



# Block Oriented Network Simulator - BONEs

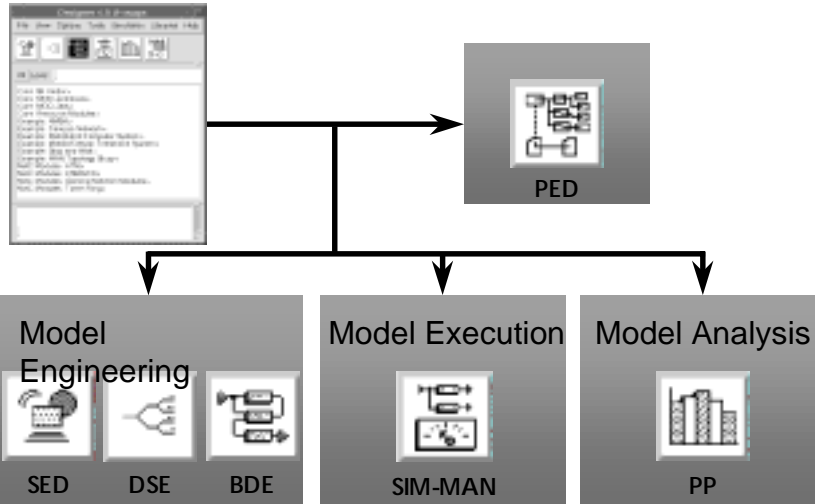
Marko Luoma

## BONEs



- ◆ BONEs is a part of Cadence / Altagroup design tools intended to be used in top to down design paradigm.
- ◆ BONEs covers system performance analysis and some behavioural algorithm design.
- ◆ BONEs offers high level libraries which make possible to do:
  - Rapid evaluation of system ideas in early phases.
  - Through simulation of entire systems with all functionality.

# BONeS Suite of Tools

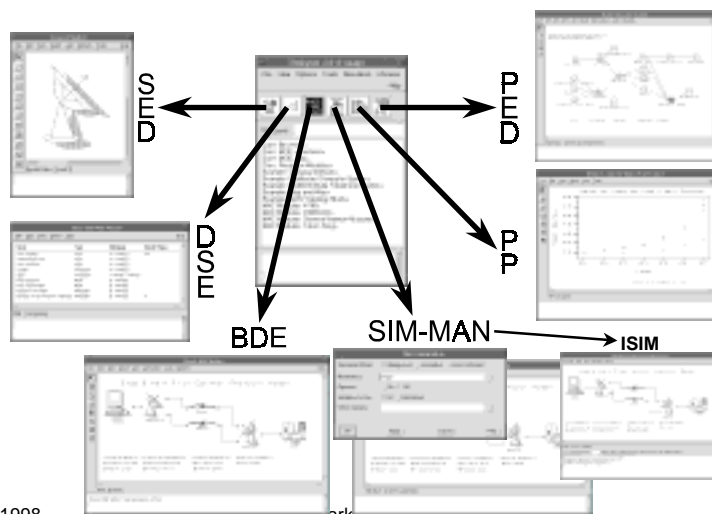


14.10.1998

(c) Marko Luoma 1998

3

# Environment and User Interface

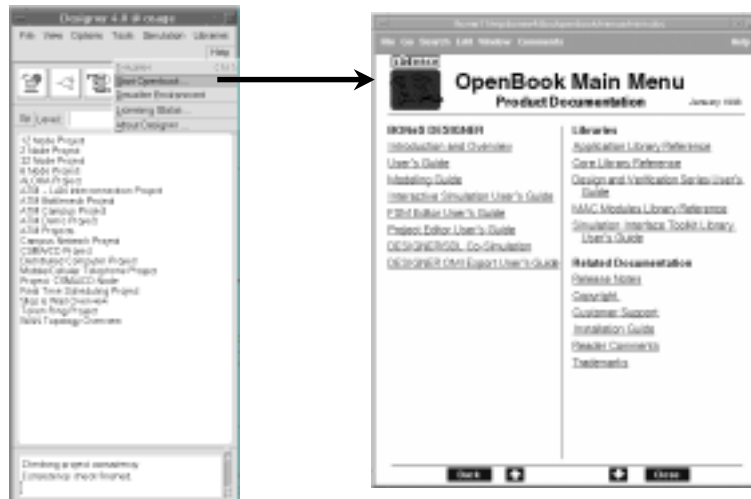


14.10.1998

(c) Marko Luoma 1998

4

# Openbook



14.10.1998

(c) Marko Luoma 1998

5

# Libraries



- ◆ Physical containers for BONEs objects
  - The BONEs design database
- ◆ Control
  - Open/closed
  - Access permissions
  - Read/write
- ◆ Environment = open libraries
  - Can select only objects from open libraries
  - Can automatically set up (Options-Save Environment)
- ◆ Corresponds to UNIX directory
- ◆ System-Supplied/User

14.10.1998

(c) Marko Luoma 1998

6



## Library operations



14.10.1998

7



## Projects

- ◆ Another kind of container
  - Strictly a logical grouping
  - Stores references to objects
  - Shows relationships
  - Orthogonal to libraries
    - » Can be several projects in a library
    - » A project can reference objects from several libraries



14.10.1998

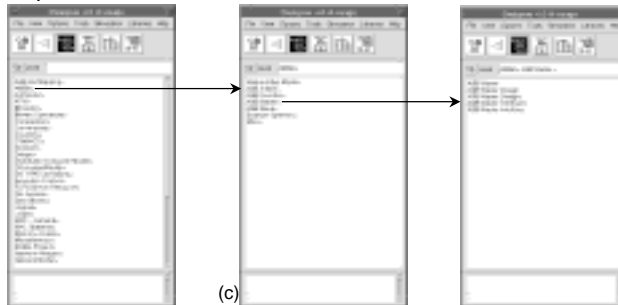
(c) Marko Luoma 1998

8



## Groups

- ◆ Another container for modules
  - Strictly logical
  - Only in BDE
  - Each object appears only once in group hierarchy
  - Groups form a tree
- ◆ Main window selection
  - By library
  - By group



14.10.1998

(c)

9



## Block Diagram Editor (BDE)



- ◆ Block diagram editor is the tool used to create modules and systems.
- ◆ With BDE you can make hierarchical implementations allowing more visual clarity.
- ◆ You select modules from list of library modules or you create new ones with:
  - Primitive editor
  - FSM editor
  - BDE editor

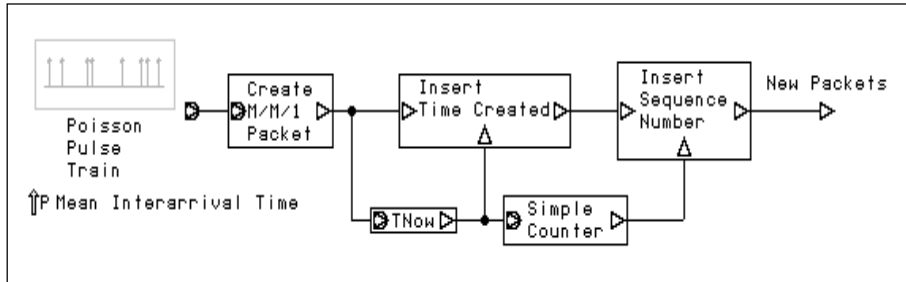
14.10.1998

(c) Marko Luoma 1998

10



# Block Diagrams



14.10.1998

(c) Marko Luoma 1998

11



# Core modules

Group	Module
Arithmetic	BV*
	BV+
	BV-
Comparison	BV==
	BV<
	BV>
Conversions	Bitvector to Int
	Int to Bitvector
Number	BVConstant
Generators	
Traffic Generators	Uniform
	PulseTrain
Vector Operations	BV Create
	BV Mem Access

Group	Module
ALU_Arithmetic	Adder
	Subtractor
	ALU 32(fixed) functions
Clock Generators	Clock
Counters	Circular Down-Counter
	Circular Up-Counter
Latch Registers	Latch
	Register
Shifters_BitOperations	And_
	Or_
	Nor_
	Left Shift by one
	Right Shift by one
Memories	DRAM
	FIFO
	ROM
	SRAM
Multipliers	Basic Multiplier

14.10.1998

(c) Marko Luoma 1998

12



## Core modules

Group	Module
DS Access/Modify	Create DS Insert Field Select Field
Number Generators	lconst Rconst TNOW Random>
Traffic Generators	Poisson Pulse Train Uniform Pulse Train
Execution Control	Execute in Order Gate
Arithmetic Loops	2 Input Expression Int Do Real Do
Delays	Abs Delay Fixed Abs Delay
Quantity-Shared Resources	Allocate Free
Server Resources	Service

- ◆ There is also a number of additional libraries for specific applications
  - MAC layer libraries
  - ATM libraries
  - TCP/IP libraries
  - Network layer libraries
  - ...

14.10.1998

(c) Marko Luoma 1998

13



## BONeS Blocks



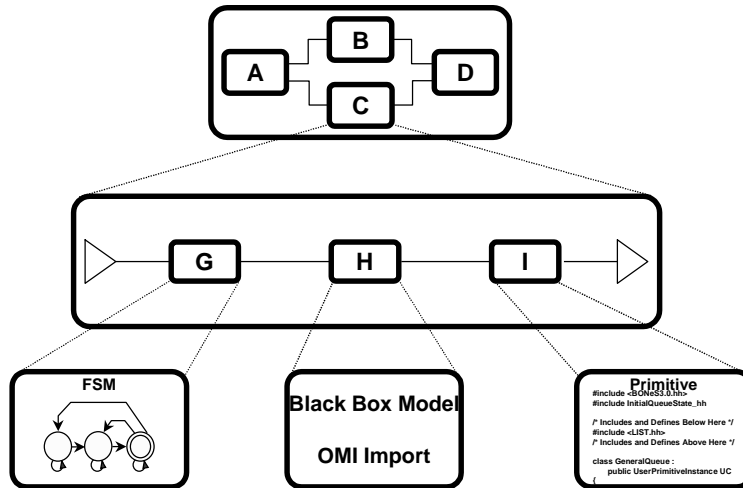
14.10.1998

(c) Marko Luoma 1998

14



## Block Diagram Hierarchy



14.10.1998

(c) Marko Luoma 1998

15



## Hierarchical implementation



14.10.1998

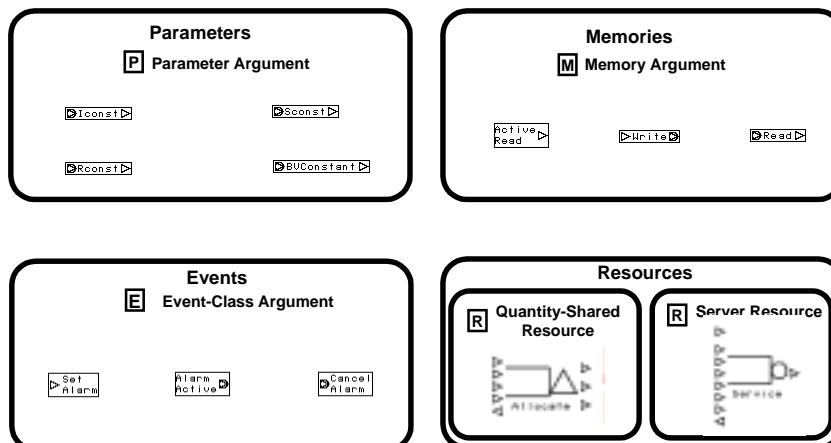
(c) Marko Luoma 1998

16





## Argument Types and Implementation



14.10.1998

(c) Marko Luoma 1998

19



## Datastructure editor (DSE)



- ◆ DSE is the tool used in creating new format for information traversing through network.
- ◆ Datastructures herit fields from their parents so that
  - Parent DS can always accommodate its child.
  - Child cannot accommodate parent cause it can not resolve all of the parameters for its fields from parent.

14.10.1998

(c) Marko Luoma 1998

20



## Datastructure

Name	Type	Subrange	Default Value
Time Created	REAL	[, infinity)	...
Sequence Number	INTEGER	[, infinity)	...
Length	INTEGER	[, infinity)	...
ACK	INTEGER	(-infinity, -infinity)	0
RO	INTEGER	(-infinity, -infinity)	...
Packet ID	INTEGER	(-infinity, -infinity)	0
Encapsulated Data	PCDT-SUBJECT		

- ◆ DS consists fields with
  - Name
  - Type of information
  - Range of values
  - Default value
- ◆ Fields can be set types, meaning that individual values are unnumbered information.
- ◆ BONEs converts set types internally number information, needed in processing.

14.10.1998

(c) Marko Luoma 1998

21



## Simulation manager (SIM)



- ◆ Simulation manager is used in executing of simulations.
- ◆ Initial values for different parameters are chosen in simulation manager.
- ◆ Each case of simulation has to saved as session of its own.

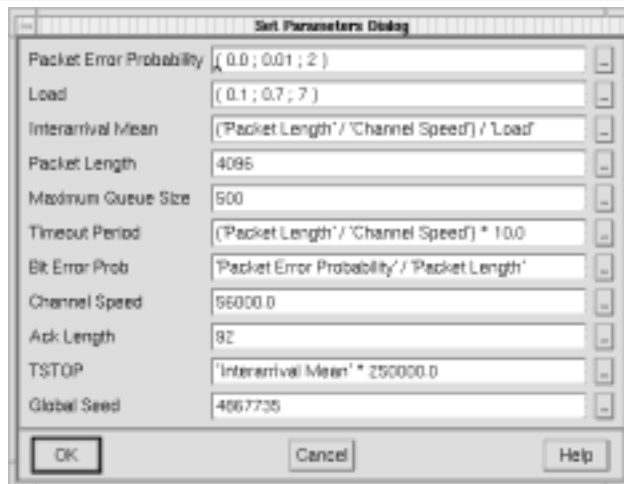
14.10.1998

(c) Marko Luoma 1998

22



## Parameter setup in SIM



14.10.1998

(c) Marko Luoma 1998

23

## Interactive simulation manager (ISIM)



- ◆ ISIM is the tool used in debugging of system.
- ◆ ISIM can also be used to build animations out of system simulation.
- ◆ With ISIM one can view development of simulation through several steps and breakpoints.

14.10.1998

(c) Marko Luoma 1998

24



# Probes



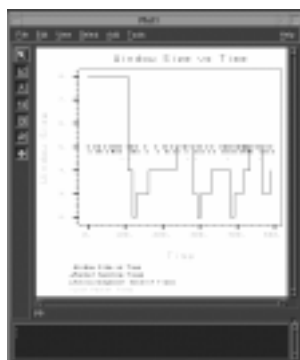
Simulation Probe



- ◆ Probes are data collection points placed on ports and arguments
- ◆ Probe placement can be made anywhere in the model hierarchy
- ◆ Probe implementation can be block diagram, FSM, or C++ primitive



# Postprocessor (PP)



- ◆ Postprocessor is tool used to create plots from datasets of probes.
- ◆ Single plot can include
  - Several datasets from different probes.
  - Trend value from one probe of many iterations



## Project editor (PE)



- ◆ Project editor is used to visualise development of project (system)
- ◆ PE can also be design canvas for blocks so that first only interfaces are created and after that implementation is done

14.10.1998

(c) Marko Luoma 1998

27



## Definition of Dynamic Instantiation

- ◆ Dynamic Instantiation (DI) lets a BONEs simulation create any number of identical instances of BONEs blocks
  - During the simulation, many instances of the block may exist simultaneously, but with varying creation and duration times
- ◆ During the simulation, each DI of a block behaves no differently than the conventional block and occupies the same location in the simulation data flow

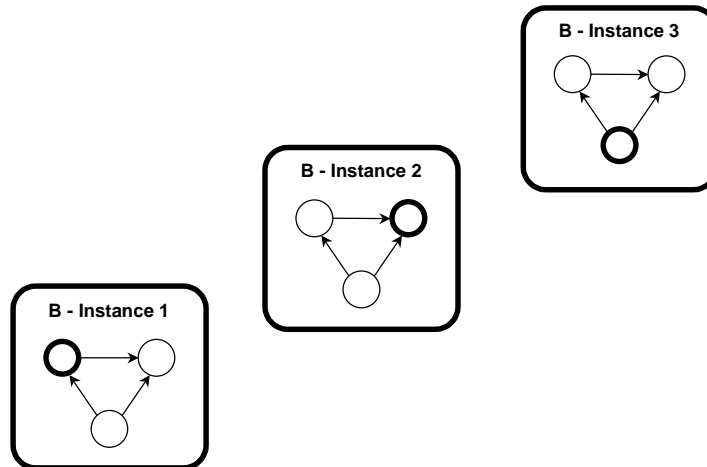
14.10.1998

(c) Marko Luoma 1998

28



## Dynamic Instances



14.10.1998

(c) Marko Luoma 1998

29



## Advantages of Dynamic Instantiation

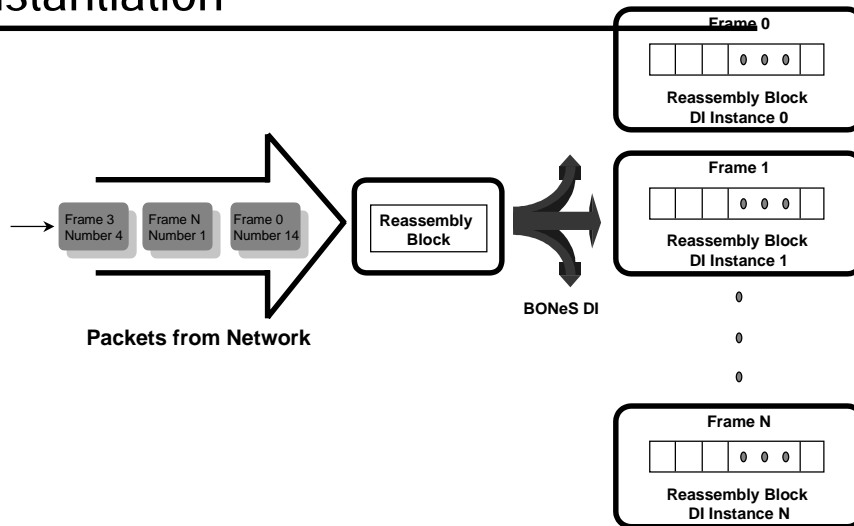
- ◆ Dynamic Instances help you solve modeling problems where objects are created, exist for some time and then are removed
- ◆ Examples
  - » Mobile units entering, traversing, and exiting a coverage area
  - » Creation and deletion of virtual circuits
  - » Peer-to-peer protocol connections

14.10.1998

(c) Marko Luoma 1998

30

# IP: Reassembly Using Dynamic Instantiation



14.10.1998

(c) Marko Luoma 1998

31