Universiteit Antwerpen

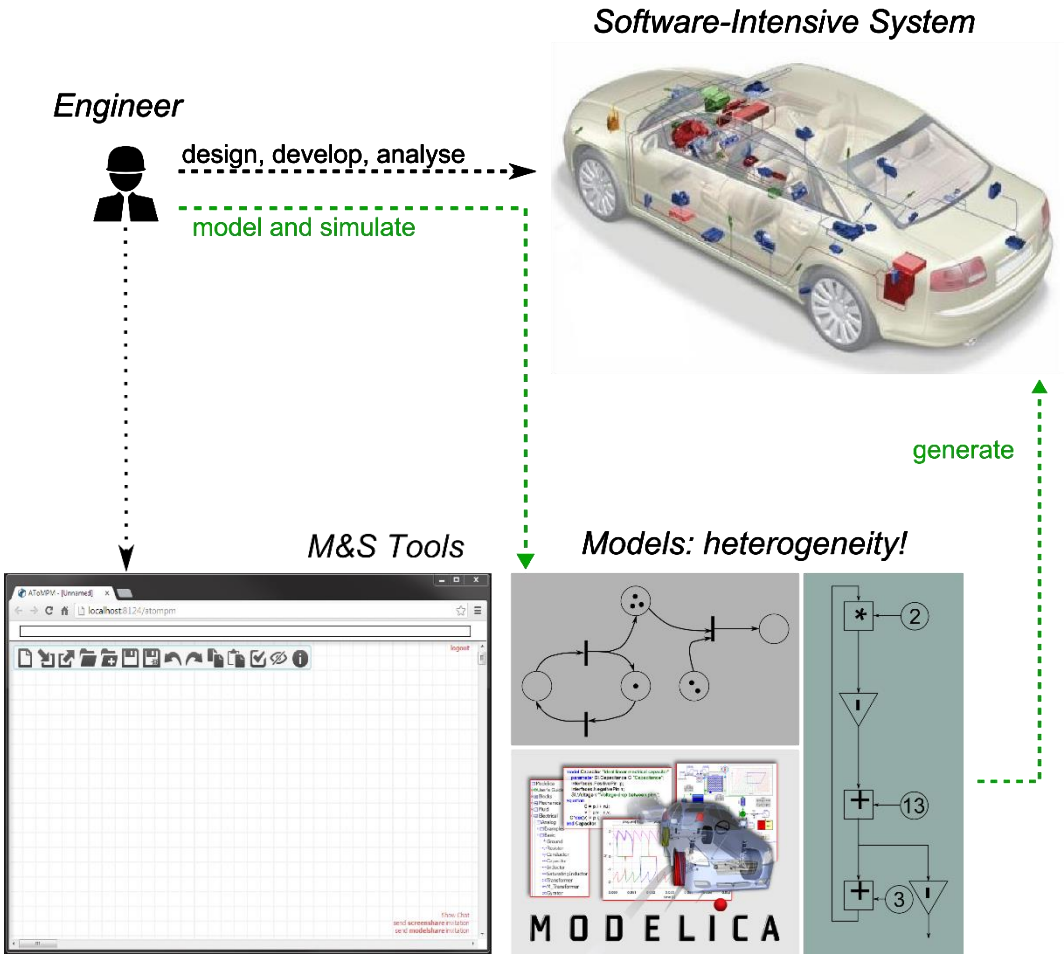
Explicit Modelling of Model Debugging Environments

Simon Van Mierlo

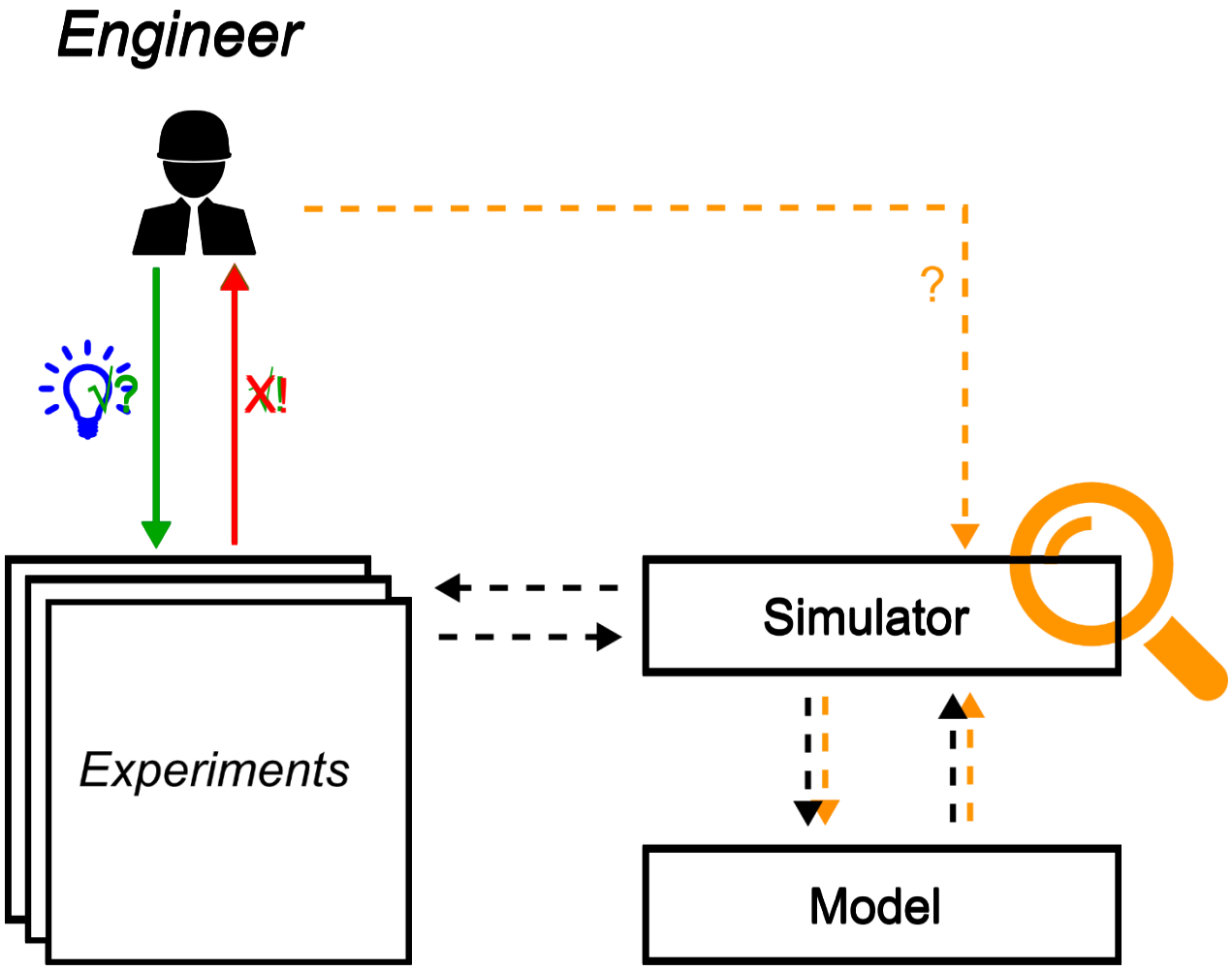
University of Antwerp

April 24, 2015

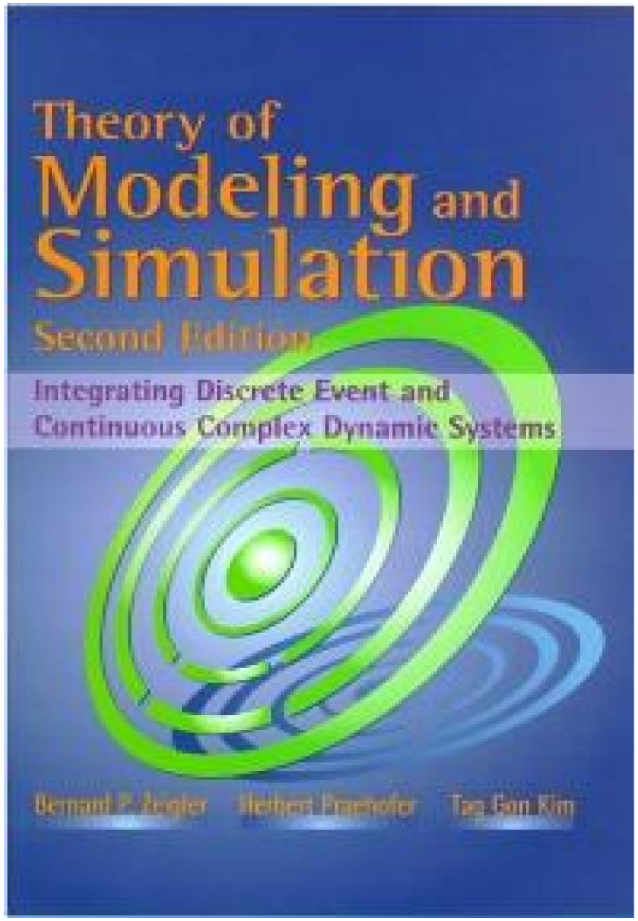
Motivation

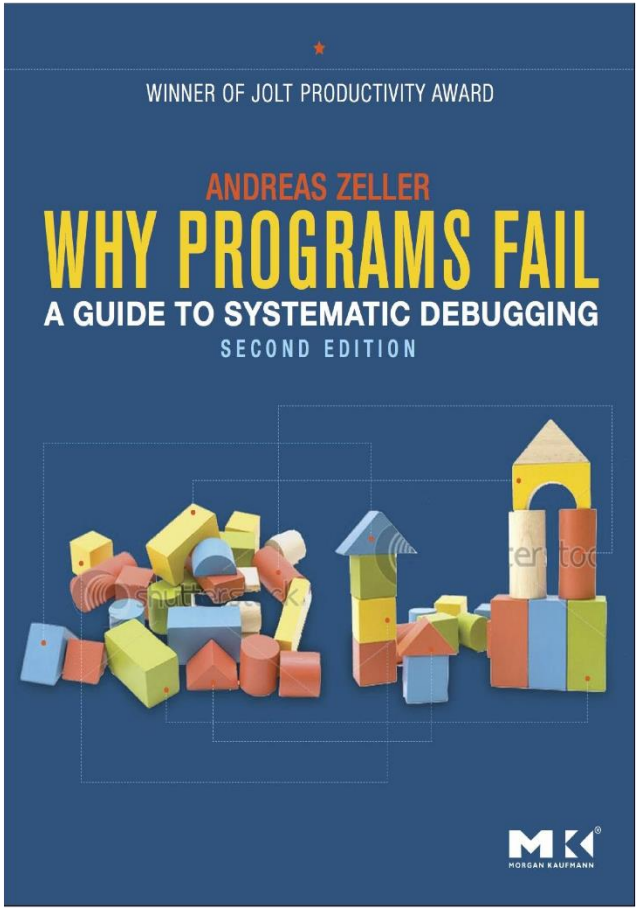

Software-Intensive System
Engineer
design, develop, analyse
model and simulate
generate
M&S Tools
Models: heterogeneity!
MODELICA

Motivation

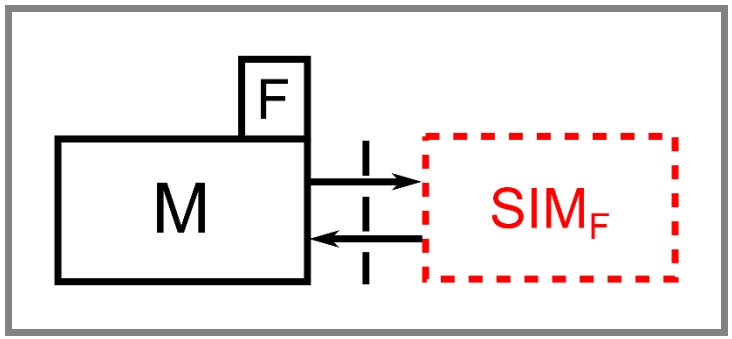


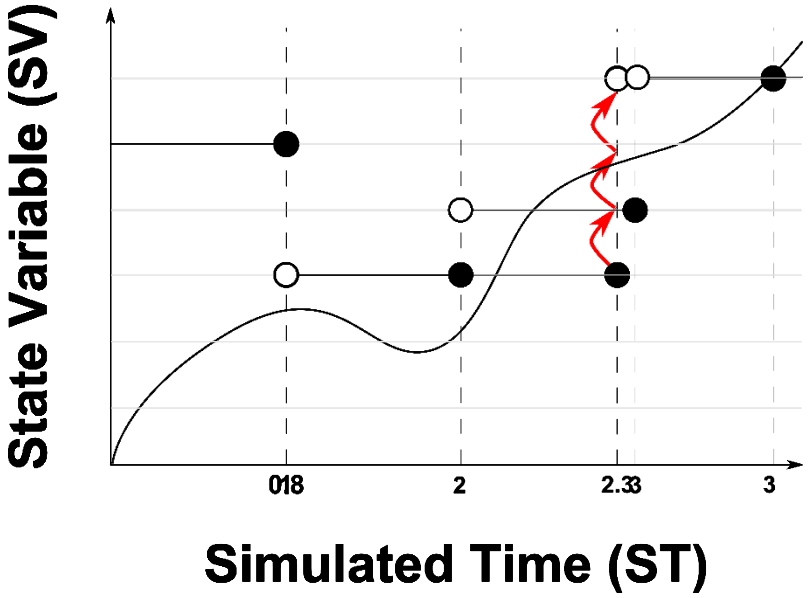
Debugging


Theory of
Modeling and
Simulation
Second Edition
Integrating Discrete Event and
Continuous Complex Dynamic Systems

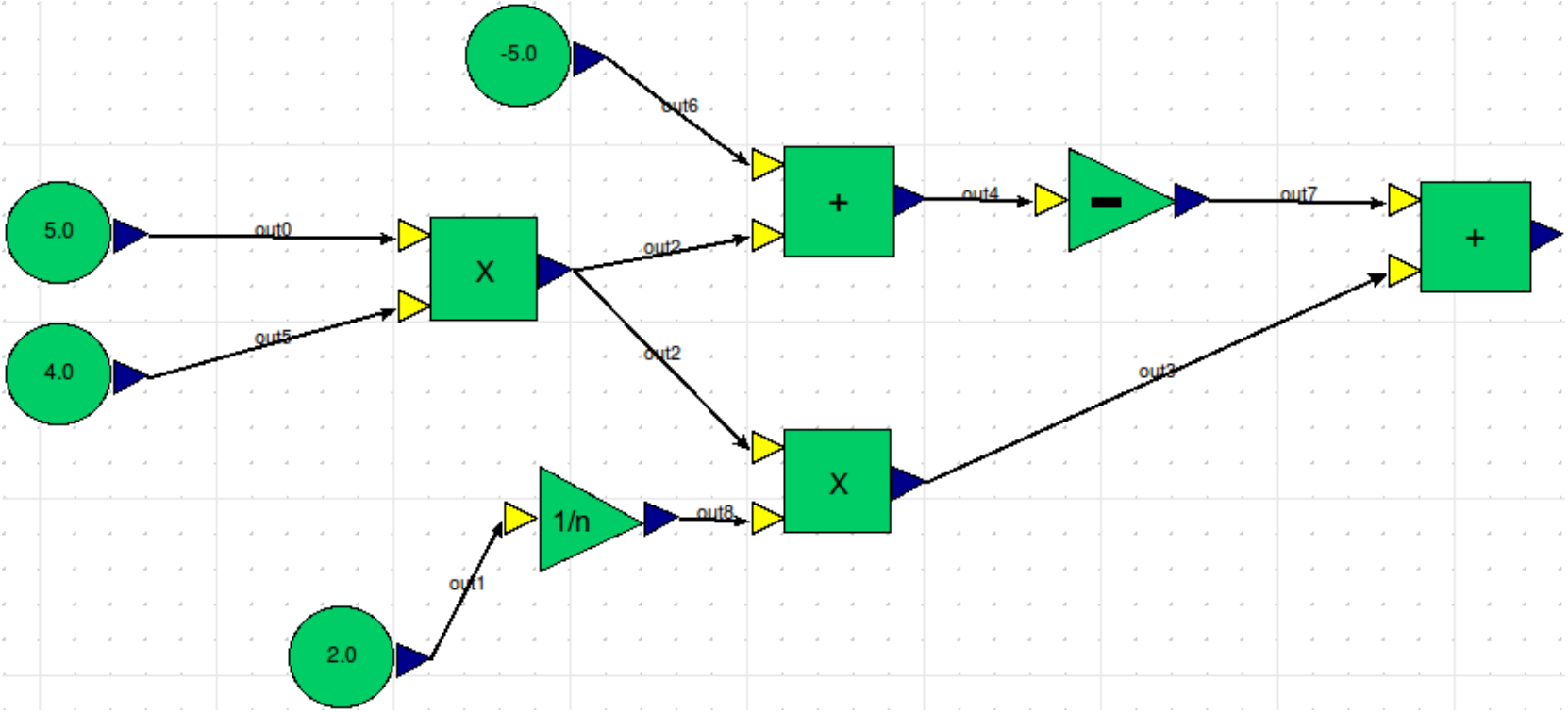

WINNER OF JOLT PRODUCTIVITY AWARD
ANDREAS ZELLER
WHY PROGRAMS FAIL
A GUIDE TO SYSTEMATIC DEBUGGING
SECOND EDITION
MORGAN KAUFMANN

Simulation





Example: CBD

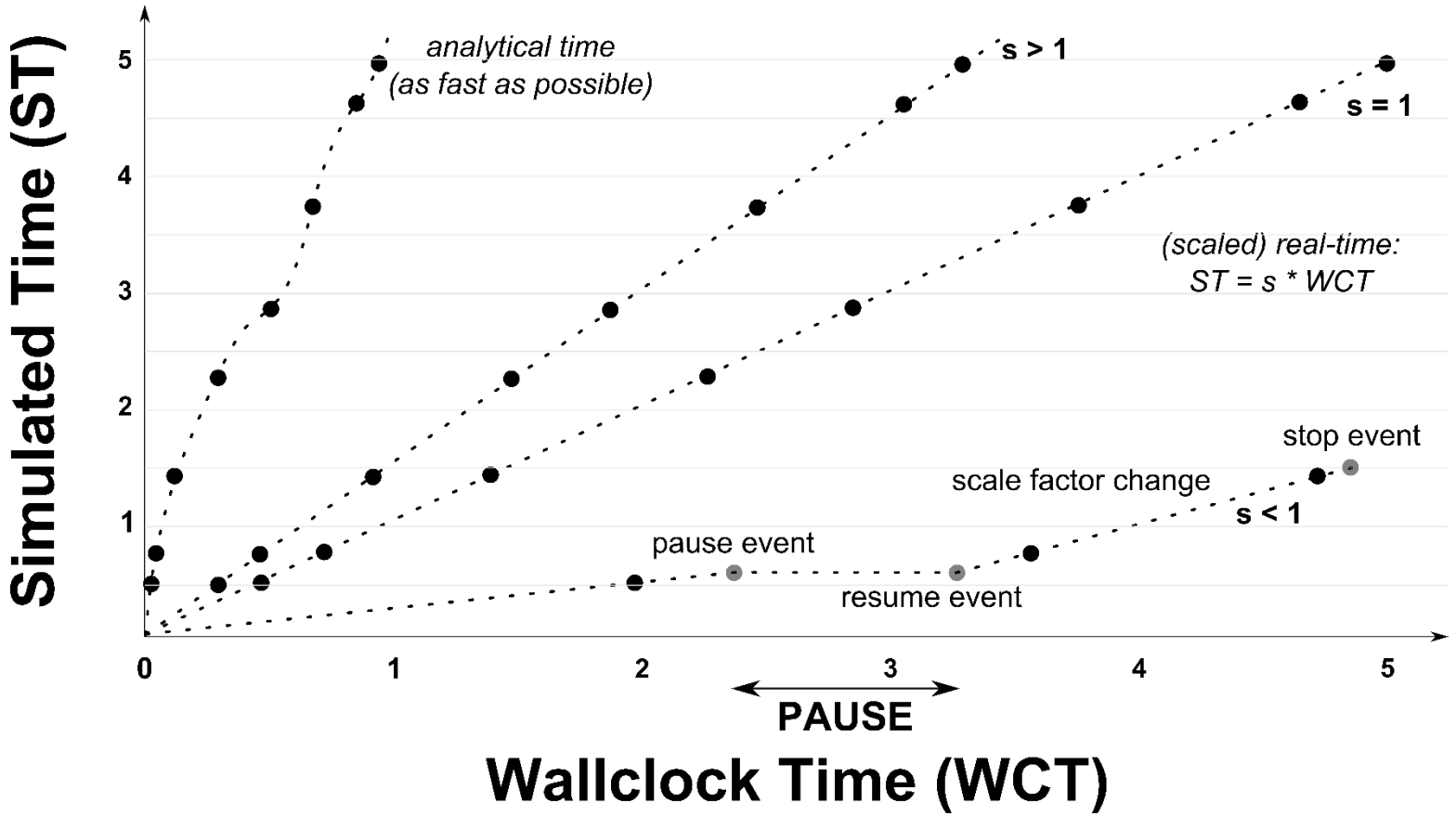

-5.0
out6
5.0
out0
X
out2
+
out4
−
out7
+
4.0
out5
out2
out3
X
1/n
out8
out1
2.0

Example: CBD

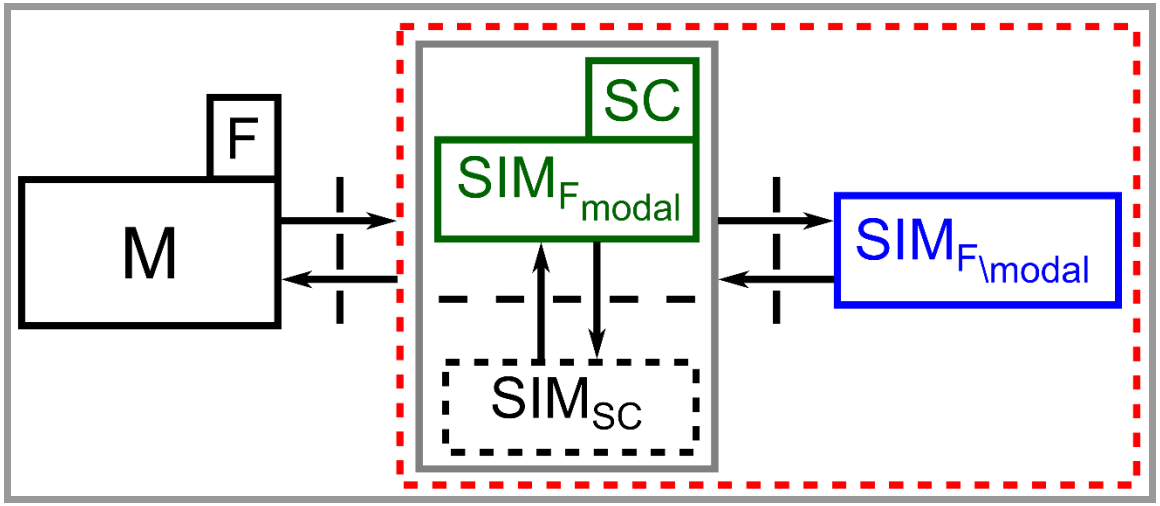
Algorithm 1 The CBD simulator's "main loop".

```
1: time ← 0
2: while not end_condition do
3:     schedule ← LOOPDETECT(DEPGRAPH(cbd))
4:     for gblock in schedule do
5:         COMPUTE(gblock)
6:     end for
7:     time ← time + δ_t
8: end while
```

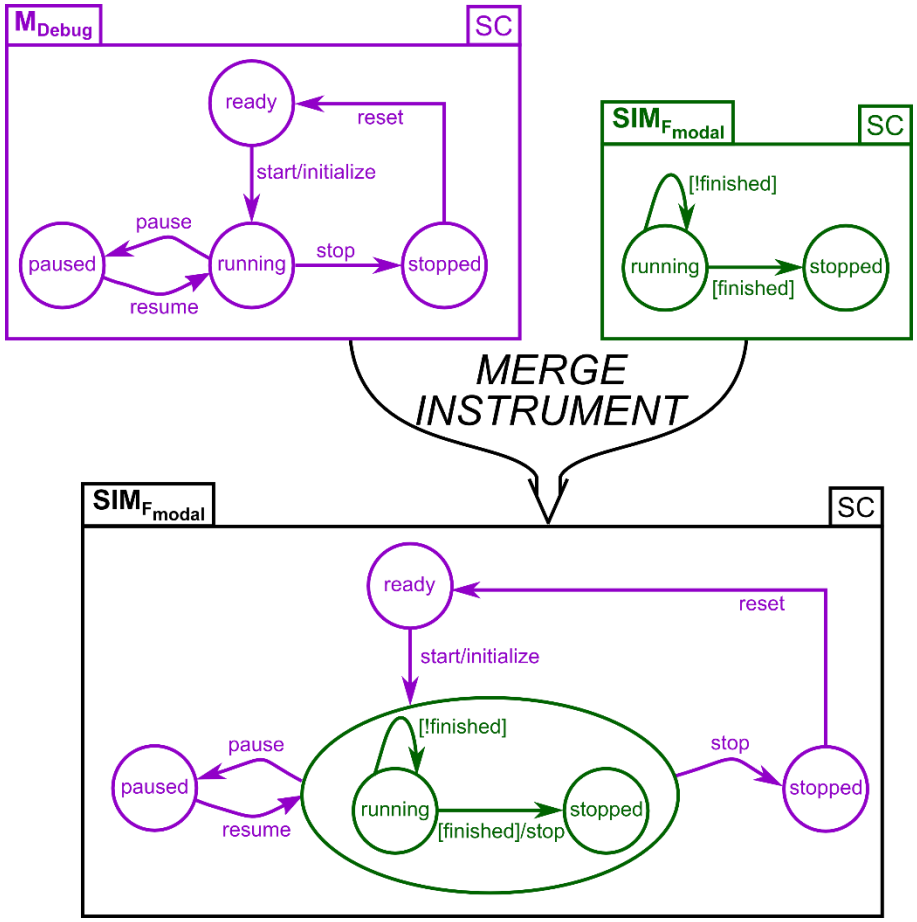
Time


Simulated Time (ST)
Wallclock Time (WCT)
analytical time
(as fast as possible)
s > 1
s = 1
(scaled) real-time:
ST = s * WCT
stop event
scale factor change
s < 1
pause event
resume event
PAUSE
0
1
2
3
4
5

De-Reconstruction



Adding Debugging



Example: CBD

Algorithm 1 The CBD simulator's "main loop".

```
1: time ← 0
2: while not end_condition do
3:     schedule ← LOOPDETECT(DEPGRAPH(cbd))
4:     for gblock in schedule do
5:         COMPUTE(gblock)
6:     end for
7:     time ← time + δ_t
8: end while
```

Example: CBD

Algorithm 2 The CBD simulator's "main loop".

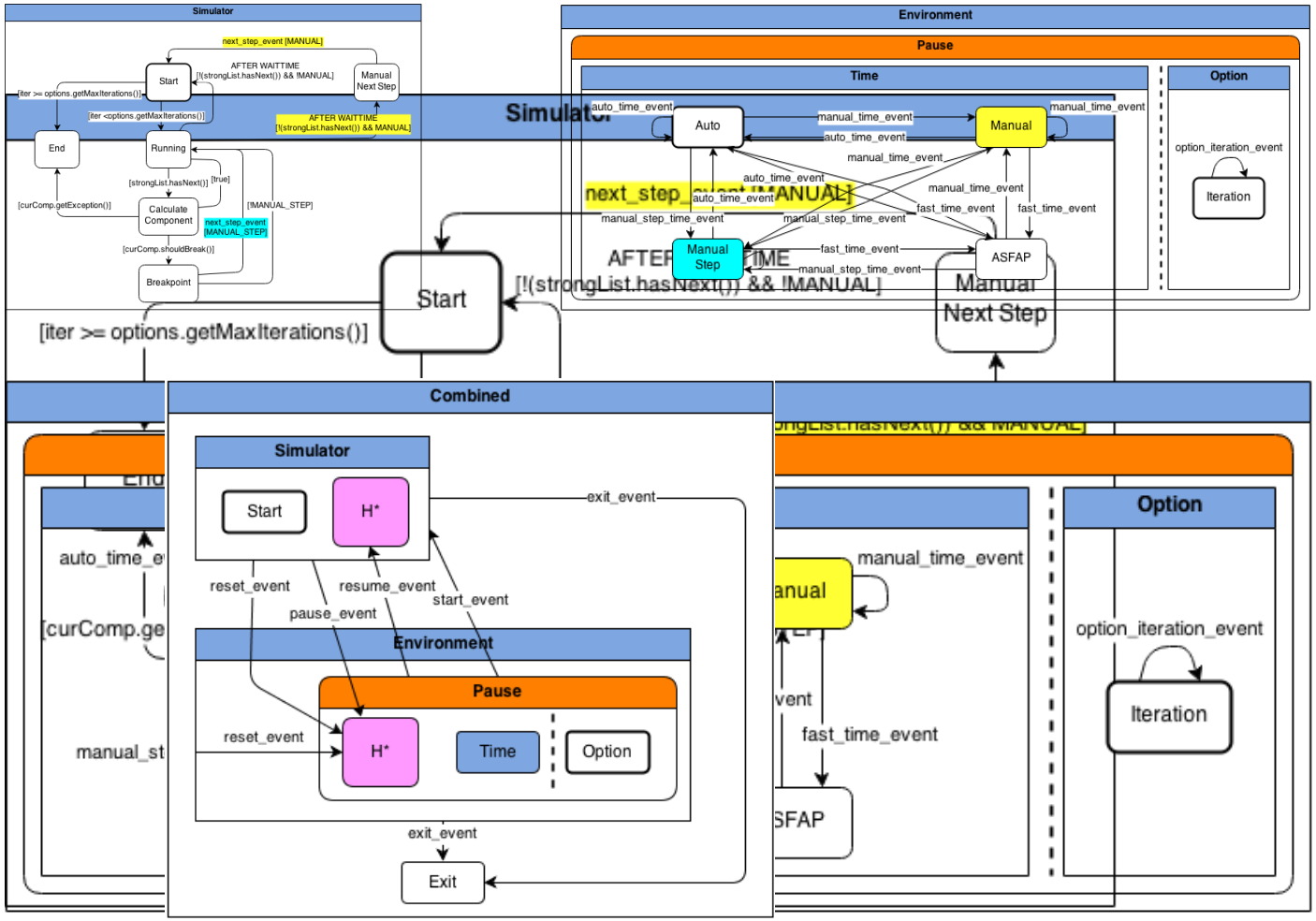
```
1: time ← 0
2: while not end_condition do
3:     schedule ← LOOPDETECT(DEPGRAPH(cbd))
4:     for gblock in schedule do
5:         COMPUTE(gblock)
6:     end for
7:     time ← time + δ_t
8: end while
```

Example: CBD

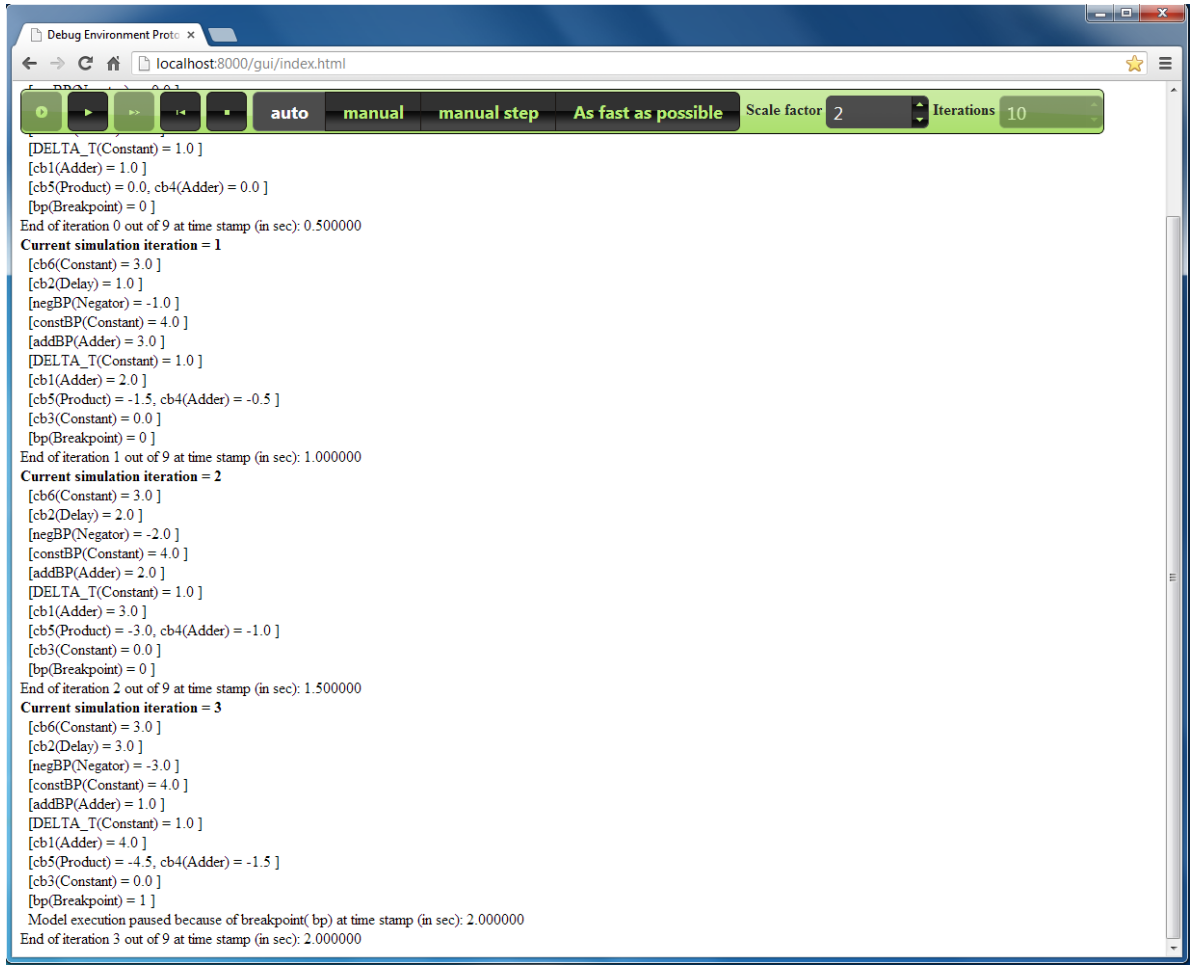
Algorithm 3 The CBD simulator's "main loop".

```
1: time ← 0
2: while not end_condition do
3:     schedule ← LOOPDETECT(DEPGRAPH(cbd))
4:     for gblock in schedule do
5:         COMPUTE(gblock)
6:     end for
7:     time ← time + δ_t
8: end while
```

Example: CBD


Simulator
Environment
Pause
Time
Option
Auto
Manual
Start
Running
End
Calculate Component
Breakpoint
Manual Step
ASFAP
Iteration
Combined
H*
Exit
reset_event
resume_event
pause_event
start_event
exit_event
option_iteration_event
manual_time_event
auto_time_event
fast_time_event
manual_step_time_event
[iter >= options.getMaxIterations()]
[!(strongList.hasNext()) && !MANUAL]
Manual Next Step

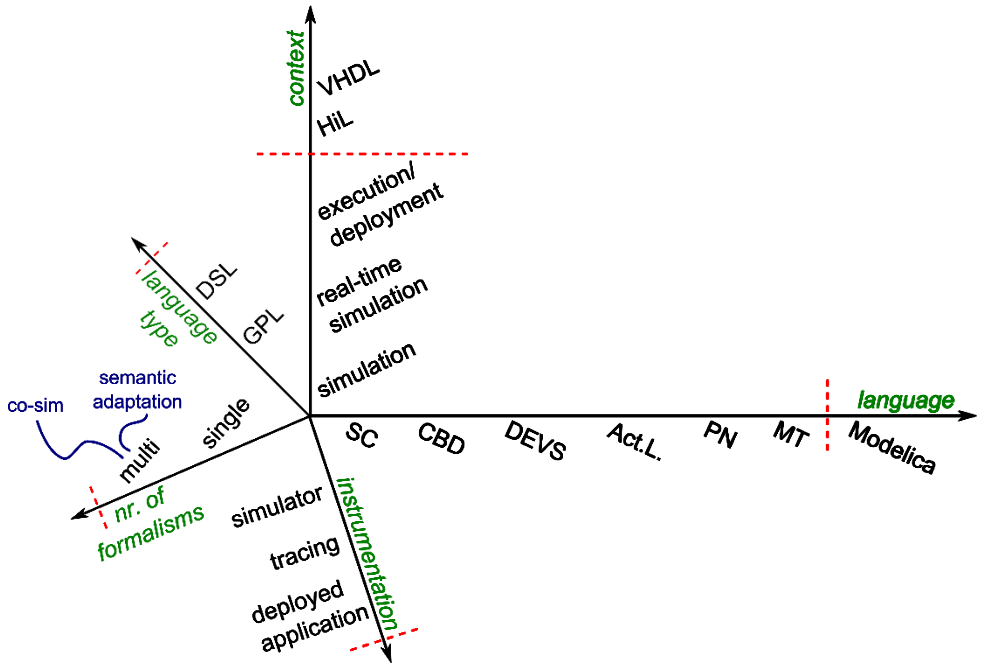
Example: CBD


Debug Environment Proto
localhost:8000/gui/index.html
auto
manual
manual step
As fast as possible
Scale factor
2
Iterations
10
[DELTA_T(Constant) = 1.0]
[cb1(Adder) = 1.0]
[cb5(Product) = 0.0, cb4(Adder) = 0.0]
[bp(Breakpoint) = 0]
End of iteration 0 out of 9 at time stamp (in sec): 0.500000
Current simulation iteration = 1
[cb6(Constant) = 3.0]
[cb2(Delay) = 1.0]
[negBP(Negator) = -1.0]
[constBP(Constant) = 4.0]
[addBP(Adder) = 3.0]
[DELTA_T(Constant) = 1.0]
[cb1(Adder) = 2.0]
[cb5(Product) = -1.5, cb4(Adder) = -0.5]
[cb3(Constant) = 0.0]
[bp(Breakpoint) = 0]
End of iteration 1 out of 9 at time stamp (in sec): 1.000000
Current simulation iteration = 2
[cb6(Constant) = 3.0]
[cb2(Delay) = 2.0]
[negBP(Negator) = -2.0]
[constBP(Constant) = 4.0]
[addBP(Adder) = 2.0]
[DELTA_T(Constant) = 1.0]
[cb1(Adder) = 3.0]
[cb5(Product) = -3.0, cb4(Adder) = -1.0]
[cb3(Constant) = 0.0]
[bp(Breakpoint) = 0]
End of iteration 2 out of 9 at time stamp (in sec): 1.500000
Current simulation iteration = 3
[cb6(Constant) = 3.0]
[cb2(Delay) = 3.0]
[negBP(Negator) = -3.0]
[constBP(Constant) = 4.0]
[addBP(Adder) = 1.0]
[DELTA_T(Constant) = 1.0]
[cb1(Adder) = 4.0]
[cb5(Product) = -4.5, cb4(Adder) = -1.5]
[cb3(Constant) = 0.0]
[bp(Breakpoint) = 1]
Model execution paused because of breakpoint(bp) at time stamp (in sec): 2.000000
End of iteration 3 out of 9 at time stamp (in sec): 2.000000

Conclusion

- Complexity of simulation experimentation environments
- Reactive and autonomous, so model explicitly, using Statecharts
- Succesfully applied to CBDs and Parallel DEVS (and Statecharts before)

Future Work


context
VHDL
HiL
execution/
deployment
real-time
simulation
simulation
language
type
DSL
GPL
semantic
adaptation
co-sim
multi
single
nr. of
formalisms
SC
CBD
DEVS
Act.L.
PN
MT
language
Modelica
simulator
tracing
deployed
application
instrumentation