

METAMODELLE FÜR HYBRIDE DYNAMISCHE SYSTEME

eingereichtes
HAUPTSEMINAR

von

cand. ing. Yuan Yuan

geb. am 13.01.1982

wohnhaft in:

Willi-Graf-Str. 7 App 151

80805 München

Tel.:

Lehrstuhl für
STEUERUNGS- und REGELUNGSTECHNIK
Technische Universität München

Univ.-Prof. Dr.-Ing./Univ. Tokio Martin Buss

Betreuer: Prof. O. Stursburg Betreuer: M. Althoff

Abgabe: 23.01.2007

Inhaltsverzeichnis

1	Einleitung	1
2	Hybride Systeme	2
2.1	Syntax Hybrider Systeme	2
2.2	Beispiele eines Hybriden Systems	3
2.3	Der Ablauf des Hybriden Systems	6
3	Informationsaustausch Format	7
3.1	Metropolis	7
3.2	Hybrid System Interchange Format	9
3.3	Der Syntaxvergleich zwischen Metropolis und HSIF	11
4	Zusammenfassung	12

1 Einleitung

Mit steigender Komplexität heutiger Regelsysteme nimmt die ganzheitliche Betrachtung von diskreten und kontinuierlichen Prozessen eine immer größere Bedeutung ein. Man denke z.B. an eine komplexe Produktionsanlage, in der unterlagerte Aufgaben kontinuierlich geregelt und von einem diskreten Regelgesetz überwacht und synchronisiert werden. Solche Systeme bezeichnet man als Hybride Systeme.

Heutzutage werden aus Gründen der Kosten, der Sicherheit und des Entwicklungszykluses viele Systeme simuliert bevor sie realisiert werden. Die Simulation hybrider Systeme wurde vor einigen Jahren begonnen. Typische Simulationtools sind z.B. Simulink, Modelica, CHECKMATE und so weiter. Ein Problem der unterschiedlichen Simulationswerkzeuge ist, dass Informationen nicht ausgetauscht werden können. Um dieses Problem zu lösen, werden Austauschformate entwickelt. Die hauptsächlichen Aufgaben der Austauschformate werden kurz zusammengefasst: Die Simulationsergebnisse eines Systems mit bestimmter Simulationssoftware, werden durch die Übersetzung des Austauschformats von anderen Simulationswerkzeuge erkannt und umgekehrt.

In dieser Arbeit, wird zuerst eine kurze Vorstellung des hybriden Systems und seines Ablaufes gegeben. Danach werden einige Simulationtools für hybride Systeme benannt. Im Hauptteil der Arbeit werden die existierenden Varianten von Metamodellen hybrider Systeme vorstellt und verglichen. Zum Schluss wird eine Zusammenfassung gegeben.

2 Hybride Systeme

2.1 Syntax Hybrider Systeme

Ein Hybrides System ist ein gemischtes System aus kontinuierlichen und diskreten Anteilen. Die Besonderheit des Systems ist, dass es kontinuierliche Evolutionen und zufälligen Sprünge hat. Die kontinuierliche Evolutionen werden durch Differenzialgleichungen beschrieben. Im Folgenden ist eine formale Definition von Lygeros gegeben. Ein Hybrides System ist ein Tupel:

$$\mathcal{H} = (Q, U_D, X, U, V, S_C, S, E, Inv, R, G)$$

- $Q = \{q_i, i \in J\}, J = \{1 \dots N\}$ ist eine Reihe von diskreten Zuständen;
- U_D ist eine Reihe von diskreten Eingänge ;
- X, U, V sind Teilreihen des begrenzten Vektorraums. Sie stellen die kontinuierlichen Zustände, Eingänge und Störgrößen in Raum dar;
- S_C ist eine Teilkategorie des kontinuierlichen dynamischen Systems.
 $S_i \in S_C$ wird definiert von folgender Gleichung
$$\dot{x} = f_i(x(t), u(t), \delta(t)) \quad i \in J$$
wobei $t \in R$, $x(t) \in X$ und f_i ist eine Funktion. Die Lösung des $x(t)$ besteht und ist einzigartig .
- $S : Q \rightarrow S_C (S : Q \rightarrow S_D)$ verbindet diskrete Zustände und kontinuierliche dynamische Systeme;
- $E \in Q \times U_D \times Q$ ist eine Ansammlung der diskreten Übergänge. ;

- $Inv : Q \rightarrow 2^{X \times U_D \times U \times V}$ ist einer Invariante;
 - $R : E \times X \times U \times V \rightarrow 2^X$ definiert den initialen Zustand des kontinuierlichen dynamischen Systems nach einem Übergang ;
 - $G : E \rightarrow 2^{X \times U \times V}$ sind Schaltbedingungen. Es gebe die Bedingungen, unter die gibt es eine Möglichkeit einen Übergang auszuführen;
- Die (Q, U_D, E) kann wie eine Finite State Machine(FSM) angesehen werden, mit Zustandsreihe Q , Eingang U_D und Übergang E . Der FMS kennzeichnet die Struktur des diskreten Übergang.

2.2 Beispiele eines Hybriden Systems

In diesem Teil, werden zwei Beispiele des Hybriden Systems gegeben.

- **Bouncing Ball**

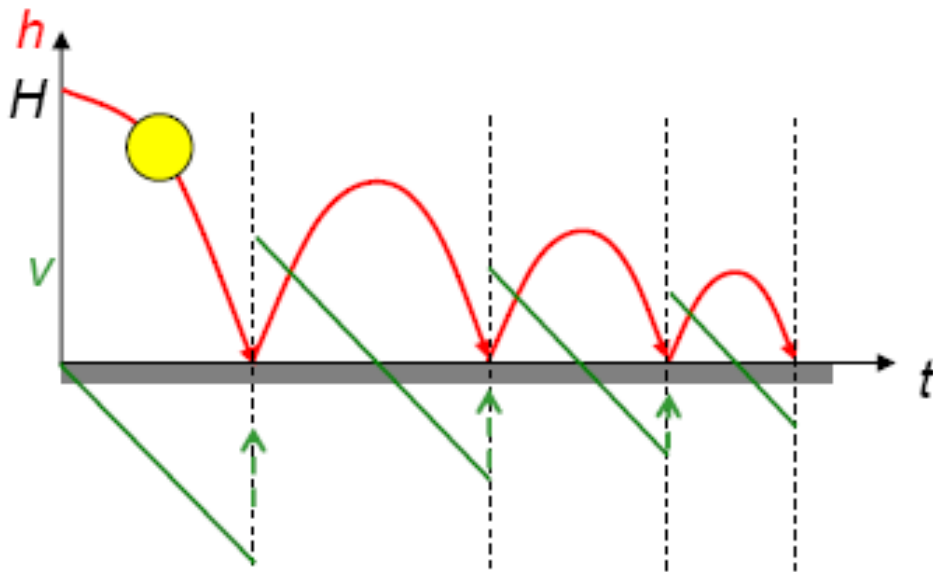


Abbildung 1: Ein Hybrides System Beispiel: Bouncing Ball, entnommen aus [1].

Der Ball fällt von der Höhe h_0 mit einer horizontalen anfänglichen Geschwindigkeit v_0 ab. Der Abfallvorgang von H_0 bis auf den Grund kann von Differenzialgleichungen beschrieben werden. Das ist ein kontinuierlicher Vorgang. Wenn der Ball auf den Grund trifft, werden kinetische Energie und die Gravitatkkräfte des Balls verändert nach das Prinzip Energieerhaltung. Das ist ein diskreter Übergang. Nach den einanderstoßen, befindet sich der Ball in einen anderen kontinuierlichen Zustand. Die Bewegungsrichtung und die Bewegungsgeschwindigkeit sind unterschiedlich vom vorherigen Zustand. Diese Phase kann auch von Differentialgleichung beschrieben werden. In der folgenden Zeit bewegt sich der Ball immer wie dieser Vorgang, bis die Bewegungsenergie null ist. Das ist ein typischer Verhaltensvorgang eines Hybriden Systems. In der folgenden Abb.2 ist ein hybrides Modell für den Bouncing Ball gegeben.

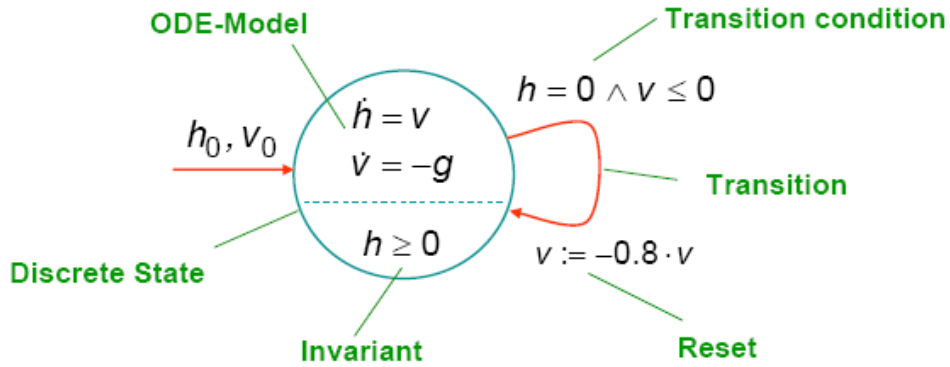


Abbildung 2: Hybrides Modell des Bouncing Ball, entnommen aus [1]

Der Kreis zeigt die Zustände des Balls. Der Obere Teil sind kontinuierliche Differentialgleichungen (*continuous State*), und der untere Teil sind Invarienzgebiete (*diskrete State*). Der Ball bewegt sich mit einer anfänglichen Höhe h_0 und einer anfänglichen Geschwindigkeit v_0 . Der kontinuierliche Zustand wird von Differentialgleichungen (*ODE-Model*) $\dot{h} = v, \dot{v} = g$ beschrieben. Mit diesen Gleichungen werden h und v gelöst. Wenn h und v die Übergangsbedingungen (*Transition condition*) erfüllen, wechselt der Ball den diskreten Zustand. In dieser Phase springt v den Wert von Geschwindigkeitseinstellung (*Reset*). Danach geht der Ball zurück in den kontinuierlichen Zustand. In den ganzen Prozess gibt es eine un-

veränderte Bedingung *invariant* $h \geq 0$.

- **Stromkreis**

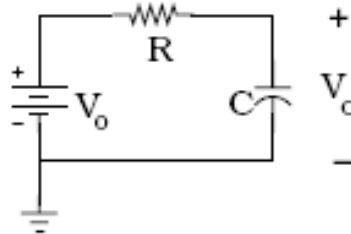


Abbildung 3: Ein Hybrides Modell eines Stromkreises, entnommen aus [2]

Abgebildet ist ein LC Stromkreis. Der physikalische Prozess des Stromkreises kann wie folgt beschrieben werden: Zuerst wird der Kondensator geladen, bis der Kondensator gesättigt oder ($v_0 = -5v$) ist. Danach fängt das Entladen an. Der Prozess läuft periodisch ab, falls es keine Verluste gibt.

Abbildung 4 zeigt das Modell der Finite State Machine mit Transitionen und

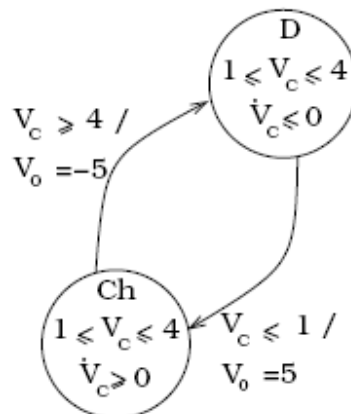


Abbildung 4: Hybrider Automat des Stromkreises, entnommen aus [2]

Invarianten. Der anfängliche Zustand ist $V_0 = 5, V_c = 1$. Der Kondensator wird geladen, wenn $\dot{V}_c \geq 0$ ist. Der Stromkreis verlässt den Zustand D, wenn die Übergangsbedingungen $V_c \geq 4 / V_0 = -5$ erfüllt sind. Der Zustand Ch wird

verlassen sobald die Übergangsbedingung $V_c \leq 1/V_0 = 5$ erfüllt sind. Danach wird der Kondensator wieder geladen. Dieser Vorgang wiederholt sich periodisch.

2.3 Der Ablauf des Hybriden Systems

Anhand der zwei Beispiele, kann man den Ablauf des Hybriden Systems erkennen. Das System verhält sich zunächst kontinuierlich wie es mit Hilfe von Gleichungen beschrieben wird. Erreicht das System Schaltbedingungen, kann es in einen anderen diskreten Zustand wechseln. In diese Phase, wird der initiale Zustand des nächsten kontinuierlichen Ablaufsvorgang eingestellt. Danach verläuft das System in einem anderen Zustand kontinuierlich. In der Zukunft wiederholt das System diesen Vorgang.

3 Informationsaustausch Format

Das Ziel des Interchangeformates ist, den Informationsaustausch zwischen unterschiedlichen Tools zu ermöglichen. In [2] wird der Informationsaustausch unterschiedlicher Tools motiviert. Die hauptsächliche Herangehensweise für Austauschformate ist das Interchangeformat, das eine von allen Tools erkennbare Daten hat. Dieses Interchangeformat ist keine Datenbank oder Datenstruktur, sondern ein Objekt, das von allen bestehenden Tools erkannt werden kann. In dem folgenden Teil werden zwei Varianten des Interchangeformat vorgestellt, und verglichen.

3.1 Metropolis

Metropolis wird von GSRC(Gigascale System Research Center) und CHESS(Center for Hybrid and Embedded Software Systems) unterstützt. Das Ziel des Projekts ist, dass das Design von Hardware und Software, von der Spezifikation bis zur Implementierung, zu unterstützen. Der Schwerpunkt in Metropolis ist die Entwicklung eines Metropolis Meta-Modells. Durch Refinement des Meta-Modells, werden andere Computationsmodelle, wie FSMs und Data Flow graphs, erstellt. In einem Meta-Modell, werden spezielle Objekte Process und Medium definiert, die die Berechnungen und Kommunikationen beschreiben. In Processes werden potentielle Aktionfolgen des Processes spezifiziert. Im Gegensatz dazu werden spezielle Kommunikationsprotokolle im Medium durchgeführt. Im Meta-Modell werden Funktion und Struktur getrennt definiert, um die Verwendung des Refinement automatisch oder manuell zu erstellen. Durch diese getrennten Definitionen, von Ansicht im [3], werden die Meta-Modell besser wieder benutzt.

Syntax in Metropolis

- Prozesse und Medium

In einem Meta-Modell, werden spezielle Objekte Prozess und Medium definiert. Kommunikation wird von den Objekten Medien(In Abbildung 5 sind M0, M1 Medien.) modelliert. Die Aufgaben des Prozesses ist Berechnung. Ein Medium wird

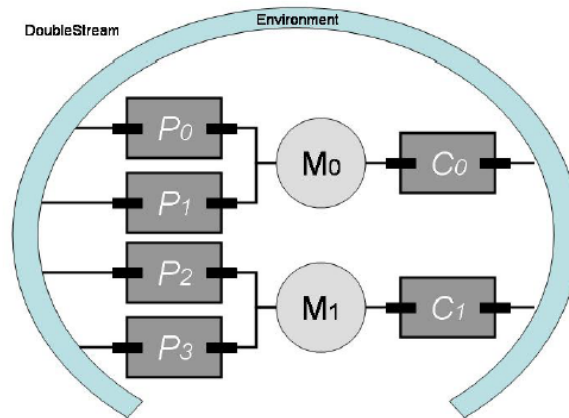


Abbildung 5: Eine Netliste von Prozessen und Medium, entnommen aus [4]

von einer Reihe von Variablen und Funktionen definiert. Die Werte der Variablen bilden die Zustände des Mediums. Die Variablen können nur von den Funktionen in das Medium zugegangen sein, und die Kommunikation zwischen diesen Prozessen wird durch den Abruf dieser Funktionen hergestellt. Das Metropolis Metamodell repräsentiert ein System als eine Reihe von Prozessen (In Abbildung 5 sind P_0, P_1, P_2, P_3 Prozesse).

- Die Schnittstelle

Die Funktionen eines Mediums setzen sich aus einer Reihe von Schnittstellen zusammen. Wenn man mit objektorientierter Sprache programmiert, erklärt eine Schnittstelle nur Funktionen. Ein Medium wird diese Funktionen durchführen. Nur die Funktionen der Schnittstelle können für Kommunikation abgerufen werden. Jedes Medium kann einen Abschluss mit bestimmten Ein- und Ausgang eines Prozesses verbinden, solange es die entsprechende Schnittstelle realisiert.

- Eigenschaften

Eine formale Spezifikation der Eigenschaften muss für das Medium gespeichert werden, während der Durchführung des Systems. Manche Eigenschaften sind statisch, z.B. die maximale Menge der Objekte, an der das Medium angeschlossen ist. Im Gegensatz sind manche dynamisch.

- Begrenzungen

In Metropolis gibt es zwei Typen Begrenzungen. Die zeitliche Begrenzung hat die Form der LTL Logik. Die quantitative Begrenzung hat die Logik der Begrenzung und die ausgedehnt Logik der Begrenzung. Begrenzungen können für die Unterschied des gesetzlichen Verhalten benutzt werden.

- Der Zeitplan und Quantity Managers

Eine Politik sollen gegeben sein, um das gleichzeitigen Vorkommen der Prozessen zu koordinieren. Das bedeutet einen Zeitplan für Prozesse zu geben. Das Metamodell führt eine spezielle Politik dafür ein. Jetzt eine Zeitplanpolitik wird von ein aufeinanderfolgendes Programm spezialisiert.

Der Quantity Manager sind verantwortlich für Kommentieren, Nachfolgen, und Synchronisieren.

3.2 Hybrid System Interchange Format

Das Hybrid System Interchange Format(HSIF) wird von der Vanderbilt Universität und der Pennsylvania Universität entwickelt. In HSIF ist ein Hybrides System eine Ansammlung von Hybriden Automaten. Jeder Hybride Automat ist eine Finite State Machine, in der kontinuierliches Verhalten abgebildet wird. Automaten tauschen Informationen durch Variablen aus. Es gibt zwei Typen von Variablen: *Signals* und *Shared Variables*. Signals werden in der Modellierung der Kommunikationen zwischen Automaten eingesetzt. Shared Variables werden für die asynchrone Kommunikation von Automaten eingesetzt.

In HSIF sind Variablen wichtige Konzepte. Im Folgenden befindet eine graphische Erklärung dafür. Mit Hilfen von Abbildung 6 wird die Syntax in HSIF erläutert.

Syntax in HSIF

- Hybride Automaten

Ein Hybrider Automat H in [4] ist ein Tupel $\langle S, s_0, V, P, T \rangle$, wobei S eine Reihe von diskreten Zuständen ist, $s_0 \in S$ der initiale Zustand ist, V eine Reihe

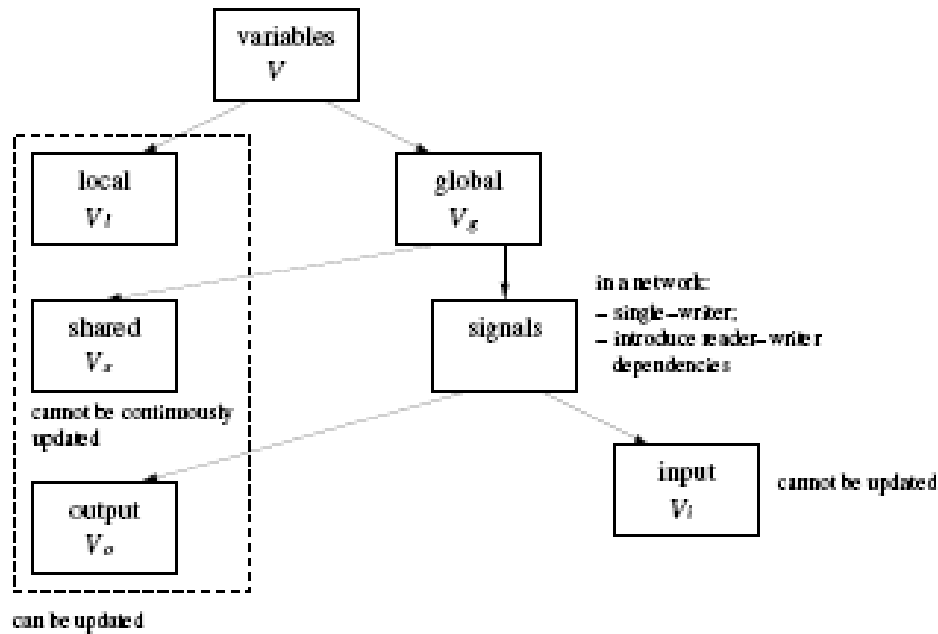


Abbildung 6: Die Variablenstruktur in HSIF, entnommen aus [4]

von Variablen ist, P eine Reihe von Parameter ist, T eine Reihe von Übergang ist.

- Variablen und Parameter

Nach dem effektiven Bereich, gliedert die Variablen sich in lokale Variablen und globale Variablen. Globale Variablen sind teilweise Signals und teilweise Shared Variablen. Signals bestehen aus Eingang- und Ausgangssignal. Es ist möglich, dass ein Eingangssignal eines Automaten ein Ausgangssignal eines anderen Automaten ist.

Parameter in ein Hybrides System sind auch teilweise lokale Parameter und teilweise globale Parameter. Sobald das System hergestellt wird, werden die Werte der Parameter festgelegt. Diese Werte werden während die Durchführung des Systems ungeändert.

- Funktionen

Das Verhalten des Automaten wird von Funktion $f(x_1, \dots, x_n)$ beschrieben, wobei x_i eine Variable oder ein Parameter des Automaten ist. Eine Funktion kann durch mathematische Gleichungen spezialisiert werden, oder im eine Tabelle gegeben

haben.

- Der Übergang

Ein Übergang eines Hybriden Systems ist ein Tupel $\langle s, g, \alpha, s' \rangle$, wobei s und s' die Quellen- und Zielzustand des Übergang sind. Bevor der Übergang sich ereignet, muss die logische Berechnung des Schutz g recht ist. Die Aktion α wird durchgeführt während den Übergang. Die Aktionen sind meistens Einstellungen der Automatvariablen. Eine Einstellung ist $x = f(x_1, \dots, x_n)$, wobei x nicht Eingangsvariable ist.

3.3 Der Syntaxvergleich zwischen Metropolis und HSIF

Die fundamentale Elemente in Metropolis sind Prozess, Medium und Quantity Manager. Der Prozess wird definiert für Berechnungen, Das Medium wird definiert für Kommunikation zwischen Prozesse. Der Quantity Manager ist für den Zeitplan des Prozessen verantwortlich. Diese drei Teile bestimmen das Verhalten der Zustände. Ein Prozess ist ein geschlossener Programmblock. In den Block werden Algorithmen definiert. Ein Prozess kann nicht direkt mit anderen Prozessen Austausch. Es kann nur durch einen Anschluss seine Variablen oder Parameter bekommen oder ausgeben. Ein Anschluss hat die Eigenschaft der Schnittstelle. Eine Schnittstelle erklärt eine Reihe von Funktionen, um die Eingang und Ausgang unterschiedartiger Variablen zu realisieren.

In HSIF sind diskrete Zustände, Übergang, Variablen und Parameter wichtige Konzepte. Ein Zustand wird von Differenz- und Logikgleichungen, und ein Invariant definiert. Er ist ein kontinuierlicher Vorgang. Der Übergang beschreibt den Vorgang, dass der Hybride Automat einen Zustand verlässt und zu einem anderen Zustand läuft. Variablen können von Differenzgleichungen erneut werden.

4 Zusammenfassung

Die hauptsächtliche Varianten des Interchangeformats haben schön vorgestellt worden. Aber diese Varianten sind nicht reife Werkzeuge. Das bedeutet, dass es noch Fehler in diese Varianten gibt, z.B. HSIF unterschützt manche Eigenschaften des Simulink/Stateflow nicht. Das bedeutet, wenn man dieses Interchangeformat einsetzt, um die Informationen in Simulink/Stateflow zu erhalten oder umgekehrt, gehen einige Informationen verloren. Und auch der Metropolis sind nicht ausreichend, um die Informationen von allen bestehenden Tools zu beschreiben. Es braucht noch mehr Bemühungen, um eines reife und gut funktioniert Interchangeformat zu bekommen. Ein gut Interchangeformat von Ansicht [4], sollen folgende Eigenschaften haben:

- Es unterschützt alle bestehenden Simulationstools, Modellierungsweisen, und Entwurfsprachen;
- Es ist frei zugänglich;
- Es unterschützt unterschiedliche Input- und Outputmechaniken;
- Es unterschützt Hierarchie- und Objektorientierungsdesign.

Literatur

- [1] O. Stursberg, “Ereignisdiskrete und hybride systeme,” tech. rep., Lehrstuhl fuer Steuerungs- und Regelungstechnik, Technische Universitaet Muenchen, 2006.
- [2] L. R. A.Pinto, A.Sangiovanni-Vincentelli, “Interchange format for hybrid systems-review and proposal,” *Hybrid Systems:Computation and Control*, vol. LNCS 3414, pp. 528–541, 2005.
- [3] A.Dubey, “@bookfoll:98, publisher = Oldenbourg, year = 1998, author = O. Föllinger, address = München-Wien, edition = 8 ,” Master’s thesis, Vanderbilt university, 2005.
- [4] C. IST-Project, “Modeling techniques,programming languages,design toolsets and interchange formats for hybrid systems,” tech. rep., 2004.