Hybrid System Modeling in VDM

MARCEL VERHOEF
CHESS & Radboud University Nijmegen – NL

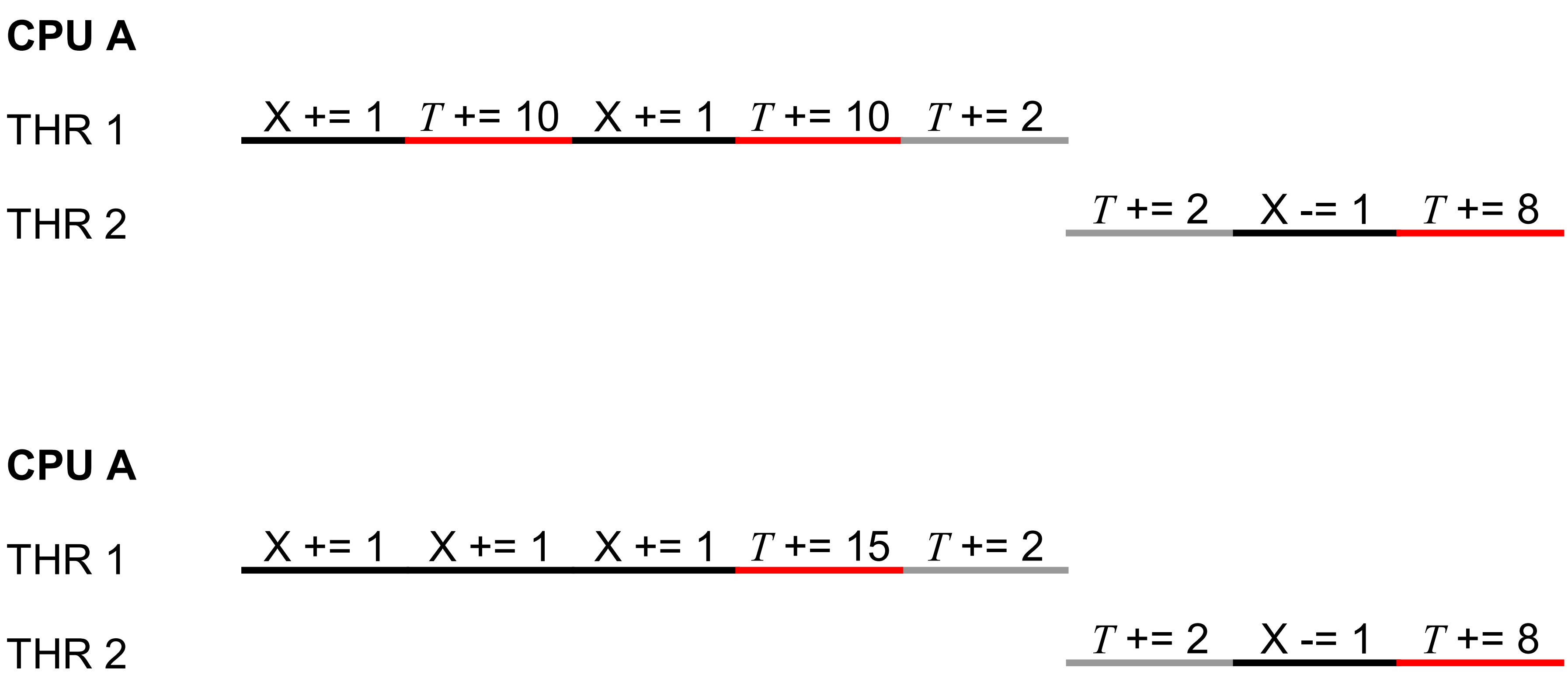
joint work with **PETER VISSER**
University of Twente – NL

and **Zoe Andrews**
School of Computing Science
Newcastle University - UK

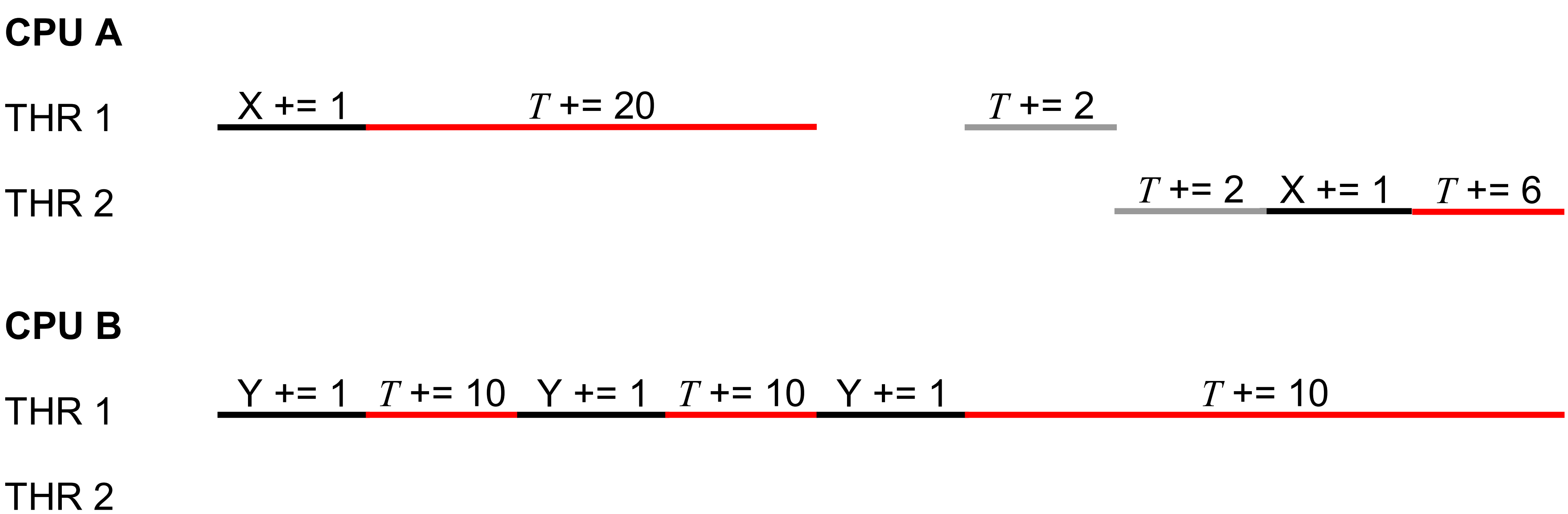
Contents

- Timed VDM++ dynamic semantics
- Distributed Timed VDM++ dynamic semantics
- Continuous Time simulations
- Interfacing DE and CT simulations
- Proof of concept – architecture overview
- Example and demonstration
- Conclusions and future work

Timed VDM++ dynamic semantics



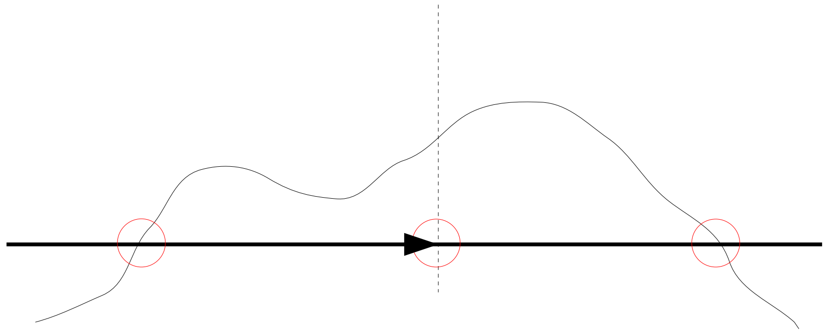
Distributed Timed VDM++ dynamic semantics



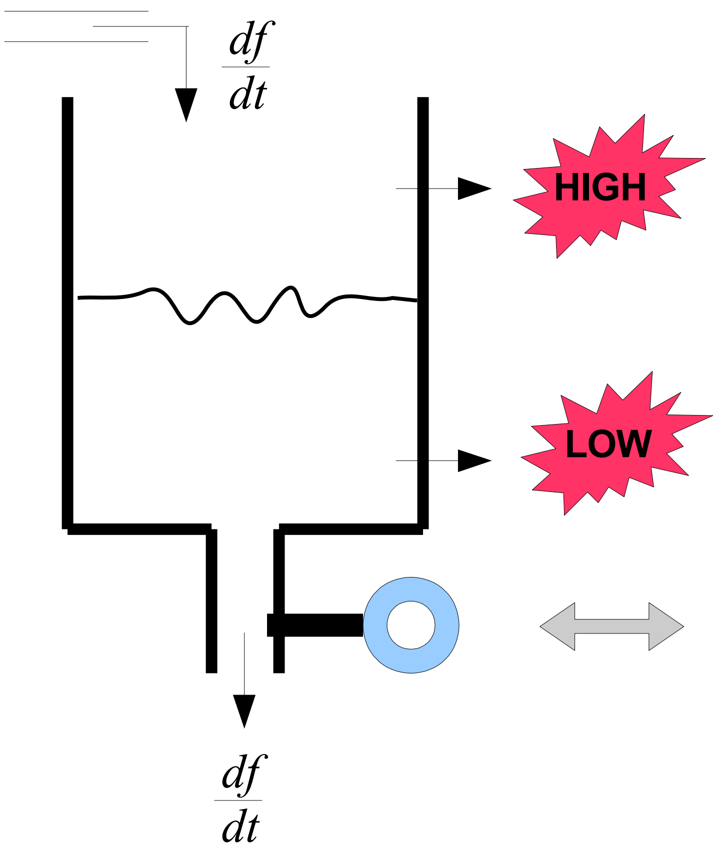
Caveat: BUS activity is dealt with as time steps

Simulation of Continuous Time models

- Sets of differential equations
- Solved numerically using some solver (e.g. Euler)
- State and Time events can be captured
  - state: zero-crossing detection (rising and falling edge)
  - time: proceed to point in the future

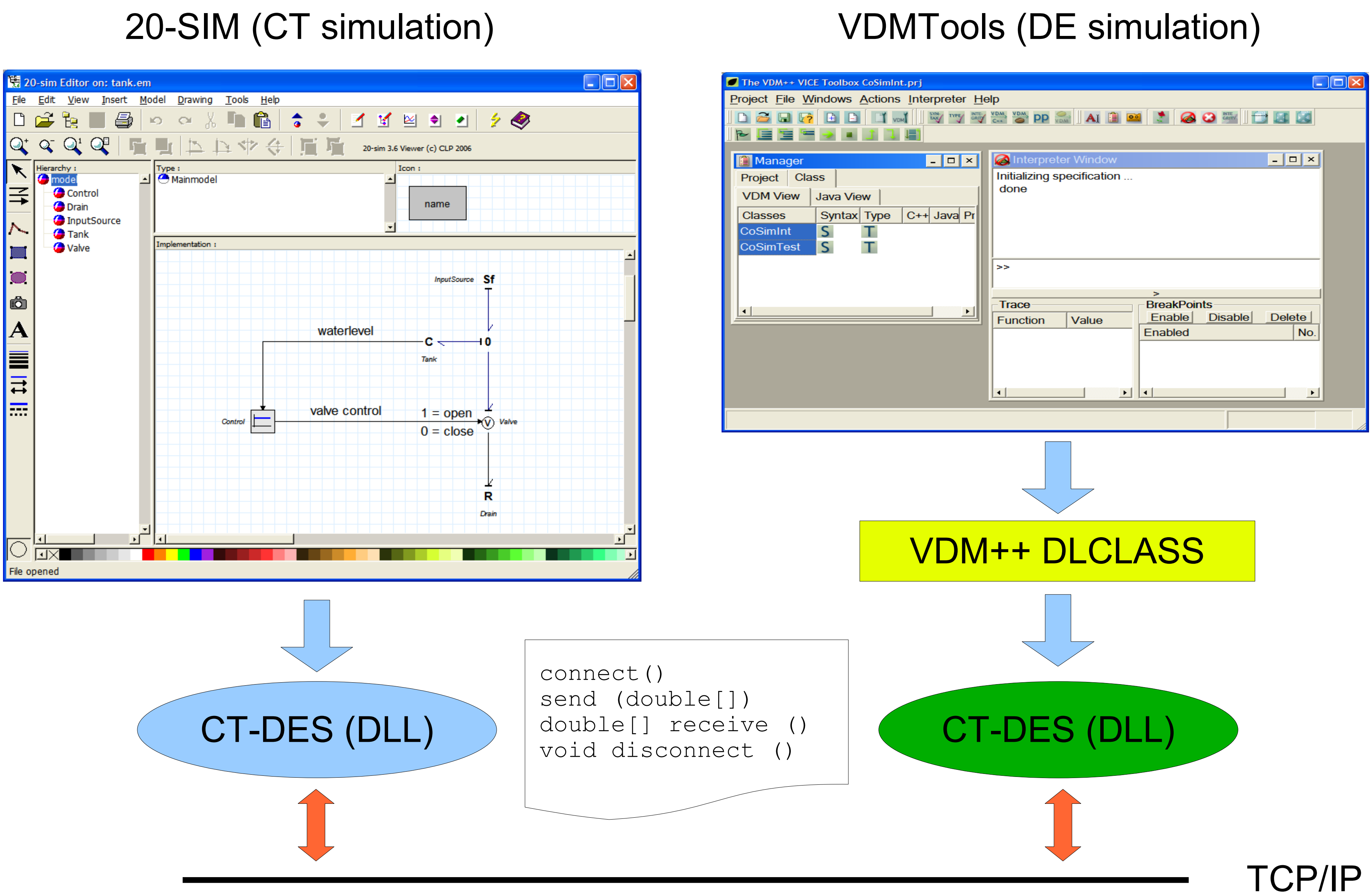


Example – Controlling the waterlevel in a tank




```
class Controller
instance variables
  level : real := 0.0;
  valve : bool := false

operations
  public async open: () ==> ()
  open () == valve := true;

  public async close: () ==> ()
  close () == valve := false;

  public async update: () ==> ()
  update () ==
    if level < 2.0 then close()
    else if level > 3.0 then open()

threads
  periodic (1000) (update)

end Controller
```

Proof of Concept – Architecture Overview

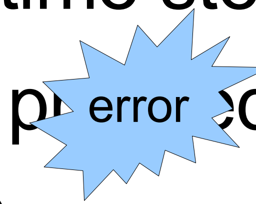


How is integration achieved?

- DE simulator is in control of time, CT is "slave"
- Perform DE until all CPUs are ready to make time step
- Determine smallest time step (including bus), proceed
- Inform CT solver (shared variables, time step)
- CT solver proceeds until new time is reached
- Inform DES (occurred events, shared variables)
- Events are processed by DES
- Iterate (step 2)

How is integration achieved?

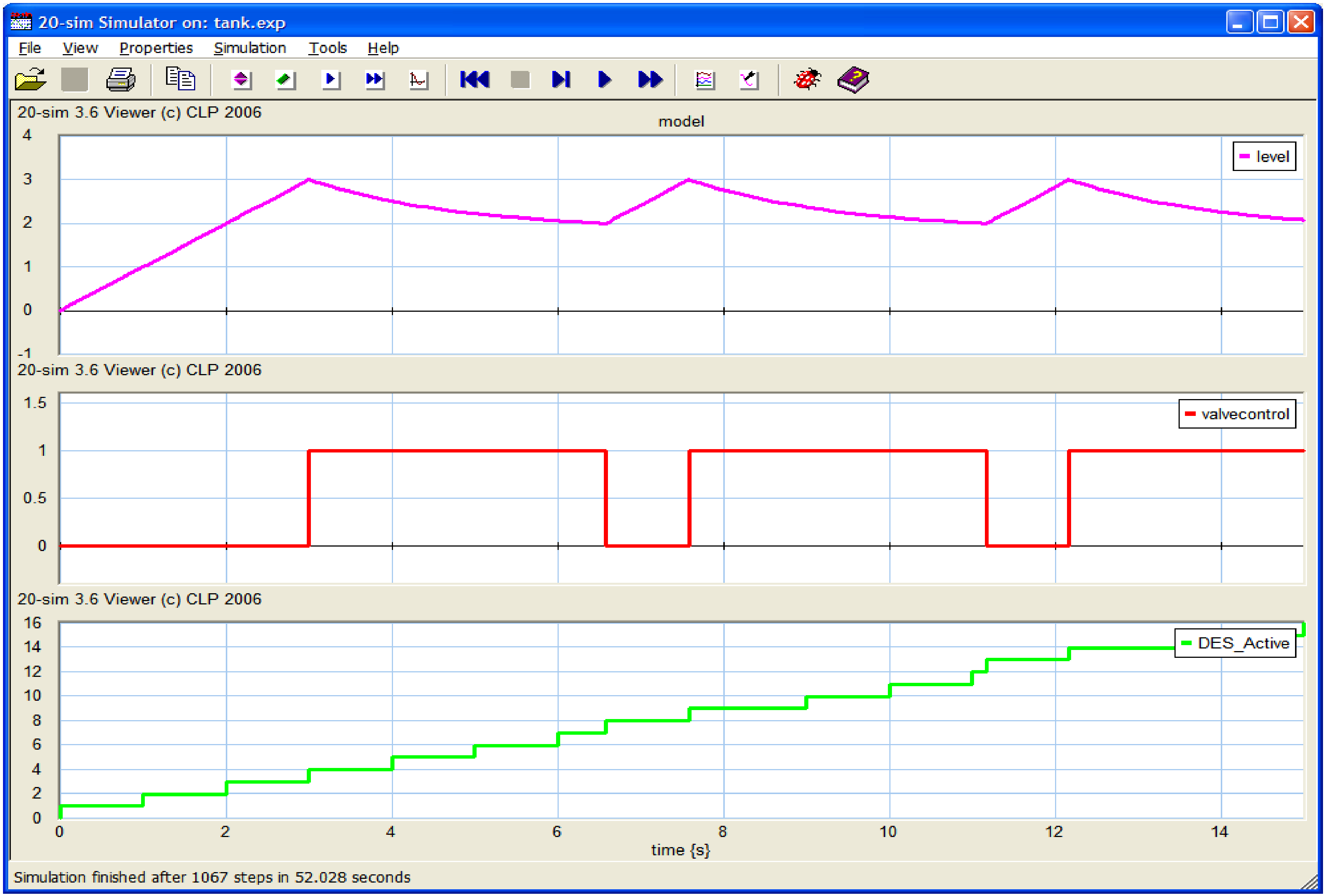
- DE simulator is in control of time, CT is “slave”
- Perform DE until all CPUs are ready to make time step
- Determine smallest time step (including bus), p[illegible]ed



- Inform CT solver (shared variables, time step)
- CT solver proceeds until new time is reached
- Inform DES (occurred events, shared variables)
- Events are processed by DES
- Iterate (step 2)

caveat: both simulator can only move forward in time

Simulation Result

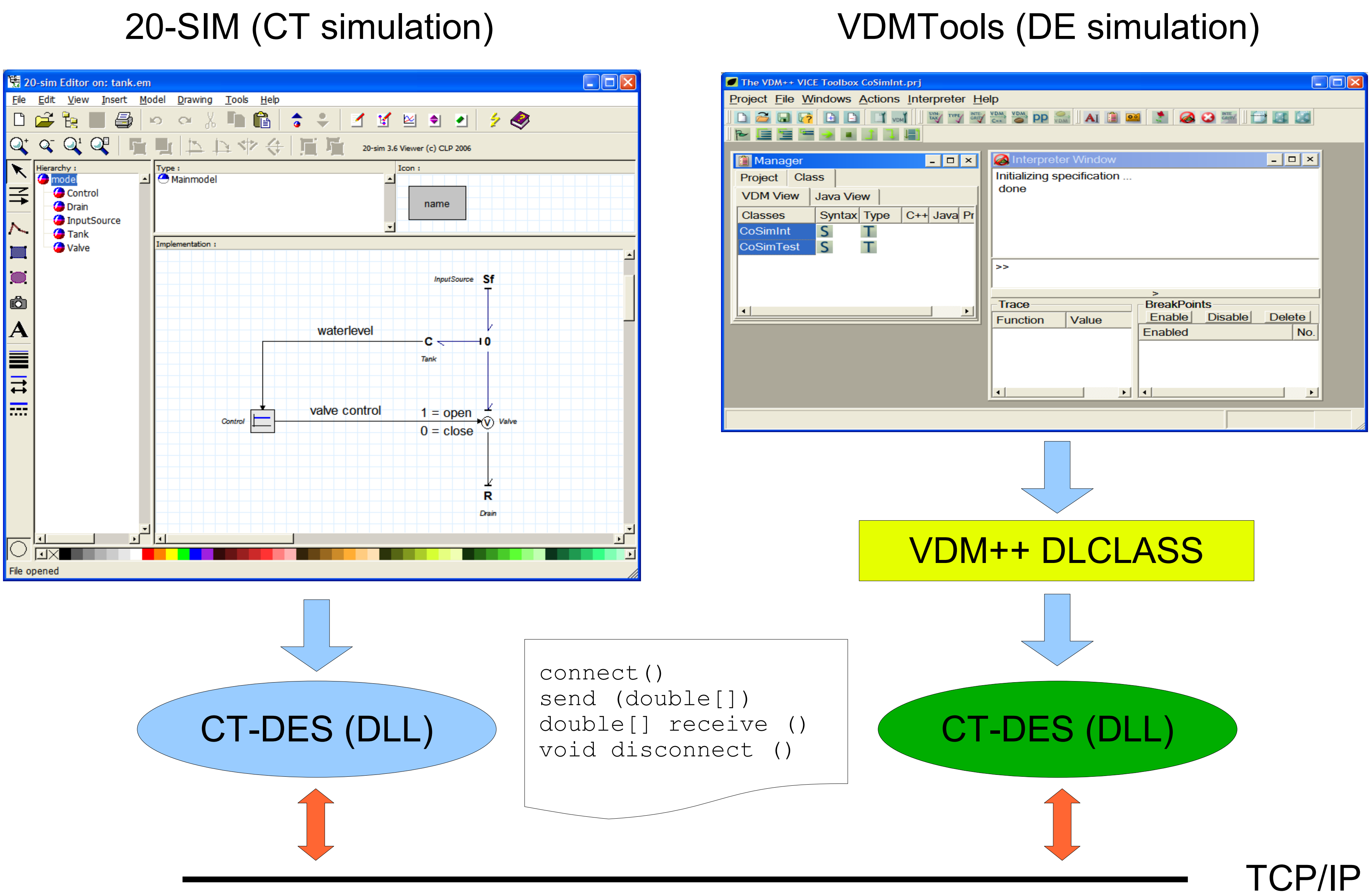


Conclusions and Future Work

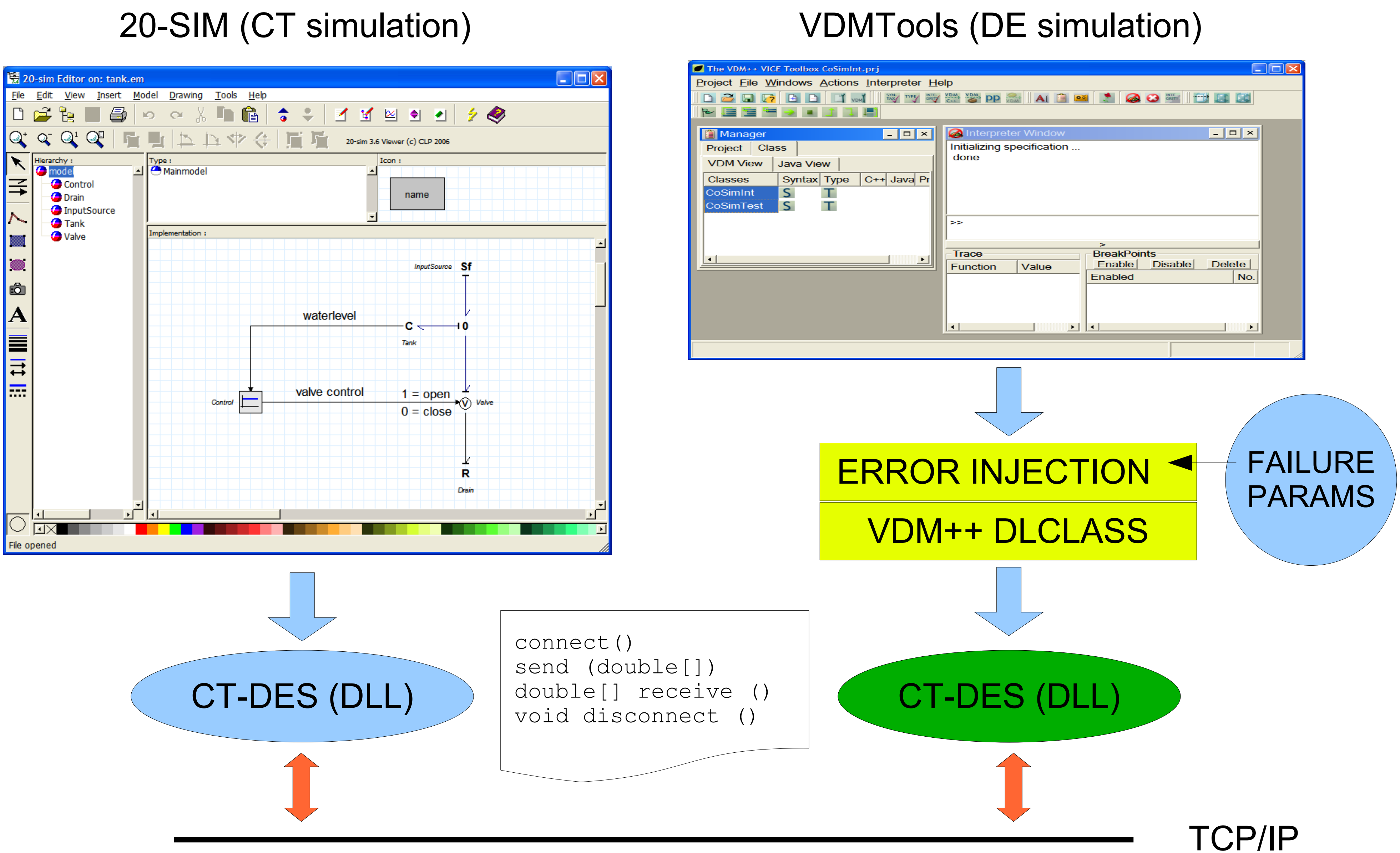
- Proof of concept successful
- Integration with new distributed timed VDM++ dynamic semantics is technically feasible
- Larger case study, involving distributed control (alignment unit for a high-volume printer)
- Extensions for failure analysis

Printer Paper Path

Modeling System Reliability (1)



Modeling System Reliability (2)



Modeling System Reliability (3)

- Start with idealized controller model in VDM
- Execute, observe correct idealized behavior
- Specify failure and error rates
- Implement failure mode handling in VDM
- Execute, observe system behavior
- Stop if satisfied else repeat step 4