A Distributed-Discrete Event Simulation Testbed

Bruno R. Preiss

University of Waterloo
Department of Electrical Engineering
Waterloo, Ontario, N2L 3G1

Systems on Silicon

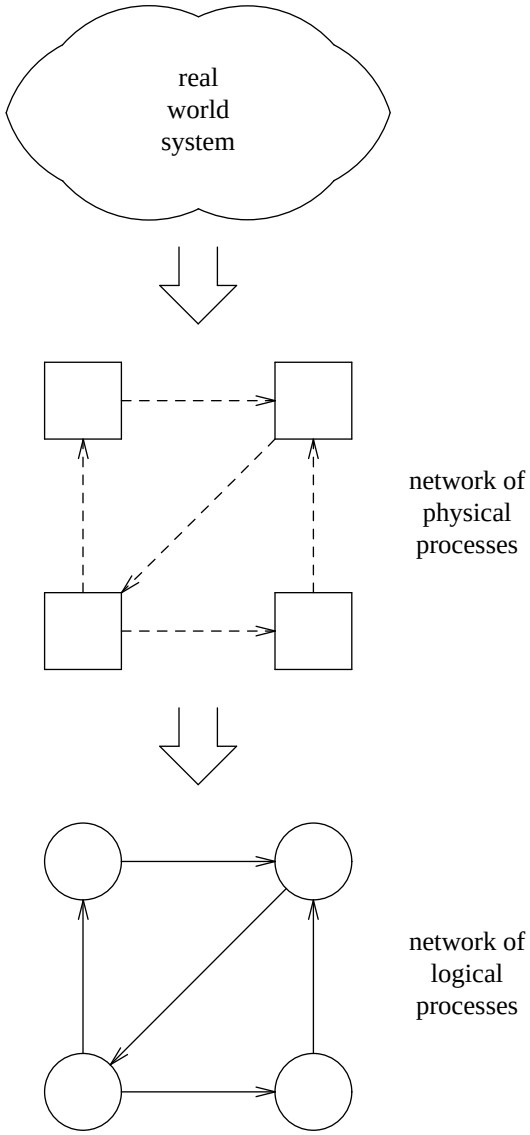
Motivation

- **parallel processor systems**
- **tightly-coupled**
- **message-passing rather than shared memory**
- **fine process granularity**
- **fine communications granularity**
- **ambitious processes [Spektor]**

Approach

- **application oriented**
- **special purpose rather than general purpose**
- **need a large amount of inherent, irregular parallelism**
- **distributed-discrete event simulation [Chandy-Misra]**
- **logic simulation, system simulation**

Distributed-Discrete Event Simulation



Systems on Silicon

Real World System

- **large amount of inherent, irregular parallelism**
- **many independent but interacting components**

Model

- **consists of** *physical processes*
- **physical processes periodically exchange information**
- **information exchange occurs at discrete points in time —** *events*

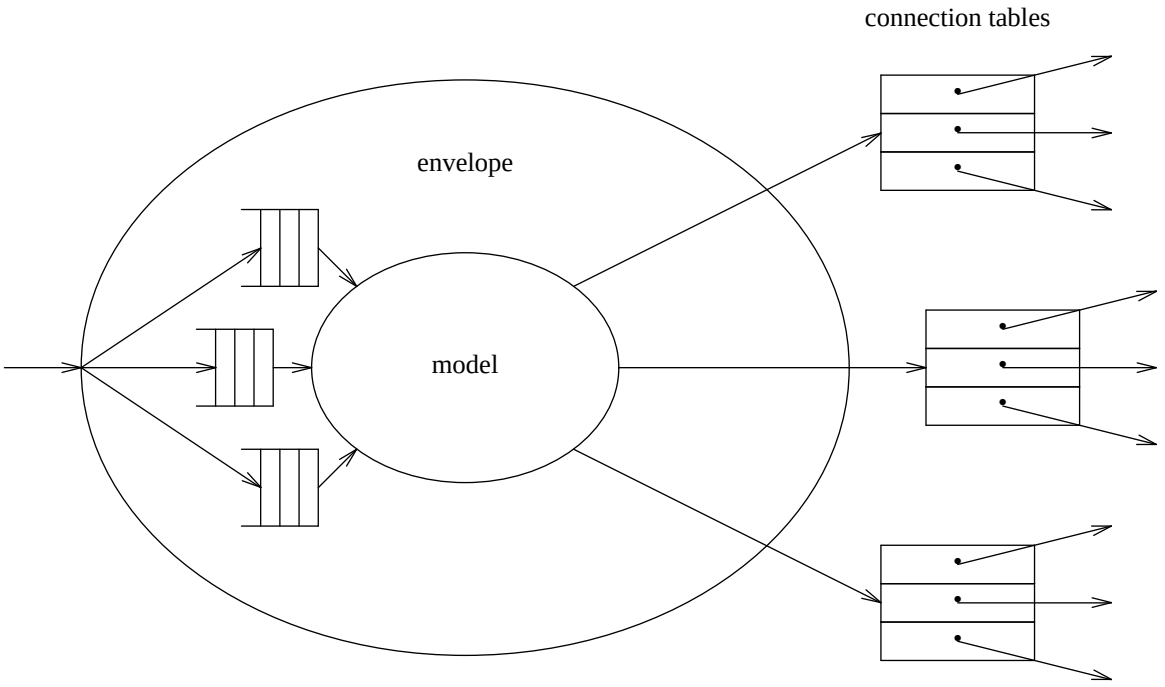
Simulation

- **consists of** *logical processes*
- **communicating sequential processes [Hoare]**
- **events are** *time-stamped messages*

Models

- **a formalism for specifying a distributed-discrete event simulation**
- **a** *model* **describes the behaviour of a physical process**
- **a model** *specifies* **the behaviour of a logical process**
- **logical processes are instantiations of models**
- **each logical process has a unique state** *state*
- **models describe the state** *transitions* **that occur in response to events**
- **state transitions are called** *actions*
- **actions are invoked by event** *combinations*
- **an event combination is a set of** *simultaneous* **events**
- **actions may generate other events**
- **events are messages sent to other logical processes (model instantiations)**

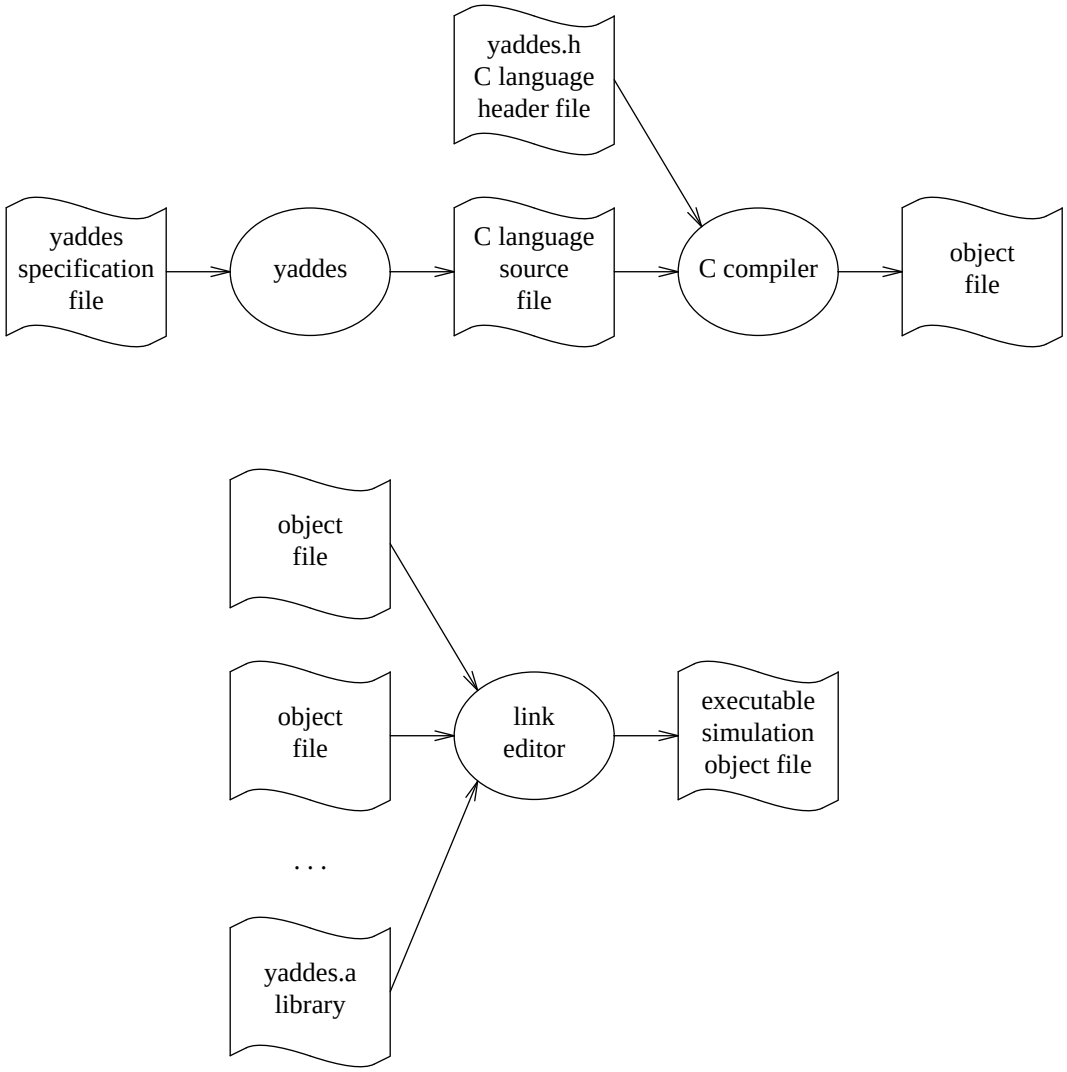
The Envelope



The Envelope (continued)

- **a logical process consists of an envelope and a model instantiation**
- **the envelope isolates the model from the simulation environment**
- **the envelope receives messages from other logical processes**
- **the envelope has a separate queue for each model input**
- **the envelope constructs event combinations and passes them to the model**
- **the model sends messages to other logical processes via the connection table**

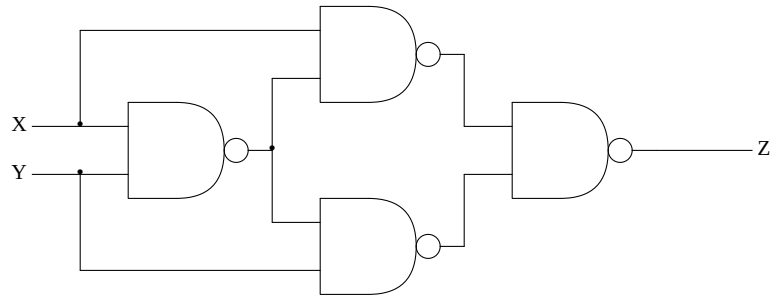
Yaddes (Yet Another Distributed-Discrete Event Simulator): System Overview



Yaddes Specification Example: Two Input NAND Gate

```
%%
model TwoInputNand
    inputs in0, in1
    outputs out
    state
    {
      int input0;
      int input1;
    }
    initial state { 0, 0 }
    action initial
    {
      OUTPUT ($out, $time + 10,
          NAND ($state->input0, $state->input1));
    }
    action in0
    {
      $state->input0 = $event [$in0];
      OUTPUT ($out, $time + 10,
          NAND ($state->input0, $state->input1));
    }
    action in1
    {
      $state->input1 = $event [$in1];
      OUTPUT ($out, $time + 10,
          NAND ($state->input0, $state->input1));
    }
    action in0, in1
    {
      $state->input0 = $event [$in0];
      $state->input1 = $event [$in1];
      OUTPUT ($out, $time + 10,
          NAND ($state->input0, $state->input1));
    }
    action default
    {
    }
    action final
    {
    }
end model
```

Yaddes Specification Example: XOR Circuit




```
%%
external model Input
    inputs none
    outputs out
    state
    {
      FILE *fp;
    }
    initial state { NULL }
end model

external model Output
    inputs in
    outputs none
    state
    {
      FILE *fp;
    }
    initial state { NULL }
end model
```

Yaddes Specification Example: XOR Circuit (continued)

```
external model TwoInputNand
    inputs in0, in1
    outputs out
    state
    {
      int input0;
      int input1;
    }
    initial state { 0, 0 }
end model

process X : Input
process Y : Input
process Gate0 : TwoInputNand
process Gate1 : TwoInputNand
process Gate2 : TwoInputNand
process Gate3 : TwoInputNand
process Z : Output

connect X.out to Gate0.in0, Gate1.in0
connect Y.out to Gate0.in1, Gate2.in1
connect Gate0.out to Gate1.in1, Gate2.in0
connect Gate1.out to Gate3.in0
connect Gate2.out to Gate3.in1
connect Gate3.out to Z.in
```

Uniprocessor Experiments

- **fully distributed simulation using a multitasking kernel (working)**
- **fully serial simulation based on a global timeline (working)**
- **distributed simulation based on multiple timelines with global synchronization (work in progress)**
- **distributed simulation based on multiple timelines with distributed time (work in progress)**

Multiprocessor Experiments

- **fully distributed simulation (future)**
- **distributed simulation based on multiple timelines with global synchronization (future)**
- **distributed simulation based on multiple timelines with distributed time (future)**

Future Directions

- **partitioning of the network of logical processes**
- **allocation of logical processes to processors**
- **investigation of architectures for simulation**
- **investigation of hardware assists for messages passing**
- **investigation of hardware assists for the envelope**