

In: T. Schwatinski, T. Pawletta: Ontologische Modellierung und Modellgenerierung in der MATLAB/Simulink Umgebung: Die "Tiny SES Toolbox". In: Proc. ASIM-Treffen STS/GMMS 2013, Düsseldorf, ARGESIM Report 41, ASIM Mitteilungsnummer AM 145, ARGESIM/ASIM Pub. TU Vienna, Austria, 02/2013, Seite 57-64. (Print ISBN 978-3-901608-41-4)

## Ontologische Modellierung und Modellgenerierung in der MATLAB/Simulink Umgebung: Die „Tiny SES Toolbox“

Tobias Schwatinski, Thorsten Pawletta

Hochschule Wismar, University of Applied Sciences

*tobias.schwatinski@hs-wismar.de*

Der „System Entity Structure and Model Base“ (SES/MB) Formalismus ist eine Ontologie, die vornehmlich im Bereich des Data-Engineering und der Simulationstechnik verwendet wird. Es wird anhand der „Tiny SES Toolbox“ praktisch gezeigt, wie der SES/MB Formalismus für die Meta-Modellierung von Systemen und für eine automatische Modellsynthese verwendet werden kann. Dazu werden verschiedene Systeme in einer SES spezifiziert und daraus, mittels der Tiny SES Toolbox, ausführbare Matlab/Simulink Modelle generiert.

### 1 Problemstellung

Komplexe Systemstrukturen lassen sich oft durch Graphen darstellen. Die Systemkomponenten beziehungsweise ganze Subsysteme werden als Knoten und deren Beziehungen untereinander durch verbindende benannte Kanten abgebildet. Ein Spezialfall der allgemeinen Strukturierung stellt die Hierarchisierung dar [1]. Im Allgemeinen sind in einem hierarchischen System alle Elemente nach einer bestimmten Rangordnung angeordnet, wobei die Elemente gleicher Rangordnung eine gemeinsame Schicht innerhalb der Hierarchie bilden. Auf der einen Seite können dadurch chaotische Strukturen verhindert werden, auf der anderen Seite werden aber alle möglichen Systemstrukturen auf definierte gerichtete Strukturen beschränkt [1]. Weiterhin werden alle Elemente innerhalb einer Hierarchie, welche nicht weiter zerlegt werden können, als Systemelemente bezeichnet.

Neben der Hierarchisierung spielt die Modularisierung eine bedeutende Rolle innerhalb sämtlicher Ingenieurwissenschaften. Dies gilt im Besonderen für die Softwaretechnik als auch für die Modellbildung und Simulation (M&S). In [1] wird ein Modul als eine weitgehend abgeschlossene Bau- oder Funktionsgruppe definiert. Derartige Module können, innerhalb der M&S, einzelnen Systemelementen einer Hierarchie zugeordnet werden. Somit ist, systemtheoretisch betrachtet, ein Modul die Abstraktion des Ein-/Ausgangsverhaltens eines Systemelements durch eine „Black Box“. Das dynamische Verhalten der „Black Box“ ist entweder mittels Zustandsvariablen, Ein-/Ausgangsfunktionen

beziehungsweise Zustandsüberföhrungsfunktionen oder durch ein Submodell zu konkretisieren.

Die beiden genannten Prinzipien werden, in Form von Signalflussgraphen (CBD: Causal Block Diagrams), in vielen grafischen Modellierungswerkzeugen verwendet, wie beispielsweise in Scilab/Xcos, Matlab/Simulink oder AnyLogic. Ein CBD ist ein Graph, der aus miteinander verbundenen Operationsblöcken gebildet wird. Dabei stellt jede Verbindung eine Signalleitung dar, während jeder Operationsblock eine entsprechende Funktionseinheit definiert. Die Operationsblöcke sind in Bibliotheken katalogisiert und können sowohl zeitbasierte Funktionalitäten wie Integrieren oder Differenzieren, als auch einfache algebraische Funktionalitäten wie Summieren oder Multiplizieren umsetzen.

Der „System Entity Structure and Model Base“ (SES/MB) Formalismus [2] ist eine Ontologie, deren Entwicklung in den 80'er Jahren durch Zeigler begonnen [3] und heute neben der Simulationstechnik insbesondere im Bereich des Data-Engineering verwendet wird [4]. Dabei stellen, ähnlich den CBD, die Hierarchisierung und die Modularisierung zentrale Prinzipien (Axiome) des SES/MB Formalismus dar. Die SES ist in erster Näherung ein Baum mit Knoten, Kanten und Attributen. Darüber hinaus werden auch Variablen, Kopplungen, Zwangsbedingungen und Auswahlregeln definiert. Die Entwicklung der SES ist eng mit der Entwicklung des Discrete Event System Specification (DEVS) Formalismus gekoppelt [3, 5]. Ähnlich den CBD können bei DEVS einzelne Operationsblöcke (atomic DEVS, coupled DEVS) modelliert und durch Signalleitungen (DEVS couplings) miteinander

verbunden werden. Darüber hinaus werden in [6, 7] unterschiedliche DEVS Modelle, in Form eines Metamodells, mit einer SES spezifiziert.

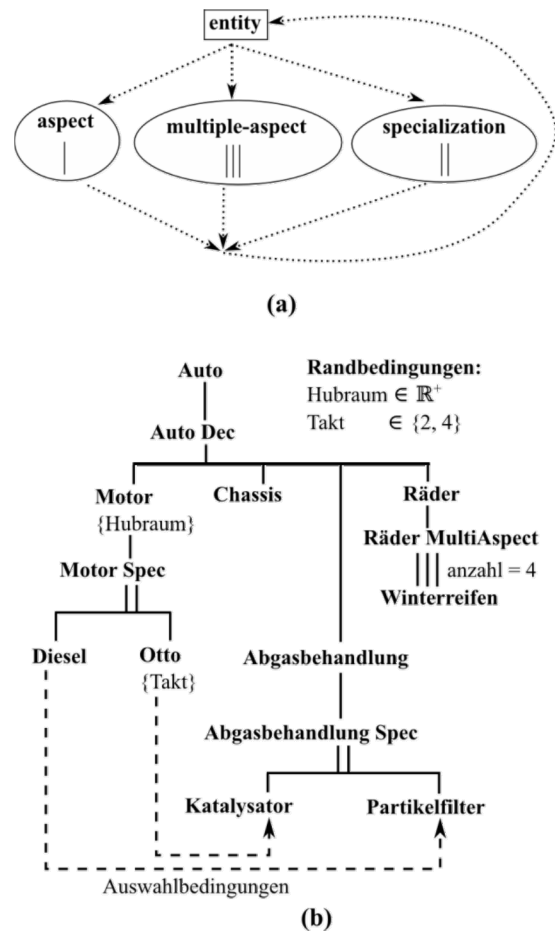
In neuerer Zeit wird durch verschiedene Arbeiten versucht, den formalen SES/MB Formalismus auch im direkten Zusammenhang mit CBD und allgemeinen Blockdiagrammen zu nutzen. In [8] werden beispielsweise fertigungstechnische Prozesskettenvarianten in einer SES spezifiziert und daraus automatisch Matlab/SimEvents Modelle generiert. Weiterhin wird in [9] untersucht, ob anstelle der formalen graphischen Beschreibungsmittel der SES auch die Verwendung adäquater Blockdiagramme möglich ist. Im Rahmen dieses Beitrags wird die praktische Anwendung des SES/MB Formalismus im Zusammenhang mit Matlab/Simulink Modellen thematisiert.

Dazu werden im nächsten Abschnitt die unterschiedlichen Arten von Beschreibungsmitteln einer SES aufgeführt und deren Verwendung für eine ontologische Systemmodellierung diskutiert. Danach wird am Beispiel des Standardregelkreises gezeigt, wie der SES/MB Formalismus für die Meta-Modellierung von dynamischen Systemen und für eine automatische Modellsynthese verwendet werden kann. Im vierten Abschnitt wird exemplarisch die SES von einem einschleifigen Regelkreis mit verschiedenen LTI-Systemen spezifiziert und daraus, mittels der „Tiny SES Toolbox“ [10], ausführbare Matlab/Simulink Modelle generiert.

## 2 Ontologische Systemmodellierung mit der SES

Die System Entity Structure (SES) ist eine Ontologie. Sie bildet einen Baum, dessen Knoten in vier unterschiedlichen Ausprägungen auftreten können [2]. Diese Ausprägungen heißen Entity-, Aspect-, Multiple-Aspect- und Specialization-Knoten. Die prinzipielle Abfolge der Knoten innerhalb einer SES ist in Abbildung 1 (a) dargestellt. Entity-Knoten repräsentieren die Elemente einer realen oder imaginären Welt. Aspect-Knoten dienen der Zerlegung eines Entity-Knotens in feiner aufgelöste Strukturen. Multiple-Aspect-Knoten definieren die Vielfachheit des nachfolgenden Entity-Knotens und Specialization-Knoten repräsentieren Kategorien oder Familien bestimmter Ausprägungen eines Entity-Knotens. Darüber hinaus können bestimmten Knoten oder Kanten Attribute zugewiesen werden.

Abbildung 1 (b) zeigt beispielhaft die SES zur Spezifikation verschiedener Automodelle. Jede Entity *Auto* besteht entsprechend dem Aspect-Knoten *Auto Dec* aus den Entities *Motor*, *Chassis*, *Abgasbehandlung* und *Räder*.



**Abbildung 1:** (a) Allgemeine Struktur einer SES [2],  
(b) Beispiel einer SES

Die Entity *Motor* wird dabei entweder spezialisiert zu der Entity *Diesel* oder der Entity *Otto*. Darüber hinaus wird der Entity *Otto* ein Attribut *Takt* zugewiesen, welches entsprechend den Randbedingungen den Wert 2 oder 4 annehmen kann. Die SES unterstützt auch das Konzept der Vererbung. So wird beispielsweise das Attribut *Hubraum* der Entity *Motor* an die Entities *Diesel* und *Otto* vererbt. Weiterhin sind die jeweiligen Spezialisierungen der Entities *Motor* und *Abgasbehandlung* über Auswahlbedingungen miteinander verknüpft. So führt beispielsweise die Auswahl des Dieselmotors auch zur Auswahl des Partikelfilters. Auf die Entity *Räder* folgt ein Multiple-Aspect-Knoten mit dem Parameter *anzahl = 4*. Deshalb wird dieser Knoten durch vier

Entities des Typs *Winterreifen* aufgelöst. Die SES in Abbildung 1 (b) beschreibt demnach zwei strukturell unterschiedliche Automodelle:

- Dieselmotor, Chassis, Partikelfilter, 4 Winterreifen
- Ottomotor, Chassis, Katalysator, 4 Winterreifen

Außerdem können die Attribute Takt und Hubraum, für beide Automodelle, mit beliebigen Werten belegt werden

Auf diese Weise lassen sich mit einer SES eine Vielzahl von verschiedenen Ausprägungen eines beliebigen Systems anschaulich darstellen. Ursprünglich entwickelt wurde die SES zur deklarativen Beschreibung von Modellstrukturen in der Simulationstechnik. Wenn die SES mit einer Modellbasis (MB) kombiniert wird, lassen sich aus ihr mit Hilfe eines Modellgenerators vollständige Simulationsmodelle ableiten. Diese erweiterte Form der SES, SES/MB genannt, weist jedem Blattknoten der Baumstruktur genau ein Softwaremodul der Modellbasis zu. Dazu ist es notwendig, dass an den Aspect-Knoten die entsprechenden Kopplungsbeziehungen der Entities spezifiziert werden. Darüber hinaus unterstützt die SES auch Auswahlregeln für Aspect- und Spezialisierung-Knoten.

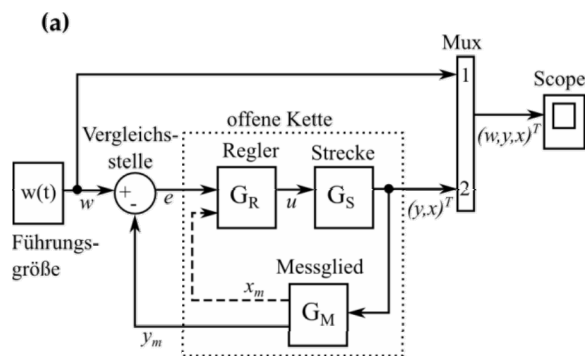
### 3 Modellierung auf Basis des SES/MB Frameworks und das Konzept der Modellsynthese

In diesem Abschnitt wird am Beispiel des Standardregelkreises praktisch gezeigt, wie unterschiedliche Modellstrukturen mittels einer SES sukzessive spezifiziert werden können. Darüber hinaus werden, aus der in diesem Abschnitt entwickelten SES, unterschiedliche Modellstrukturen synthetisiert.

Der Standardregelkreis in Abbildung 2 (a) besteht aus den drei Operationsblöcken: *Führungsgröße*, *Vergleichsstelle* und *offene Kette*. Der Operationsblock *offene Kette* besitzt einen Eingang und zwei Ausgänge und entspricht somit einem SIMO-System. Die Eingangsgröße entspricht dabei der Regeldifferenz  $e$ . Die beiden Ausgangsgrößen entsprechen zum Einen der gemessenen Regelgröße  $y_m$  und zum Anderen dem Vektor  $(y, x)^T$ , welcher sich aus den Streckenzuständen  $x$  und der Regelgröße  $y$  zusammensetzt. In diesem Zusammenhang stellt

lediglich  $y_m$  die Regelgröße für die Rückführung dar, während  $(y, x)^T$  ausschließlich informativen Charakter hat. Zum Zwecke der graphischen Bewertung von Simulationsergebnissen, wie z.B. der Auswertung von Führungsgröße  $w$ , Regelgröße  $y$  und Streckenzuständen  $x$ , sind außerdem ein *Mux*- und ein *Scope*-Operationsblock in Abbildung 2 (a) vorhanden.

Abbildung 2 (b) zeigt die erste Schicht der SES des Standardregelkreises. Dabei sind die relevanten Operationsblöcke als Entity-Knoten und die Kopplungen zwischen den Operationsblöcken als Kantenattribute des Aspect-Knotens *Standardregelkreis Dec* spezifiziert. Das Kantenattribut "Führungsgröße.w, Vergleichsstelle.+" definiert beispielsweise, dass der Ausgang  $w$  des Blocks *Führungsgröße* mit dem Eingang  $+$  des Blocks *Vergleichsstelle* verbunden ist.



Führungsgröße  $w$ , Regeldifferenz  $e$ , Stellgröße  $u$ , Zustände  $x$ , Regelgröße  $y$

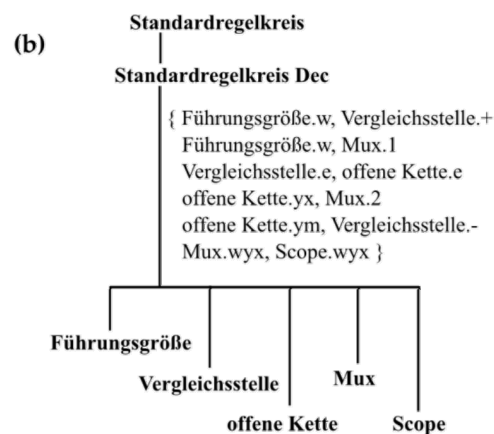


Abbildung 2: Standardregelkreis mit SES

Im Folgenden werden die Entity-Knoten *Führungsgröße* und *offene Kette* der SES aus Abbildung 2 (b) sukzessive erweitert. Zunächst wird in Abbildung 3 (a) die Entity *Führungsgröße* um einen Teilbaum erweitert. Dieser Teilbaum definiert

eine Taxonomie und ordnet verschiedene Signalgeneratorblöcke, in Form von spezialisierten Entities, der allgemeinen Entity *Führungsgröße* zu. Außerdem sind die Entities *Sinus* und *Sprung* durch Attribute parametrisiert. Weiterhin können Auswahlregeln, für die Auswahl einer konkreten Spezialisierung, als Kantenattribut des Specialization-Knotens *Führungsgröße Spec* definiert werden.

Abbildung 3 (b) zeigt, in Form eines weiteren Teilbaums, die Taxonomie der Entity *offene Kette*. Diese unterscheidet zwischen einem Zustandsraummodell *ZR-Modell* und einer allgemeinen Systembeschreibung *allgemeines System*. Die Entity *allgemeines System* setzt sich dabei, entsprechend dem Aspect-Knoten *allgemeines System Dec*, aus den Entities *Regler*, *Strecke* und *Messglied* zusammen. In diesem Zusammenhang können die Komponenten des allgemeinen Systems sowohl linear als auch nichtlinear beschrieben werden. Abbildung 2 (a) zeigt dazu exemplarisch die Komposition der offenen Kette aus einzelnen Übertragungsfunktionen. Weiterhin sind Auswahlregeln und Kopplungsbeziehungen als Kantenattribute angegeben.

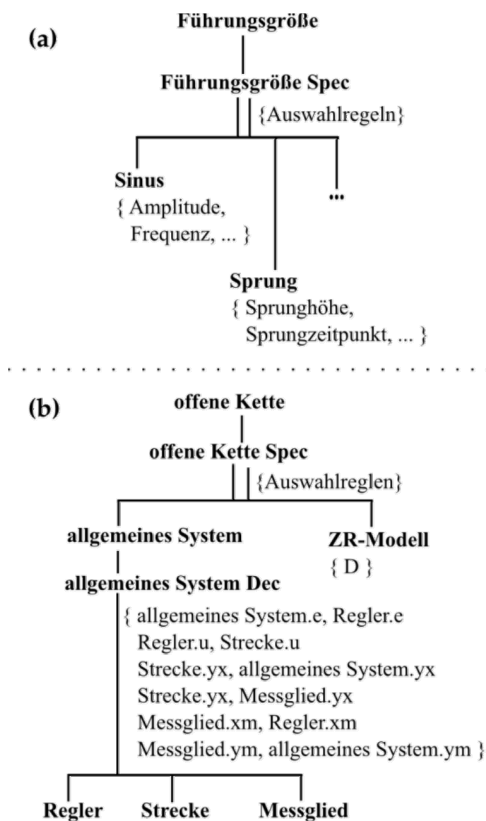


Abbildung 3: SES des Standardregelkreises, 2. Teil

In Abbildung 4 werden sämtliche Blattknoten der SES in Abbildung 3 (b) um Teilbäume erweitert. Abbildung 4 (a) zeigt die Taxonomie der Entity *ZR-Modell*. Diese umfasst verschiedene *SIMO*-Systeme, welche durch die Attribute Systemmatrix *A*, Eingangsmatrix *B*, Ausgangsmatrix *C* und der Durchgangsmatrix *D* hinreichend spezifiziert werden. Im vorliegenden Fall wird die Durchgangsmatrix *D* von der Entity *ZR-Modell* vererbt und ist somit für alle *SIMO*-Entities identisch.

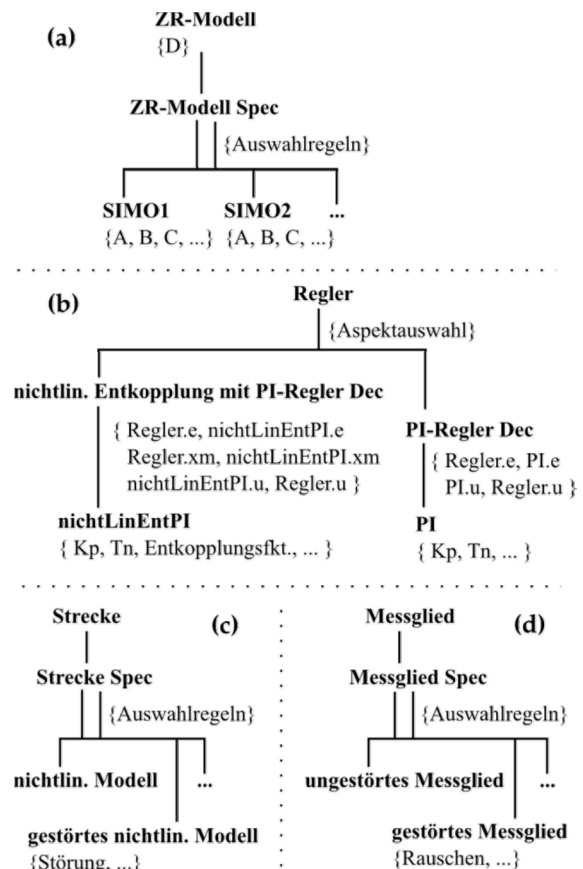


Abbildung 4: SES des Standardregelkreises, 3. Teil

Abbildung 4 (c) und (d) stellen Taxonomien der Entities *Strecke* und *Messglied* dar. In beiden Fällen umfassen sowohl die Regelstrecke als auch das Messglied verschiedene nichtlineare mathematische Modelle zzgl. deren Attribute. Die konkrete Struktur dieser Modelle ist nicht näher spezifiziert.

Abbildung 4 (b) spezifiziert für die konkrete Reglerstruktur der Entity *Regler* zwei unterschiedliche Aspekte zzgl. zugehöriger Auswahlregeln. Zum Einen kann der Regler in Form eines gewöhnlichen *PI*-Reglers, mit den beiden Attributen Proportionalbeiwert *Kp* und Nachstellzeit

$T_n$ , ausgeführt werden. Zum Anderen können die nichtlinearen Streckenmodelle *Strecke* durch die nichtlineare Entkopplung linearisiert und somit einfach geregelt werden, wenn zusätzlich die Messung der Streckenzustände  $x$  möglich ist. In diesem Zusammenhang soll die Schätzung von Streckenzuständen durch einen Beobachter nicht berücksichtigt werden. Beim vorliegenden Anwendungsbeispiel wird innerhalb der offenen Kette, entgegen dem üblichen Aufbau, der durch das Messglied bestimmte Streckenzustand  $x_m$  direkt an den Regler übermittelt. Einen derartigen nichtlinearen Regler stellt die Entity *nichtLinEntPI* dar, welche eine Kombination aus nichtlinearer Entkopplung und PI-Regler repräsentiert und neben den Attributen  $K_p$  und  $T_n$  auch eine *Entkopplungsfunktion* definiert.

Die einfache SES des Standardregelkreises, aufgespalten in die Abbildungen 2 bis 4, spezifiziert 20 unterschiedliche Modellstrukturen. Darüber hinaus werden 18 Attribute für die Parametrierung der einzelnen Modellstrukturen definiert. Durch die Beschneidung der SES, *Prunen* genannt, kann aus der Menge an möglichen Modellstrukturen einer SES genau eine Modellstruktur synthetisiert und parametrisiert werden. Dazu muss sämtlichen Attributen ein konkreter Wert zugewiesen und alle Entscheidungsknoten, wie Specialization-Knoten, Aspect-Knoten und Multiple-Aspect-Knoten aufgelöst werden. Während des Prunens werden unter anderem Entities umbenannt, Kopplungsbeziehungen aktualisiert und Teilbäume bzw. Parameter vererbt. Weitere Details dazu sind in [2, 4] zu finden. Das Ergebnis des Prunens ist die Pruned Entity Structure (PES), welche genau eine Systemstruktur der ursprünglichen SES darstellt. Die Abbildungen 5 und 6 zeigen beispielhaft die PES zweier möglicher Modellstrukturen, welche aus der zuvor betrachteten SES des Standardregelkreises abgeleitet wurden.

#### 4 „Tiny SES Toolbox“ und deren Verwendung in Matlab/Simulink

In diesem Abschnitt wird die praktische Verwendung der „Tiny SES Toolbox“ von der Spezifikation von Metamodellen bis hin zur automatischen Modellgenerierung von Matlab/Simulink Modellen thematisiert. Die theoretische Grundlage dafür bildet der SES/MB Formalismus, welcher jedem Blattknoten einer SES genau ein Softwaremodul aus einer Modellbasis (MB) zuordnet und darüber hinaus die Ableitung ausführbarer simulatorspezifischer Simulationsmodelle aus einer PES ermöglicht.

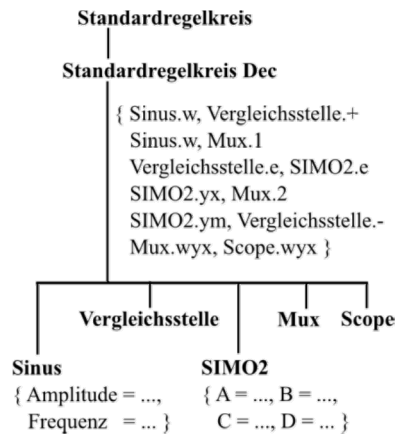


Abbildung 5: PES Beispiel 1

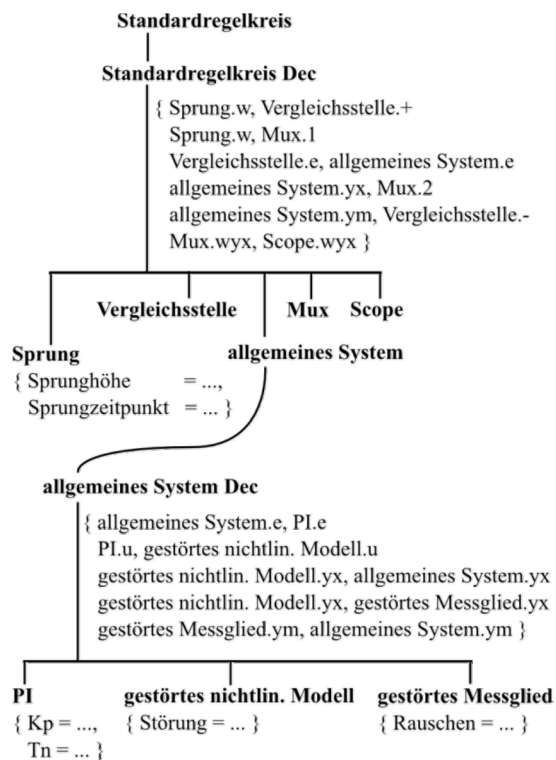


Abbildung 6: PES Beispiel 2

Das Konzept der Modellgenerierung, auf Grundlage des SES/MB Formalismus, zeigt Abbildung 7. Als Anschauungsbeispiel dient im Folgenden der einschleifige Regelkreis in Abbildung 8. In diesem Beispiel können die allgemeinen Operationsblöcke *Führungsgröße* und *offene Kette* durch andere konkrete Operationsblöcke ausgetauscht werden. Im vorliegenden Fall soll anstelle des Operationsblocks *Führungsgröße* entweder ein Sinusgenerator oder ein Sprunggenerator verwendet werden. Weiterhin soll der Operationsblock *offene Kette* durch zwei

unterschiedlich parametrisierte LTI-SISO-Systeme modelliert werden. Die Modellbasis des einschleifigen Regelkreises zeigt Abbildung 9. Die zugehörige SES zeigt Abbildung 10.

Die SES in Abbildung 10 spezifiziert 4 unterschiedliche Modellstrukturen und definiert 16 Attribute. Weiterhin sind an den beiden Specialization-Knoten in Abbildung 10 (b) und (c), in Form von Kantenattributen, verschiedene Auswahlregeln definiert. Darüber hinaus definiert die SES in Abbildung 10 (d) die drei SES-Variablen: *auswahlFührung*, *auswahlLTI* und *parameterFührung*. Diese Variablen sind ausschließlich für das Prunen von Bedeutung und werden z.B. für die logische Auswertung von Auswahlregeln oder für die Wertzuweisung an

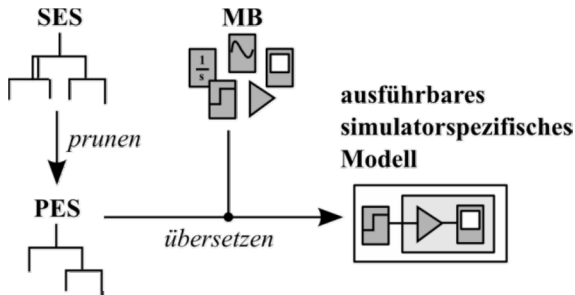


Abbildung 7: Modellgenerierung nach SES/MB Formalismus

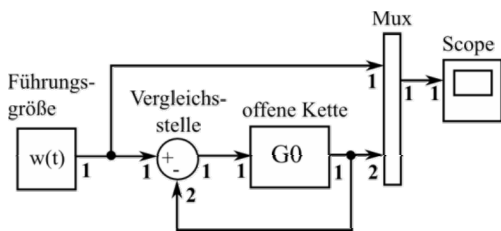


Abbildung 8: Einschleifiger Regelkreis

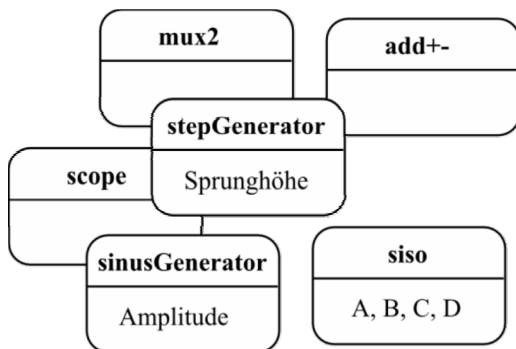


Abbildung 9: Modellbasis des einschleifigen Regelkreises

parametrisierbare Attribute verwendet. Die semantischen Bedingungen in Abbildung 10 (d) definieren zusätzlich die zulässige Wertemenge für jede einzelne Variable. In Abbildung 10 (b) bestimmt die Variable *auswahlFührung* die konkrete Ausführung der Entity *Führungsgröße*. Außerdem dient die Variable *parameterFührung* zur

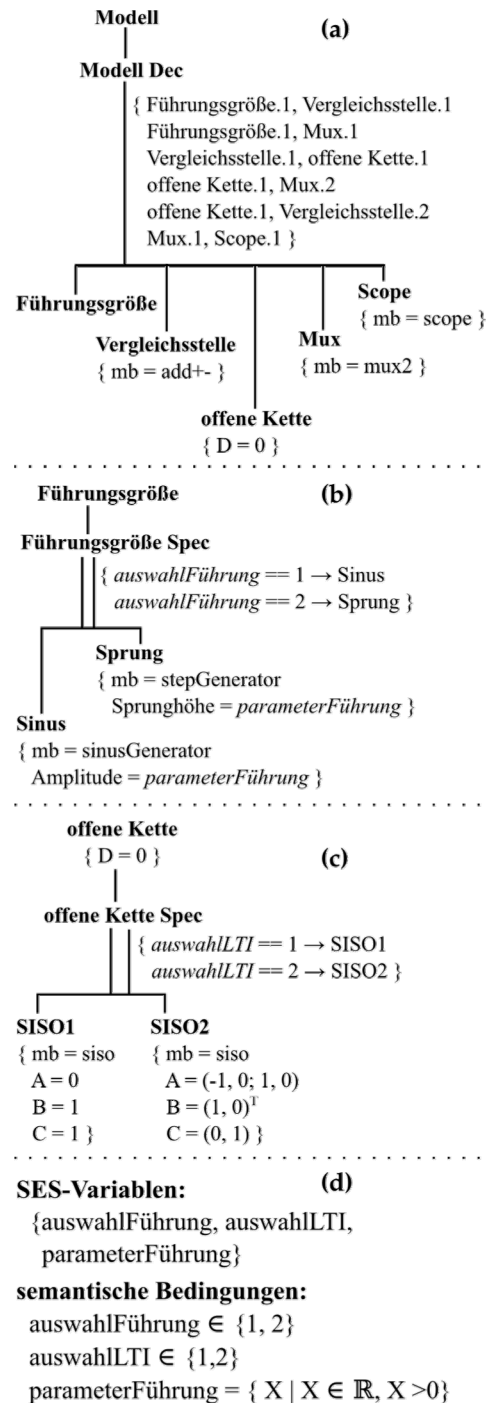


Abbildung 10: SES des einschleifigen Regelkreises

```

1  /*Definition von Knoten*/
2  node(modell, [ ]).
3  node(modellDec, [ ]).
4  /*Definition von Kanten*/
5  edge(modell, modellDec).
6  edge(modellDec, fuehrungsgroesse).
7  /*Definition von Kopplungen*/
8  decomp(modellDec, modell,
    [[fuehrungsgroesse,1,vergleichsstelle,1],
     [fuehrungsgroesse,1,mux,1],
     [vergleichsstelle,1,offeneKette,1],
     [offeneKette,1,mux,2],
     [offeneKette,1,vergleichsstelle,2],
     [mux,1,scope,1]
    ]).
9  /*Definition von Attributen*/
10 parameter(mux, [value(mb, mux2)]).
11 parameter(scope, [value(mb, scope)]).
12 /*Definition von Auswahlregeln*/
13 specrule(offeneKetteSpec, offeneKette,
    [[sisol, [value(auswahlLTI, 1)]],
     [siso2, [value(auswahlLTI, 2)]]
    ]).
14 /*Definition von semantischen
    Bedingungen*/
15 semanticconds(
    (value(parameterFuehrung,X), X>0 )
    ).
    
```

Abbildung 11: Auszug aus der prädikatenlogisch formulierten SES

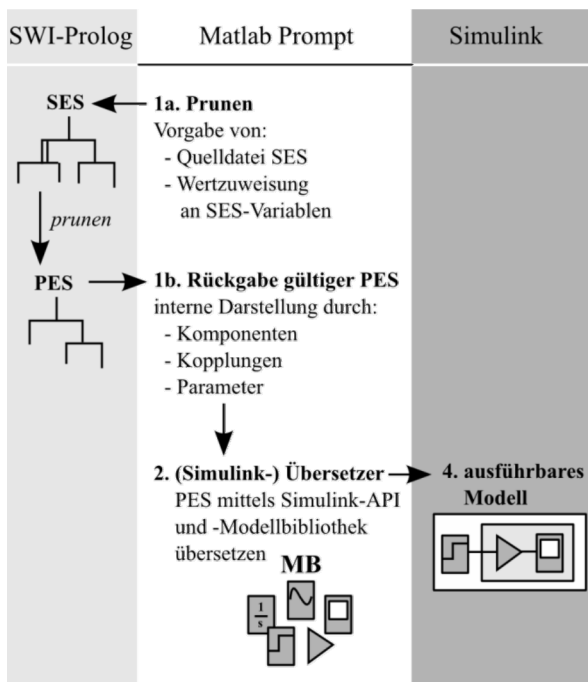


Abbildung 12: Verwendung der „Tiny SES Toolbox“

Parametrierung der beiden Attribute *Amplitude* und *Sprunghöhe*. Weiterhin bestimmt die Variable *auswahlLTI* die konkrete Ausführung der Entity *offene Kette* in Abbildung 10 (c). Das Attribut *mb* weist jedem Blattknoten der SES genau ein Softwaremodul aus der Modellbasis in Abbildung 9 zu.

Die vorliegenden SES muss für die Verwendung mit der „Tiny SES Toolbox“ in einer separaten Datei codiert werden. Dazu sind sämtliche Elemente, wie z.B. Knoten, Kanten und Attribute, prädikatenlogisch zu formulieren. Einige Beispiele zeigt Abbildung 11. Die genaue Syntax ist in [10] dokumentiert.

Darüber hinaus implementiert die „Tiny SES Toolbox“ einen Prune-Algorithmus in SWI-Prolog und einen (Simulink-)Übersetzer in Matlab. Die prinzipielle Verwendung der „Tiny SES Toolbox“ ist in Abbildung 12 dargestellt.

Im ersten Schritt wird die Prune-Funktion in Matlab gerufen und ihr die Quelldatei einer SES zzgl. deren SES-Variablen übergeben. Die Rückgabewerte der Prune-Funktion sind Matlab-Cell-Arrays, welche die Komponenten, Kopplungen, Attribute und Gültigkeit der PES beinhalten. Falls die PES gültig ist, wird im zweiten Schritt der (Simulink-)Übersetzer gerufen. Dieser kann die PES, mittels der Simulink-API und der Simulink-Modellbibliothek, in ein ausführbares Simulink-Modell übersetzen, welches anschließend ausgeführt werden kann. Abbildung 13 zeigt dazu beispielhaft ein Matlab-Skript. Das mittels des Skripts erzeugte Simulink-Modell ist in Abbildung 14 dargestellt.

## 5 Zusammenfassung und Ausblick

Die SES kann für die Spezifikation von unterschiedlichen CBD verwendet werden. Darüber hinaus können im Rahmen des SES/MB Formalismus (i) einzelne Modellstrukturen aus einer SES synthetisiert und (ii) diese unter Verwendung einer Modellbasis in ausführbare simulatorspezifische Modelle übersetzt werden. In diesem Beitrag wurde am Beispiel des Standardregelkreises gezeigt, dass dies auch für Matlab/Simulink-Modelle möglich ist.

Die in diesem Beitrag vorgestellte „Tiny SES Toolbox“ implementiert beide Aspekte des SES/MB Formalismus. Eine Besonderheit der Toolbox ist, dass die Programmierung und Verarbeitung der SES mittels SWI-Prolog realisiert wird. Dazu stellt die „Tiny SES Toolbox“ eine geeignete Matlab-SWI-Prolog Schnittstelle bereit. Daneben beinhaltet die

Toolbox den Prototyp eines Übersetzers, der einzelne Modellstrukturen einer SES in ausführbare Simulink-Modelle übersetzt. Somit kann die „Tiny SES Toolbox“ durchgängig von der Meta-Modellierung von Systemen bis hin zur automatisierten Modellgenerierung, innerhalb von Matlab/Simulink, verwendet werden.

Zukünftig wird die „Tiny SES Toolbox“ an verschiedenen Stellen erweitert werden. Ein wichtiges Ziel ist beispielsweise die Übersetzung von SES-Modellstrukturen in Matlab/SimEvents-Modelle.

```

1 %SES-Quelldatei und SES-Variablen
  definieren
2 inputSES = 'standardRegelkreisSES.pl';
3 v_in = { {'auswahlFuehrung', '1'},...
          {'auswahlLTI', '1'},...
          {'parameterFuehrung','0.5'} };
4 %SES prunen
5 [Components,Couplings,Parameter,Valid]= ...
   Matlab_Interface_SES(inputSES, v_in);
6 %(Simulink-)Übersetzer
7 if strcmp(Valid,'valid')
8   %Simulinkmodell initialisieren
9   nameSystem='pesTemp';
10  load_system('Simulink');
11  h=new_system(nameSystem);
12  open_system(h);
13  set_param(h, 'Solver', 'ode45',...
             'StopTime', '10');
14  %PES nach Simulink übersetzen
15  modelBuilder(nameSystem, Components,...
               Couplings, Parameter);
16  %Modell simulieren
17  sim(nameSystem);
18 else
19  disp(['PES not valid!']);
20 End
    
```

Abbildung 13: Modellgenerierung und Modellausführung in Matlab



Abbildung 14: Einschleifiger Regelkreis in Simulink

## 6 Quellenangaben

- [1] Balzert: *Lehrbuch der Softwaretechnik: Basiskonzepte und Requirements Engineering*, 3. Auflage, Spektrum Akademischer Verlag Heidelberg, 2009
- [2] Zeigler, Hammonds: *Modeling and Simulation-Based Data Engineering*, Elsevier Academic Press, 2007
- [3] Zeigler: *Multifaceted Modelling and Discrete Event Simulation*. Academic Press, 1984
- [4] Kim, Won, Sim, Kim: *System Integration Method Based on System Entity Structure*, United States Patent Application Publication, US 2008/0092207 A1
- [5] Zeigler, Sarjoughian: *Guide to Modeling and Simulation of Systems of Systems*. Springer Pub., 2012
- [6] Zeigler, Praehofer, Kim: *Theory of Modeling and Simulation*, Second Edition, Elsevier Academic Press, 2000
- [7] Hagendorf, Pawletta: *A Framework for Simulation Based Structure and Parameter Optimization of Discrete Event Systems*. In: Discrete-Event Modeling and Simulation: Theory and Applications, CRC Press Inc. of Taylor & Francis Group, USA, 2011
- [8] Larek, Brinksmeier, Pawletta, Hagendorf: *Model-Based Planning of Resource Efficient Process Chains Using System Entity Structures*. In: Future Trends in Production Engineering - Proc. of the 1st Conf. of the German Academic Soc. for Prod. Eng. (WGP), Springer Pub., 2013
- [9] Shin, Jung, Lee, Chi, Han: *SES-based Structure Modeling Method Using the Block Diagram*, In: Proc. of The 2012 Int. Conf. on Foundations of Comp. Science (FCS'12), Las Vegas/Nevada, USA, 2012
- [10] Schwatinski, Schmidt, Pawletta: *Tiny SES Toolbox*, Website 2012, [http://www.mb.hs-wismar.de/cea/SES\\_Tbx/sesToolboxMain.html](http://www.mb.hs-wismar.de/cea/SES_Tbx/sesToolboxMain.html)