

**Tommy Baumann / Ronny Krüger / Michael Reeßing /  
Prof. Beat Brüderlin / Prof. Horst Salzwedel**

# **Computergestützter Konzeptentwurf dynamischer und mechatronischer Systeme mit Regelkreisen**

## **EINFÜHRUNG**

Um komplizierte Sachverhalte besser zu verstehen und untersuchen zu können, ist es von großem Nutzen diese zu modellieren und zu systematisieren. Vor allem erspart man sich dadurch teure Tests in der realen Welt. Mit der fortschreitenden Entwicklung werden solche Systeme immer komplexer. Zum Beispiel verdoppelt sich die Komplexität elektronischer Systeme alle 1,5 Jahre. Daraus folgt, dass man zum Erhalt der Produktivität immer mehr „Human Resources“ benötigt, oder man findet einen Weg um diese effizienter einzusetzen. Mit Hilfe von Electronic Design Tools ist dies möglich, da Problemstellungen virtuell modelliert und danach simuliert werden können. Die Entwicklung solcher Software vollzog sich in mehreren Schritten, parallel zum Anwachsen der Komplexität in der realen Welt. Angefangen in den Siebziger Jahren mit dem „Physical Design“ (CAD), in den Achtzigern dem „Logic Level Design“ (Verilog, VHDL), in die Neunziger dem „Functional Level Design“ bis in die heutige Zeit zum „Mission Level Design“ [4]. Das Softwaresystem „MLDesigner“ [2] nutzt erstmals die Möglichkeit zur Systemmodellierung auf Basis dieser neunten Abstraktionsebene. Eine weitere Anforderung an solche Softwaresysteme ist die Erweiterbarkeit und Anpassbarkeit, weswegen es sinnvoll ist universelle Schnittstellen zum Datenaustausch mit der Umgebung bzw. mit angrenzenden Domänen zu schaffen. Dieser Vortrag behandelt die Konzeption, Umsetzung und Verwendung einer solchen Schnittstelle.

## **MLDESIGNER**

„MLDesigner“ [2] (Abbildung 1) ist ein Simulations- und Modellierwerkzeug auf der Systemebene, welches beide Hauptbereiche (Architektur und Funktion) und die hauptsächlichsten

Simulationsdomänen integriert. Die Modellierdomänen umfassen Discrete Event, Finite State Machine, drei Arten von Data Flow (Dynamic, Synchronous, Boolean) und Continuous Time/Discrete Event. Diese Domänen können einzeln oder gekoppelt (für Mehrdomänen-Modelle) verwendet werden. Modelle werden graphisch, durch hierarchische Block Diagramme definiert. Alle Blöcke können parametrisiert werden, für einfache „what-if-Analyse“ um die Wiederverwendbarkeit zu maximieren. „MLDesigner“ bietet eine vollständige Entwurfsumgebung zum Modellieren von komplexen Systemen von einem Missions-Szenario, über Netzwerke, bis hin zur Signalform [3]. Außerdem ist „MLDesigner“ einfach erweiterbar. Anwender können Entwurfsdomänen hinzufügen und Schnittstellen zu existierenden Werkzeugen bilden. Somit können eigene neue Werkzeuge entwickelt werden. Es werden hierarische Datenstrukturen zur Verfügung gestellt. Sockets bieten die Möglichkeit, dynamisch über Schnittstellen mit anderen Entwurfswerkzeugen mit internet-residenten Anwendungen, oder direkt über die Hardware zu kommunizieren. Eine Vielzahl von Konvertern ermöglicht den Datenaustausch mit „First-Generation-Products“, wie „Ptolemy“, „BONeS“ und „COSSAP“. Des Weiteren existiert eine Anbindung an das CASE-Tool „Rational Rose Realtime“. Die mitgelieferte Anwendungsbibliothek, welche in der Grundausrüstung aus über 2000 Design-Blöcken besteht, kann beliebig erweitert werden.

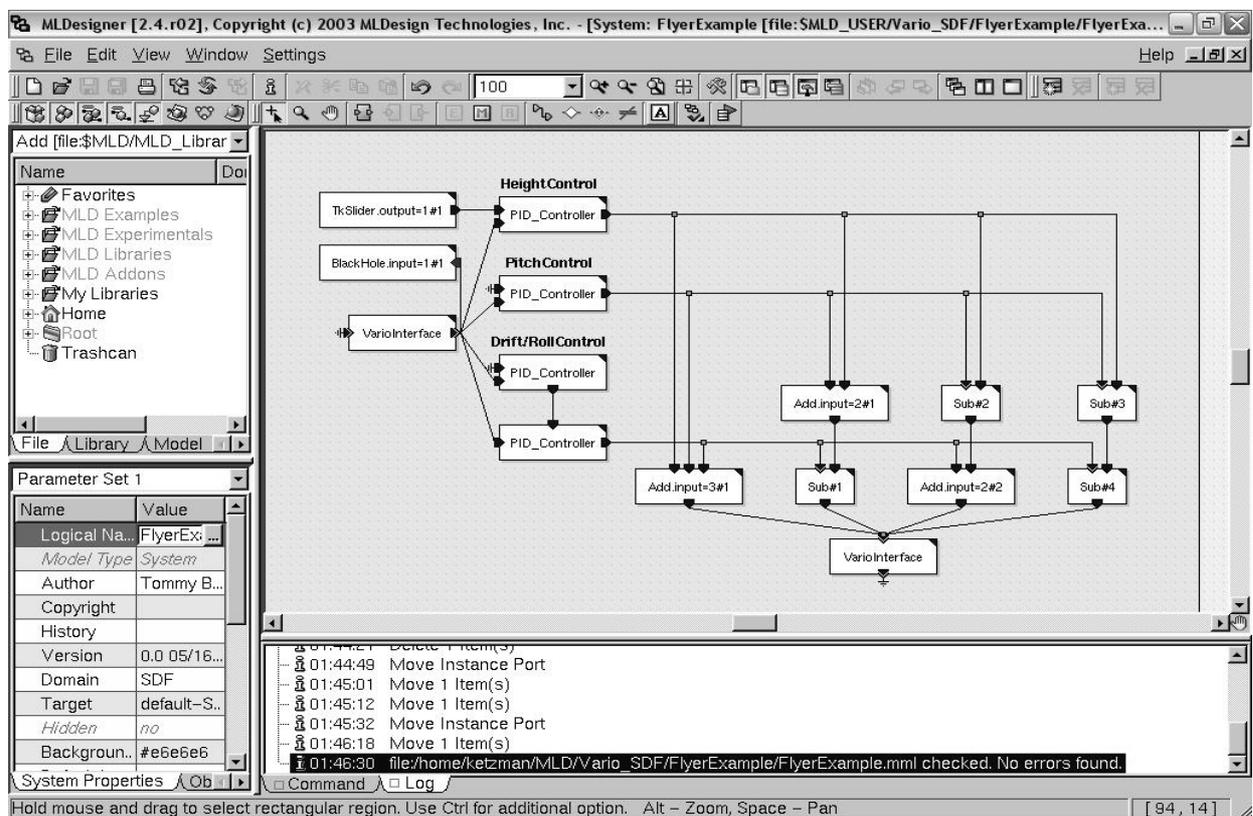


Abbildung 1: Modellierung der Regelkreise im MLDesigner

## MOPS3D

„MOPS3D“ (Abbildung 2) ist eine 3D-Umgebung mit einer integrierten Physik-Simulation. Es können beliebige Objekte eingefügt werden und mit physikalischen Eigenschaften ausgestattet werden. Das System erlaubt eine Echtzeit-Simulation von Massen und Federn, sowie äußeren Kräften (Antriebe), welche auf die Objekte einwirken. Ebenfalls können Kräfte als Folge von Kollisionen behandelt werden [1]. Dieses System erlaubt es außerdem, Sensoren und Aktoren sowie einfache Regelkreise zu modellieren, mit denen die Kräfte gesteuert werden können. Für komplexere Steueraufgaben ist eine Schnittstelle zur Middleware „VARIO“ vorhanden, um einen Datenaustausch zu ermöglichen.

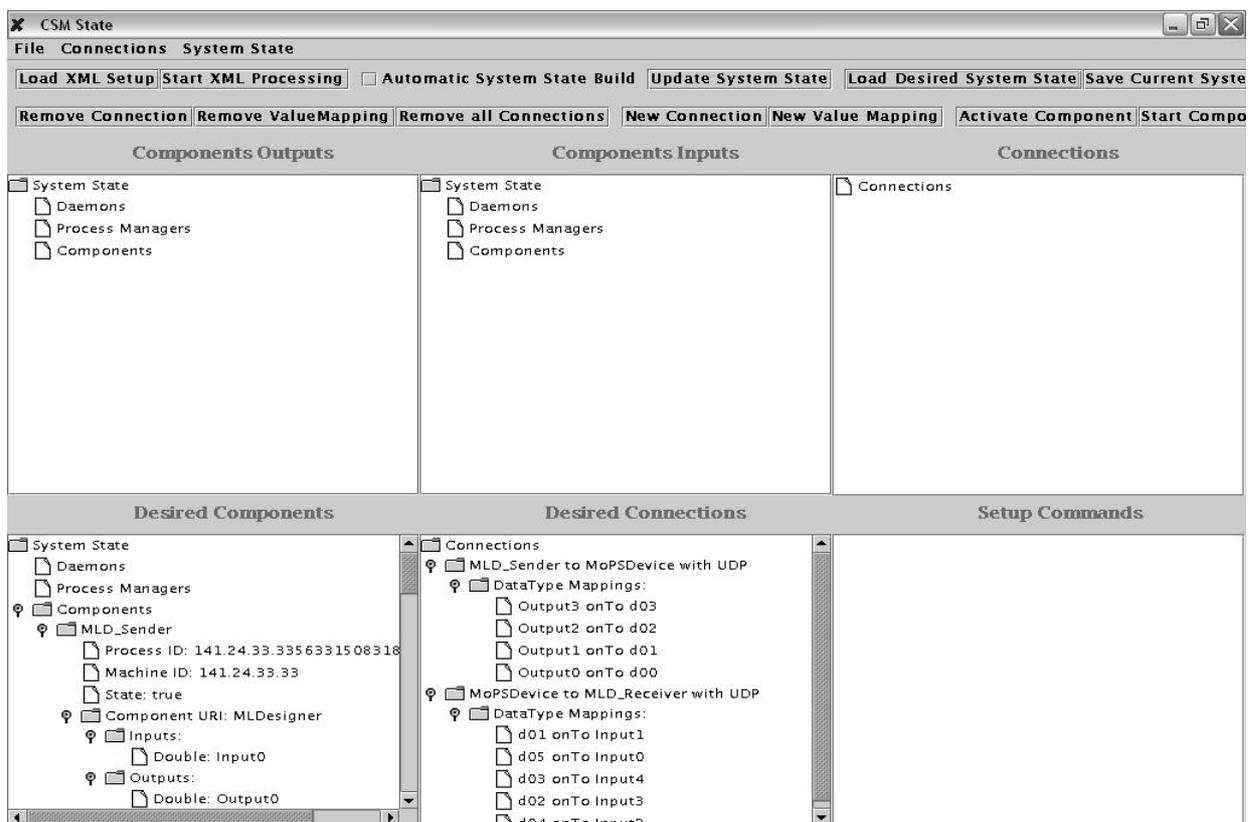


**Abbildung 2: Flugsimulation eines virtuellen Raketenfahrzeugs in MOPS3D**

## VARIO

VARIO (Abbildung 3) ist eine als verteiltes System implementierte Middleware zum Anschluss beliebiger Ein- und Ausgabegeräte an eine Anwendungssoftware [5]. Die Zustände der

angeschlossenen Geräte (Achsenauslenkungen, Buttons usw.) können als logisches Gerät einer Anwendung zur Verfügung gestellt werden. Die Protokollkommunikation erfolgt XML-basiert. Die Datenkommunikation wird durch die Versendung von Byte-Arrays realisiert. Der verwendete Ansatz garantiert gleichermaßen Plattformunabhängigkeit wie Effizienz. Die Verteiltheit erlaubt den Zugriff auf eine Vielzahl von Geräten, die an verschiedene Computer angeschlossen sein können. Für treiberintensive Geräte können entsprechend leistungsfähige Einzelrechner eingesetzt werden. Auf diese Weise lassen sich zusätzliche Schnittstellen verschiedenster Arten einfach hinzufügen. VARIO ist zur Laufzeit voll konfigurierbar. Von der Applikation vorgegebene logische Geräte können so durch verschiedene physikalische Geräte repräsentiert werden, was insbesondere bei der Entwicklung neuartiger Eingabehardware von großem Nutzen ist. Zu diesem Zweck existieren zahlreiche Möglichkeiten, um auf den Datenfluss Einfluss zu nehmen (Filterung, Skalierung, Datentypkonvertierung, Zusammenfassung mehrerer physikalischer Geräte zu einem logischen Gerät). Damit wird es möglich, ein und dasselbe von der Anwendung vorgegebene logische Gerät mittels verschiedener Arten von physikalischen Geräten zu realisieren.



**Abbildung 3: Verbindungsaufbau mittels VARIO**

**KONZEPT**

Das hier vorgestellte Konzept nutzt zur Systemmodellierung das Softwaresystem „MLDesigner“, zur Datenübertragung die Middleware „VARIO“ und zur Benutzer-Interaktion inklusive Visualisierung das Softwaresystem „MOPS3D“ [1]. Die Verbindung zwischen dem „MLDesigner“ und „VARIO“ wurde im Rahmen dieses Projektes entwickelt. Diese Schnittstelle ist in einem hohen Maß variabel bezüglich der zu modellierenden Systeme. Hervorzuheben ist, dass unter anderem, verschiedene Datentypen unterstützt werden. Folgendes Bild zeigt die Interaktion zwischen den einzelnen allein stehenden Applikationen, welche zur Laufzeit miteinander gekoppelt werden (Abbildung 4).



**Abbildung 4: Konzept**

## **ANWENDUNGSBEISPIEL**

Am Beispiel einer einfachen Flugsimulation wird nun die Arbeitsweise der Schnittstelle erläutert. Das physikalische Modell beschreibt ein Flugobjekt mit vier Navigationsdüsen (Abbildung 2), welche über „MLDesigner“ gesteuert werden. Diese vier Düsen sind die Aktoren des zu steuernden Systems. Als Sensoren werden von MOPS3D Werte für Pitch, Roll, Drift und Höhe zur Verfügung gestellt. Über VARIO (Abbildung 3) werden die Werte an „MLDesigner“ (Abbildung 1) gesendet und dort unter anderem über mehrere modellierte PID-Regler verarbeitet. Die daraus resultierenden Werte für die 4 Düsen werden wieder über VARIO an MOPS3D zurückgesendet. Eine besondere Herausforderung in der Systemintegration war die asynchrone Koppelung der Ereignisse. Die Physiks simulation generiert mehrere tausend Ereignisse pro Sekunde, von denen nur ein Bruchteil von VARIO (welches den Datenbus simuliert) übertragen werden kann. Diese Messdaten müssen im Regler wieder synchronisiert werden. Die Regelwerte werden zu den Aktoren (hier Düsen) zurückübertragen. Die Steuerung der Geschwindigkeit, Höhe und Richtung des Flugobjektes kann z. B. über ein Force-Feedback-Lenkrad erfolgen und über eine getracktes Head Mounted Display visualisiert werden. VARIO ist hier ebenfalls zur Koppelung der Eingabegeräte

hilfreich. Die Demo-Anwendung erlaubt simulierte Flüge über den Campus der TU-Ilmenau. Die Regler sind in der Lage auch mit unvorhergesehenen Situationen (z. B. auch bei einer, dem Schwebefahrzeug angehängten, pendelnden Last, oder auch bei einer Kollision dieser Last mit Objekten) umzugehen.

## **ABSCHLUSSBEMERKUNG**

Im Rahmen dieser Publikation wurden erste Versuche zur ganzheitlichen Modellierung und Simulation virtueller Produkte in der frühen Entwurfsphase durchgeführt. Dadurch wird es dem Entwickler ermöglicht Designfehler früher zu erkennen, um auf diese zu reagieren. Außerdem eröffnet die für den „MLDesigner“ entwickelte VARIO-Anbindung eine Vielzahl von neuen Anwendungsmöglichkeiten, da über diese universelle Schnittstelle sehr einfach Daten mit anderen Applikationen ausgetauscht werden können.

### **Literatur- bzw. Quellenhinweise:**

- [1] Ronny Krüger, Michael Reeßing. *Computeranimation auf der Basis physikalischer Simulation und Regelungstechnik*. Hauptseminar, TU-Ilmenau, 2001.
- [2] *MLDesigner Dokumentation*, Version 2.5, 2004. <http://www.mldesigner.com>.
- [3] Horst Salzwedel. *Design Technology Development Towards Mission Level Design*. 49th International Scientific Colloquium TU-Ilmenau, 2004.
- [4] Gunar Schorch. *Entwurf integrierter Mobilkommunikationssysteme auf Missionsebene*. Dissertation, TU-Ilmenau, 2000.
- [5] R. Schönfelder, G. Wolf, M. Reeßing, R. Krüger, B. Brüderlin. *A Pragmatic Approach to a VR/AR Component Integration Framework for Rapid System Setup*. 1th Workshop „Augmented und Virtual Reality in der Produktentstehung“ Heinz Nixdorf Institut Paderborn, 2002.

### **Autorenangabe(n):**

Tommy Baumann  
Prof. Beat Brüderlin  
Technische Universität Ilmenau, Max-Planck-Ring 14, PF 100565  
98684, Ilmenau  
Tel.: (+49)3677/69-2767  
Fax: (+49)3677/69-1285  
Email: tommy.baumann@gmx.de