Simulation Techniques in Metropolis

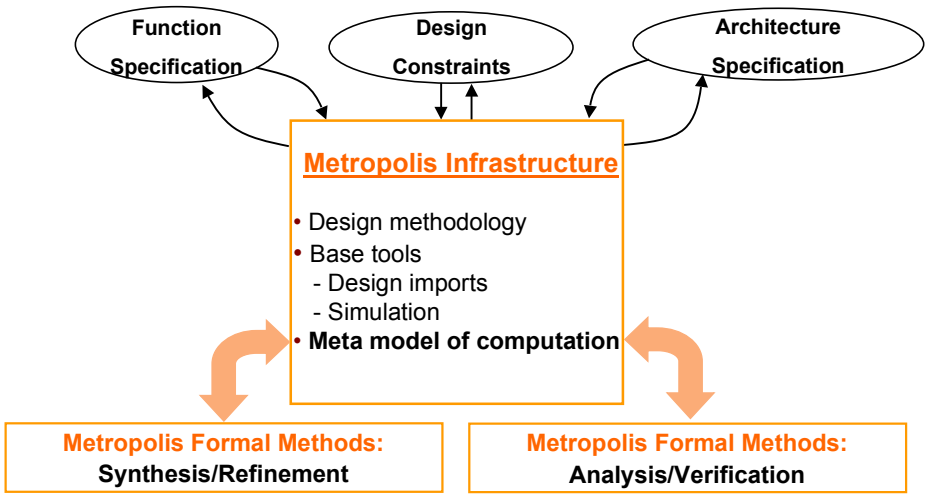
Claudio Passerone

DUSD(S&T)

Outline

- Simulation in the Metropolis Infrastructure
- Simulation strategy and algorithm
- Current status
- What's missing?
- Demo

Simulation in the Metropolis Infrastructure



June 18, 2001 3

Simulation strategy and algorithm

- Multithreaded concurrent simulation
  - a thread for each process
  - function calls for communication media interfaces
- Managing the simulation
  - process threads run until constraints or **await** need to be evaluated
  - a *manager* allows only selected processes to proceed
    - high abstraction level: satisfy constraints (using schedulers as help)
    - low abstraction level: verify properties
- Prototype written in Java 1.2

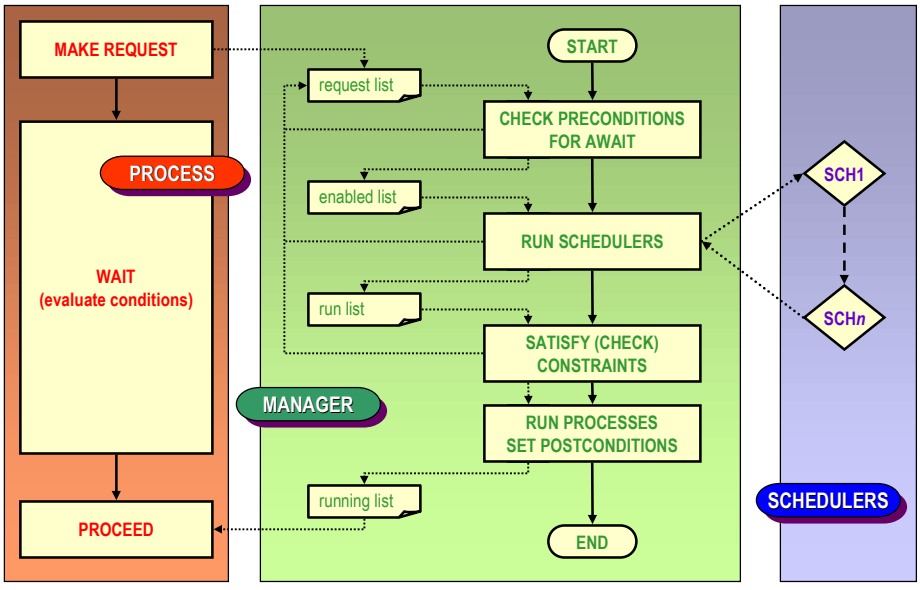
June 18, 2001 4

Simulation strategy and algorithm

- **Processes**
  - instances of a specialized class of **process**
  - **thread** ⇒ **run** function of a thread
  - **port** ⇒ variable (type is an interface)
- **Communication Media**
  - instances of a specialized class of **medium**
  - implement interfaces using member functions
- **await**
  - translated into a simulation model, interacting with the manager

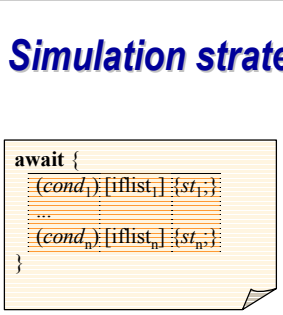
June 18, 2001 5

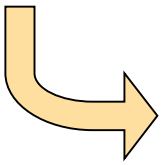
Simulation strategy and algorithm



June 18, 2001 6

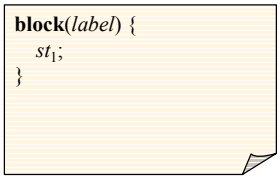
Simulation strategy and algorithm

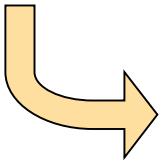




```
{
  Lock[][] lock = {{iflist1}, ..., {iflistn}};
  setProgramCounter(pc);
  myManager.addRequest(pc, lock, await);
  do {
    thread.wait();
    boolean[] cond = {cond1, ..., condn};
    myManager.notify();
  } while (state != RUNNING);
  selected = myManager.getSelected();
}
switch(selected) {
  case 1: st1; break;
  ....
  case n: stn; break;
}
{
  setProgramCounter(pc);
  myManager.endRequest(pc);
}
```

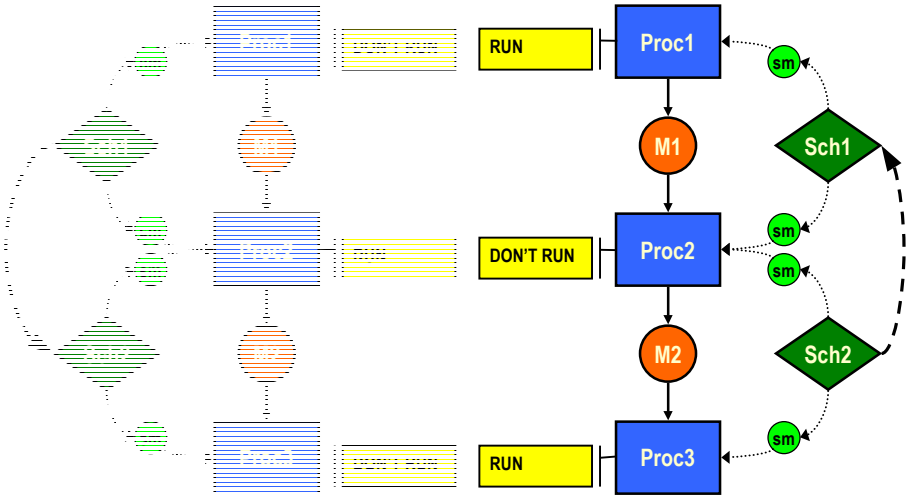

Simulation strategy and algorithm





```
{
  setProgramCounter(pc);
  myManager.addRequest(pc, constraint);
  do {
    thread.wait();
  } while (state != RUNNING);
}
st1;
{
  setProgramCounter(pc);
  myManager.endRequest(pc);
}
```

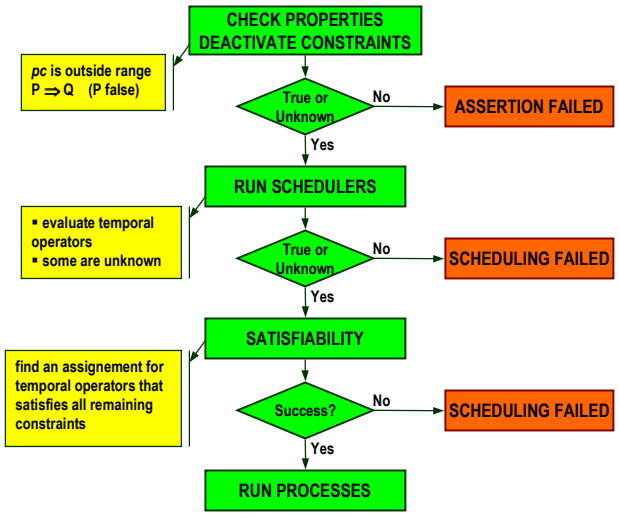
Simulation strategy and algorithm



June 18, 2001 9

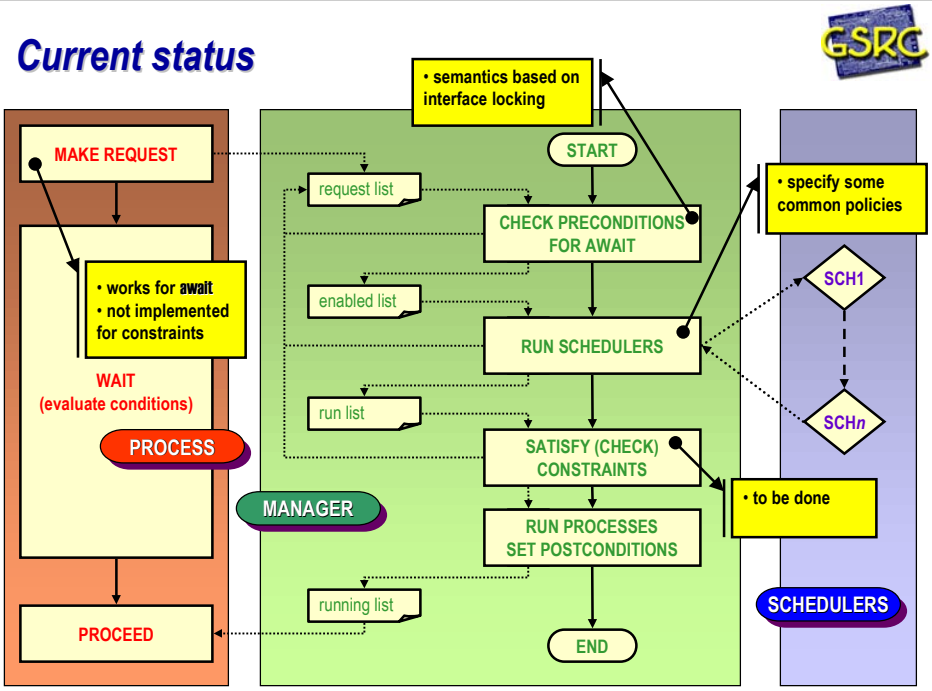
Simulation strategy and algorithm

CONSTRAINTS	
$proposition_1$	True
$proposition_2$	True
$proposition_3$	True
$proposition_4$	True
$proposition_5$	True
$proposition_6$	True
$proposition_7$	True
$proposition_8$	True
$proposition_9$	True
$proposition_{10}$	True
$proposition_{11}$	True
$proposition_{12}$	True
$proposition_{13}$	True
$proposition_{14}$	True



June 18, 2001 10

Current status



June 18, 2001 11

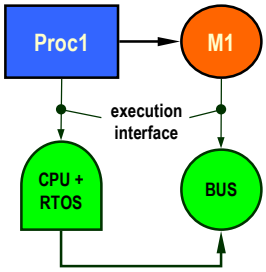
What's missing?

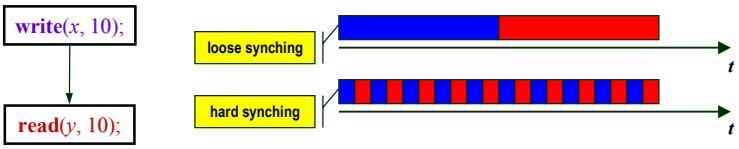
- Automatic translation from meta-model to simulation
  - meta-model parser, macro expansion, ...
- Support for refinement
  - generate entire new netlist and simulate
- Support for architecture
  - architectural elements are specified in the meta-model
    - processes (RTOS schedulers)
    - media (Bus)
  - architecture introduces time
    - need for synchronization

June 18, 2001 12

What's missing?

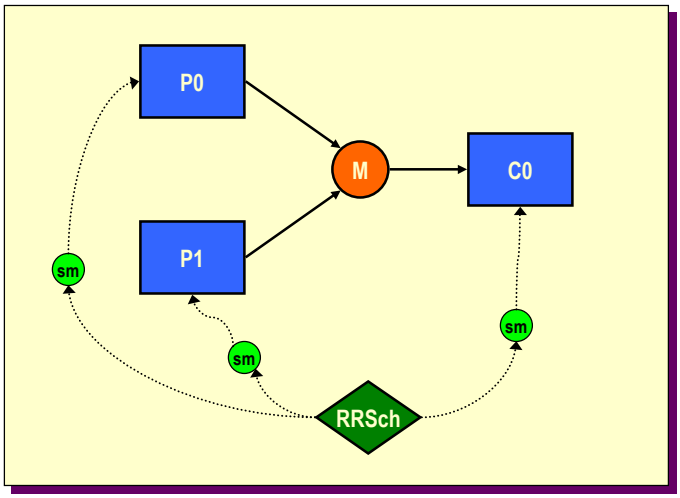
- Two primitives for architectures
  - delay(annotation)
    - interface between functional elements and architectural elements
    - may carry any information, but ultimately it's time
  - synch()
    - synchronize all annotations with a global time





June 18, 2001 13

Demo



June 18, 2001 14