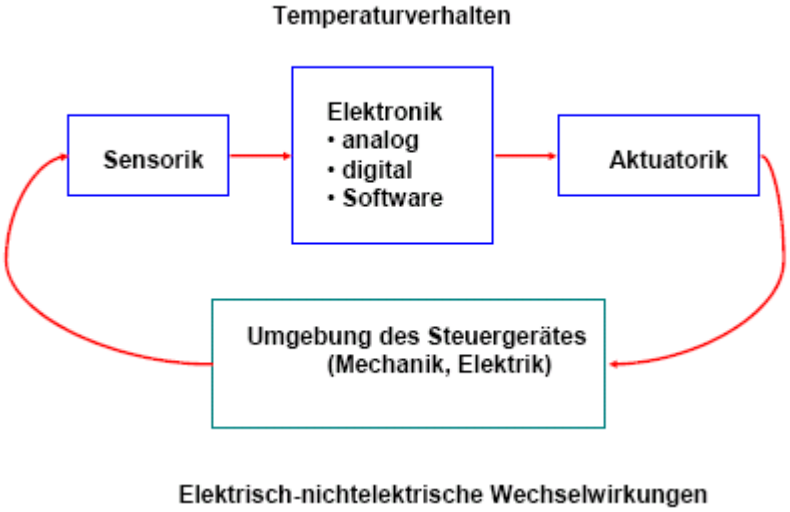

Abbildung 1: Zu betrachtende Wechselwirkungen bei der Systemsimulation

Bei der Simulation des Bordnetzes spielen elektrisch-nichtelektrische Wechselwirkungen innerhalb von Komponenten ebenso wie zur Umgebung sowie die gemeinsame Betrachtung von Hard- und Softwarebestandteilen eine Rolle. In Abb. 1 ist diese Situation kurz skizziert. Auch ist speziell bei der Fahrzeugentwicklung die Einbeziehung realer Objekte oder materieller Modelle davon in die gemeinsame Simulation von Bedeutung.

Der Erfolg einer Simulation hängt neben der Qualität der verwendeten Simulationswerkzeuge wesentlich von den eingesetzten Simulationsmodellen ab. Modelle entstehen im Entwicklungsprozess an den unterschiedlichsten Stellen, für die unterschiedlichsten Zwecke, auf unterschiedlichsten Abstraktionsebenen und für die unterschiedlichsten Simulationswerkzeuge, um nur einige Gesichtspunkte zu nennen. Einige dieser Fragestellungen sind in Abb. 2 aufgeführt.

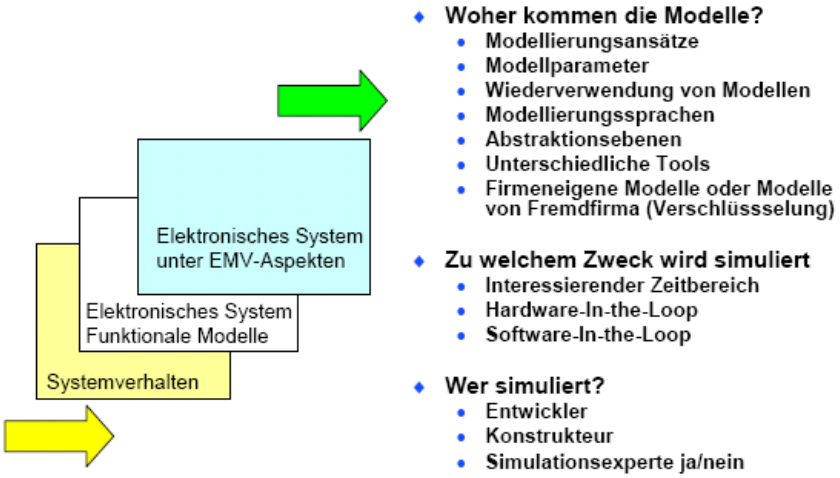

Abbildung 2: Fragestellungen bei der Systemsimulation

Für den Einsatz der Systemsimulation im Entwicklungsprozess spielen die Verfügbarkeit von Modellen und die Beurteilung ihrer Eignung für eine spezielle Aufgabe eine wichtige Rolle. Typische Merkmale in diesem Zusammenhang sind dabei zum einen die Frage, welches Objekt modelliert wird (Abbildungsmerkmal), zum anderen aber auch, welche Eigenschaften des Objektes durch das Modell beschrieben werden und welche nicht (Verkürzungsmerkmal) und für welchen Zweck ein Modell erstellt worden ist (Pragmatisches Merkmal) [1]. Beispiele für unterschiedliche Abstraktionsebenen – Beschreibung des Systemverhaltens, Beschreibung des funktionalen Verhaltens, Beschreibung des Verhaltens unter EMV-Aspekten - von Modellen eines Objektes sind in Abb. 2 skizziert.

Einerseits liegen auf Grund von Arbeiten in Zusammenhang mit der Entwicklung von Komponenten eine Reihe von Modellen für diese vor, andererseits ist es für den Nutzer oft schwer, ein geeignetes Modell zu finden und zu entscheiden, ob es sich für seine Aufgabe eignet. Durch die sehr unterschiedlichen Simulationsumgebungen ist ein weiteres Problem die Verwendung von Modellen, die für eine Umgebung erstellt worden sind, und in einer anderen verwendet werden sollen.. Auch die Bereitstellung von Ausgangsdaten für die Modell-erstellung ist häufig problematisch. Die unterschiedlichen Aspekte in Zusammenhang mit

Modellerstellung und Modellaustausch aus Sicht der Systemsimulation sollen im Folgenden kurz skizziert werden.

2 Modellanforderungen

Üblich ist bei der Modellierung für die genannten Aufgaben ein objektorientierter Ansatz. Dabei wird zwischen der Beschreibung des Modellrandes (Verbindungspunkte und Modellparameter) und des eigentlichen Modellverhaltens unterschieden. An den Klemmen werden entweder Fluss- und Differenzgrößen (bei Netzwerkmodellen) oder Ein- und Ausgangsverläufe (bei Signalflussmodellen) betrachtet. Eine Wechselwirkung zwischen den Modellen wird durch die Zusammenschaltung und in der Regel nur über die Signalverläufe an den Klemmen hergestellt.

Das rein *analoge Verhalten* wird dabei über Differentialgleichungen oder auch durch lineare oder nichtlineare Abbildungen vorgegeben. Durch das Abstraktionsniveau ergibt sich ein Kompromiss zwischen Modellgenauigkeit und Simulationsgeschwindigkeit (siehe auch Abb. 3). Modelle für die Systemsimulation werden in der Regel nur langsamere Änderungen berücksichtigen können als Modelle, die für den Entwurf von Komponenten benötigt werden. Werden Modelle im Rahmen einer Echtzeitsimulation benötigt, ist die Auswertung der Modellgleichungen innerhalb einer definierten Zeit (WCET – worst case execution time) zu sichern. Letztlich kann dies im Allgemeinen nur durch Modellbeschreibungen, die in Form gewöhnlicher Differentialgleichungen (ODE – ordinary differential equations) vorliegen, erfolgen. Die Überführung eines in DAE-Beschreibung (DAE – differential algebraic equations) vorliegenden Modells in eine (konsistente) ODE-Darstellung ist eine üblicherweise anspruchsvolle Aufgabe [2]. Diese Aussage trifft auch zu, wenn als Ausgangsbeschreibung ein auf partiellen Differentialgleichungen beruhender Ansatz vorliegt.

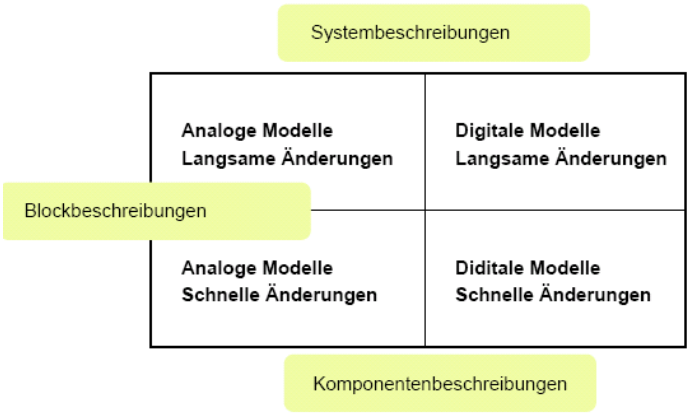

Abbildung 3: Anforderungen unter Genauigkeits- und Rechenzeitaspekten [3]

Unter Konsistenz von Modellen ist dabei üblicherweise qualitativ annähernd ähnliches Verhalten zu verstehen, wenn an den Klemmen eine Anregung mit gleichen Signalverläufen auftritt. Diese Ähnlichkeit formal zu beschreiben und zu prüfen ist, zumindest für das analoge Verhalten, eine aus unserer Sicht nicht abschließend gelöste Aufgabe (siehe dazu auch [4]).

Die Konsistenz zu sichern ist insbesondere dann von Interesse, wenn Modelle unterschiedlicher Abstraktionsebenen (siehe auch Abb. 4) miteinander verbunden werden sollen.

Abbildung 4: Abstraktionsebenen und Beschreibungsmittel

Die Beschreibung des *diskreten Verhaltens* läuft in vielen Fällen auf die Beschreibung mit zeitdiskreten Zustandsgleichungen hinaus [5]. Ereignisgesteuerte Simulationsalgorithmen oder auf Datenflussbeschreibungen basierende Algorithmen werden zur Simulation eingesetzt. Dabei sind (in den meisten Fällen) die Signalwerte Elemente endlicher Mengen. Digitale Hardware wird üblicherweise so beschrieben.

Modelle, die sowohl analoges als auch diskretes Verhalten beschreiben, werden als *hybride Modelle* oder mixed-signal Modelle bezeichnet. Die analoge Verhaltensbeschreibung greift auf die Beschreibung des diskreten Verhaltens zu und umgekehrt. Dieses Wechselspiel unterscheidet sich im Detail vielfach in Abhängigkeit von den verwendeten Modellbeschreibungssprachen und Simulationswerkzeugen. Eines der vielen Probleme, das bei der Modellerstellung vermieden werden muss, ist das Auftreten sehr vieler (theoretisch unendlich vieler) Ereignisse in einem endlichen Zeitintervall, auch als Zeno-Verhalten bezeichnet [6]. Zur Erklärung des Phänomens wird oft der springende Ball herangezogen. Das Problem kann aber auch bei der Beschreibung des Übergangs von Gleit- zu Haftreibung oder bei schwellwerthängigem Verhalten auftreten. Zur Vermeidung sind in der Regel von der Modellbeschreibungssprache und dem eingesetzten Simulationswerkzeug abhängige Vorkehrungen zu treffen.

Die dritte für die Systemsimulation interessante Komponente ist die *Software*, die beispielsweise in den Steuergeräten zum Einsatz kommt. Stand bei den in Hardware eingebetteten Systemen (embedded systems) historisch gesehen die effiziente Nutzung der beschränkten Ressourcen wie geringer verfügbarer Speicher, kleine Wortbreite und relativ geringe Taktfrequenz im Vordergrund, so verschiebt sich die Fragestellung, auch unter dem Gesichtspunkt parallel arbeitender Softwaresysteme, stärker in Richtung Vorhersagbarkeit und Zuverlässigkeit [7], Fehlertoleranz, Sicherheit, Reduzierung des Energieverbrauch usw. Der neuen Qualität hat man versucht mit der Einführung des Begriffs Cyber-Physical Systems (CPS) Rechnung zu tragen. Damit ergeben sich auch neue Anforderungen an Entwurf und Verifikation der Software.

Benötigt werden in diesem Zusammenhang Modelle, die analoges und diskretes Verhalten von Komponenten und der Umgebung eines Systems beschreiben. Ein erfolgreiches Vorgehen verlangt, den Aufwand an dieser Stelle zu reduzieren, neue Modelle einfach erstellen zu können und existierende Modelle schnell nutzbar zu machen. Forderungen in diesem Zusammenhang sind, auf Standardlösungen für die Modellierung („commercial-off-the-shelf“ modeling environments) zugreifen zu können, eine automatisierte Transformation von Modellen von einer Abstraktionsebene in eine andere zu unterstützen und eine zeitaufwändige und fehleranfällige Validierung der Modelle zu vermeiden [8].

3 Modellnutzung in heterogenen Simulationsumgebungen

3.1 Modellerstellung

Bei der Erstellung von Modellen sind einerseits die Anforderungen an die Modelle zu beschreiben, andererseits die für die Modellierung verfügbaren Daten wie Messergebnisse oder Datenblattangaben bereit zu stellen. Beim Übergang von einer Abstraktionsebene zur anderen baut die Modellerstellung auch vielfach auf Simulationsergebnissen auf (z. B. unter Anwendung von Methoden der Systemidentifikation, siehe auch [9]). Versuche, diesen Prozess zu systematisieren haben sich allgemein verbindlich - wie z. B. in [10] angeregt - bisher offenbar nicht durchsetzen können.

Dabei tragen klare Festlegungen durchaus zur Reduzierung des Modellierungsaufwandes und zur Vermeidung von Missverständnissen bei. Alleine die Verfügbarkeit von Diagrammen aus Datenblättern in maschinenlesbarer Form vereinfacht diesen Prozess. Die Beschreibung der Anforderungen an das Modell (siehe auch Abb. 3) beeinflusst die Auswahl des mathematischen Modellansatzes (linear, nichtlinear, statisch, dynamisch, …). Auch ist festzulegen, für welche Simulationsarten (Nominalwertanalyse, statistische Simulationen, Zeitbereichs- und/oder Frequenzbereichssimulationen) das zu erstellende Modell eingesetzt werden soll. Ist beispielsweise nur eine Frequenzbereichsanalyse mit dem Modell vorgesehen, braucht ggf. kein Zeitbereichsmodell erstellt werden.

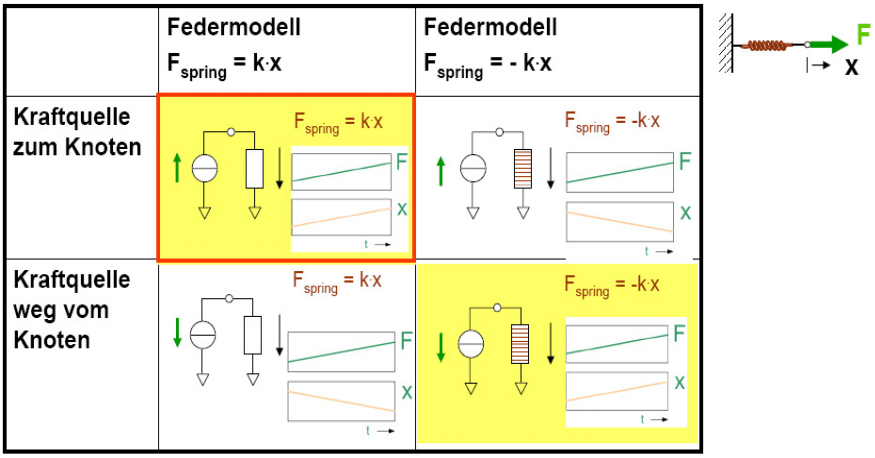

Abbildung 5: Varianten für Zählrichtungen bei mechanischen Netzwerkmodellen

Der Dokumentation der Modelle ist insofern besonderes Gewicht beizumessen, weil bestimmte Eigenschaften der Modelle (verwendete Maßeinheiten, Zählrichtungen – siehe Abb. 5, …) sich teilweise nicht aus rein formalen Kriterien bei Verwendung verbreiteter Beschreibungsmitteln wie VHDL-AMS, Modelica, Matlab/Simulink und Simscape, Verilog-AMS ableiten lassen (siehe auch [11]).

Bestandteil der Dokumentation muss letztlich auch eine Beschreibung des Vorgehens bei der Parametrisierung und zur Validierung eines Modells sein. Einzuschätzen oder zu Klassifizieren ist dabei auch die Qualität des Modells – der Grad des Vertrauens, den man dem Modell entgegenbringen kann. Eine Weiterverwendung von Modellen verlangt zudem eine nachvollziehbare Versionierung. Trotz gewünschter Automatisierung dieser Schritte sind diese in vielen Fällen „von Hand“ durchzuführen. Als ein Fazit ergibt sich an dieser Stelle: Eine Voraussetzung für eine bessere Unterstützung ist eine stärkere Formalisierung der Beschreibung der der Modellierung zugrunde liegenden Ausgangsdaten und der Anforderungen an das zu erstellende Modell. Eine derartige formale Beschreibung kann auch zum Wiederauffinden von Modellen u. U. sinnvoll eingesetzt werden.

3.2 Modellbibliotheken

Modellierungsaufwand kann reduziert werden, wenn auf Modelle aus Modellbibliotheken (siehe z. B. [12]) zurückgegriffen werden kann. Es bleibt dann „nur“ die Aufgabe für einen Anwendungsfall, die passenden Modelle zu finden und zu parametrisieren. Zu unterscheiden ist u. a. zwischen Tool-spezifischen, unternehmensübergreifenden und unternehmensweiten sowie nutzereigenen Bibliotheken. Die unter 3.1 angesprochenen erforderliche Konsistenz kann bei der Erstellung der Modell für eine Bibliothek sicher gestellt werden. Durch bekannte Struktur der Modelle einer Bibliothek wird die Bereitstellung von Parametrisierungs-vorschriften und –werkzeugen erleichtert. Diese verfügbar zu machen trifft in erster Linie auf Modelle zu, die in breiterem Umfang eingesetzt werden. Ein weiterer Vorteil der Verwendung von Modellbibliotheken besteht darin, dass Modelle „en bloc“ von einer Modellierungssprache in eine andere oder für eine andere Simulationsumgebung umgesetzt werden können. Wenn sich diese Schritte nicht automatisieren lassen, ist sicher ein einmaliger manueller Aufwand akzeptabel. Allerdings bleibt dann das Problem, dass bei Änderung von Modellen in einer Bibliothek die Modelle in anderen Bibliotheken nachgezogen werden müssen. Als Fazit ergibt sich: Modellbibliotheken sind ein möglicher Weg, den Modellierungsaufwand beim Nutzer zu reduzieren und Modelle von einer Simulationsumgebung oder einer Modellbeschreibungssprache in eine andere zu überführen. Voraussetzung für die Nutzung der Vorteile ist die Auswahl von Grundmodellen, deren wiederholte Verwendung gesichert ist, die erprobt sind und über einen längeren Zeitraum stabil genutzt werden können. Unter dem Aspekt Systemsimulation ist die Festlegung des Umfangs der für diese Zwecke zu verwendenden Modelle und die Festlegung auf zugrunde liegende Modellierungsansätze eine nicht abgeschlossene Aufgabe.

3.3 Modelltransformation und Modellaustausch

Im vorangegangenen Abschnitt ist auf die Frage der Modelltransformation kurz eingegangen worden. Bei der Umsetzung ganzer Modellbibliotheken, die wiederholt verwendet werden, ist sicher ein gewisser Aufwand akzeptabel. Etwas anders sieht die Situation aus, wenn einmalig einzelne Modelle umgesetzt werden müssen. In Sonderfällen mag die Umsetzung von Modellen aus einer Modellbeschreibungssprache in eine andere automatisierbar sein. In der Regel ist dies aber eine Aufgabe, die manuelle Eingriffe verlangt. Ein Grund mag darin zu suchen sein, dass Eigenheiten der jeweils genutzten Simulationsumgebung berücksichtigt werden müssen. Im Zusammenhang mit hybriden Modellen ist auf einen diesbezüglichen Aspekt im Abschnitt 2 kurz eingegangen worden.

Günstiger erscheint es, als Ausgangspunkt für eine automatisierte Umsetzung eine formale Beschreibung zu wählen, die im Wesentlichen die dem Modell zugrunde liegende mathematische Beschreibung repräsentiert. Aus dieser Beschreibung können dann Modelle in unterschiedlichen Beschreibungssprachen generiert werden. Erste Standardisierungsbemühungen hierzu hat es unter der Bezeichnung „Common Model eXchange"-Format [13] gegeben. Sehr viel erfolgversprechender scheinen die Aktivitäten zum Functional MockUp Interface [14] zu sein, die im Umfeld der Modelica-Standardisierung angesiedelt sind. Möglichkeiten und Grenzen des FMI in Zusammenhang mit der Nutzung anderer Beschreibungssprachen, z. B. VHDL-AMS, zu untersuchen, ist sicher eine interessante Problemstellung. Als kurzes Fazit in Zusammenhang mit der Modelltransformation ist festzustellen: Die Transformation von Modellen von einer Beschreibungssprache in eine andere ist nach vorliegenden Erfahrungen ein Problem, dass sich in der Regel ohne manuelle Eingriffe nicht lösen lässt. Günstiger erscheint es, von einer formalen Beschreibung zugrunde liegender mathematischer Zusammenhänge auszugehen und darauf aufbauend Modelle in unterschiedlichen Beschreibungssprachen und für unterschiedliche Simulationsumgebungen zu generieren.

3.4 Co-Simulationslösungen

Umgangen werden kann die Modelltransformation, wenn Simulationswerkzeuge die gleichzeitige Verwendung von Modellen in unterschiedlichen Beschreibungssprachen unterstützen. Das ist bei kommerziellen Werkzeugen häufig der Fall. Speziell für die Verwendung von Modelica-Beschreibungen in im Elektronikentwurf typischerweise eingesetzten Werkzeugen sind aber beispielsweise keine Lösungen bekannt.

Daneben ist der Einsatz von Co-Simulationslösungen eine Möglichkeit, Modelle, die für unterschiedliche Simulationswerkzeuge erstellt worden sind, gemeinsam im Rahmen einer Systembeschreibung zu nutzen. Solche Co-Simulationslösungen sind u.a.

- TISC Suite [15]
- Functional Digital Mock-Up FDMU [16]
- System Architect Designer SyAD [17]

Die zugrunde liegenden Ansätze unterscheiden sich u.a. hinsichtlich der Kopplung der Simulatoren und der Beschreibung des zu untersuchenden Gesamtsystems.

4 Weiterführende Schritte

Die zu untersuchenden Systeme werden immer komplexer. Analoge und digitale Hardware müssen gemeinsam mit Softwarekomponenten und Beschreibungen der Umgebung betrachtet werden. Ein entscheidender Punkt für die Systemsimulation ist die Verfügbarkeit geeigneter Modelle. Verschiedene in diesem Zusammenhang zu beachtende Aspekte sind in diesem Beitrag erwähnt worden. Einerseits existiert eine Reihe von Modellen, die in unterschiedlichen Schritten des Entwicklungsprozesses entstanden sind. Andererseits erweist sich deren Nutzung für Zwecke der Systemsimulation – auch auf Grund der Vielfalt der eingesetzten Lösungen – und unter Umständen schon ihr Auffinden teilweise als schwierig. So besteht der – zumindest subjektive - Wunsch nach etwas mehr Systematik als Grundlage für formalisierte Beschreibungen, die dann auch dazu beitragen können, durch automatisierte Lösungen, den teilweise hohen Aufwand zu reduzieren. Folgende Möglichkeiten erscheinen u.a. als zweckmäßig:

Eine formale Beschreibung von Ausgangsdaten für die Modellierung, Klassifizierung von Modellanforderungen und Modellierungsansätzen kann helfen, den Aufwand bei der Erstellung von Modellen zu reduzieren und gleichzeitig dazu beitragen, die Suche nach existierenden Modellen zu erleichtern. Ein Ansatz könnte in diesem Zusammenhang sein zu prüfen, ob sich für diese Fragestellung vorhandene XML Schemata erweitern lassen oder die Definition eines neuen sinnvoll ist.

Allgemein akzeptierte Modellbibliotheken können sich als wichtiges Hilfsmittel beim Modellaustausch erweisen. Die Definition von Bestandteilen derartiger Bibliotheken, speziell auch für die Systemsimulation im Automobilbau, kann dazu beitragen, Werkzeuge für den automatisierten Übergang von einer für den Komponentenentwurf genutzten Abstraktionsebene auf eine für die Systemsimulation zweckmäßige verfügbar zu machen.

Für eine Transformation von Modellen erscheint als Ausgangspunkt die Beschreibung der mathematischen Zusammenhänge zwischen den Klemmengrößen ein vielversprechender Ansatz. Die Konsequenzen von Standardisierungsbestrebungen wie FMI für die Nutzung im Umfeld anderer Modellbeschreibungssprachen wie VHDL-AMS bedürfen aber noch einer näheren Untersuchung.

Die dargestellte Sicht ist sicher sehr subjektiv geprägt, kann aber vielleicht trotzdem Ausgangspunkt für weitere Überlegungen und Diskussionen, auch im Rahmen der ASIM, sein.

Literatur

[1] Stachowiak, H.: Allgemeine Modelltheorie. Springer-Verlag, 1973

[2] Faure, C.; Gaid, M. B. et al: Methods for real-time simulation of Cyber-Physical Systems : application to automotive domain. 7th Int. Wireless and Mobile Computing Conference (IWCMC), 2011, pp. 1105-1110. DOI 10.1109/IWCMC.2011.5982695

[3] Schwarz, P.: Requirements for the Simulation of Complex Heterogeneous Systems. 8th Modelica Conference Dresden, 2011. Online: https://www.modelica.org/events/modelica2011/Proceedings/pages/keynote/keynote_schwarz.pdf

[4] AMS assertion subcommittee des Accellera Verilog-AMS Technical Subcommittee. Online: http://www.eda.org/twiki/bin/view.cgi/VerilogAMS/AmsAssertions

[5] Abdallah, A.; Feron, E. M. et al: Hardware/Software Codesign for Aerospace and Automotive Systems. Proc. of the IEEE, vol. 98 (2010), issue 4, pp. 584 – 602. DOI 10.1109/JPROC.2009.2036747

[6] Derler, P.; Lee, E. A.; Sangiovanni-Vincentelli, A.: Modeling Cyber-Physical Systems. Proc. of the IEEE, vol. 100 (2012), issue 1, pp. 13-28. DOI 10.1109/JPROC.2011.2160929

[7] Lee, E.A.: Cyber Physical Systems: Design Challenges. 11th IEEE International Symposium on Object Oriented Real-Time Distributed Computing (ISORC), 2008, pp. 363 – 369. DOI 10.1109/ISORC.2008.25 (auch verfügbar als Technical Report UCB/EECS-2008-8, University of California at Berkeley, 2008)

[8] Cook, J.: Introduction: Cyber-physical systems and the twenty-first century automobile. NSF Workshop On Cyber-Physical Systems, October 2006. Online: http://varma.ece.cmu.edu/CPS/Position-Papers/Ford.pdf

[9] Senger, P.; Dölling, R.; Rosenstiel, W.: Overview of an environment for automated electrical behavioural modelling using data-based methods. Proc. 2010 Conference on Research in Microelectronics and Electronics (PRIME), 2010, pp. 1-5.

[10] SAE J2546: Model Specification process Standard. SAE Electronic Design Automation Standards Committee, 2002. Online: http://standards.sae.org/j2546_200202/

[11] Richtlinien für die Entwicklung von VHDL-AMS-Modellbibliotheken. In: Modellbasierte Systementwicklung. FAT-Schriftenreihe Nr. 226, 2009. Online: http://www.vda.de/de/publikationen/publikationen_downloads/detail.php?id=781

[12] Erstellung einer VHDL-AMS-Modellbibliothek für die Simulation von Kfz-Systemen. FAT-Schriftenreihe Nr. 207, 2006. Online: http://www.vda.de/de/publikationen/publikationen_downloads/detail.php?id=897

[13] Common Model eXchange Format (CMX) Version Beta 1.1.0, 2006. Online: http://cmx.sourceforge.net/

[14] Functional Mockup Interface (FMI) definition. Online: http://www.modelisar.com/

[15] TISC Suite – Software zur Kopplung mehrerer Simulationswerkzeuge. TLK Thermo GmbH Braunschweig. Online über: http://www.tlk-thermo.com/

[16] Functional Digital Mock-Up FDMU. Projekt der Fraunhofer-Gesellschaft e.V. Online: http://www-past.igd.fraunhofer.de/igd-a2/fdmu/

[17] System Architect Designer SyAD: A tool for system simulation and design. CISC Semiconductor GmbH Klagenfurt. Online: http://www.cisc.at/system-architect-designer-2.html