Simulation in Cyber Security

Andres Ojamaa

Institute of Cybernetics, Tallinn University of Technology

CS Theory Days, 25 Jan 2008, Põlva

Outline

What is this thing called Cyber Security?

- Information security
  - Confidentiality, Availability, Integrity
  - ... also Forensics, Auditing, Disaster recovery, ...
- Information assurance
- Cyber security
- Cyber defence

Laws? Military?

Simulation Needs and Tasks

- Education
- Planning and design of computer networks
- Application performance analysis
- Simulation of attacks on attack trees
- Simulation of worms, viruses
- Denial of Service, hardware failures
- Network models
- Visualization

Simulation: Motivation

- Learn by playing "what-if" games
- Optimize budget
- Discover design flaws earlier
- Visualize fast and invisible processes
- Save resources

Simulation: Shortcomings

- Not always reliable
- Models can be expensive to build and maintain
- Not trivial to get it right: performance, accuracy, level of abstraction
- There is no single tool to answer all questions
- The Internet is constantly changing
- Hard to get real data, adaptivity of network protocols

Discrete Event Simulation

Most of the simulators are just class libraries and frameworks to build the simulation program upon.

- Infrastructure: tracing, events, objects, connections, ...
- Event queue and queue manager
- Class libraries of standard components: clocks, protocol implementations, traffic generators
- Tools for postprocessing and visualizing the traces (Network ANimator)

ns-2

Ns-2 is a discrete event simulator targeted at networking research.

- Supports: TCP, routing, ...
- Implemented in C++, Tcl
- Portable, free software
- Single threaded, no distributed coputations
- Separate tools for model construction and processing the output

Using ns-2: hello, world

```
% set ns [new Simulator]
% $ns at 1 "puts \"hello, world\""
% $ns at 2 "exit"
% $ns run
hello, world
```

Basic ns-2 Simulation

- Create scheduler
- Build network and connections
- Generate traffic
- Analyze traces

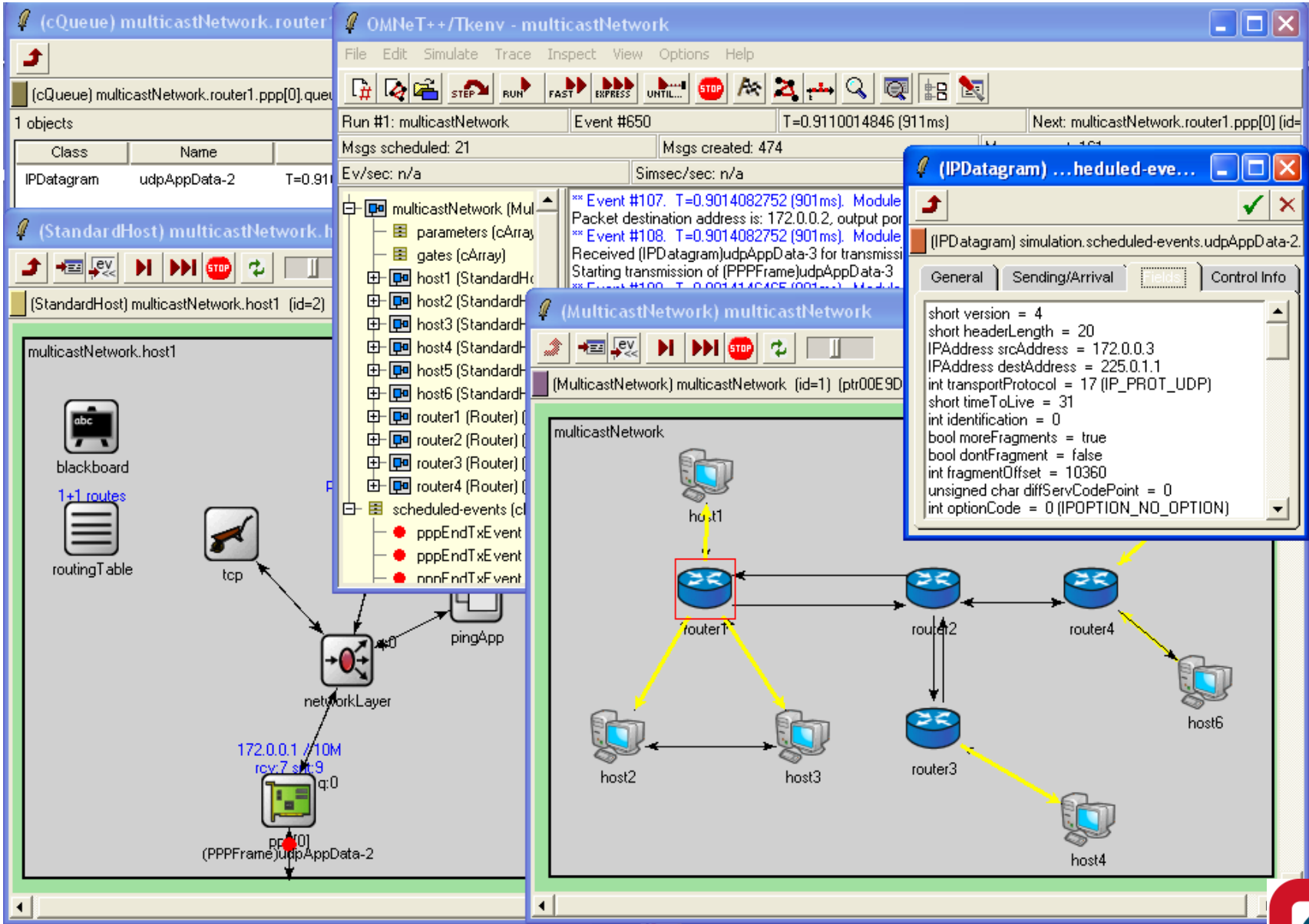
Ns-2 Script Examples

```
set n0 [$ns node]
$ns duplex-link $n0 $n1 5Mb 2ms DropTail
set tcp [$ns create-connection
      TCP $n0 TCPSink $n1 0]
set ftp [new Application/FTP]
$ftp attach-agent $tcp
```

OPNET®

- Really fancy: integrated software and hardware appliances
- Graphical user interface
- Tools for various tasks: design and planning, auditing, monitoring
- Lots of pre-built components
- Really expensive

OMNeT++ [3]

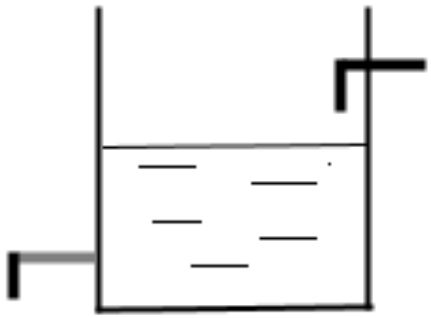

(cQueue) multicastNetwork.router1
(cQueue) multicastNetwork.router1.ppp[0].queue
1 objects
Class
Name
IPDatagram
udpAppData-2
(StandardHost) multicastNetwork.host1
(StandardHost) multicastNetwork.host1 (id=2)
multicastNetwork.host1
blackboard
1+1 routes
routingTable
tcp
pingApp
networkLayer
172.0.0.1 /10M
rcv:7 snt:9
q:0
(PPPFrame)udpAppData-2
OMNeT++/Tkenv - multicastNetwork
File Edit Simulate Trace Inspect View Options Help
STEP RUN FAST EXPRESS UNTIL... STOP
Run #1: multicastNetwork
Event #650
T=0.9110014846 (911ms)
Next: multicastNetwork.router1.ppp[0] (id=
Msgs scheduled: 21
Msgs created: 474
Ev/sec: n/a
Simsec/sec: n/a
multicastNetwork (Mul
parameters (cArray
gates (cArray)
host1 (StandardH
host2 (StandardH
host3 (StandardH
host4 (StandardH
host5 (StandardH
host6 (StandardH
router1 (Router)
router2 (Router)
router3 (Router)
router4 (Router)
scheduled-events (cl
pppEndTxEvent
pppEndTxEvent
** Event #107. T=0.9014082752 (901ms). Module
Packet destination address is: 172.0.0.2, output por
** Event #108. T=0.9014082752 (901ms). Module
Received (IPDatagram)udpAppData-3 for transmissi
Starting transmission of (PPPFrame)udpAppData-3
(MulticastNetwork) multicastNetwork
(MulticastNetwork) multicastNetwork (id=1) (ptr00E9D
multicastNetwork
host1
router1
router2
router4
host6
host2
host3
router3
host4
(IPDatagram) ...heduled-eve...
(IPDatagram) simulation.scheduled-events.udpAppData-2
General
Sending/Arrival
fields
Control Info
short version = 4
short headerLength = 20
IPAddress srcAddress = 172.0.0.3
IPAddress destAddress = 225.0.1.1
int transportProtocol = 17 (IP_PROT_UDP)
short timeToLive = 31
int identification = 0
bool moreFragments = true
bool dontFragment = false
int fragmentOffset = 10360
unsigned char diffServCodePoint = 0
int optionCode = 0 (IPOPTION_NO_OPTION)

Rich Components

Rich Components — Concepts of (simulation) domains

- *Visual representation* for visual programming
- *Logical part* for abstract properties
- *Program component* implementing computation algorithms
- *Daemon part* for interactive properties

Rich Component: Example



```
1  public class BoilerDaemon
2                implements RunnableDaemon,... {
3     public void run() {
4        ...
5     }
6  }
```

```
1  public class Boiler {
2     /*@ specification Boiler {
3        double inFlow, outFlow;
4        double q, T, minQ, qinit;
5        void done;
6        q = qinit + inFlow - outFlow;
7        q, minQ -> done {test};
8     }@*/
9
10    public void test(double amount, double min) {
11       if (amount < min)
12          beep();
13    }
14 }
```


Workflow in CoCoViLa

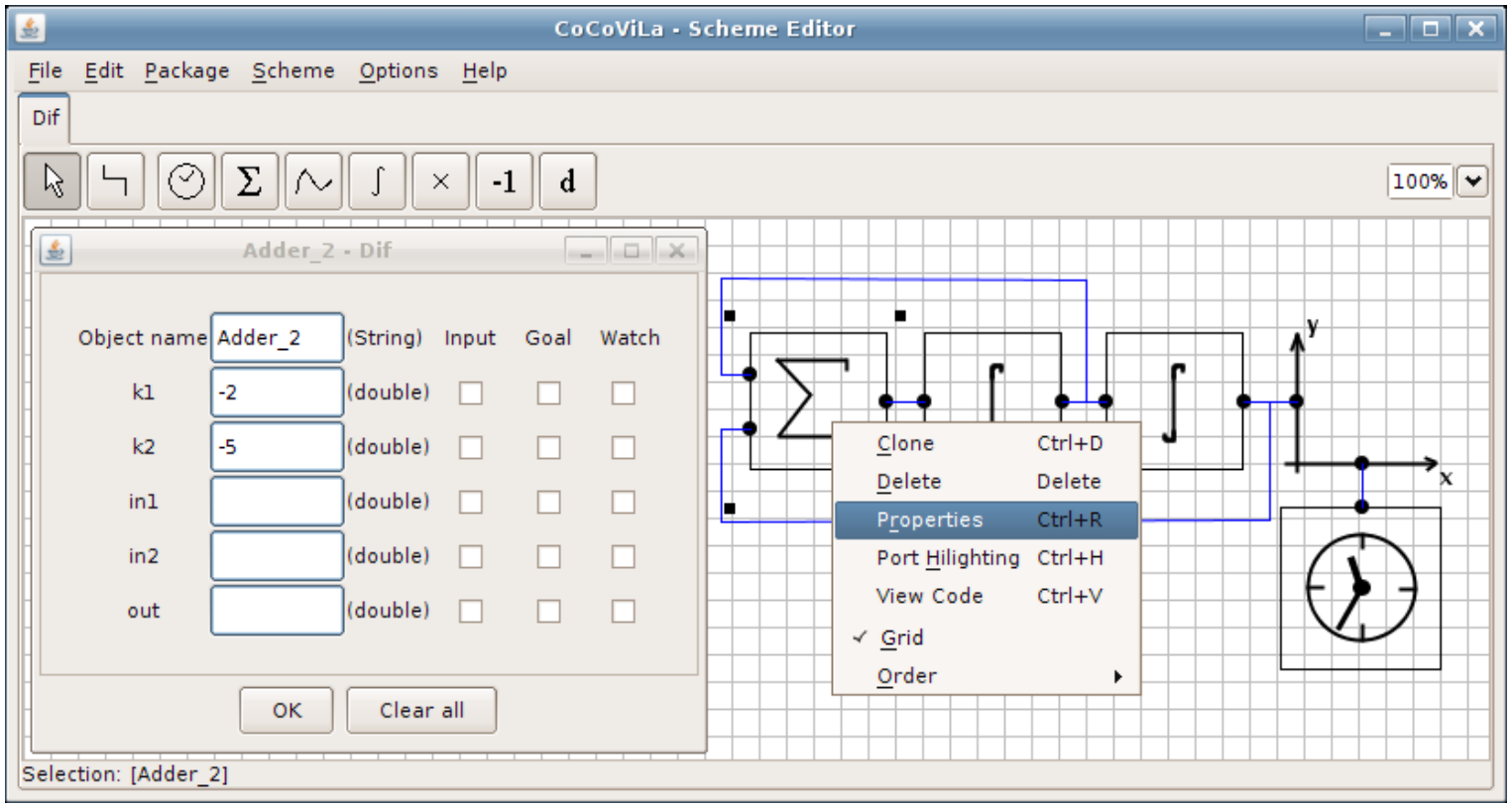
1. **Scheme**
2. Textual specification
3. Internal representation
4. Proof = Algorithm
5. Java program
6. **Output**

Interactivity in CoCoViLa Simulations

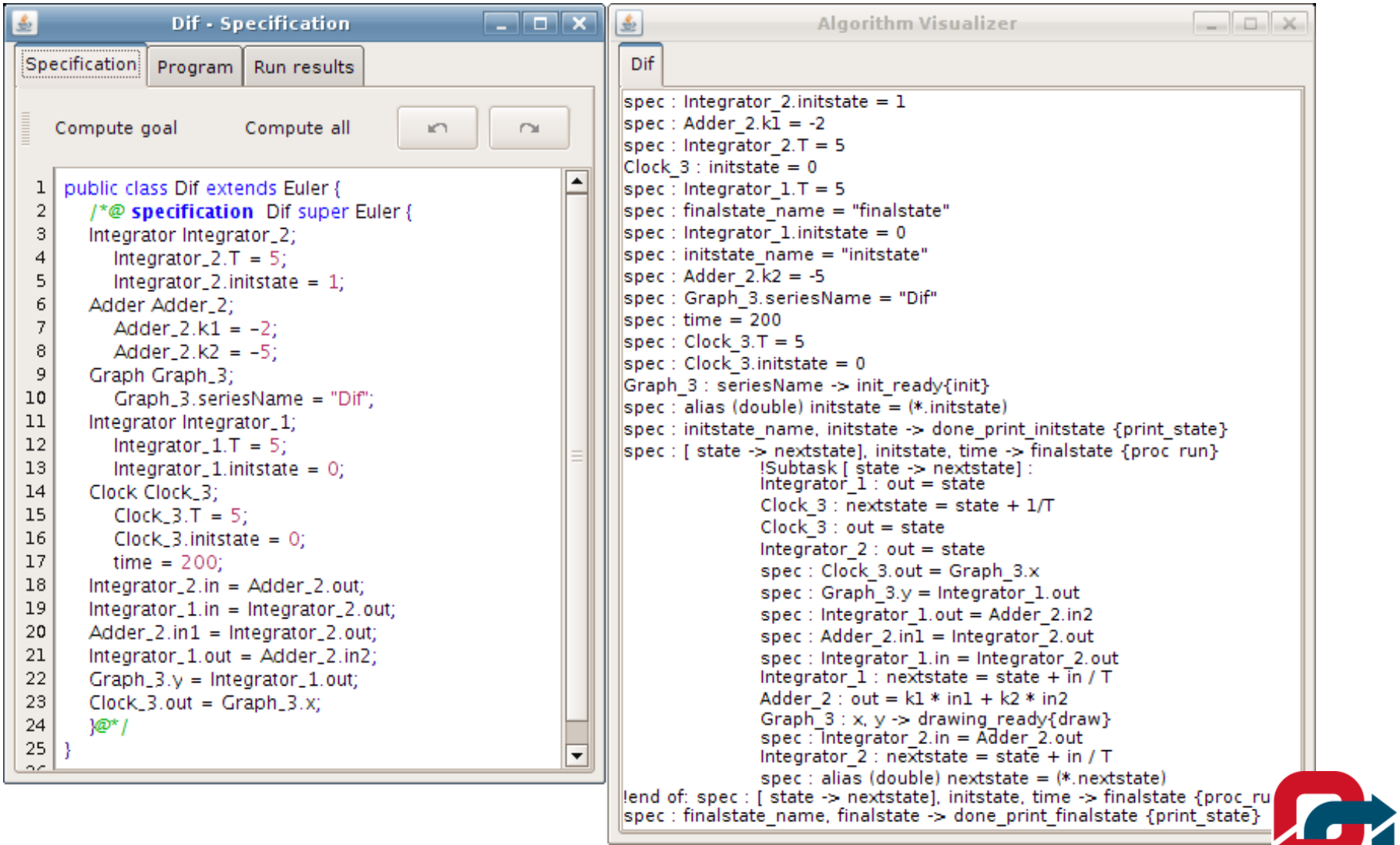
A programming interface between the generated Java program and the scheme was needed...

```
ProgramContext.getFieldValue(objectName,
                             fieldName);
```

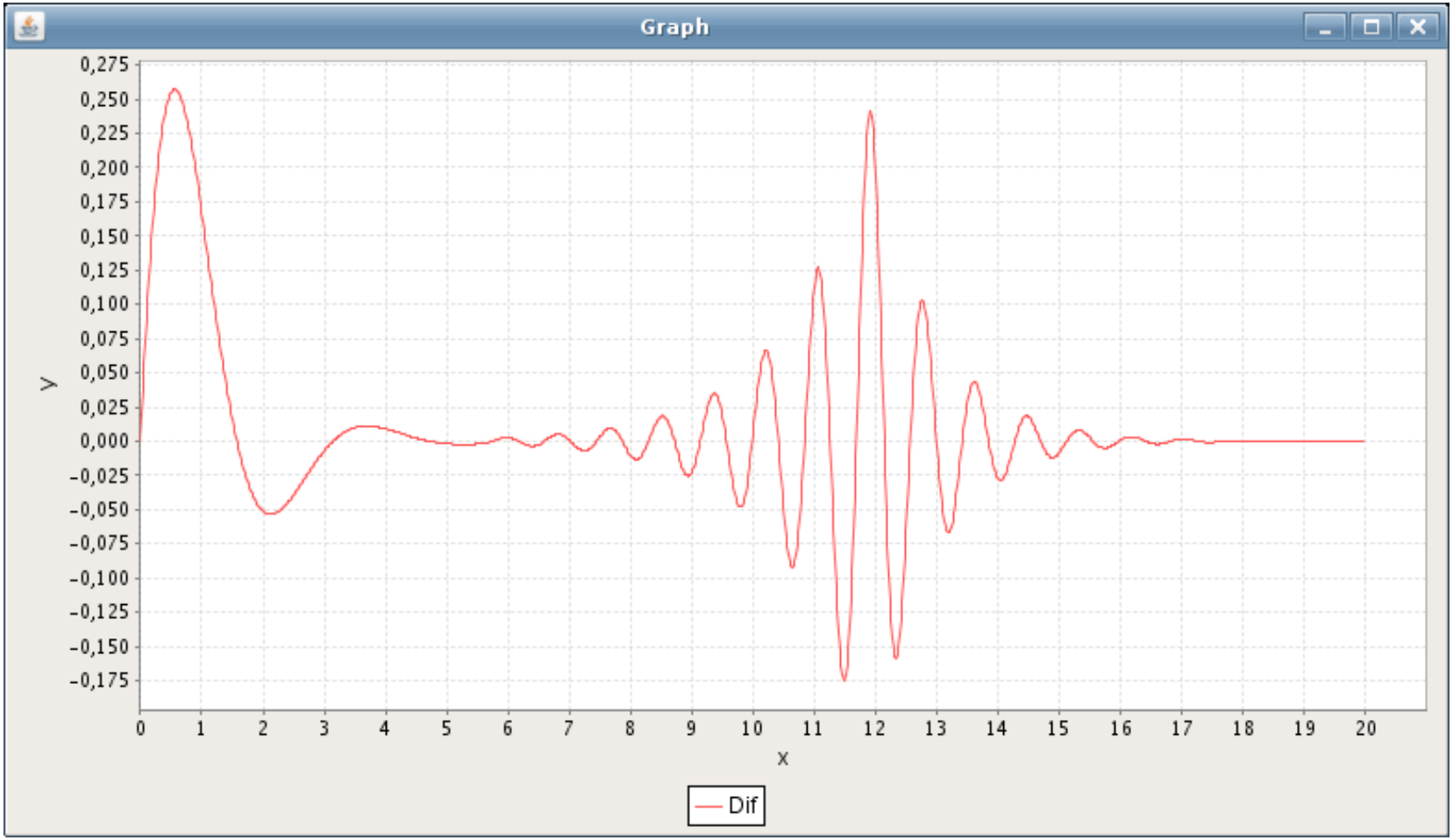
Scheme Editor: Oscillator



Oscillator: Textual Representation and Algorithm

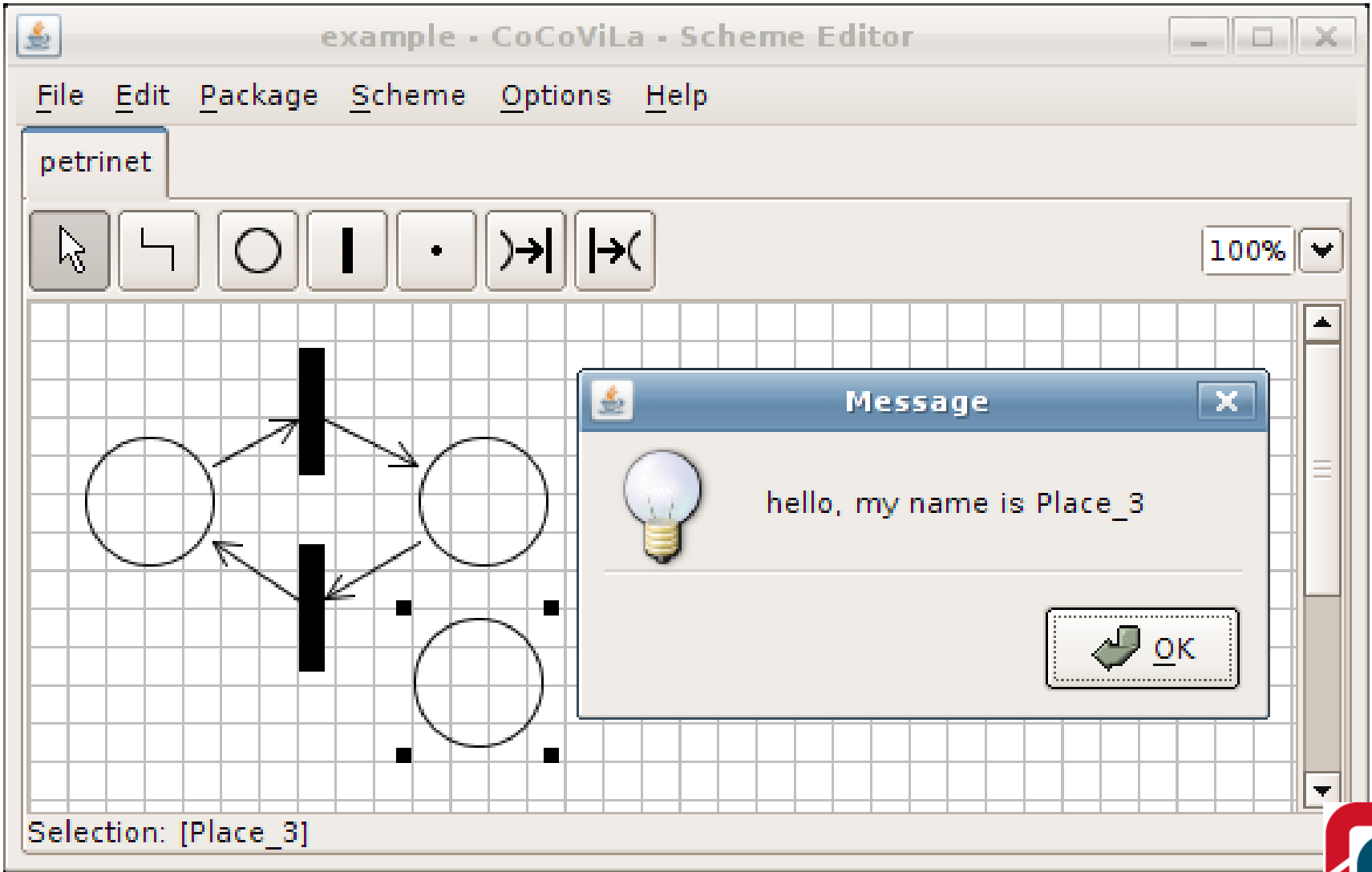


Oscillator: Simulation Result



Fields `k1` and `k2` of the Adder were changed.

Interactive Rich Component



Future Work

- Develop an expert system shell for CoCoViLa
- Implement a simple simulation engine as a rich component
- Build an AS level model of the Estonian Internet
- Collect and accumulate expert knowledge and real data
- Experiments with real data
- Implement interfaces to other simulation packages

Summary

Thank you for your attention!

Supporters:

- Institute of Cybernetics
- Estonian Information Technology Foundation
- Tiger University

References

- CoCoViLa — Compiler Compiler for Visual Languages, `http://www.cs.ioc.ee/~cocovila/`
- ns-2 — Network Simulator, `http://www.isi.edu/nsnam/ns/`
- OMNeT++ — Discrete event simulator, `http://www.omnetpp.org/`
- OPNET®, `http://www.opnet.com/`