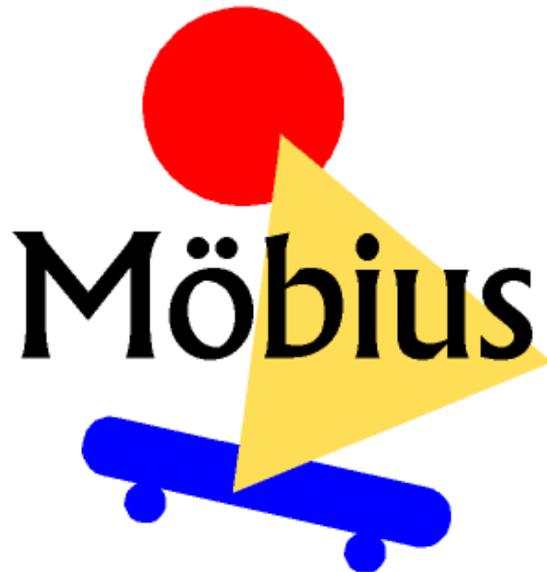


The Moebius Modelling Framework and its ZDD-based Solvers

Kai Lampka

*VOSS-Workshop: Two Decades of Probabilistic Verification
- Reflections and Perspectives -
Leiden, 12th - 16th November 2007*

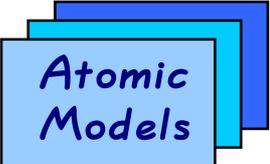




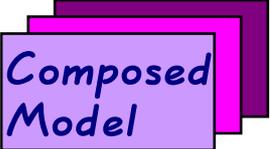
ILLINOIS
UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN

- ★ *Tool for the model-based analysis of discrete-event systems (here focus on CTMCs)*
- ★ *Developed under the lead of Bill Sanders, where other groups contributed.*
- ★ *Releases can be obtained from www.mobius.uiuc.edu*

Editors for constructing high-level models



Atomic
Models



Composed
Model

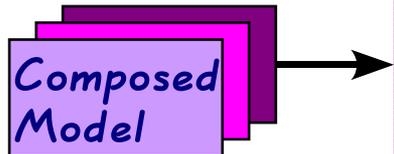
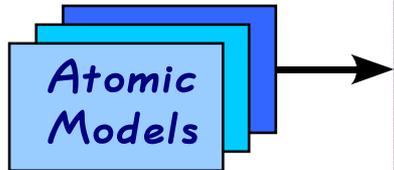


Reward-
definition



Para-
meters

Editors for
constructing
high-level
models



AFI-Objects

SANModel

PEPAModel

⋮

Activities

State Vars

Perf. vars

Abstract
Functional Interface

Editors for
constructing
high-level
models

Atomic
Models

Composed
Model

Reward-
definition

Para-
meters

AFI-Objects

SANModel

PEPAModel

⋮

Activities

State Vars

Perf. vars

CTMC
Generators
(executable)

MxD

Flat

ZDD

Abstract
Functional Interface

Editors for
constructing
high-level
models

Atomic
Models

Composed
Model

Reward-
definition

Para-
meters

AFI-Objects

SANModel

PEPAModel

⋮

Activities

State Vars

Perf. vars

CTMC

Generators
(executable)

MxD

Flat

ZDD

State-
Level AFI

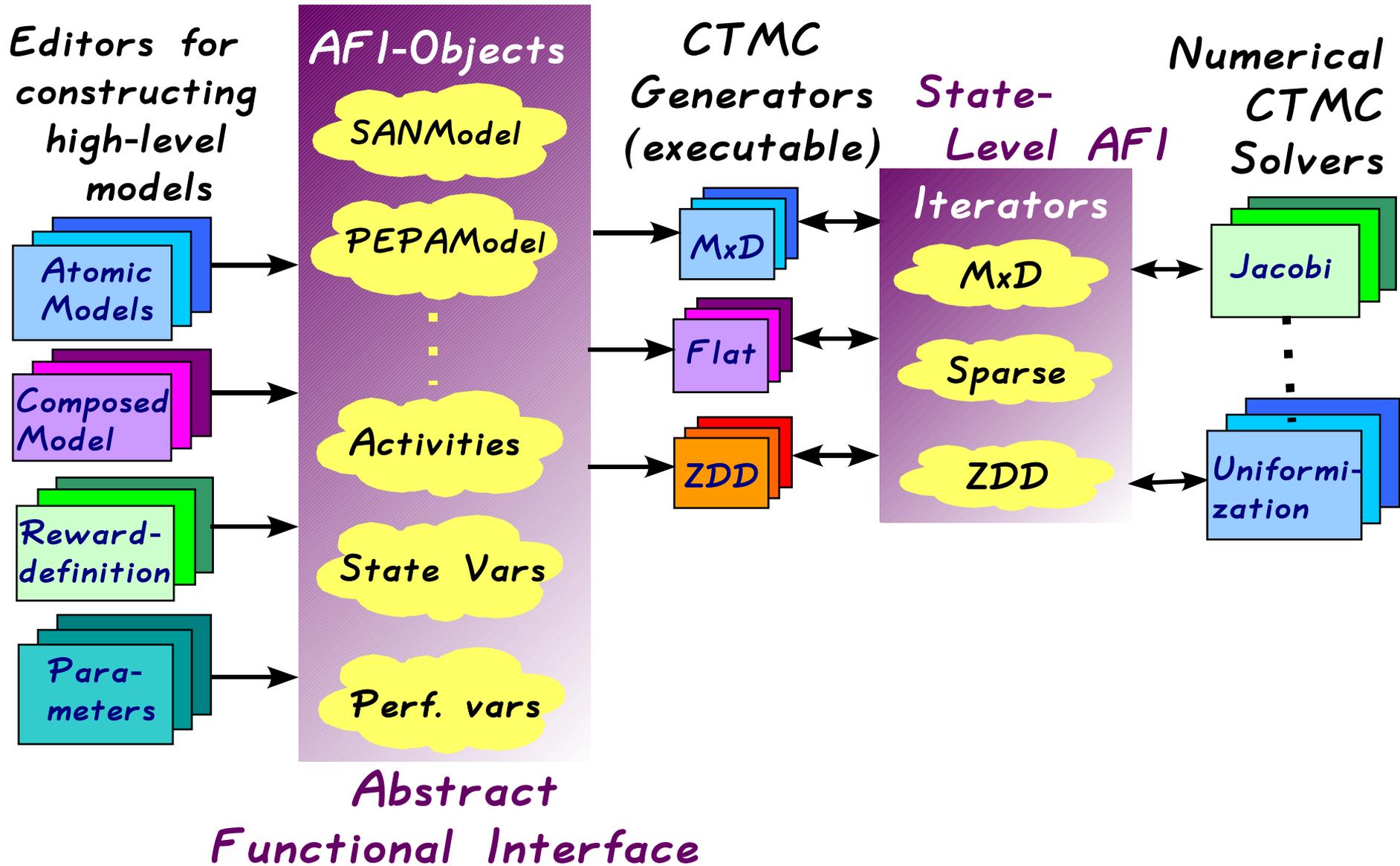
Iterators

MxD

Sparse

ZDD

Abstract
Functional Interface



Editors for
constructing
high-level
models

Atomic
Models

Composed
Model

Reward-
definition

Para-
meters

AFI-Objects

SANModel

PEPAModel

⋮

Activities

State Vars

Perf. vars

Abstract
Functional Interface

CTMC
Generators
(executable)

MxD

Flat

ZDD

State-
Level AFI

Iterators

MxD

Sparse

ZDD

proprietary
Hybrid Solvers

Numerical
CTMC
Solvers

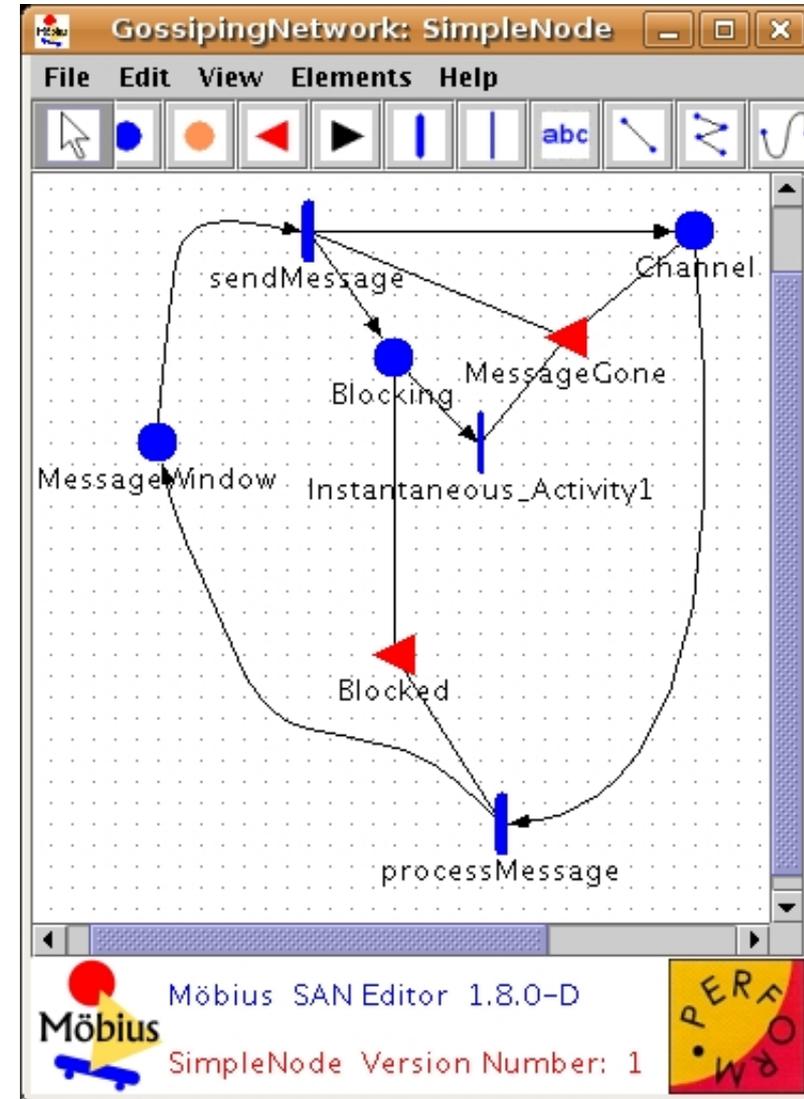
Jacobi

Uniformi-
zation

- ZDD-based solvers based on approach of Parker et. al.
- Reward-local scheme

The Editors: Atomic Models

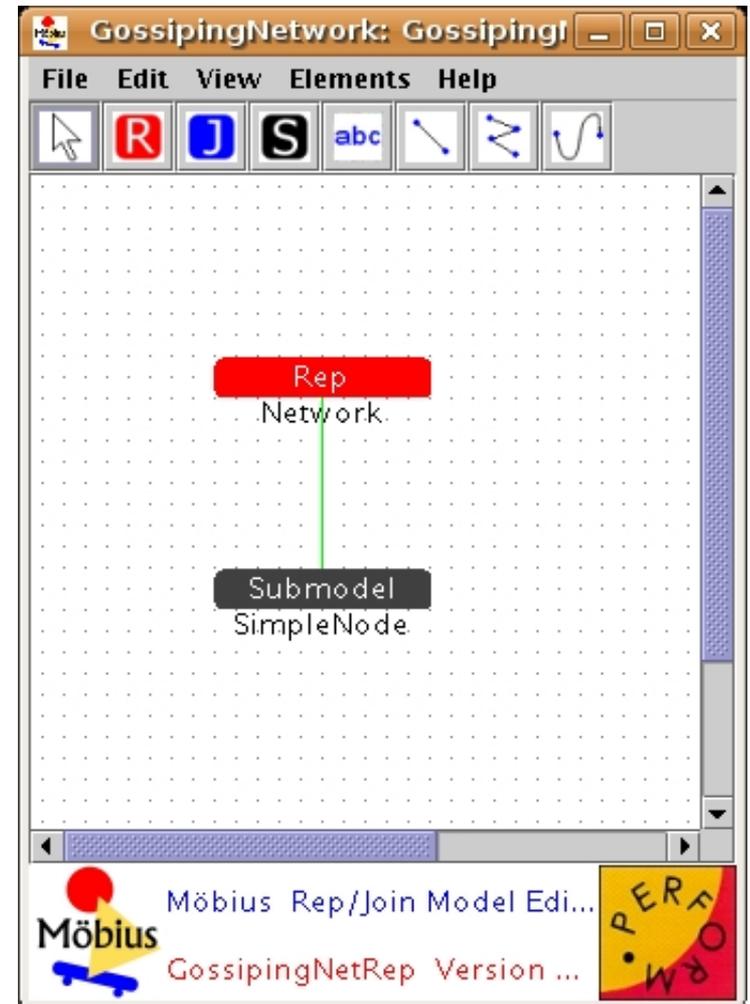
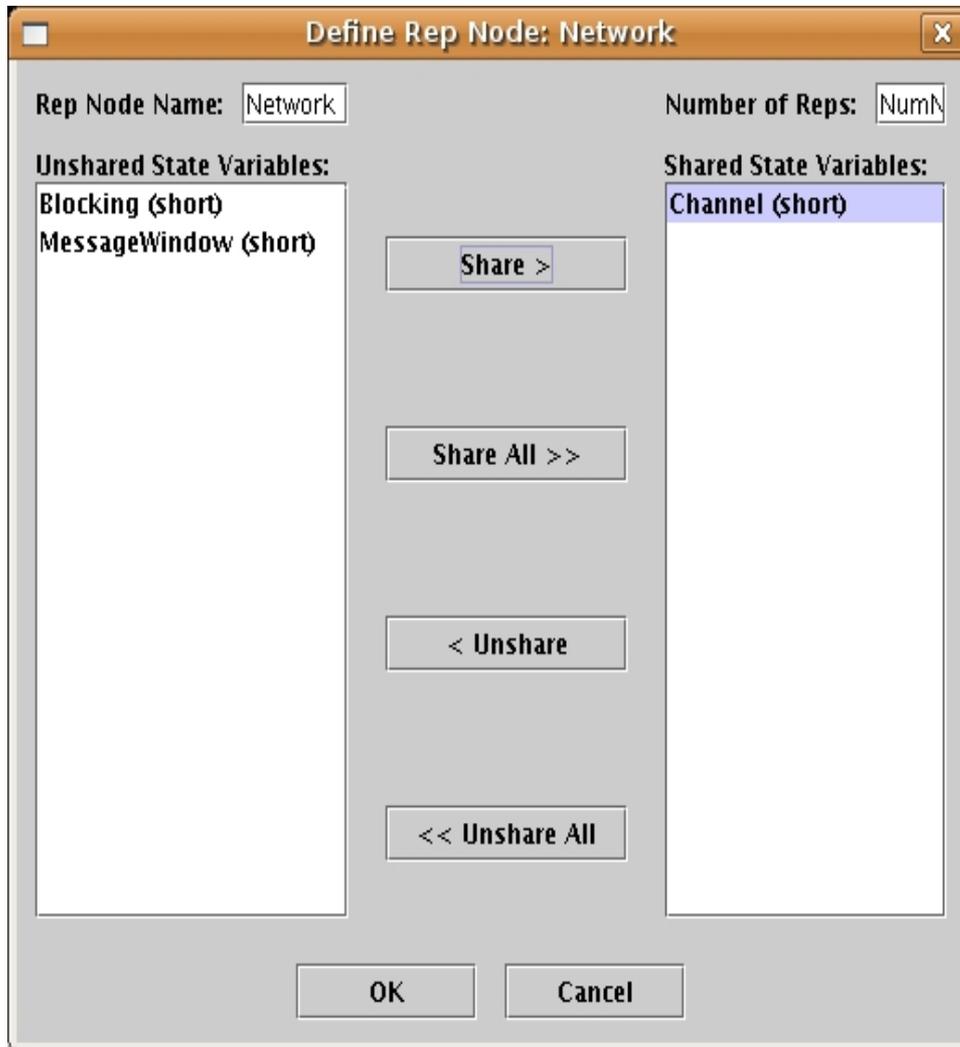
- ★ **Supported formalisms:**
SAN, PEPA, Bucket&Balls, Modest, etc..
- ★ Employed for specifying **elementary units** of the system under analysis
- ★ **Atomic models** combined within one executable model can be of different kind



The Editors: Model Composer

- ★ **Supported composition schemes:** Synchronize, Sharing of variable and Replication
 - ★ Sub-units of composed models are **atomic** or **composed models**
- ⇒ **Complex high-level model of system**

The Editors:



The Editors: Performance Variable

Complex measures for
evaluating system behavior

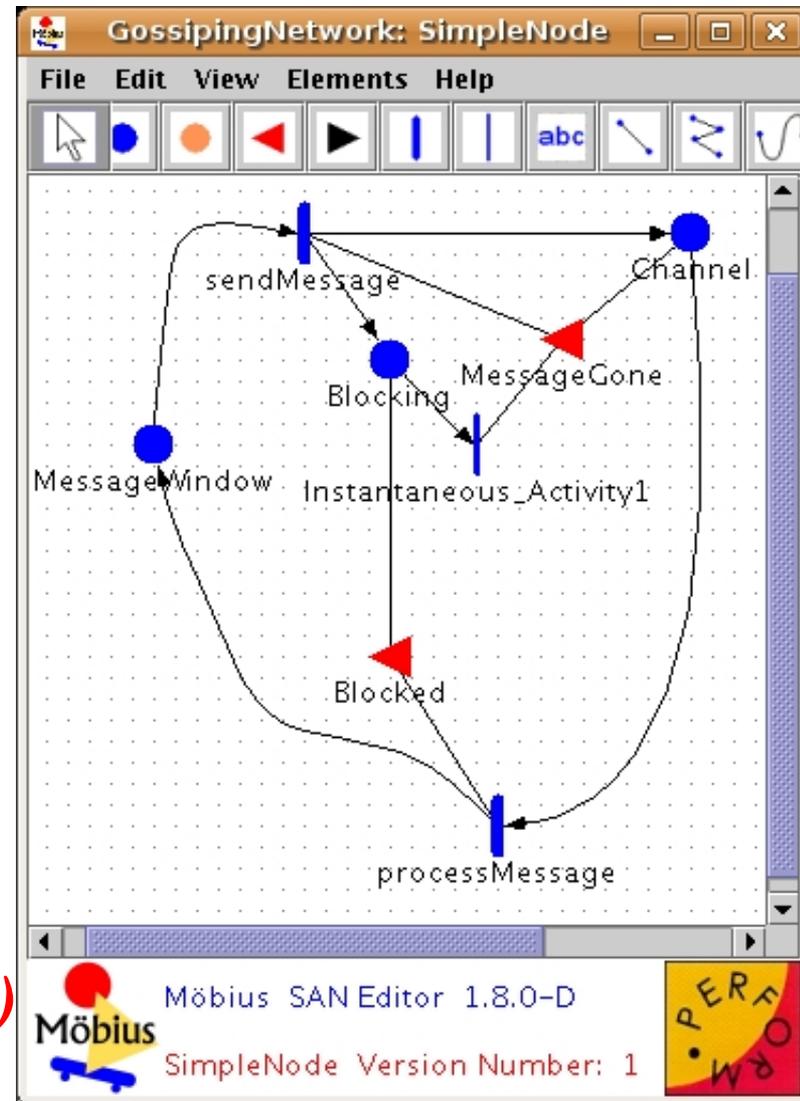
★ **Rate Rewards:**

ChannelAvailability:

```
if (Channel->Mark() == 0)
    return 1;
else return 0;
```

★ **Impulse Rewards:**

```
if (send->Enabled()) return 1;
(can also be state dependent)
```

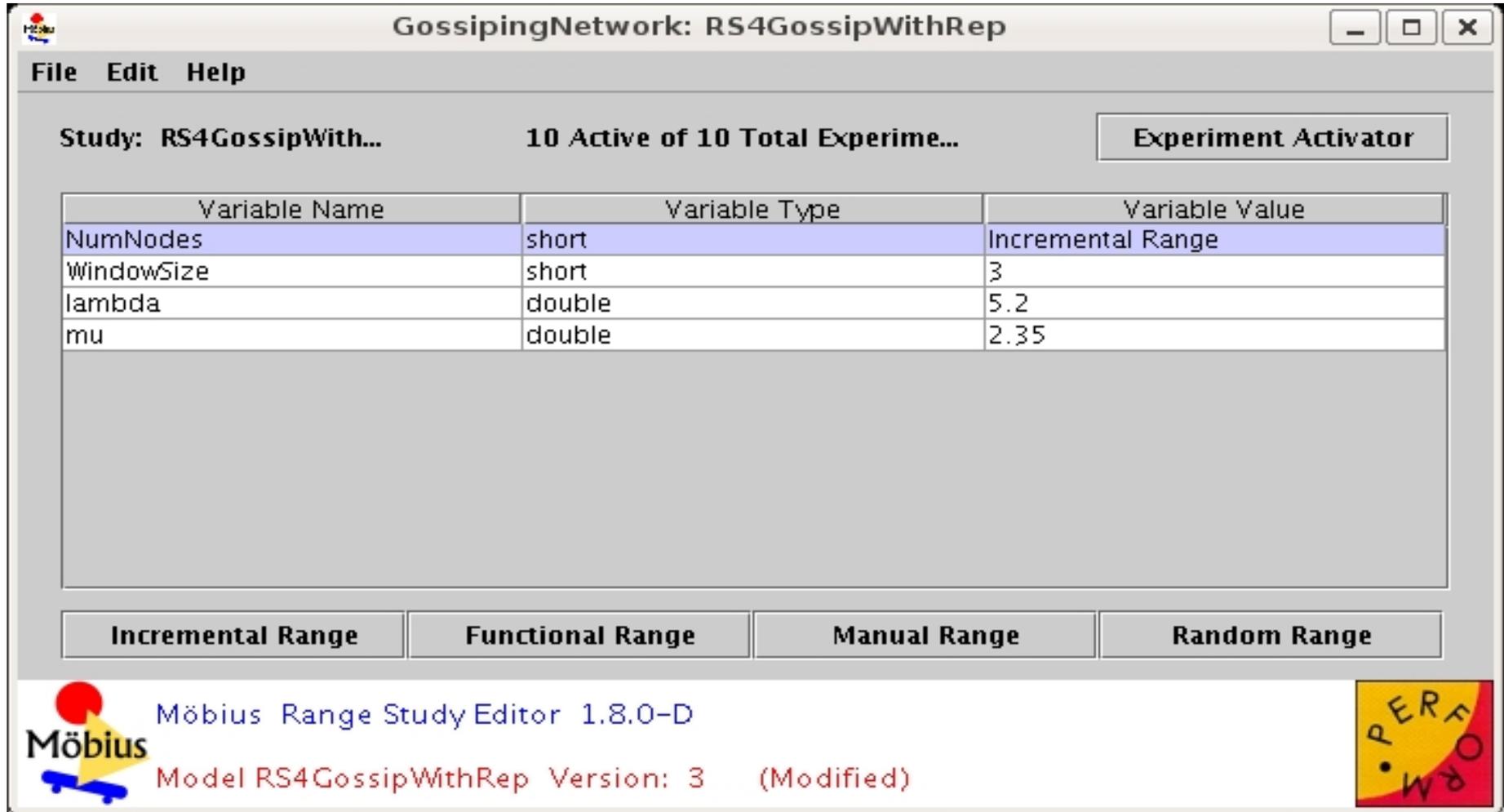


The Editors: Design Experiments

Define fixed values or ranges for the parameters (global variables) of the model, e.g.:

- ★ Varying values for the activity rates*
- ★ Varying values for number of tokens or initial value of state variable*
- ★ Varying values for number of replicas*

The Editors: Design Experiments



The screenshot shows the 'GossipNetwork: RS4GossipWithRep' window in the Möbius Range Study Editor. The window title bar includes standard OS controls (minimize, maximize, close). The menu bar contains 'File', 'Edit', and 'Help'. The main area displays the study name 'Study: RS4GossipWith...', the number of active experiments '10 Active of 10 Total Experieme...', and an 'Experiment Activator' button. A table lists the variables being edited:

Variable Name	Variable Type	Variable Value
NumNodes	short	Incremental Range
WindowSize	short	3
lambda	double	5.2
mu	double	2.35

At the bottom of the window, there are four buttons for selecting the range type: 'Incremental Range', 'Functional Range', 'Manual Range', and 'Random Range'. The status bar at the bottom left shows the Möbius logo and the text 'Möbius Range Study Editor 1.8.0-D'. The status bar at the bottom right shows the text 'Model RS4GossipWithRep Version: 3 (Modified)' and a 'PERFORM' logo.

Solvers: Generate reward-annotated CTMCs

- ★ **Traditional Generator**
(Hash Table + Sparse Matrix)
- ★ **Symbolic Generators:**
 - **MDD-based approach [Derisavi⁺03],**
inspired by Saturation Technique of Ciardo et al.
 - x run-time sometimes not satisfactory
 - x usage of instantaneous activities limited,
 - x impulse rewards not supported
 - **ZDD-based activity-local scheme**
[Lam03,Lam06]
 - x instantaneous activities not yet supported

Solvers: Numerical Solution

On the basis of the transition rate matrix of each set of parameters **transient state and steady state probabilities** are computed

- ★ Moebius offers a wide range of numerical solution methods
 - Jacobi, Gauss-Seidel, Takahashi, . . .
 -
- ★ Solvers employ **state-level AFI** for accessing transition rate matrices
- ★ **Proprietary solvers for ZDDs [Lam06b]**

Moebius (Key features)

- ★ **Multi-solution:** Deterministic or stochastic delays
=> Numerical (with CTMCs) or simulation-based analysis of models
- ★ **Compositional modelling style:** User-defined symmetries are exploited (**Replicate**), leading to much smaller CTMCs
- ★ **Multi-formalism:** Different model description methods can be combined within one executable model
- ★ **Extendability: Abstract Interfaces:**
 - ★ **AFI:** Formalism-independent model descriptions in C++
 - ★ (State-Level AFI: Layout of transition matrix is independent of solvers.)