13. SYMPOSIUM „DESIGN FOR X“
NEUKIRCHEN, 10. UND 11. OKTOBER 2002

PRODUKT UND PROZESSE AUS SICHT DER MODELLBILDUNG UND SIMULATION

Willy Schweiger, Kristin Paetzold, Baijun Shi

Zusammenfassung

Für die in den frühen Phasen besonders wichtigen Parametermodelle zur Produkt- und Prozeßbeschreibung konnte gezeigt werden, daß das Automatenparadigma geeignet ist, alle erforderlichen formalen Zusammenhänge einheitlich zu beschreiben. Dies gilt sowohl in Hinblick auf domänenübergreifende Systeme (Mechatronik) als auch zur geschlossenen Darstellung zeitkontinuierlicher, zeitdiskreter und insbesondere reaktiver Prozesse. Bestehende kommerzielle Simulationswerkzeuge erlauben auf dieser Basis eine effiziente Modellierung und Simulation. Es sei aber nicht zuletzt daran erinnert, daß eine sorgfältige ingenieurmäßige Vorbereitung eines Modells unvermeidbar ist.

1 Einleitung

Formale Produkt- und Prozeßbeschreibungen waren und sind Gegenstand vieler Forschungsvorhaben. Die Ergebnisse dieser Forschungen schlagen sich in einer kaum überblickbaren Anzahl von Veröffentlichungen nieder. Dennoch herrscht, mit steigender Tendenz nach wie vor Handlungsbedarf für weitere einschlägige Forschungsaktivitäten.

Der vorliegende Beitrag will dies begründen und daraus resultierend versuchen, einige Konturen dieser Beschreibungen aus formaler Sicht schärfer nachzuzeichnen. Die Begründungen liefern die Motivation dazu. Ziel ist eine möglichst einheitliche, formale Beschreibung von einem Produkt und von den, ihm assoziierten Prozessen zu erreichen.

In erster Linie sollen hier die sensitiven frühen Phasen der Produktentwicklung angesprochen werden. Hier ist der Bedarf an massiver Modellbildung und Simulation besonders hoch und gleichzeitig aufgrund des abstrakten Niveaus auch besonders schwierig. Der Wunsch nach formalen Beschreibungsmitteln als Basis für eine umfassende Rechnerunterstützung ist, nicht nur hier, nach wie vor unerfüllt.

Der Beitrag ist wie folgt gegliedert. Nach der Begründung für weitere Forschungsaktivitäten wird kurz das Produktdatenmodell *mfk* des Lehrstuhls für Konstruktionstechnik skizziert. Ihm folgt der Einstieg in die Prozeßbeschreibung vor dem Hintergrund des Produktlebenszyklus. Es wird eine allgemeine Prozeßdefinition diskutiert, welche Kernpunkt der Beschreibung technischer Ein/Ausgabesysteme ist. Die Notwendigkeit der simultanen Berücksichtigung zeitkontinuierlicher, zeitdiskreter und ereignisdiskreter Prozesse wird erläutert. Der einfache Roboter *Sherpa* (Namensgebung KTmfk) dient als Beispiel der dargelegten Erörterungen.

2 Motivation

Die Argumente für die Notwendigkeit von Modellbildung und Simulation in der Produktentwicklung - in diesem Beitrag soll ausschließlich die *digitale* Modellbildung in Betracht gezogen werden - sind vielfach erörtert worden und sollen hier nicht wiederholt werden. In rascher Folge tauchen jedoch immer neue Aspekte auf, welche eine

Intensivierung weiterer Forschung stark motivieren. Hier sollen nur die wichtigsten dieser neuen Aspekte angesprochen werden:

- Technische Produkte werden in ihrem Aufbau und in ihrer Funktionalität immer komplexer – struktureller Aspekt

- Neue Produkte entstehen interdisziplinär (Mechatronik, Medizintechnik, Biotechnik) – domänenübergreifender Aspekt

- Notwendigkeit der Einbettung der Produkte in einen größeren Kontext (z.B. Techno- und Biosphären) – raumübergreifender Aspekt

- Notwendige Betrachtung des gesamten Produktlebenszyklus – domänen-, raum- und zeitübergreifender Aspekt.

Mit einem Produkt ist eine Vielzahl von Prozessen assoziiert. Ein Produkt formal beschreiben bedeutet also in jedem Falle auch diese Prozesse formal erfassen zu können. Der Zwang zu massiver und durchgängiger Modellbildung und Simulation wächst demnach.

3 Das Produkt

Aus Sicht der Modellbildung - die anschließende Simulation wird im weiteren als selbstverständlich vorausgesetzt - muß ein Produkt formal beschreibbar sein. Als Datenmodell wird hier das Modell *mfk* des Lehrstuhls für Konstruktionstechnik verwendet. Bei diesem Datenmodell handelt es sich um ein semantisches relationsbasiertes Modell, welches ausführlich im Beitrag von M. Koch [4] in diesem Symposium diskutiert wird. Im vorliegenden Kontext ist entscheidend, daß in diesem Datenmodell der Begriff der *Funktion* vorhanden und verankert ist. Dieser Funktionsbegriff weist allerdings unter verschiedenen Sichten auch unterschiedliche Interpretationen auf.

Aus der hier interessierenden Sicht der Modellbildung ist es dieser Funktionsbegriff, welcher den Zusammenhang mit den Prozessen herstellt, die mit dem Produkt assoziiert sind. Genau genommen handelt es sich immer um ein Funktionentupel: die Transitionsfunktion und die Verhaltensfunktion (vgl. Abschnitt 4.2).

Das Produktdatenmodell ist dynamisch, d.h. es wächst mit der Entwicklung. Es ist damit hervorragend geeignet, die in Abschnitt 2 geforderten Erweiterungen abzubilden und somit eine umfassende Produktbeschreibung ermöglichen.

4 Prozesse

4.1 Der Produktlebenszyklus

Als Ausgangspunkt für eine Modellbildung technischer Prozesse soll der Produktlebenszyklus (PLZ) dienen. Dieser zentrale Prozeß ist durch folgende Teilprozesse in der beschriebenen zeitlichen Abfolge definiert:

- Produktentwicklungsprozeß "E" mit den Subprozessen Planen, Konzipieren, Entwerfen und Ausarbeiten (VDI 2221), *intellektuelle Prozesse*

- Produktionsprozeß "P" mit den Subprozessen Arbeitsvorbereitung, Produktion (z.B. DIN 8580), Qualitätssicherung und Verteilung, *morphogenetische und logistische Prozesse*

- Produktnutzungsprozeß "N" (Einführung, Gebrauch, Ausmusterung (*Funktionalitäts- oder Verhaltensprozesse parallel mit geriartrischen Prozessen*) und schließlich,

- Produktentsorgungsprozeß "R" (Demontage, Separation, Recycling und Deponieren), *mortalitäre und regenerative Prozesse*.

Eine Detaillierung der Prozeßkette (E, P, N, R) wurde in [11] diskutiert. Dabei wurde im Sinne einer hierarchischen Erhöhung der Granularität eine Dreiebenenhierarchie eingeführt, welche in ihrer Makroebene die Prozeßkette selbst beschreibt, in der ihr folgenden Mesoebene z.B. die Komplexität von Produktionsprozessen und deren Modellierung aufzeigt (z.B. durch Petri-Netze) und schließlich in einer Mikroebene bis hin zur Modellierung von Modellen mit vielen Freiheitsgraden physikalische Sachverhalte mit maximal möglicher (sinnvoller) Detaillierung erfaßt. Es wurde weiterhin aufgezeigt, daß die angesprochenen Prozesse durch die im folgenden Abschnitt gegebene allgemeine Prozeßdefinition voll erfaßbar sind.

Die Beschreibungen der einzelnen Teilprozesse unterscheiden sich natürlich hinsichtlich der Art ihrer *Zustände* und hinsichtlich deren zeitlichen *Verhaltens*. Das bedeutet, daß die jeweiligen abstrakten Prozeßmodellierungen durchaus verschiedene Repräsentationen haben können und werden.

Aus der Sicht des Produktes lassen sich PLZ-Prozesse auch in aktive und passive Prozesse einteilen. So werden unter passiven Prozessen all jene zusammengefaßt, die etwas „an einem Produkt tun“, während die aktiven Prozesse des Produktes im wesentlichen auf dessen Gebrauchsphase konzentriert sind. Es sind die Funktionalitätsprozesse, also jene, welche „das Produkt tut“. Letztere sind damit mit dem Verhalten des Produktes gleichzusetzen.

Bevor jedoch auf die unterschiedlichen Paradigmen technischer Prozesse eingegangen wird, soll zunächst eine allgemeine und für die Beschreibung der fokussierten Prozesse zielführende Darstellung dessen, was unter Prozessen verstanden werden angegeben werden.

4.2 Modellierung von Prozessen

Nach DIN 66201 versteht man unter einem Prozeß eine Umformung und/oder den Transport von Materie, Energie und/oder Information. Die Liste dieser Größen findet sich direkt wieder im Paradigma eines abstrakten technischen Systems in Form eines Eingabe/Ausgabe-Systems [8]. Mit der Akzeptanz dieses E/A-Systems (MIMO-System, <u>m</u>ultiple <u>I</u>nput <u>m</u>ultiple <u>O</u>utput) folgt man der allgemeinen Konstruktionsmethodik.

Die Gesamtheit beliebiger Abhängigkeitsbeziehungen zwischen zwei Mengen (X Eingabe, Y Ausgabe) läßt sich immer als zweistellige Relation

(1) $S \subset X \times Y$

beschreiben, wobei S in der Regel nicht abzählbar ist. Eine kompaktere Beschreibung gelingt durch Einführung einer Operatorformulierung

(2) $Y = S(X)$

Der Operator S muß dabei zwei wesentliche Forderungen erfüllen:

- seine Kausalität, d.h. der Ausgang darf nicht von zukünftigen Eingangsgrößen abhängen

- er muß sich ohne Bezugnahme auf die Eingangsgrößen erklären lassen (Separation des Systemsverhaltens von den Eingangsgrößen).

Wegen allfälliger Trägheiten, sowohl in der Mechanik als auch in der Elektrodynamik wird zur allgemeinen Systembeschreibung eine weitere Variable, der sogenannte Zustandsvektor Z benötigt. Dieser repräsentiert das Erinnerungsvermögen oder die zeitliche Vorgeschichte des Systems und stellt somit eine sogenannte Speichergröße dar. Damit ist der Anschluß an die Automatentheorie gewonnen (vgl. [12]. Der abstrakte endliche Automat (mit *endlichem* Erinnerungsvermögen, entsprechend der Dimension des Zustandsvektors)

(3) $A = (X, Y, Z, z_0, f_T, g, T)$; $f_T: X \times Z \to Z$; $g : X \times Z \to Y$

stellt demnach das Paradigma für ein technisches System dar (zur Erklärung der Zeitvariablen T siehe unten). Das Bild 1 zeigt die Struktur eines derartigen (MEALY-) Automaten. Die Funktion f_T heißt Aktions-, Transfer- oder Evolutionsfunktion (next state map), die Funktion g ist die Ergebnisfunktion, sie beschreibt das erwartete Verhalten (read out map). Substituiert man hierin die allgemeingültige Definition eines Prozesses (in Anlehnung an Duden [2])

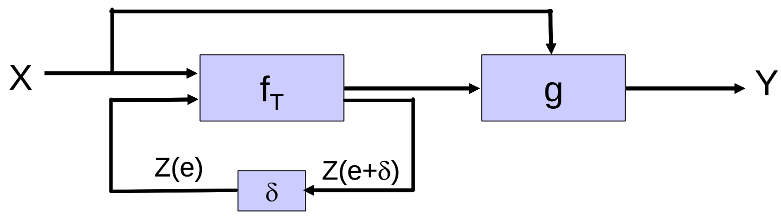

Bild 1: MEALY - Automat

(4) $P = (Z, f_T, z_0, T)$

so erkennt man den direkten Zusammenhang zwischen Automat und Prozeß

(5) $A = (X, Y, P, g)$

Der Zustand $z_0 \subset Z$ definiert einen Startzustand. Die Anwendung des *Aktionsoperators* f_T auf einen Zustand liefert stets einen neuen Zustand. Der Operator wird entweder aufgrund externer Ereignisse (state events) oder aufgrund zeitlicher Ereignisse (time events) aktiviert. Dieser Operator f_T ist damit von fundamentaler Bedeutung für das Fortschreiten (Evolution) eines Prozesses. Wird zur Vereinfachung ein trägheitsloses (gedächtnisloses, sogenanntes kombinatorisches) System angenommen, dann verschwindet P wegen $Z = \emptyset$ und (5) heißt dann trivialer Automat.

Die Elemente $t \in T$ beschreiben monoton wachsende, zeitliche Variable. Für reaktive, ereignisgetriebene Prozesse ist $T \subseteq R^e$, wobei R^e die (i.a. nicht abzählbare) Menge von Ereignissen umfaßt. Für zeitdiskrete Prozesse ist $T \subseteq Z$, mit Z gleich der Menge der ganzen Zahlen. Schließlich ist für zeitkontinuierliche (i.a. physikalische) Prozesse $T \subseteq R^1$, mit R^1 gleich der überabzählbaren Menge der reellen Zahlen (Zeitkontinuum).

Mit der Definition (4) ist noch keine Prozeßbeschreibung verbunden. Diese wird erst erhalten, wenn man einem so definierten Prozeß eine *Dynamik* wie folgt zuordnet. Im einfachsten, wenn auch sehr häufig verwendeten Fall *folgt ein neuer Zustand aus einem vorangegangenen Zustand durch Evolution*, also durch Hinzufügen einer zeitlokalen Wachstumsgröße f*[…]

(6) $z(e_{j+1}) = z(e_j) + f^* [z(e_j), x(e_j)]; z(\ldots) \in Z; \quad x(\ldots) \in X$

Mit der Definition eines Ereignisabstandes (Episode)

(7) $e_{j+1} - e_j = \delta_j$

folgt im Sinne einer verallgemeinerten TAYLOR-Entwicklung (mit $f^* = \delta_j f_T$)

(8) $z(e_j+\delta_j) - z(e_j) = \delta_j f_T [z(e_j), x(e_j)]$

bzw. im Sinne eines verallgemeinerten Differentialquotienten

(9) $[z(e_j+\delta_j) - z(e_j)]/ \delta_j = f_T [z(e_j), x(e_j)]$

Es kann gezeigt werden, daß die allgemeinste Form einer Evolutionsgleichung [9, 13] durch eine verallgemeinerte Bilanzgleichung

$$f_1 (\sigma z, z, x) = f_2 (\sigma z, z, x)$$

gegeben ist. Darin ist σ ein sogenannter backward shift, also eine Zeitverschiebung um einen konstanten Betrag für zeitdiskrete bzw. ein entsprechender zeitlicher Differentialoperator für zeitkontinuierliche Prozesse. Diese Form zeigt, daß Zustandsgleichungen von 1. Ordnung in z und von in 0-ter Ordnung in x sind. Es besteht demnach keine Notwendigkeit, diese Gleichungen in der expliziten Form (wie (6), (8) oder (9)) $\sigma z = f (z, x)$ zu verwenden. Damit gehören implizite Systeme, Deskriptorsysteme (Differential-Algebra-Systeme) usw. ebenfalls zur Klasse der Zustandsmodelle.

Für verschiedene Prozeßarten kann aus der allgemeingültigen Evolutionsgleichung (9) durch Spezialisierung des verallgemeinerten Differentialquotienten eine entsprechend angepaßte Beschreibung erhalten werden.

4.2.1 Reaktive Prozesse

Mit (9) sind ereignisgetriebene oder reaktive Prozesse im allgemeinen beschreibbar. Man kann allerdings in vielen Fällen den verallgemeinerten Differentialquotienten durch logische Operatoren ausdrücken.

Der in (9) verwendete verallgemeinerte Differentialoperator wird dabei durch den XOR-Operator (exklusives OR, Antivalenz) zu ersetzen sein. Damit erhält man die Zustandsraumdarstellung reaktiver Systeme in der Form

(10) $dz(e_k) \sim dt = f_T [z(e_k), x(e_k)]$

mit den diskreten, nicht zeitäquidistant auftretenden Ereignissen $e_k(t)$ (asynchroner Automat, s.u.). Dabei ist die Ableitung "~" im Sinne der binären XOR-Funktion zu interpretieren [1, 10], d.h. $dz[e_k(t)] \sim dt = \lim (\Delta t \rightarrow 0) \{z(t+\Delta t) \text{ XOR } z(t)\}$. Gleichungen wie (10) werden auch als BOOLEsche Differentialgleichungen angesprochen.

4.2.2 Zeitdiskrete und zeitkontinuierliche Prozesse

Bei der Beschreibung zeitdiskreter Prozesse unterstellt man zeitäquidistante Taktschritte (z.B. bei getakteten Produktionsprozessen). Damit erhält man einen konstanten Ereignisabstand Δt und damit aus (9) anstelle des verallgemeinerten Differentialquotienten einen Differenzenquotienten

(11) $[z(t+\Delta t) - z(t)]/ \Delta t = f_T [z(t), x(t)]$

Verläuft der Prozeß zeitkontinuierlich (wie viele physikalische Prozesse), so erhält man durch den Grenzübergang lim ($\Delta t \rightarrow 0$) schließlich eine gewöhnliche Differentialgleichung als Prozeßbeschreibung (Zustandsgleichung)

(12) $dz(t)/dt = f_T [z(t), x(t)]$

Die beiden kanonischen Funktionen f_T und g des Automaten sind also die entscheidenden Beschreibungen eines technischen Systems. Ihre Ermittlung stellt die zentrale Aufgabe der Modellbildung dar.

5 Prozessfelder und Datenreduktion

Beim Übergang von der nahezu geometrielosen, meist rein topologischen Konzeptphase zur folgenden Entwurfsphase nimmt der Anteil an geometrischen Informationen über das Produkt stetig zu. Damit wird der Einsatz geometriegebundener Modelle möglich. Ihre Beherrschung ist für alle physikalischen Modelle die Domäne der finiten Verfahren, allen voraus die universelle Methode der finiten Elemente. Ihr Einsatz erlaubt eine wesentlich feinere Granularität der Information über die gesuchten Größen (Deformation, mechanische Spannungen, Temperaturen, elektromagnetische Feldgrößen usw.). Dies geschieht allerdings auf Kosten einer deutlich größeren erforderlichen und erzeugten Datenmenge bei Anwendung derartiger Modelle. Die Methode der Finiten Elemente erweist sich demnach als Grundlage einer einheitlichen Beschreibung mechanischer, thermodynamischer und elektrodynamischer Feldmodelle.

Sollen derartige Modelle als Teilmodelle eines Gesamtmodells verwendet werden (z.B. sogenannte Beobachtermodelle), ist meist eine Datenreduktion unabdingbar. Eine allgemeine Methode zur Datenreduktion ist die sogenannte Karhunen-Loève Dekomposition (z.B. Krysl [5]). Diese beruht auf der Singularwertzerlegung eines algebraischen Gleichungssystems und bedeutet letztendlich die Erzeugung eines orthogonalen Unterraums mit deutlich geringerer Dimension als der des Ausgangsproblems auf Basis einer Minimierung von Fehlerquadraten.

Für linear-elastische Strukturen geht dies in die Darstellung durch Eigenvektoren über. Dabei treten nicht selten Reduktionen der Ordnung 10^3 und höher auf. Ein sorgfältiger Vergleich des Eigenwertspektrums einer Struktur mit jenem der externen Anregung erlaubt die Festlegung der Höhe der Reduktion.

Für lineare Systeme, welche durch Zustandsgleichungen vom Typ

(13) $dz(t)/dt = f_T [z(t), x(t)] = A\, z(t) + B\, x(t)$

(A konstante Systemmatrix, B konstante Steuer- oder Eingangsmatrix) repräsentiert sind, gelingt die Modellreduktion durch das Verfahren von L. Litz (z.B. Föllinger [3]), welches ebenfalls auf einer modalen Reduktion beruht. In einschlägigen kommerziellen Programmen (z.B. Matlab/Toolboxes) sind hierfür entsprechende Routinen vorkonfektioniert

6 Anwendungsbeispiel: der autonome Roboter SHERPA

Der autonome Roboter SHERPA (Bild 2) ist ein einfach strukturiertes Kerlchen, dessen Intelligenz hier gerade mal 4 innere Zustände aufweist. Er verfügt in der Symposiumsversion aber auch nur über zwei Berührungssensoren, welche ihm Hindernisse von links oder rechts signalisieren. Seine Intelligenz wird durch eine SIMULINK/STATECHART gemäß Bild 3 repräsentiert (nach MathWorks [7]).

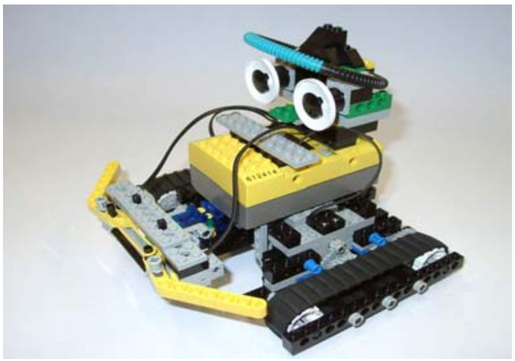

Bild 2: Der Roboter SHERPA

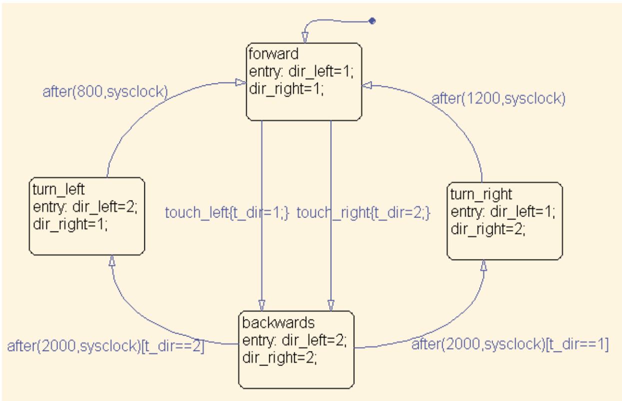

Bild 3: SHERPA's Intelligenztest

Das System hat drei Eingänge (einen Energiestrom über eine in zeitlichen Abständen zu erneuernde Batterie) und zwei Signalströme (die beiden Berührungssensoren). Die beiden Motoren bilden zwei Aktoren. Es handelt sich also um ein MIMO-System (multiple Input, multiple Output System) (vgl. Bild 4). Auf eine detaillierte Prozeßbeschreibung von SHERPA (Fahrdynamik, beschleunigen, verzögern usw.) wurde verzichtet. Der vereinfachte Prozeß beschreibt damit lediglich eine Folge von sich ändernden Richtungen in der Ebene. Der Zustandsraum besitzt 4 Dimensionen, die x-, y-Komponenten der Position in dieser Ebene und die jeweiligen zugehörigen Geschwindigkeitskomponenten. Die Verhaltensfunktion ist die Identität, der Ausgabevektor ist demnach identisch mit dem Zustandsvektor. Das heißt, daß SHERPA einen MEDWEDJEW-oder Semi-Automaten darstellt.

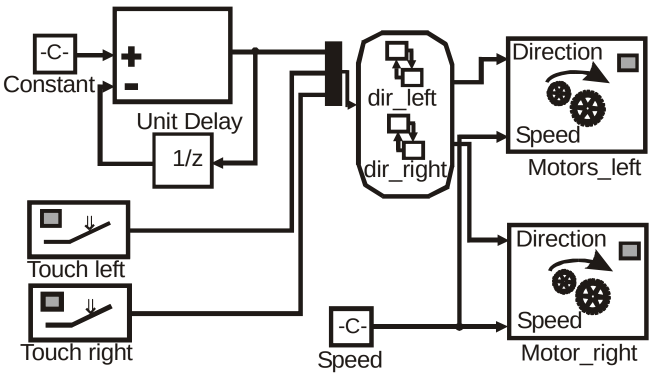

Bild 4: SIMULINK-Modell von SHERPA

SHERPA's Handlungsstrategie ist sehr einfach *forward* (vgl. Bild 3). Diese Strategie wird durch Verwendung des temporalen Logikoperators *after[time]* erreicht. Mit derartigen temporalen Logikoperatoren (näheres z. B. Manna/Pnueli [6]) lassen sich reaktive Prozesse, also solche mit vorgegebenem Zeitablauf modellieren.

7 Literatur

[1] Bochmann, D., Posthoff, C.: Binäre dynamische Systeme. Oldenburg Verlag, 1981.

[2] Duden-Informatik, 2. Aufl. Mannheim, 1993.

[3] Föllinger O.: Regelungstechnik, 6. Aufl. Hüthig Verlag Heidelberg, 1990.

[4] Koch, M., Meerkamm, H.: Durchgängige Funktions-modellierung in den frühen Konstruktionsphasen. Meerkamm H. (Hrsg.) Beiträge zum 13. Symposium DFX, Neukirchen 2002.

[5] Krysl P., Lall S., Marsden J. E.: Dimensional Model Reduction in Non-linear Finite Element Dynamics of Solids and Structures, Int. J. Numer. Meth. Engng. 00, 2000.

[6] Manna Z., Pnueli A.: The Temporal Logic of Reactive and Concurrent Systems, Springer, 1992.

[7] Robot Car: An Embedded Example; The Mathworks Inc, 2001; http://www.mathworks.com/

[8] Pahl G., Beitz W.: Konstruktionslehre, 4. Aufl. Springer 1997.

[9] Polderman J. W., Willems J. C.: Introduction to Mathematical Systems Theory Springer, 1998.

[10] Scheuring R., Wehlan H.: Der Boolesche Differentialkalkül - Eine Methode zur Analyse und Synthese von Petri-Netzen. Automat.-Technik, 39, 1991

[11] Schweiger W., Paetzold K.: **Hierarchische Modellierung von Prozessen.** http://www.sfb396.uni-erlangen.de/

[12] Schweiger W., Paetzold K.: From Protomodelling to Automata Equations, eingereicht bei J. Eng. Design.

[13] Willems J. C.: Paradigms and Puzzles in the Theory of Dynamical Systems; IEEE Trans. Autom. Control, Vol. 36, No. 3, 1991.

Prof. Dr.-Ing. Willy Schweiger,
Dipl.-Ing. Kristin Paetzold,
MSc. Baijun Shi
Lehrstuhl für Konstruktionstechnik
FAU Erlangen-Nürnberg
Martensstraße 9 - D-91058 Erlangen
Tel: xx49-9131-85-23221
Fax: xx49-9131-85-23223
Email: schweiger@mfk.uni-erlangen.de
paetzold@mfk.uni-erlangen.de
shi@mfk.uni-erlangen.de
URL: http://www.mfk.uni-erlangen.de