Computer Aided Control System Design

Anchorage, Alaska

25 September, 2000

DEVS as a Common Denominator for Multi-formalism Hybrid Systems Modelling

Hans Vangheluwe

School of Computer Science

McGill University

Montréal, Canada

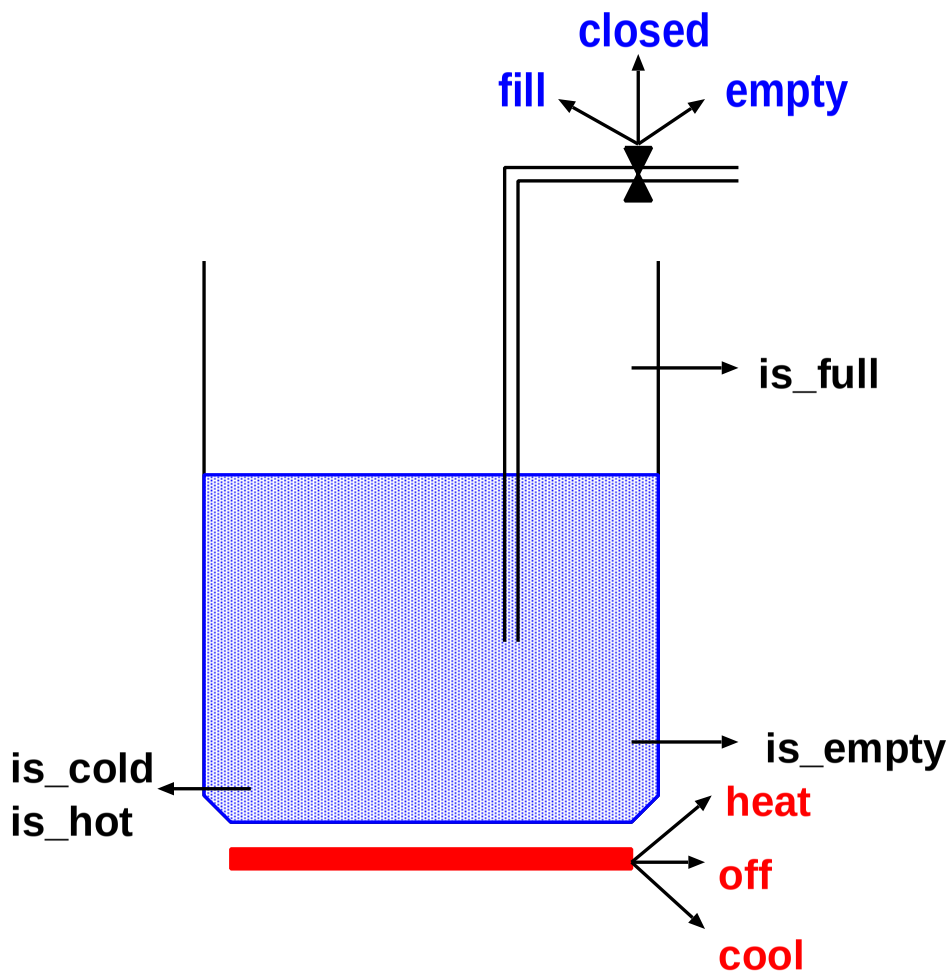
Overview

1. Multi-formalism Modelling and Simulation
   - What ?
   - Formalism Transformation Graph
   - Formalism Transformation
2. DEVS
3. Mapping ODEs onto DEVS
4. Conclusions/Future Work

Multi-formalism Modelling and Simulation

- Different levels of abstraction
- Answering different questions
- Mixing different formalisms

System under study: T, h controlled liquid



Detailed (continuous) view, ALG + ODE formalism

Inputs (discontinuous $\rightarrow$ hybrid model):

- Emptying, filling flow rate ϕ
- Rate of adding/removing heat W

Parameters:

- Cross-section surface of vessel A
- Specific heat of liquid c
- Density of liquid ρ

State variables:

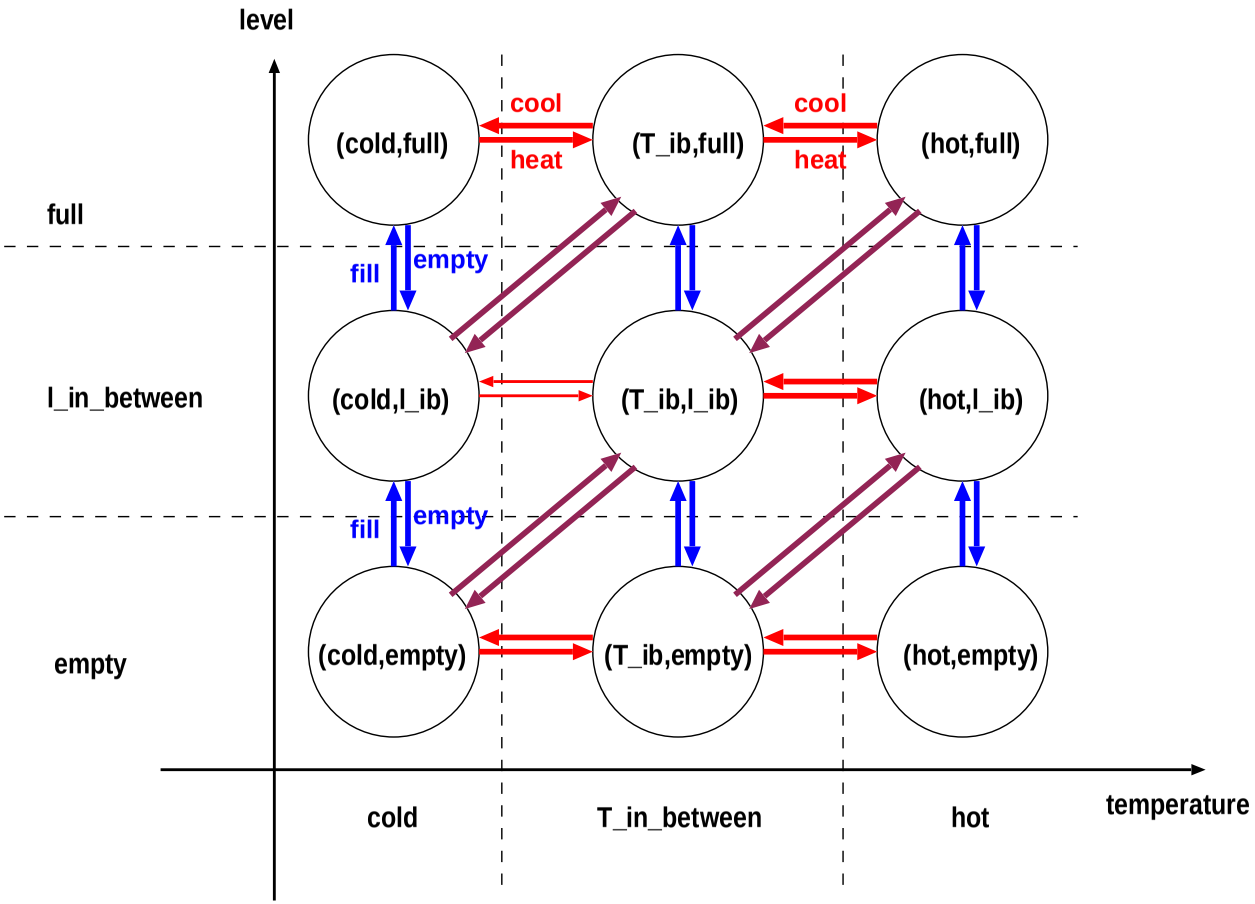
- Temperature T
- Level of liquid l

Outputs (sensors):

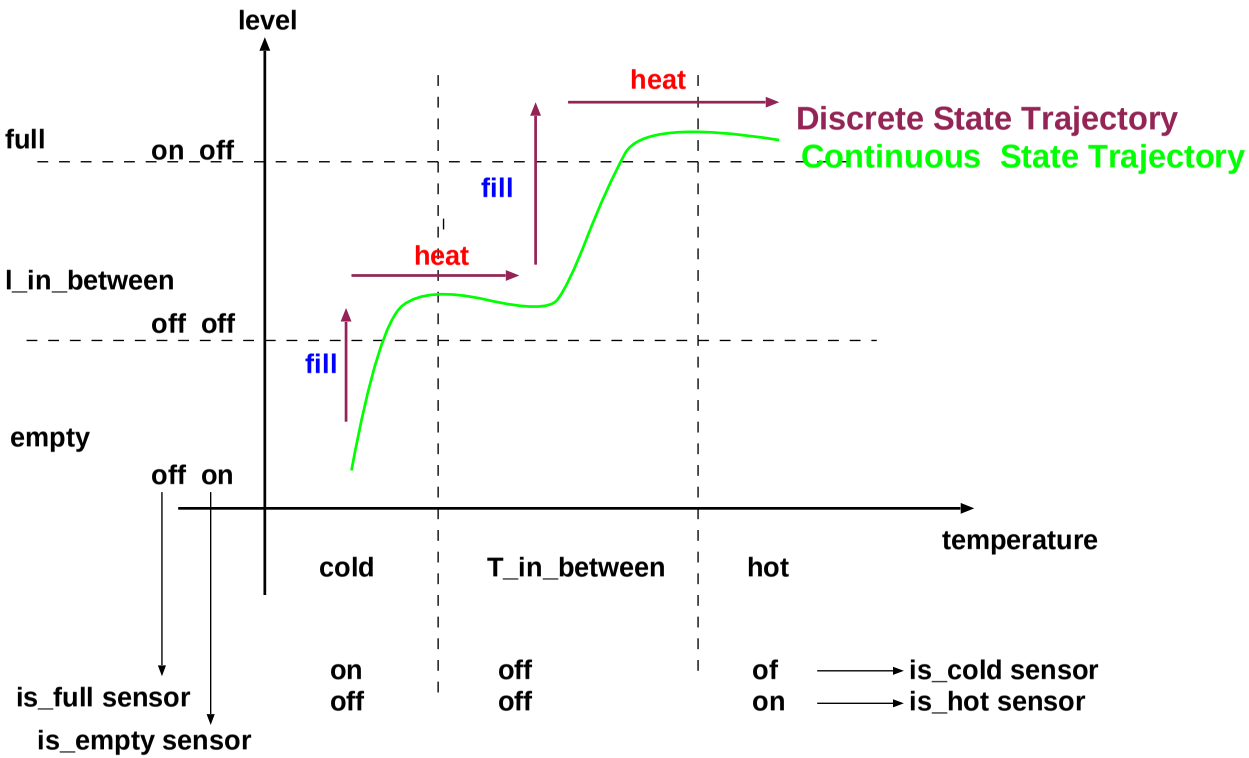
- $is_low, is_high, is_cold, is_hot$

$$\begin{cases} \frac{dT}{dt} & = & \frac{1}{l}[\frac{W}{c\rho A} - \phi T] \\ \frac{dl}{dt} & = & \phi \\ is_low & = & (l < l_{low}) \\ is_high & = & (l > l_{high}) \\ is_cold & = & (T < T_{cold}) \\ is_hot & = & (T > T_{hot}) \end{cases}$$

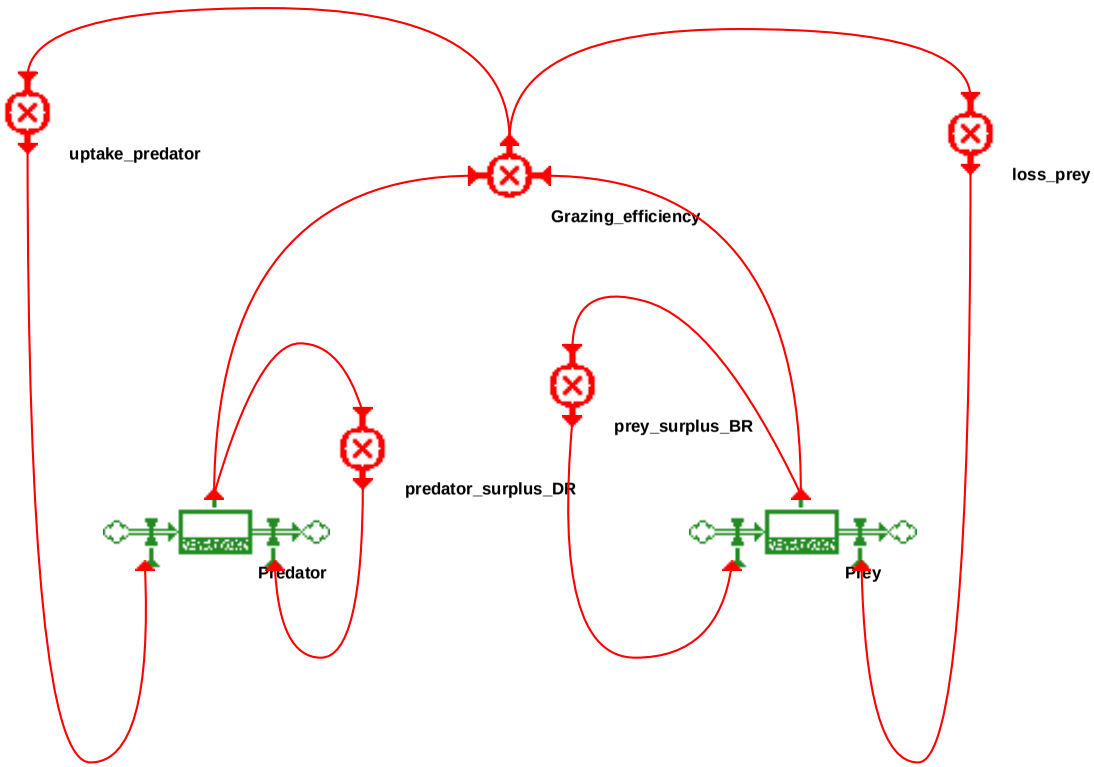
High-level (discrete) view, FSA formalism



Levels of abstraction/views: trajectories



Predator-prey, Forrester System Dynamics

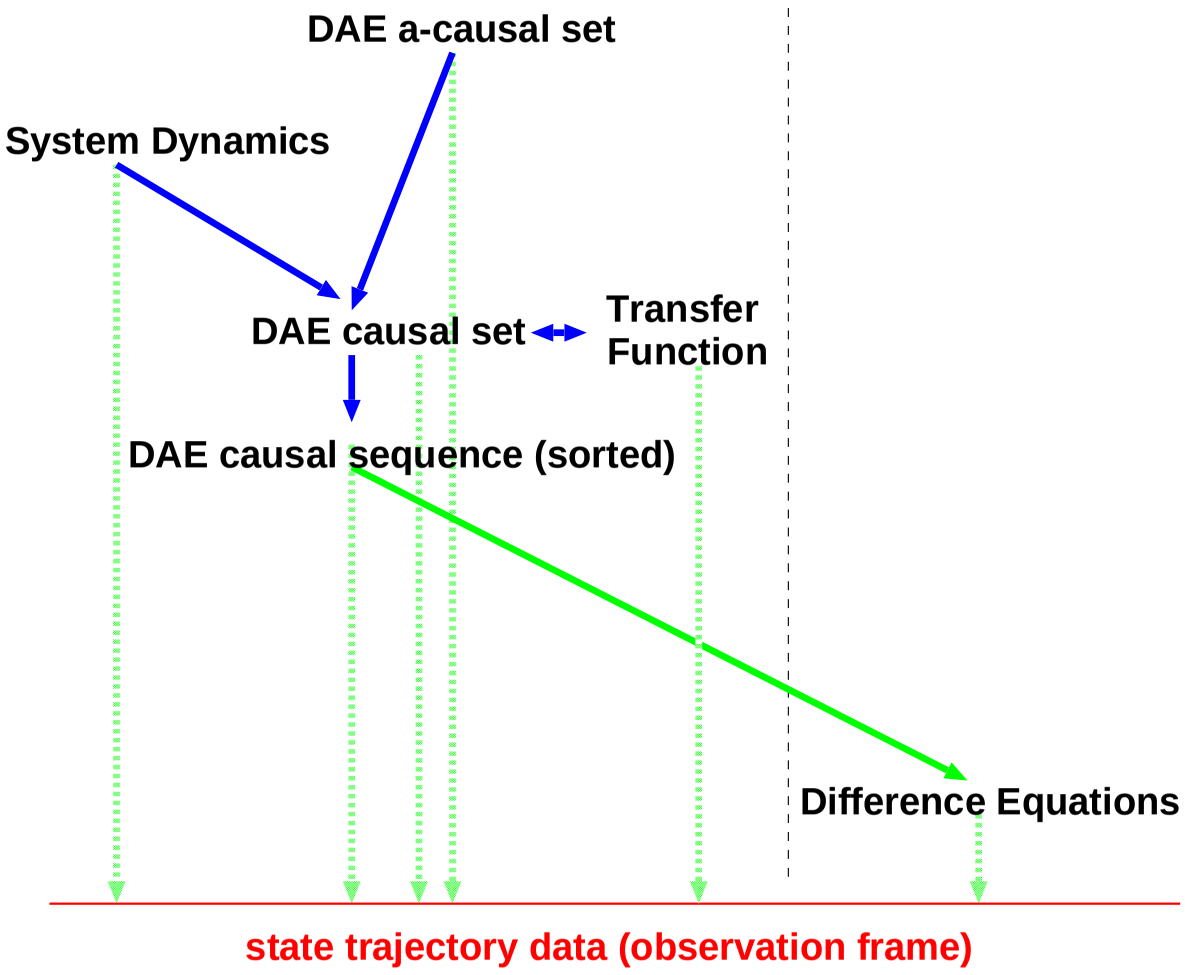


2−species predator−prey system

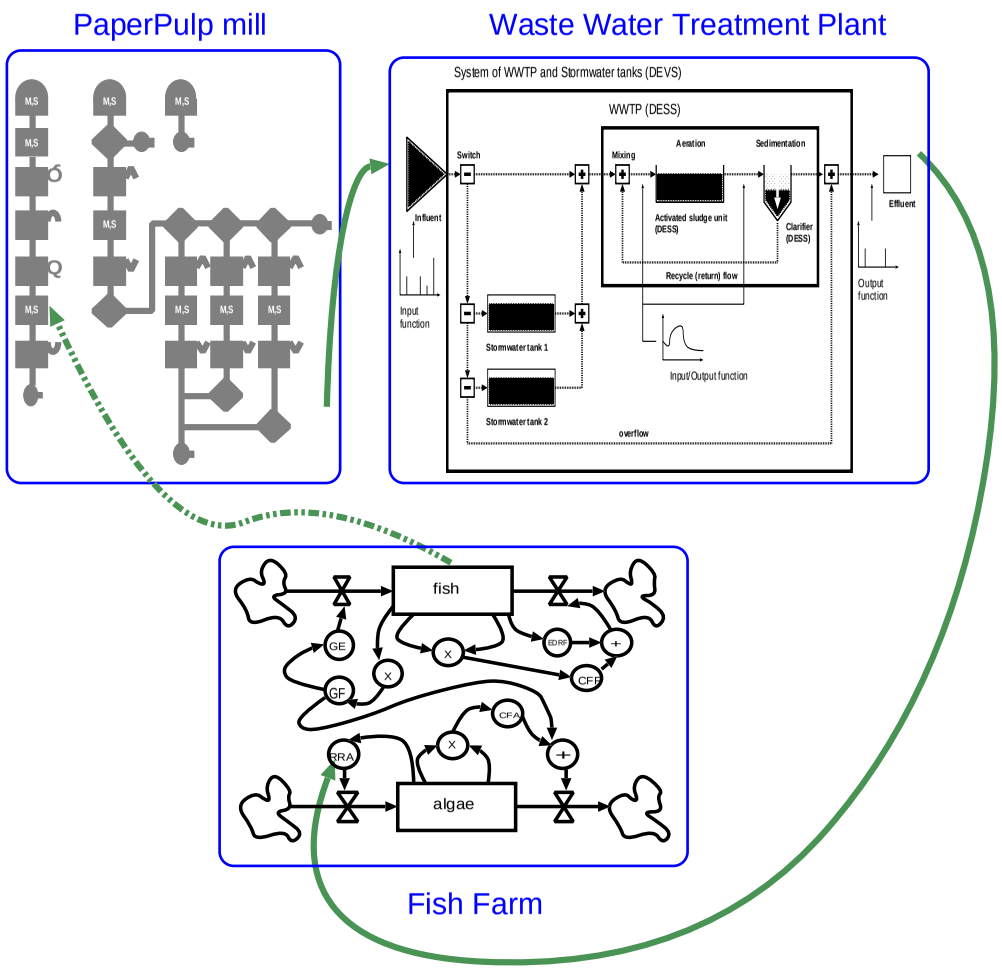
Predator-prey, ODE + ALG

$$\frac{d\ level}{dt} = BR - DR.$$

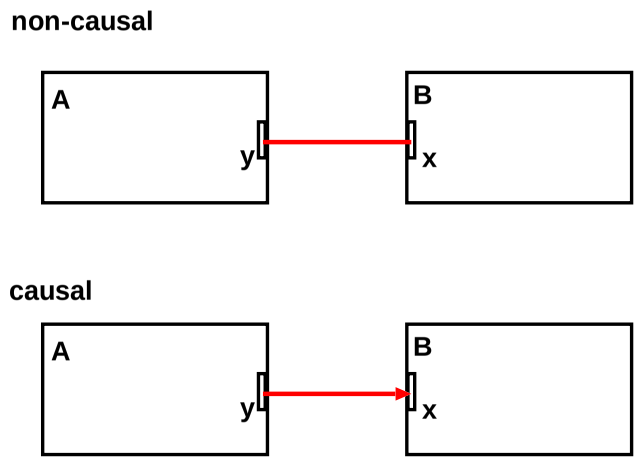
Formalism Transformation



Complex System: Coupling Different Formalisms



Closure under Coupling/Composition: Block Diagram



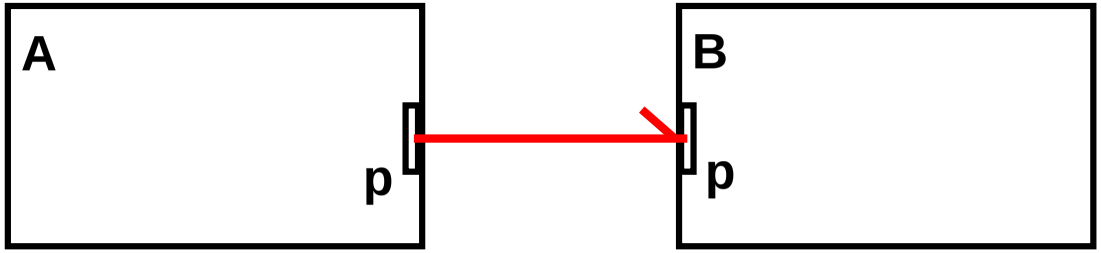
Non-Causal:

$$A.y = B.x$$

Causal:

$$B.x := A.y$$

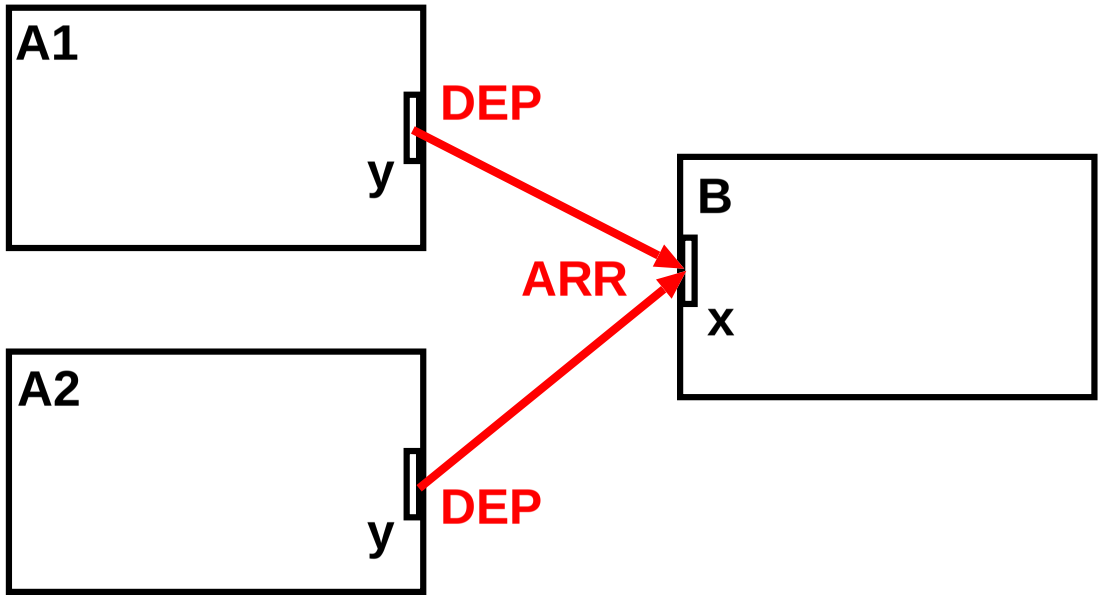
Closure under Coupling/Composition: non-causal Bond Graph



$$A.p.effort = B.p.effort$$

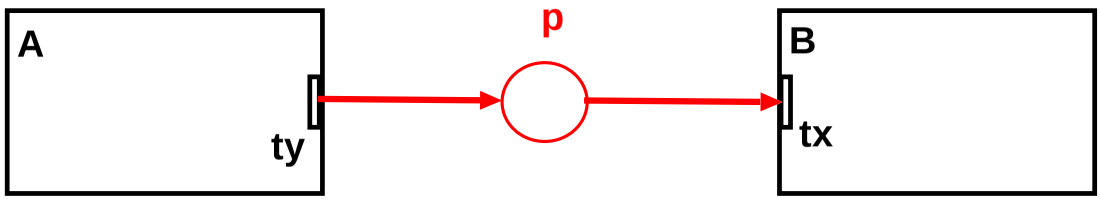
$$A.p.flow = B.p.flow$$

Closure under Coupling/Composition: Discrete Event



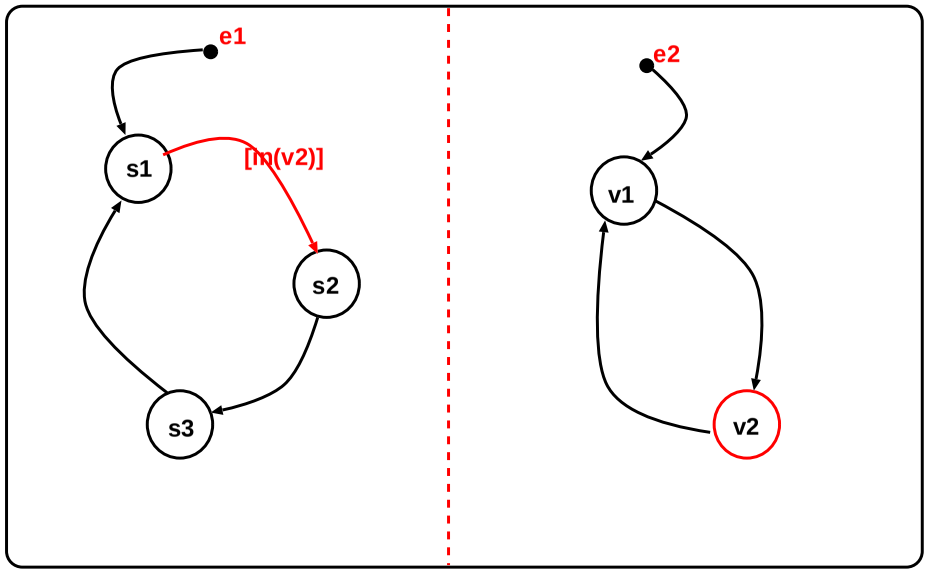
- *schedule* ARRivals
- *resolve* collisions

Closure under Coupling/Composition: Petri Net



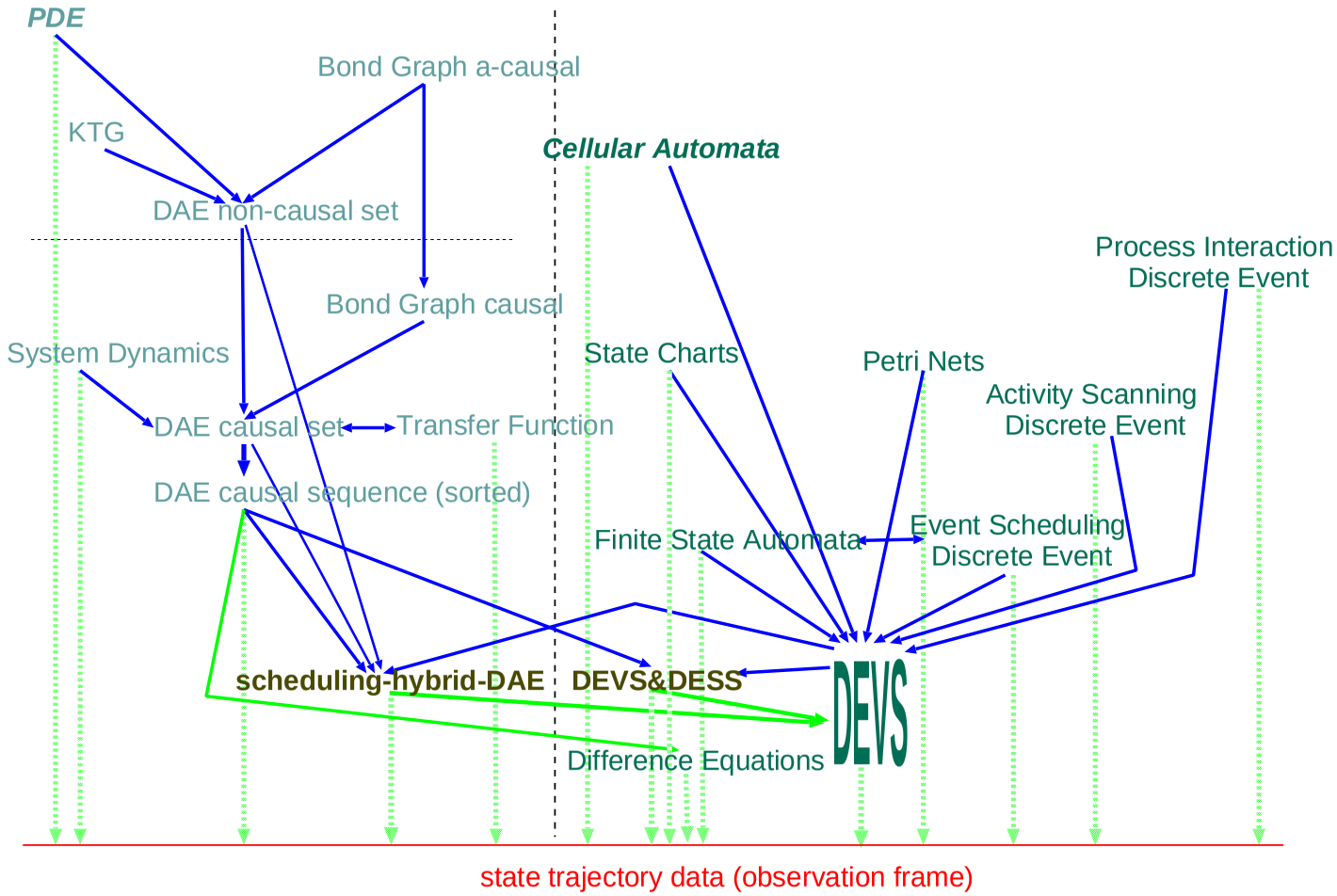
- Transitions $A.ty, B.tx$ are used as ports
- Coupling between ports by means of a place p

Closure under Coupling/Composition: State Charts



- Concurrency (dotted vertical line)
- Synchronisations by means of guards ($[in(v2)]$)
- Hierarchy through entry points ($e1, e2$)

Formalism Transformation Graph



Use of Formalism Transformation

- Answer questions at “optimal” level
- Re-use lower level modeller/simulator
- Potential for optimization
- New formalisms
- Semantics of coupled multi-formalism models

Semantics of Coupled Models

- Meta-formalism subsumes all formalisms
- Transform to common formalism
- Co-simulation (transform to trajectory)

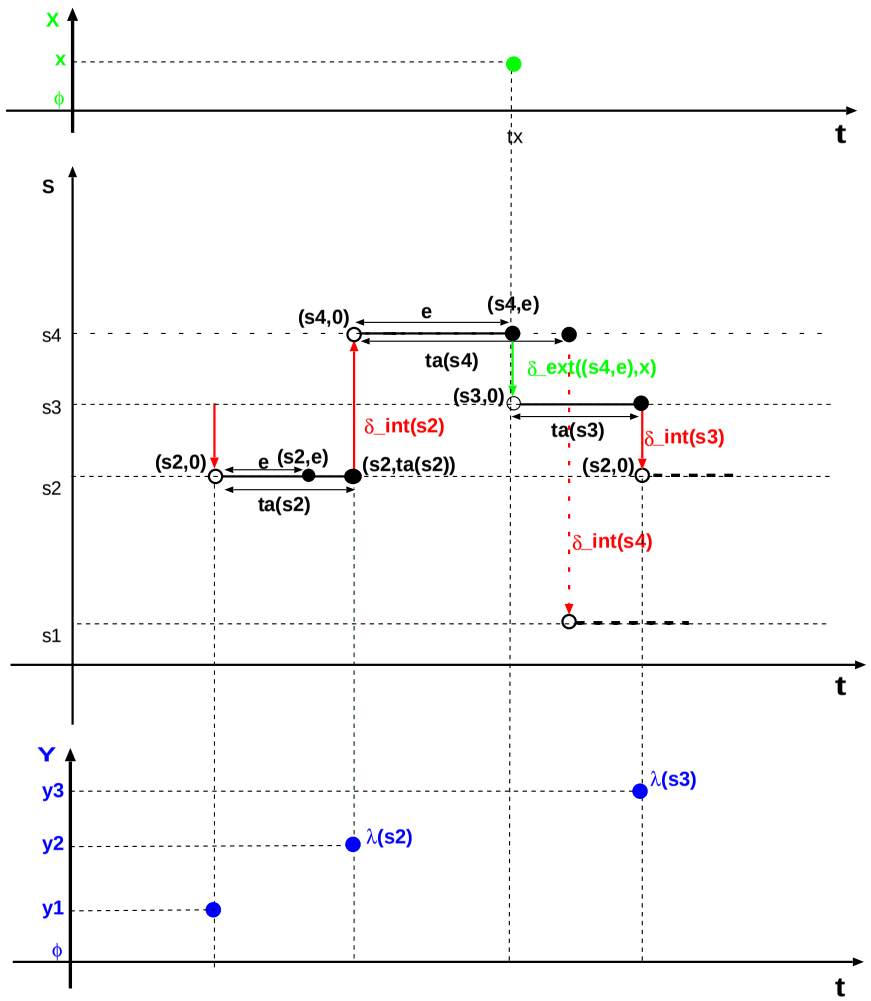
Transformation to Common Formalism

1. Start from a coupled multi-formalism model. Check consistency of this model (*e.g.,* whether causalites and types of connected ports match).
2. Cluster all formalisms described in the same formalism.
3. For each cluster, implement closure under coupling.
4. Look for the best common formalism in the Formalism Transformation Graph all the remaining different formalisms can be transformed to. Worst case: trajectory level.
5. Transform all the sub-models to the common formalism.
6. Implement closure under coupling of the common formalism.

$$atomicDEVS \equiv \langle X, S, Y, \delta_{int}, \delta_{ext}, \lambda, ta \rangle$$

$T = \mathbb{R}$	time base
X	input set
$\omega : T \rightarrow X \cup \{\phi\}$	input segment
S	state set
Y	output set
$\delta_{int} : S \rightarrow S$	internal transition function
$ta : S \rightarrow \mathbb{R}^{+}_{0,\infty}$	time advance function
$Q = \{(s,e) \mid s \in S, 0 \leq e \leq ta(s)\}$	total state, e is elapsed time
$\delta_{ext} : Q \times X \rightarrow S$	external transition function
$\lambda : S \rightarrow Y$	output function

State Trajectory of a DEVS-specified Model



Coupled DEVS

$$coupledDEVS \equiv \langle X_{self}, Y_{self}, D, \{M_i\}, \{I_i\}, \{Z_{i,j}\}, select \rangle$$

$$M_i = \langle S_i, ta_i, \delta_{int,i}, X_i, \delta_{ext,i}, Y_i, \lambda_i \rangle, \forall i \in D.$$

$$\{Z_{i,j} | i \in D \cup \{self\}, j \in I_i\},$$

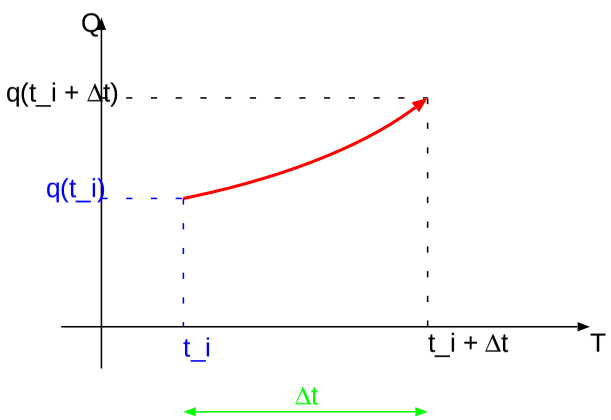
$$\begin{aligned} Z_{self,j} &: X_{self} \rightarrow X_j \quad , \forall j \in D, \\ Z_{i,self} &: Y_i \rightarrow Y_{self} \quad , \forall i \in D, \\ Z_{i,j} &: Y_i \rightarrow X_j \quad , \forall i, j \in D. \end{aligned}$$

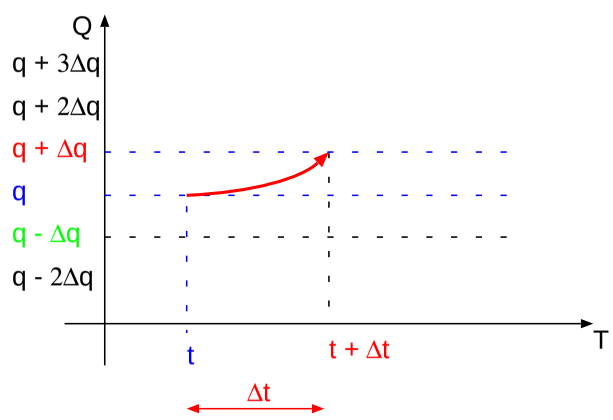
parallel DEVS

$$atomic\ P-DEVS \equiv \langle S, ta, \delta_{int}, X, \delta_{ext}, \delta_{conf}, Y, \lambda \rangle,$$

Mapping ODE onto DEVS

$$\begin{cases} \frac{dq}{dt} = f(q,x,t) & q \in Q \\ y(t) = g(q,t) & y \in Y \end{cases}$$





Quantization

$$atomicDEVS \equiv \langle \hat{S}, ta, \delta_{int}, \hat{X}, \delta_{ext}, \hat{Y}, \lambda \rangle.$$

$$\delta_{int}((\hat{q}, \hat{x}, t)) = (\hat{q} + sgn(f(\hat{q}, \hat{x}, t))\Delta_q, \hat{x}, t + ta(\hat{q}, \hat{x}, t)).$$

$$ta((\hat{q}, \hat{x}, t)) = \left| \frac{\Delta_q}{f(\hat{q}, \hat{x}, t)} \right|$$

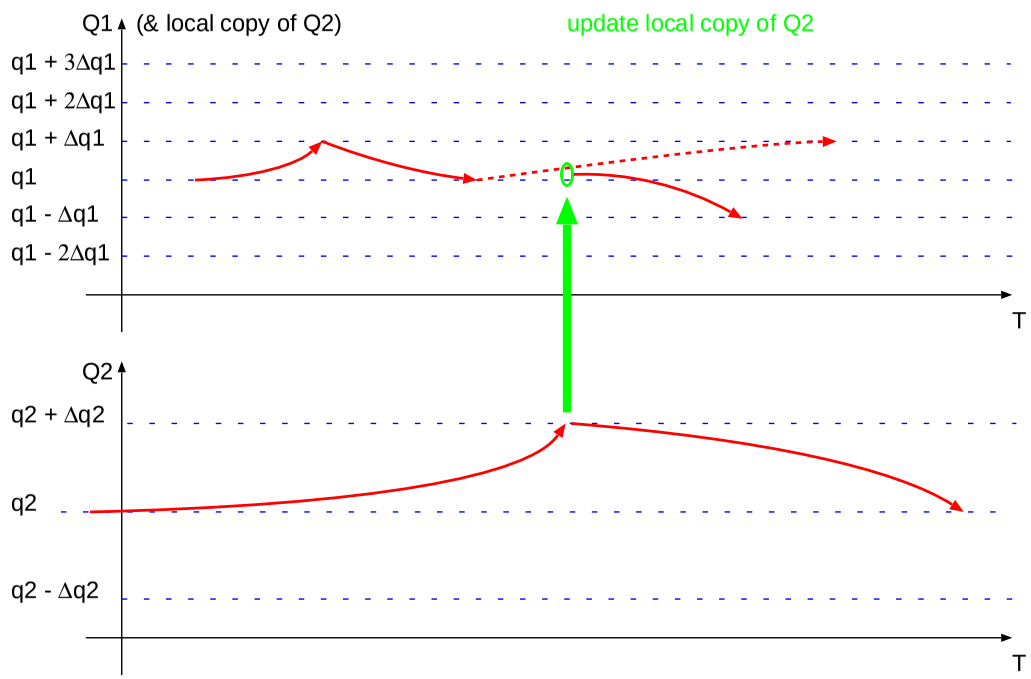
$$\delta_{ext}((\hat{q}, \hat{x}, t), e, \hat{x}') = (\hat{q}, \hat{x}', t + e).$$

$$\delta_{confl}((\hat{q}, \hat{x}, t), \hat{x}') = (\hat{q} + sgn(f(\hat{q}, \hat{x}, t))\Delta_q, \hat{x}', t).$$

$$\lambda((\hat{q}, \hat{x}, t)) = \lfloor g(\hat{q}, t)/\Delta_y \rfloor.$$

Coupling

$$\begin{cases} \frac{dq_1}{dt} = f_1(q_1, q_2, t) & q_1 \in Q_1 \\ \frac{dq_2}{dt} = f_2(q_2, x, t) & q_2 \in Q_2 \end{cases}$$



Conclusions/Future Work

- Need for Multi-formalism Modelling and Simulation
- Formalism Transformation
- DEVS (discrete event)
- Mapping ODE onto DEVS
- Error analysis, prototype performance