STATE EVENTS IN CONTINUOUS SYSTEMS - CLASSIFICATION AND MODELICA IMPLEMENTATION

Felix Breitenecker,
Inst. f. Analysis and Scientific Computation,
Mathematical Moddeling and Simulation, TU Vienna

Günther Zauner, Nikolas Popper;
DWH – Drahtwarenhandlung Simulation Services

die Drahtwarenhandlung

ASC - MMS
TU VIENNA

CSSL Standard 1968

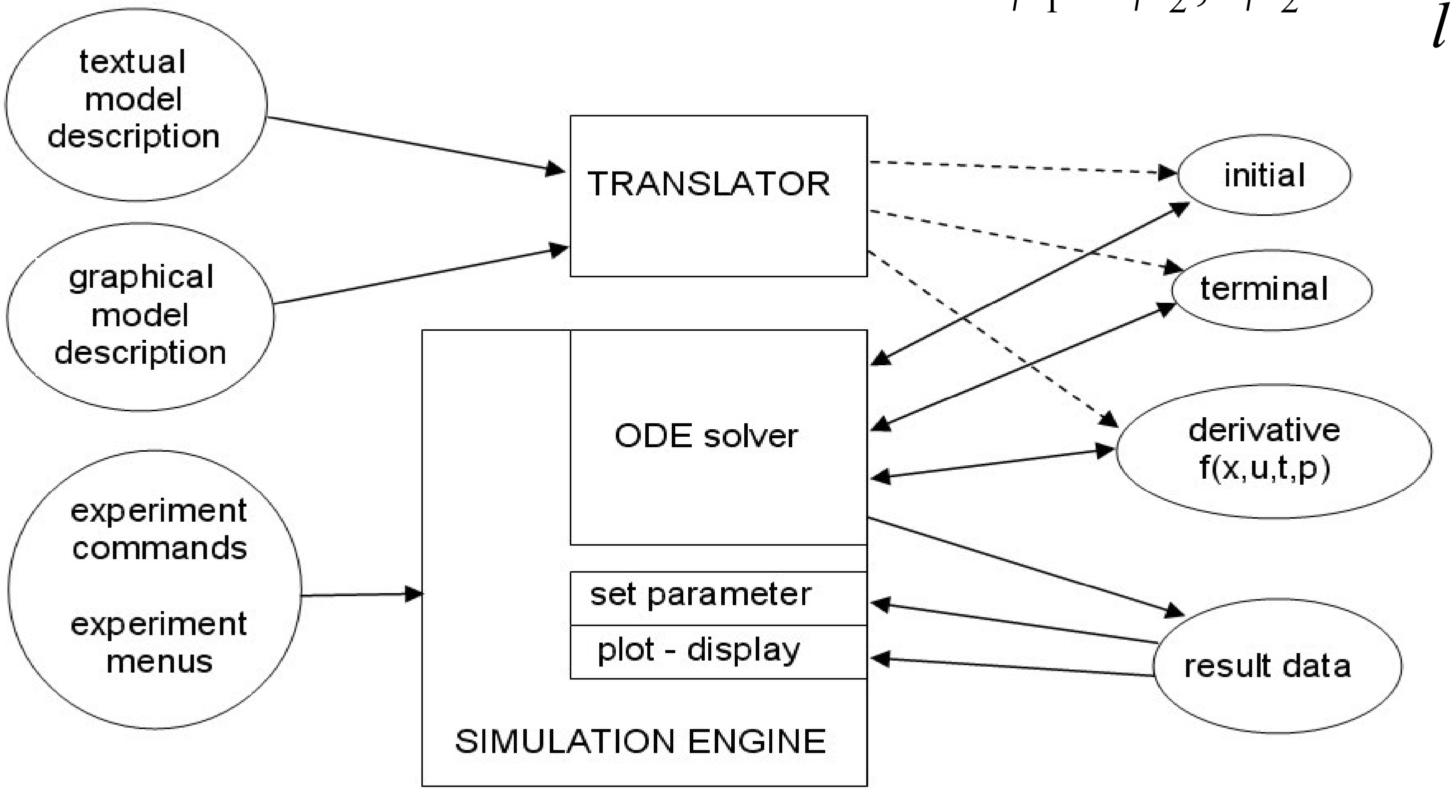
$$\dot{\vec{x}}(t) = \vec{f}(\vec{x}(t), \vec{u}(t), \vec{p}, t)$$

$$\dot{\varphi}_1 = \varphi_2, \ \dot{\varphi}_2 = -\frac{g}{l}\sin\varphi_1 - \frac{d}{m}\varphi_2$$

CSSL Standard 1968

$$\dot{\vec{x}}(t) = \vec{f}(\vec{x}(t), \vec{u}(t), \vec{p}, t)$$

$$\dot{\varphi}_1 = \varphi_2, \quad \dot{\varphi}_2 = -\frac{g}{l}\sin\varphi_1 - \frac{d}{m}\varphi_2$$



die Drahtwarenhandlung

ASC - MMS
TU VIENNA

State Events

$$\dot{\vec{x}}(t) = \vec{f}(\vec{x}(t), \vec{u}(t), \vec{p}, t), \quad h(\vec{x}(t), \vec{u}(t), \vec{p}, t) = 0, \quad E(h(0))$$

$$\dot{\varphi}_1 = \varphi_2, \ \dot{\varphi}_2 = -\frac{g}{l}\sin\varphi_1 - \frac{d}{m}\varphi_2, \quad h(\varphi_1, \varphi_2) = \varphi_1 - \varphi_p = 0$$

$$l_l \to l_s, \quad l_s \to l_l$$

$$\dot{\varphi}_1 \to \dot{\varphi}_1 \frac{l_l}{l_s}, \dot{\varphi}_2 \to \dot{\varphi}_2 \frac{l_s}{l_l}$$

die Drahtwarenhandlung

ASC - MMS
TU VIENNA

State Events

$$\dot{\vec{x}}(t) = \vec{f}(\vec{x}(t), \vec{u}(t), \vec{p}, t), \quad h(\vec{x}(t), \vec{u}(t), \vec{p}, t) = 0, \quad E(h(0))$$

$$\dot{\varphi}_1 = \varphi_2, \; \dot{\varphi}_2 = -\frac{g}{l}\sin\varphi_1 - \frac{d}{m}\varphi_2, \quad h(\varphi_1, \varphi_2) = \varphi_1 - \varphi_p = 0$$

- type 1: parameter change - **SE-P**
- type 2: one or more inputs change discontinuously - **SE-I**
- type 3: one or more states change discontinuously - **SE-S**
- type 4: the dimension of the state vector changes discontinuously - **SE-D**

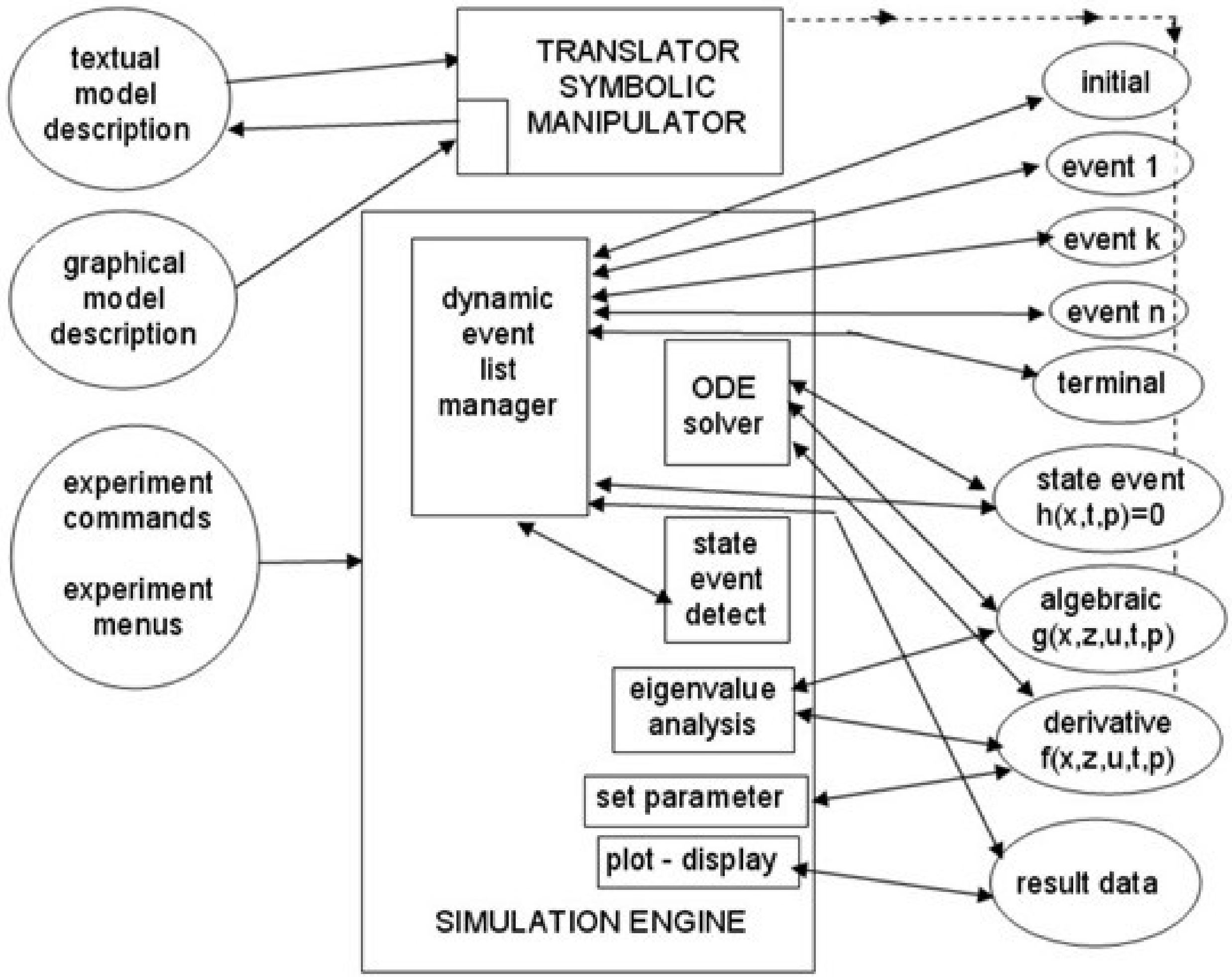
$$l_l \to l_s, \quad l_s \to l_l$$

$$\dot{\varphi}_1 \to \dot{\varphi}_1 \frac{l_l}{l_s}, \dot{\varphi}_2 \to \dot{\varphi}_2 \frac{l_s}{l_l}$$

State Events

$$\dot{\vec{x}}(t) = \vec{f}(\vec{x}(t),\vec{u}(t),\vec{p},t),$$
$$h(\vec{x}(t),\vec{u}(t),\vec{p},t) = 0,$$
$$E(h(0))$$

$$\dot{\varphi}_1 = \varphi_2,$$
$$\dot{\varphi}_2 = -\frac{g}{l}\sin\varphi_1 - \frac{d}{m}\varphi_2,$$
$$h(\varphi_1,\varphi_2) = \varphi_1 - \varphi_p = 0$$
$$l_l \to l_s, \quad l_s \to l_l$$
$$\dot{\varphi}_1 \to \dot{\varphi}_1\frac{l_l}{l_s}, \dot{\varphi}_2 \to \dot{\varphi}_2\frac{l_s}{l_l}$$



die Drahtwarenhandlung

ASC - MMS
TU VIENNA

State Events

- **State Events Type 1 – Change of Parameters: SE-P**
  formulated by IF-THEN-ELSE constructs or switches
  necessity of a state event depends on the accuracy wanted

- **State Events Type 2 – Change of Input: SE-I**
  no state events - only time events
  – listed here due to historic reasons.

- **State Events Type 3 – Change of State Variable: SE-S**
  essential state event
  condition described by IF-THEN or WHEN, or switch
  event action described in ALGEBRAIC block

- **State Events of Type 4 – Change of Dimension: SE-D**
  essential state event (e.g. change of degree of freedom)
  condition described by IF-THEN or WHEN, or switch
  event action described in ALGEBRAIC and/or new model sections

$$\dot{\varphi}_1 = \varphi_2,$$

$$\dot{\varphi}_2 = -\frac{g}{l}\sin\varphi_1 - \frac{d}{m}\varphi_2,$$

$$h(\varphi_1, \varphi_2) = \varphi_1 - \varphi_p = 0$$

$$l_l \to l_s, \quad l_s \to l_l$$

$$\dot{\varphi}_1 \to \dot{\varphi}_1 \frac{l_l}{l_s}, \dot{\varphi}_2 \to \dot{\varphi}_2 \frac{l_s}{l_l}$$

die Drahtwarenhandlung

ASC - MMS
TU VIENNA

Handling State Events

In principle, the service (handling) of a state event requires four steps:

- Detection of the event: usually by checking the change of the signum-function of *h*(*x*).
- Localisation of the event: algorithms make use of either iterative techniques, or of interpolation techniques for determining the time instant of the event with sufficient accuracy.
- Service of the event: calculating / setting new parameters, inputs and states; switching to new equations
- Restart of the ODE solver in a **‘maximal’ state vector**, or starting **another model** (hybrid decomposition)

$$\dot{\varphi}_1 = \varphi_2,$$

$$\dot{\varphi}_2 = -\frac{g}{l}\sin\varphi_1 - \frac{d}{m}\varphi_2,$$

$$h(\varphi_1,\varphi_2) = \varphi_1 - \varphi_p = 0$$

$$l_l \to l_s, \quad l_s \to l_l$$

$$\dot{\varphi}_1 \to \dot{\varphi}_1 \frac{l_l}{l_s}, \dot{\varphi}_2 \to \dot{\varphi}_2 \frac{l_s}{l_l}$$

die Drahtwarenhandlung

ASC - MMS
TU VIENNA

State Events DAEs

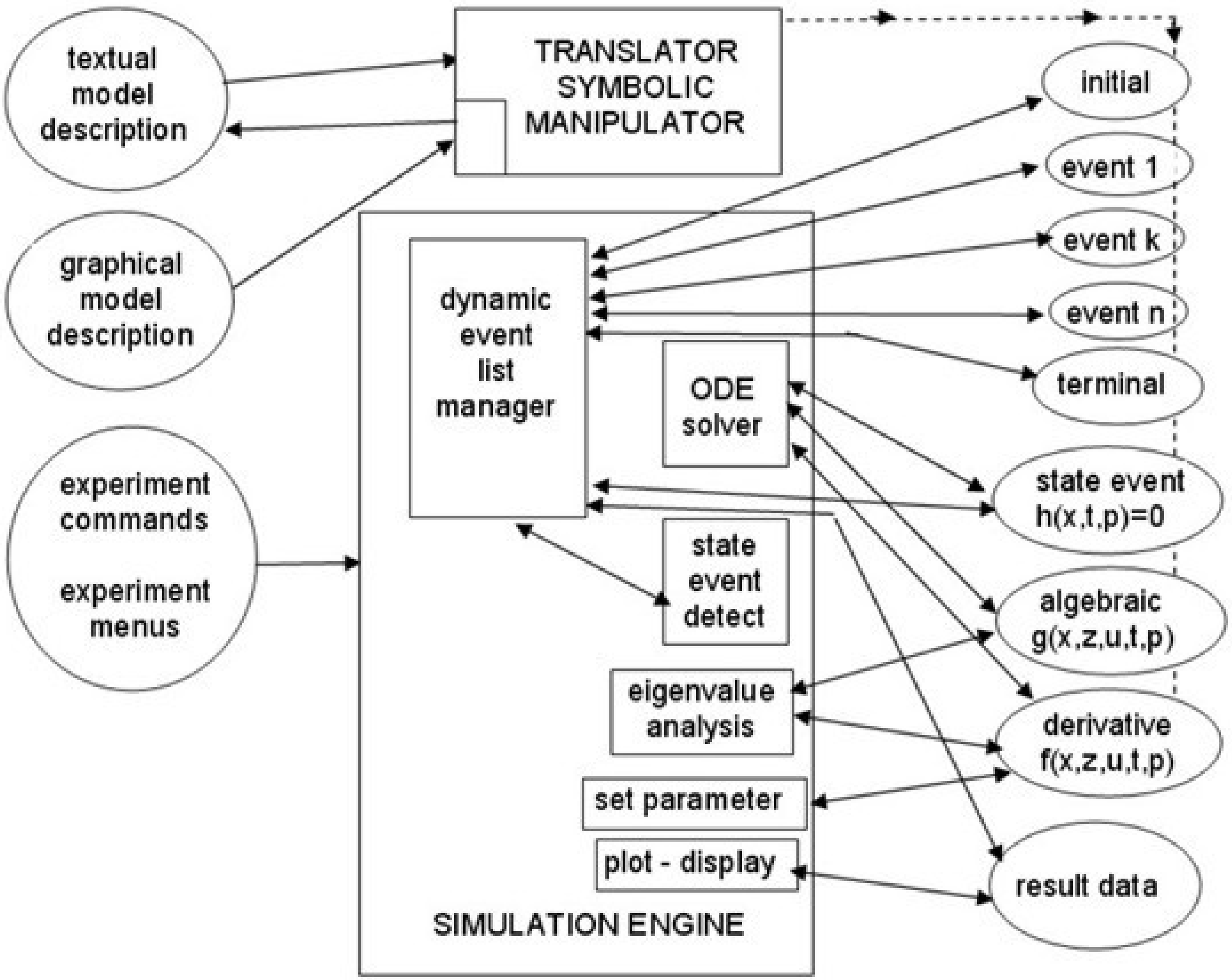
Algebraic equations –
Implicit state events

$$\dot{\vec{x}}(t) = \vec{f}(\vec{x}(t), \vec{u}(t), \vec{p}, t),$$

$$g(\vec{x}(t), \vec{u}(t), \dot{\vec{x}}(t), \vec{p}, t)$$

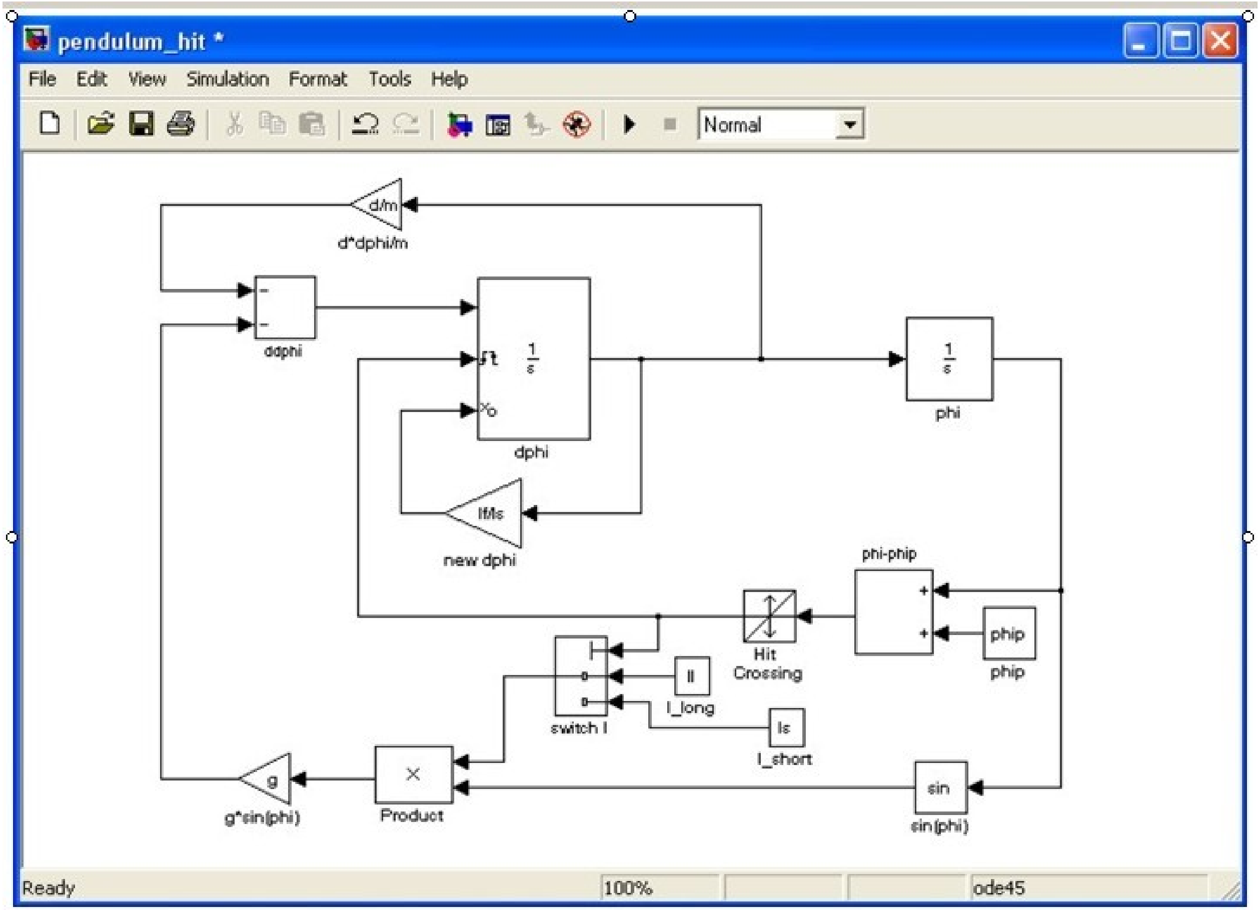
$$h(\vec{x}(t), \vec{u}(t), \vec{p}, t) = 0,$$

$$E(h(0))$$



Classic Implementations of State Events

SIMULINK



$$\dot{\varphi}_1 = \varphi_2,$$

$$\dot{\varphi}_2 = -\frac{g}{l}\sin\varphi_1 - \frac{d}{m}\varphi_2,$$

$$h(\varphi_1,\varphi_2) = \varphi_1 - \varphi_p = 0$$

$$l_l \to l_s, \quad l_s \to l_l$$

$$\dot{\varphi}_1 \to \dot{\varphi}_1 \frac{l_l}{l_s}, \dot{\varphi}_2 \to \dot{\varphi}_2 \frac{l_s}{l_l}$$

Classic Implementations of State Events

ACSL

- PROGRAM constrained pendulum
- CONSTANT m = 1.02, g = 9.81, d =0.2
- CONSTANT lf=1, lp=0.7
- DERIVATIVE dynamics
-   ddphi = -g*sin(phi)/l – d*dphi/m
-   dphi = integ (ddphi, dphi0)
-   phi   = integ (dphi, phi0)
-   SCHEDULE hit   .XN. (phi-phip)
-   SCHEDULE leave .XP. (phi-phip)
- END ! of dynamics
- DISCRETE hit
-   l = ls; dphi = dphi*lf/ls
- END ! of hit
- DISCRETE leave
-   l = lf; dphi = dphi*ls/lf
- END ! of leave
- END ! of constrained pendulum

$$\dot{\varphi}_1 = \varphi_2,$$

$$\dot{\varphi}_2 = -\frac{g}{l}\sin\varphi_1 - \frac{d}{m}\varphi_2,$$

$$h(\varphi_1,\varphi_2) = \varphi_1 - \varphi_p = 0$$

$$l_l \to l_s, \quad l_s \to l_l$$

$$\dot{\varphi}_1 \to \dot{\varphi}_1 \frac{l_l}{l_s}, \dot{\varphi}_2 \to \dot{\varphi}_2 \frac{l_s}{l_l}$$

die Drahtwarenhandlung

ASC - MMS
TU VIENNA

‘Modern’ Implementations of State Events

MODELICA

$$\dot{\varphi}_1 = \varphi_2,$$

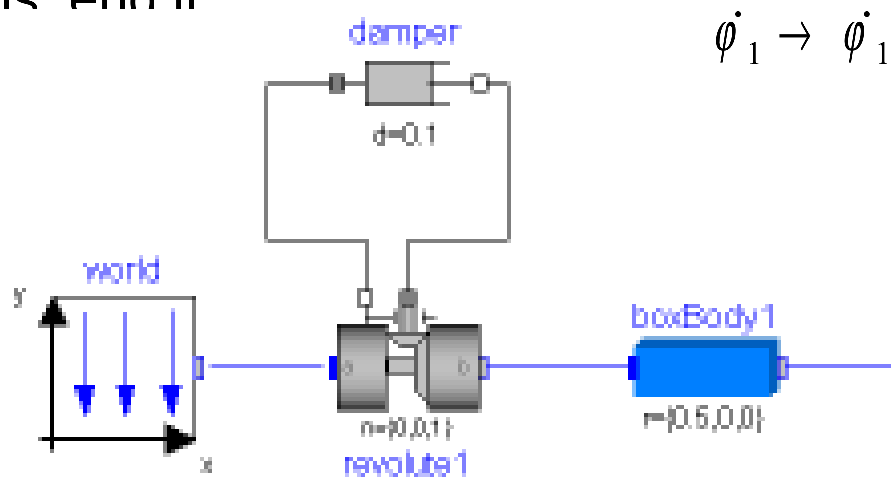
$$\dot{\varphi}_2 = -\frac{g}{l}\sin\varphi_1 - \frac{d}{m}\varphi_2,$$

$$h(\varphi_1,\varphi_2) = \varphi_1 - \varphi_p = 0$$

$$l_l \to l_s, \quad l_s \to l_l$$

$$\dot{\varphi}_1 \to \dot{\varphi}_1 \frac{l_l}{l_s}, \dot{\varphi}_2 \to \dot{\varphi}_2 \frac{l_s}{l_l}$$

```
equation /*pendulum*/
v = length*der(phi);   vdot = der(v);
mass*vdot/length + mass*g*sin(phi) + damping*v = 0;
algorithm
if (phi<=phipin) then length:=ls; end if;
if (phi>phipin) then length:=l1
```



‘Modern’ Implementations of State Events

ANYLOGIC

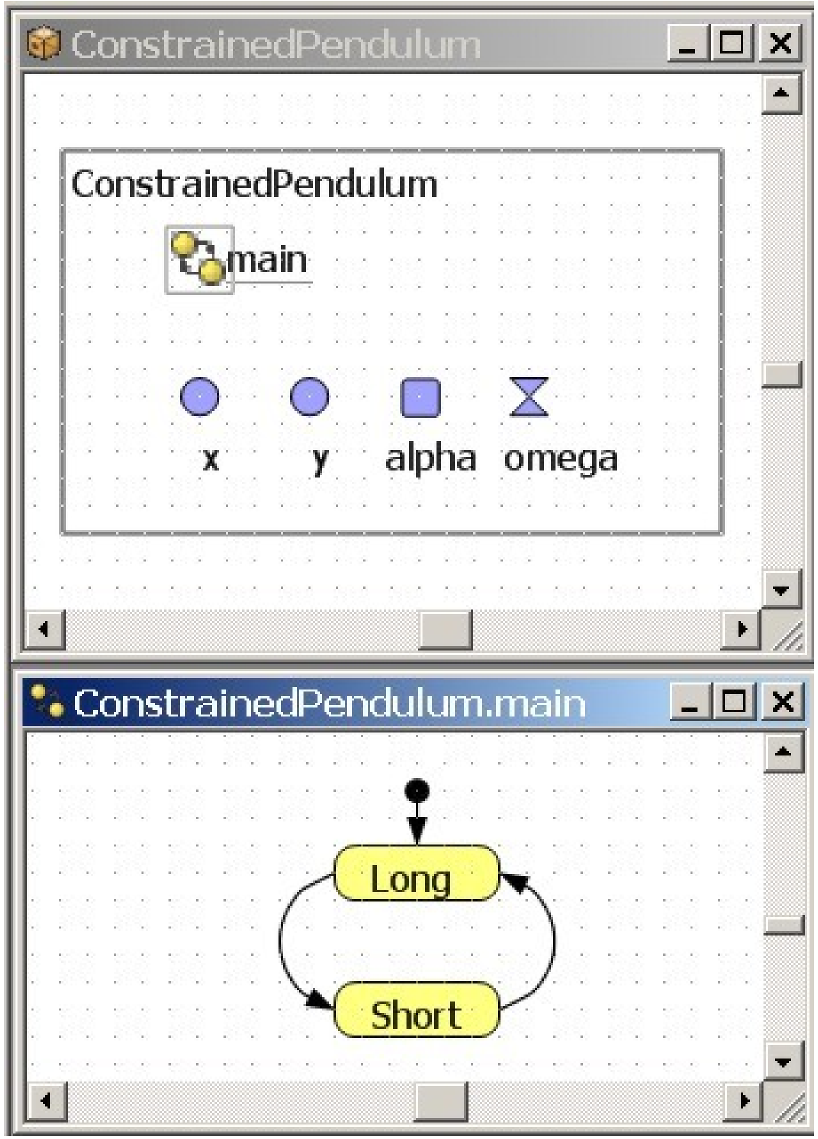
Equations
d(alpha)/dt = omega
x = l*sin(alpha), y = l*cos(alpha)

Equations
d(omega)/dt = (-g*sin(alpha)-
mu*omega)/ls

Change eventLong
(alpha>=alphaN)||(alpha<=alpha
Action
omega=omega*ls/ll
stop

Change EventShort
(alpha>=alphaN)||(alpha<=alpha)
Action
omega=omega*ll/ls
Stop

Equations
d(omega)/dt =
(-g*sin(alpha)- mu*omega)/ll



$$\dot{\varphi}_1 = \varphi_2,$$

$$\dot{\varphi}_2 = -\frac{g}{l}\sin\varphi_1 - \frac{d}{m}\varphi_2,$$

$$h(\varphi_1,\varphi_2) = \varphi_1 - \varphi_p = 0$$

$$l_l \to l_s, \quad l_s \to l_l$$

$$\dot{\varphi}_1 \to \dot{\varphi}_1 \frac{l_l}{l_s}, \dot{\varphi}_2 \to \dot{\varphi}_2 \frac{l_s}{l_l}$$

die Drahtwarenhandlung

ASC - MMS
TU VIENNA

‘Modern’ Implementations of State Events

ANYLOGIC

Equations

d(alpha)/dt = omega

d(omega)/dt =
(-g*sin(alpha)-
mu*omega)/l

x = l*sin(alpha)

y = l*cos(alpha)

Change eventLong

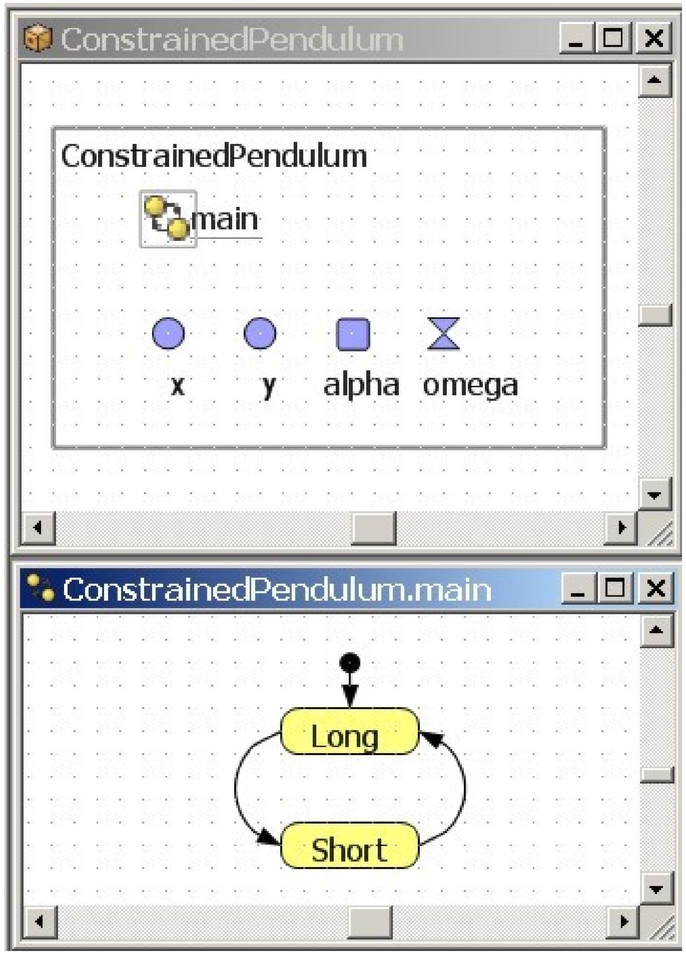
(alpha>=alphaN)||

(alpha<=alphaN)

Action

l=ll

omega=omega*ls/ll



$$\dot{\varphi}_1 = \varphi_2,$$

$$\dot{\varphi}_2 = -\frac{g}{l}\sin\varphi_1 - \frac{d}{m}\varphi_2,$$

$$h(\varphi_1, \varphi_2) = \varphi_1 - \varphi_p = 0$$

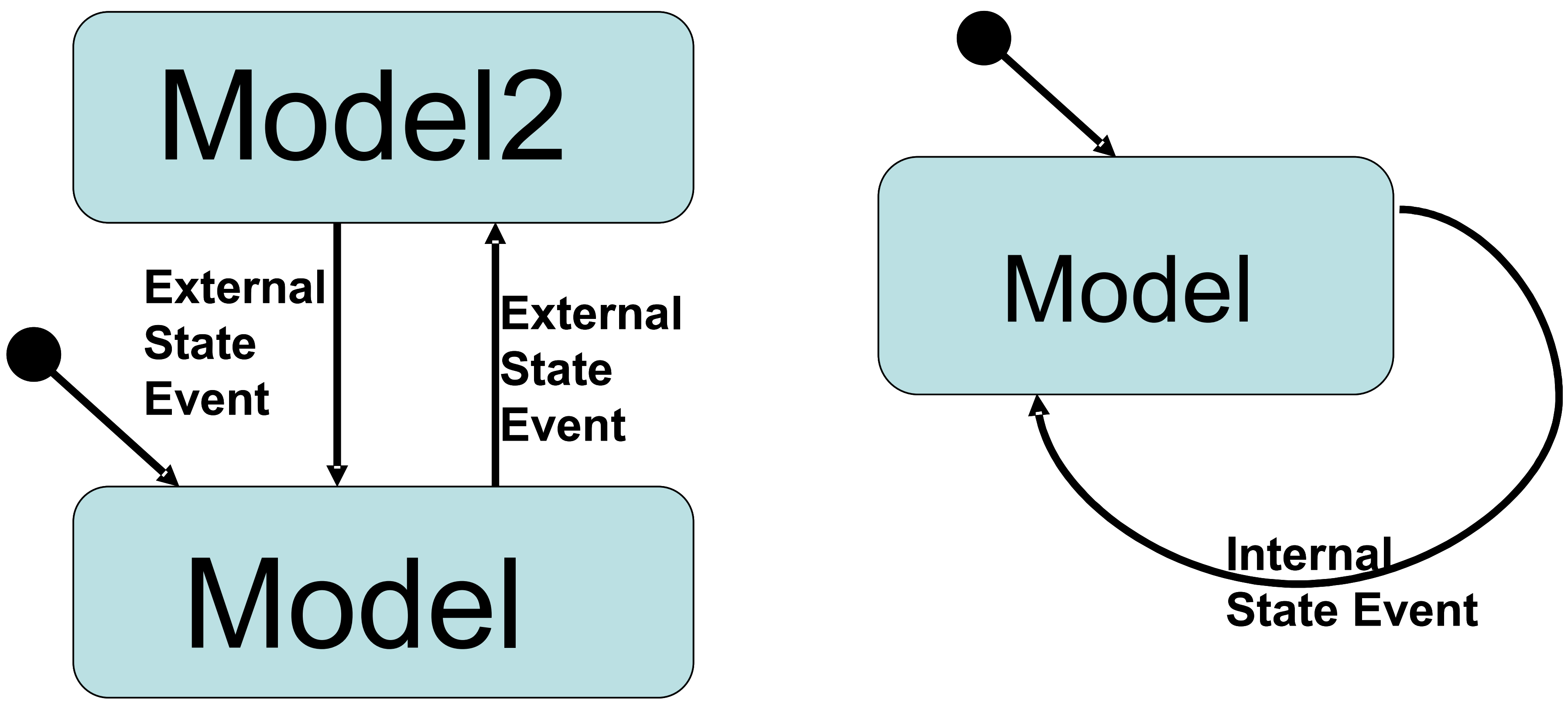
$$l_l \to l_s, \quad l_s \to l_l$$

$$\dot{\varphi}_1 \to \dot{\varphi}_1 \frac{l_l}{l_s}, \dot{\varphi}_2 \to \dot{\varphi}_2 \frac{l_s}{l_l}$$

die Drahtwarenhandlung

ASC - MMS
TU VIENNA

Hybrid and Structural Dynamic Systems

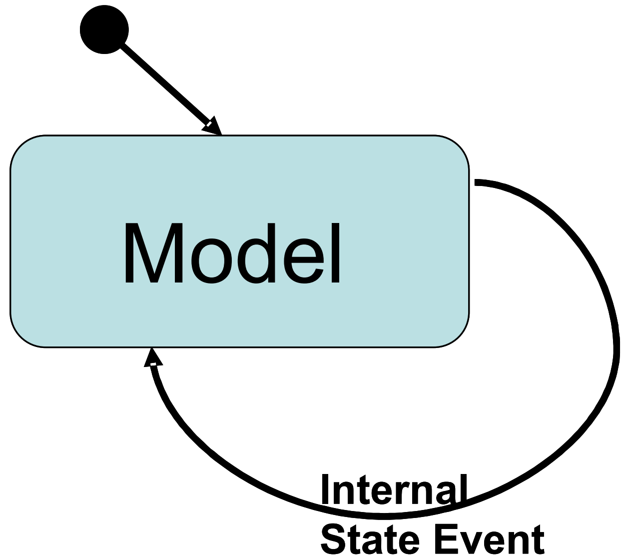


die Drahtwarenhandlung

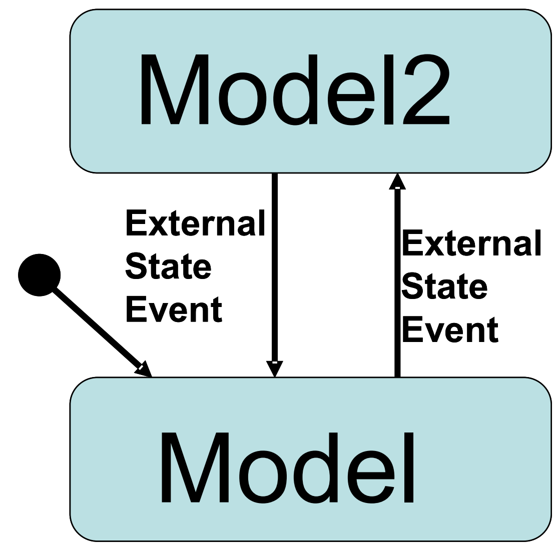
ASC - MMS
TU VIENNA

Hybrid and Structural Dynamic Systems

Maximal State Space for structural-dynamic systems – internal events

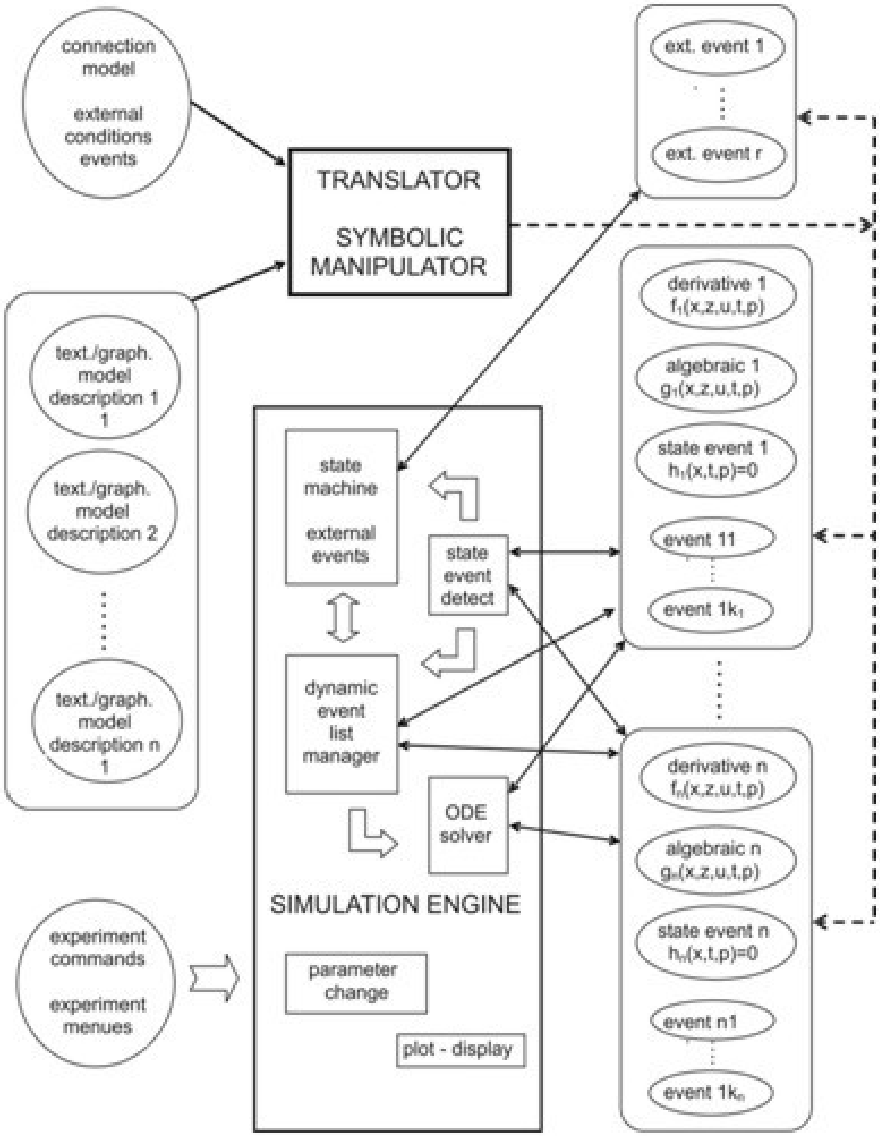


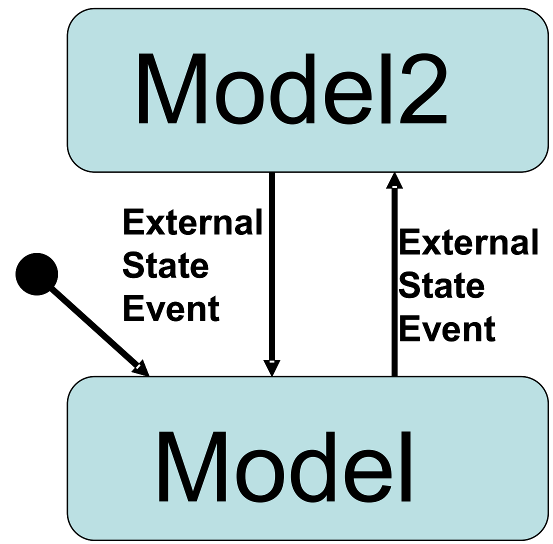
Hybrid Decomposition for structural-dynamic systems – external events



die Drahtwarenhandlung
ASC - MMS
TU VIENNA

Hybrid and Structural Dynamic Systems



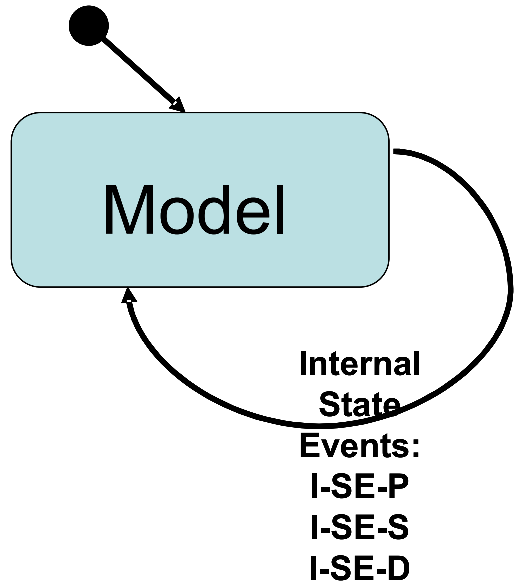


die Drahtwarenhandlung

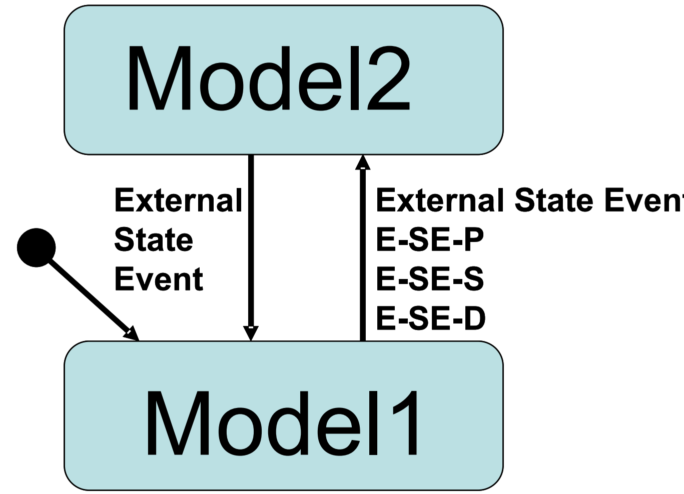
ASC - MMS
TU VIENNA

Hybrid and Structural Dynamic Systems

Maximal State Space for structural-dynamic systems – internal events I-SE

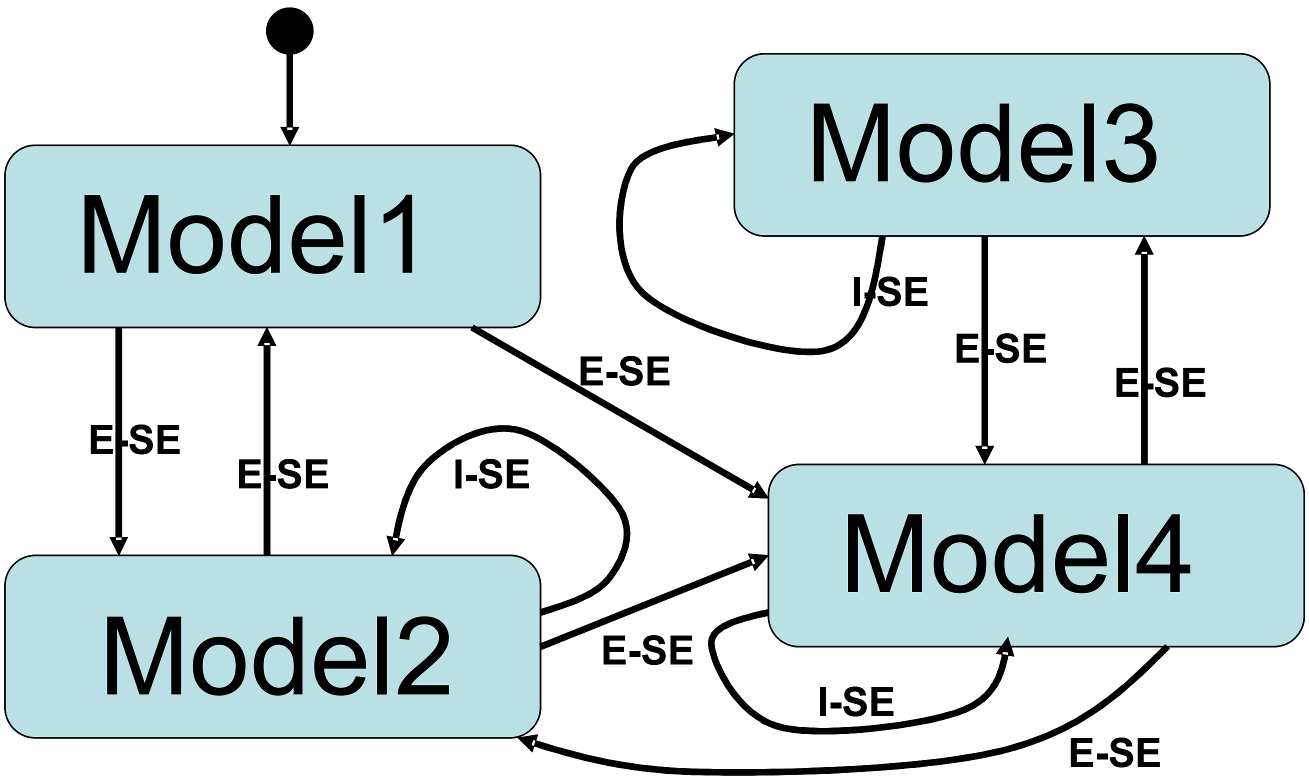


Hybrid Decomposition for structural-dynamic systems – external events E-SE



Hybrid and Structural Dynamic Systems

Mixture I-SE and E-SE



Simulators - I-SE or E-SE

Modern simulators ?

- whether a-causal physical modelling is supported,
- whether a-causal physical modelling is obeying the Modelica standard,
- Whether DAE handling is supported (index reduction, etc)
- whether external events are supported (equal to whether hybrid decomposition into independent submodels is possible),
- and whether state chart modelling or a similar construct is supported.

Simulators - I-SE or E-SE

Modern simulators ?

- whether a-causal physical modelling is supported,
- whether a-causal physical modelling is obeying the Modelica standard,
- Whether DAE handling is supported (index reduction, etc)
- whether external events are supported (equal to whether hybrid decomposition into independent submodels is possible),
- and whether state chart modelling or a similar construct is supported.

SIMULATOR	
A- causal Modelling	yes or no
MODLICA Standard	yes or no
DAE Handling	yes or no
Hybird decomposition	yes or no
State chart Modelling	yes or no

Simulators - I-SE or E-SE

Modern simulators ?

- MATLAB / Simulink
- Dymola / Modelica
- Mosilab / Modelica
- AnyLogic
- ModelVision
- Scilab/Scicos
- MAPLE - Sim

die Drahtwarenhandlung

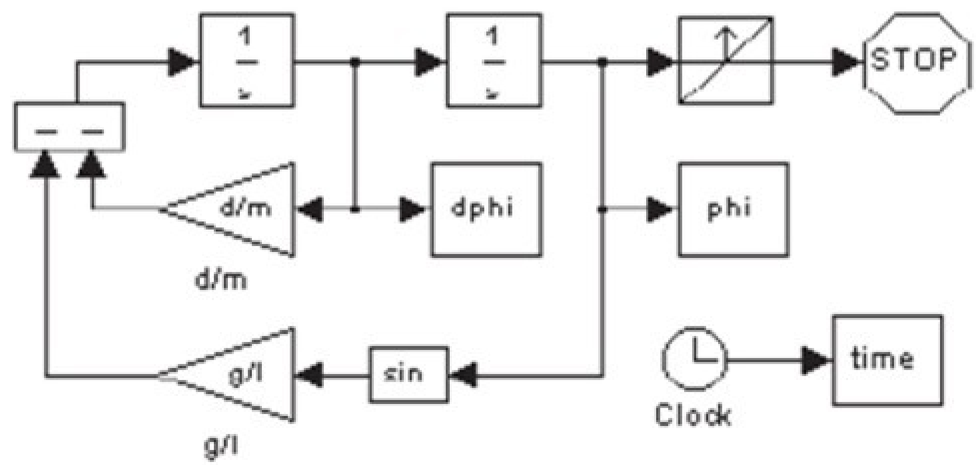
ASC - MMS
TU VIENNA

Modern simulators ? – MATLAB / Simulink

MATLAB / Simulink	
A- causal Modelling	no
MODLICA Standard	no
DAE Handling	weak
Hybrid decomposition	yes
State chart Modelling	no

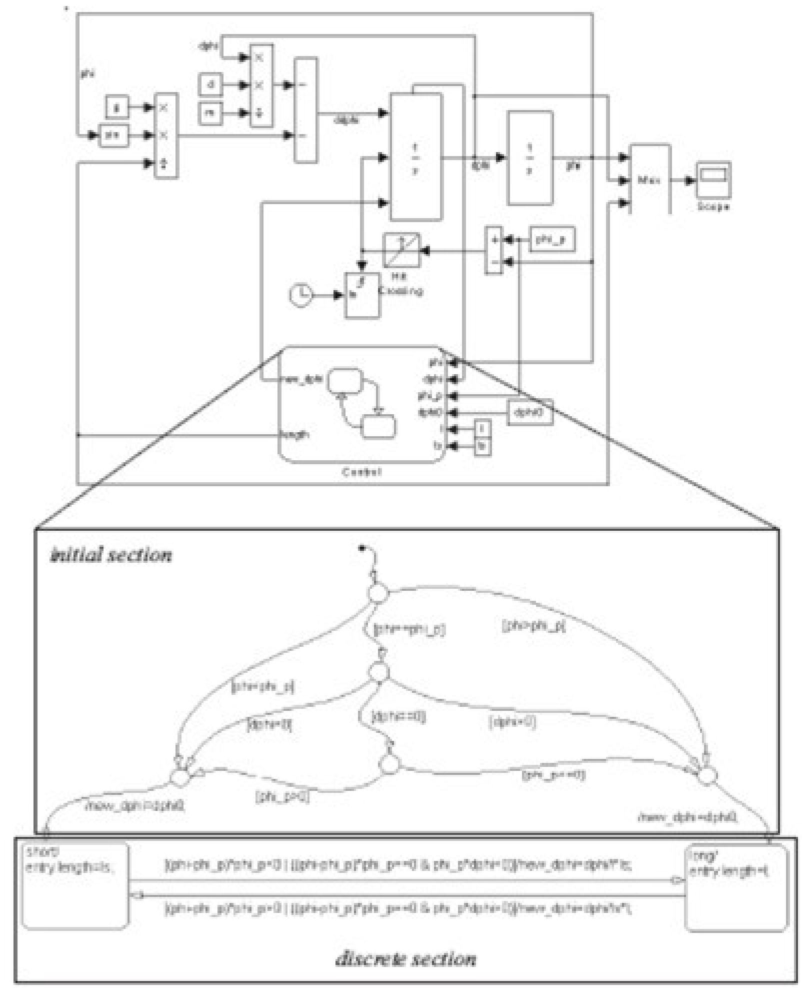
```
if ((phi_p-phi0)*phi_p<0  or
     (phi0==phi_p & phi_p*v>0))
     dphi0=v/ls;
     sim('pendulum_short',[t(length(t)),10]);
     v=dphi(length(dphi))*ls;

else    dphi0=v/l;
     sim('pendulum_long',[t(length(t)),10]);
     v=dphi(length(dphi))*l;end
```



Modern simulators ? – Simulink / STATEFLOW

Simulink / STATEFLOW	
A- causal Modelling	no
MODLICA Standard	no
DAE Handling	weak
Hybrid decomposition	no
State chart Modelling	yes



Stateflow

MODELICA / Dymola

equation /*pendulum*/ v =
length*der(phi); vdot = der(v);
mass*vdot/length + mass*g*sin(phi)
+damping*v = 0;

algorithm

if (phi<=phipin) then length:=ls; end if;

if (phi>phipin) then length:=l1; end if;

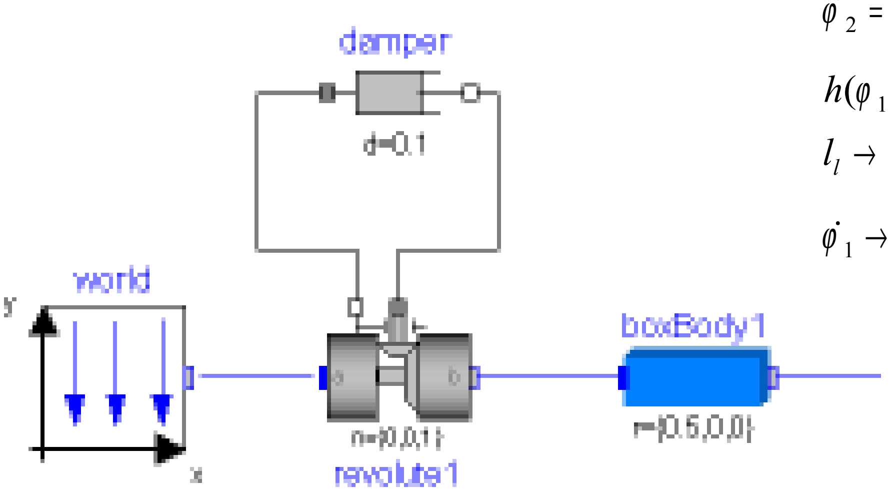
$$\dot{\varphi}_1 = \varphi_2,$$

$$\dot{\varphi}_2 = -\frac{g}{l}\sin\varphi_1 - \frac{d}{m}\varphi_2,$$

$$h(\varphi_1,\varphi_2) = \varphi_1 - \varphi_p = 0$$

$$l_l \to l_s, \quad l_s \to l_l$$

$$\dot{\varphi}_1 \to \dot{\varphi}_1 \frac{l_l}{l_s}, \dot{\varphi}_2 \to \dot{\varphi}_2 \frac{l_s}{l_l}$$

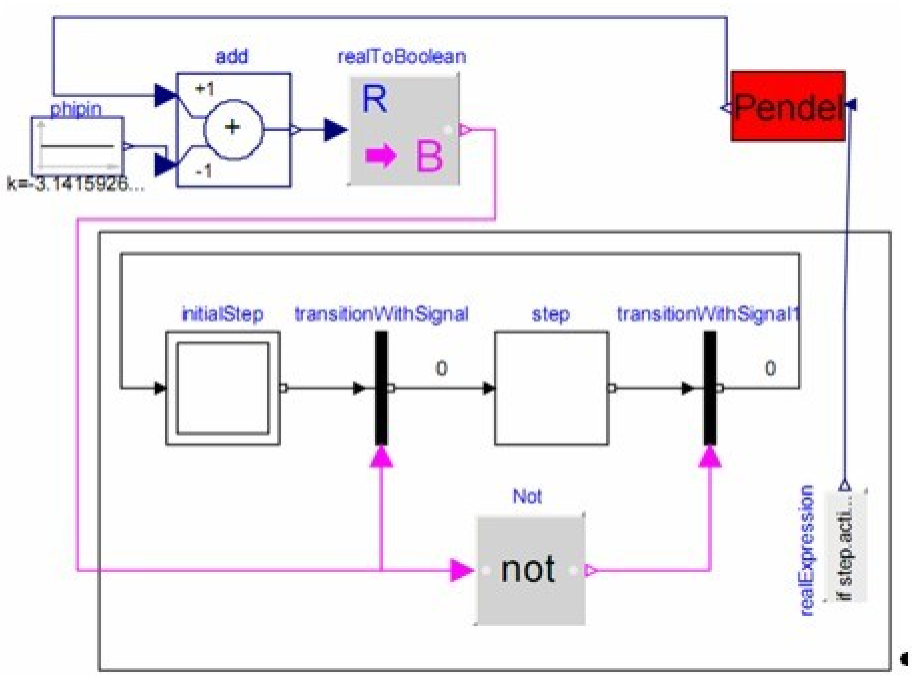


MODELICA / Dymola	
A- causal Modelling	yes
MODELICA Standard	yes
DAE Handling	yes
Hybrid decomposition	no
State chart Modelling	weak

die Drahtwarenhandlung

ASC - MMS
TU VIENNA

Modern simulators ? – Modelica / Dymola



State Graph Library

MODELICA / Dymola / State Chart Library	
A- causal Modelling	yes
MODELICA Standard	yes
DAE Handling	yes
Hybrid decomposition	no
State chart Modelling	weak

die Drahtwarenhandlung

ASC - MMS
TU VIENNA

Implementations of State Events in textual form

- classical if-statements

  if condition then expression else expression

- when construct

  when condition then end when;

- algorithm
- noEvent

  if noEvent(condition) then expression else expression

- smooth

  smooth(1, if condition then expression else expression)

- reinit

Simulators - I-SE or E-SE

Modern simulators ?
– Modelica / Mosilab

```
equation /*pendulum*/

v = l1*der(phi);
    vdot = der(v);mass*vdot/l1 +
    mass*g*sin(phi)+damping*v = 0;

algorithm
    if (phi<=phipin) then length:=ls; end
    if;if (phi>phipin) then length:=l1; end if;
    end
```

Modern simulators ? – Modelica / Mosilab

event Boolean lengthen(start=false),shorten(start = false);

equation
    lengthen=(phi>phipin); shorten=(phi<=phipin);

equation /*pendulum*/
    v = l1*der(phi); vdot = der(v); mass*vdot/l1 + mass*g*sin(phi)+damping*v= 0;

Statechart
    state LengthSwitch extends State;
    State Short,Long,Initial(isInitial=true);
    transition Initial -> Long end transition;
    transition Long -> Short event shorten action length := ls;end transition;
    transition Short -> Long event lengthen action length := l1;end transition; end LengthSwitch;

MODELICA / Mosilab	
A- causal Modelling	yes
MODELICA Standard	yes
DAE Handling	no
Hybrid decomposition	yes
State chart Modelling	yes

die Drahtwarenhandlung

ASC - MMS
TU VIENNA

Simulators - I-SE or E-SE

Modern simulators ? –

Modelica / Mosilab

model Long
equation
mass*vdot/l1 + mass*g*sin(phi)+damping*v = 0;end Long;

model Short
equationmass*vdot/ls + mass*g*sin(phi)+damping*v = 0;end Short;

event discrete Boolean lengthen(start=true),
shorten(start = false);equationlengthen =
(phi>phipin);shorten=(phi<=phipin);

Statechart
state ChangePendulum extends State;State
Short,Long,startState(isInitial=true);transition startState -> Long
actionL:=new Long(); K:=new Short(); add(L);end transition;transition ….

die Drahtwarenhandlung

ASC - MMS
TU VIENNA

Simulators - I-SE or E-SE

Modern simulators ? –

Modelica / Mosilab

```
model Long
    equation
    mass*vdot/l1 + mass*g*sin(phi)+damping*v = 0;end Long;

model Short
    equationmass*vdot/ls + mass*g*sin(phi)+damping*v = 0;end Short;


event discrete Boolean lengthen(start=true),
    shorten(start = false);equationlengthen =
    (phi>phipin);shorten=(phi<=phipin);

Statechart
    state ChangePendulum extends State;State
    Short,Long,startState(isInitial=true);transition startState -> Long
    actionL:=new Long(); K:=new Short(); add(L);end transition;transition ….
```

AnyLogic

Equations
d(alpha)/dt = omega
x = l*sin(alpha), y = l*cos(alpha)

Equations
d(omega)/dt = (-g*sin(alpha)-
mu*omega)/ls

Change eventLong
(alpha>=alphaN)||(alpha<=alpha
Action
omega=omega*ls/ll
stop

Change EventShort
(alpha>=alphaN)||(alpha<=alpha)
Action
omega=omega*ll/ls
Stop

Equations
d(omega)/dt =
(-g*sin(alpha)- mu*omega)/ll

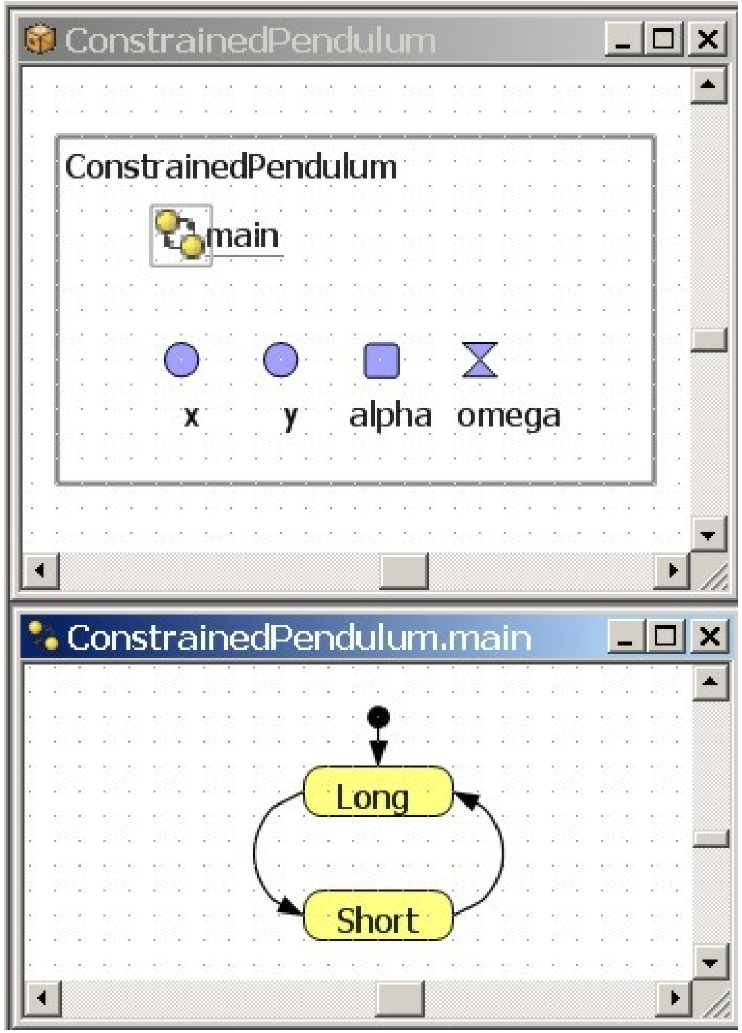
$$\dot{\varphi}_1 = \varphi_2,$$

$$\dot{\varphi}_2 = -\frac{g}{l}\sin\varphi_1 - \frac{d}{m}\varphi_2,$$

$$h(\varphi_1,\varphi_2) = \varphi_1 - \varphi_p = 0$$

$$l_l \to l_s, \quad l_s \to l_l$$

$$\dot{\varphi}_1 \to \dot{\varphi}_1 \frac{l_l}{l_s}, \dot{\varphi}_2 \to \dot{\varphi}_2 \frac{l_s}{l_l}$$

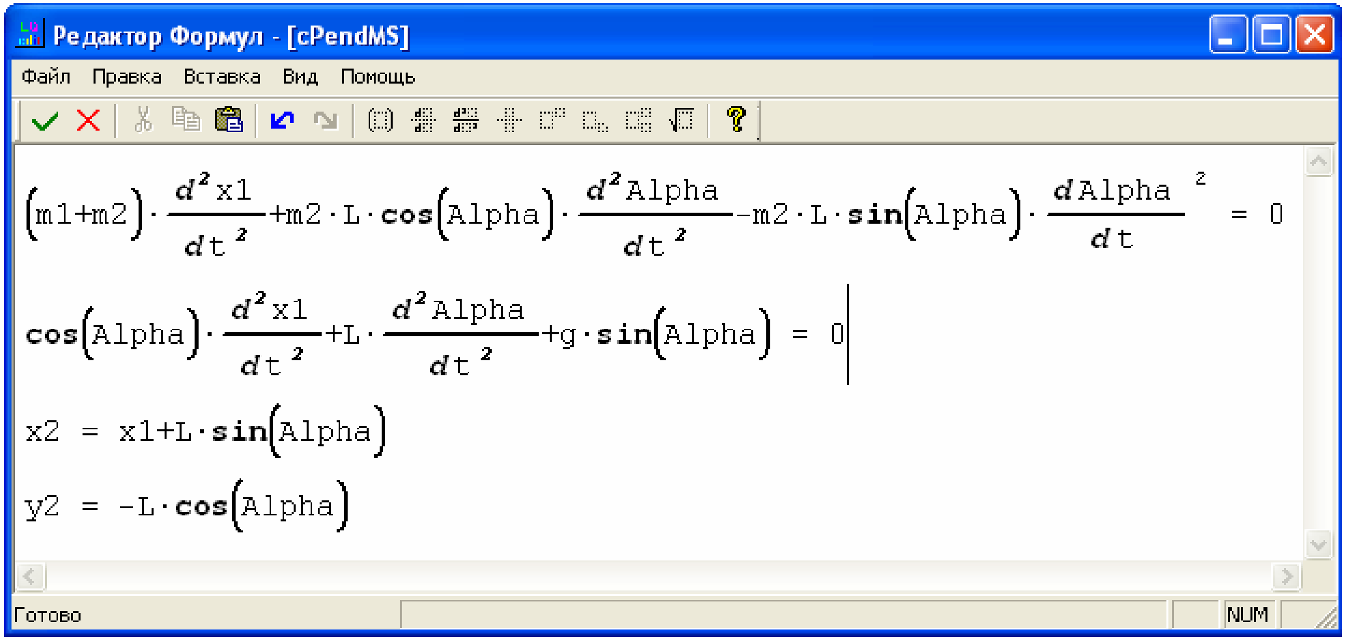


AnyLogic	
A- causal Modelling	weak
MODELICA Standard	no
DAE Handling	no
Hybrid decomposition	yes
State chart Modelling	yes

die Drahtwarenhandlung

ASC - MMS TU VIENNA

ModelVision ~ AnyLogic



Model Vision	
A- causal Modelling	yes
MODELICA Standard	no
DAE Handling	yes
Hybrid decomposition	yes
State chart Modelling	yes

die Drahtwarenhandlung

ASC - MMS
TU VIENNA

Scilab / Scicos

- **Extending the model description by Modelica models (textually and graphically), and**
- **refining the if-then-else – and when – clause by introducing different classes of associated events, resulting in clauses being as capable as state charts.**

Scilab / Scicos	
A- causal Modelling	yes
MODELICA Standard	no
DAE Handling	yes
Hybrid decomposition	weak
State chart Modelling	simulate d

die Drahtwarenhandlung

ASC - MMS
TU VIENNA

MAPLE / Sim

- **New Toolbox to Maple (2008)**
- **Modelica- and Maple-Sim Libs generate Maple DAEs**
- **Simulation with Maple**

MAPLE Sim	
A- causal Modelling	yes
MODELICA Standard	yes
DAE Handling	yes
Hybrid decomposition	no
State chart Modelling	no

die Drahtwarenhandlung

ASC - MMS
TU VIENNA

Summary

I-SE or E-SE

- **Formal Model for Modelica extension**
- **Distinction between different approaches**
- **May be translation options for SE-P, SE-S, SE-D**

- **More E-SE – less DAE ?**