http://jaslibrary.sourceforge.net

JAS 1.0: New features

Michele Sonnessa
(sonnessa@di.unito.it)
Department of Computer Science
University of Torino, Italy

SwarmFest 2004, May 9-11 Ann Arbor

Main new features

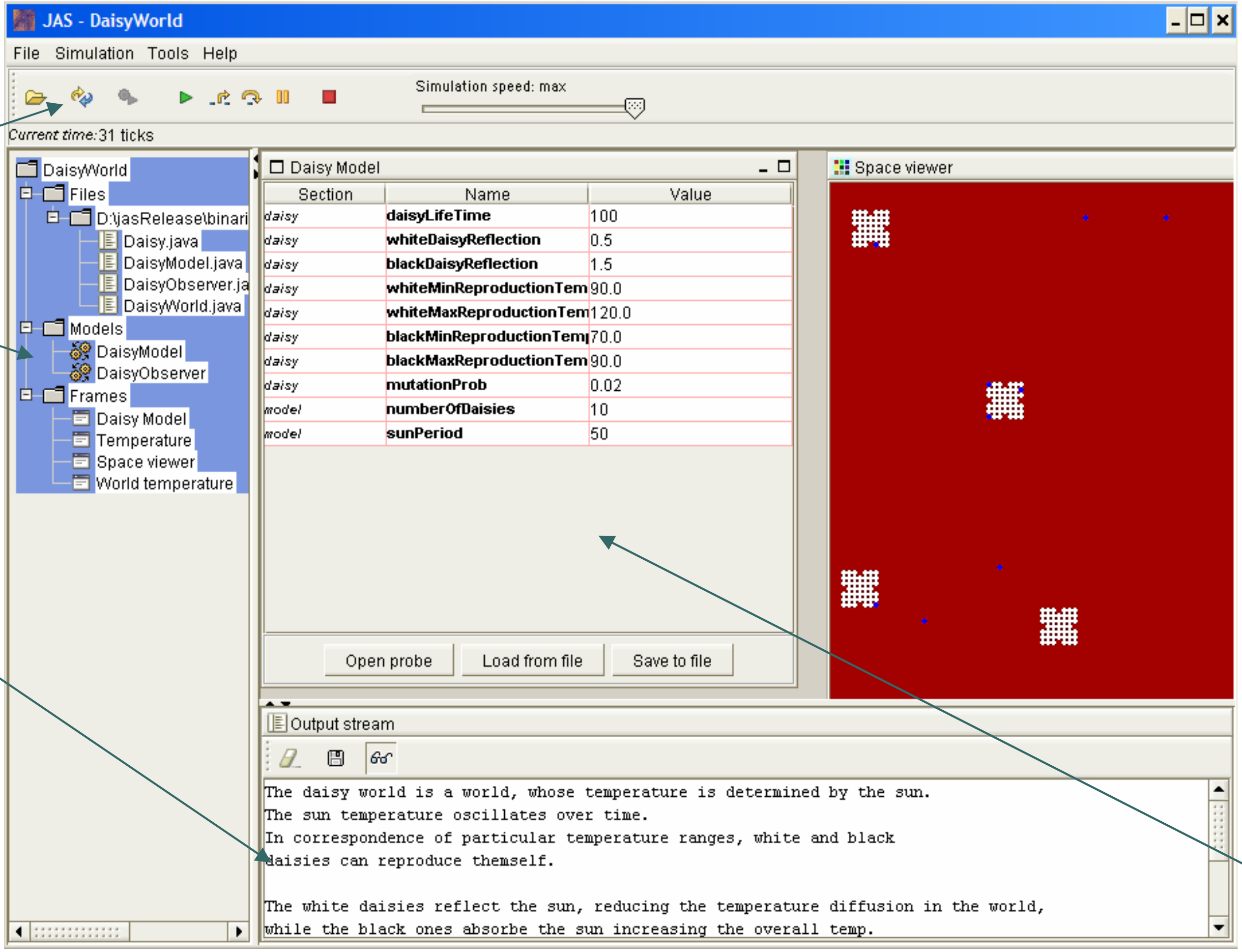
- A new GUI: the simulation environment
- A brand new statistical package
- HSQLDB database support
- Graph package with Social Network Analysis functions

The simulation environment

Simulation controls

Project tree panel

Output console

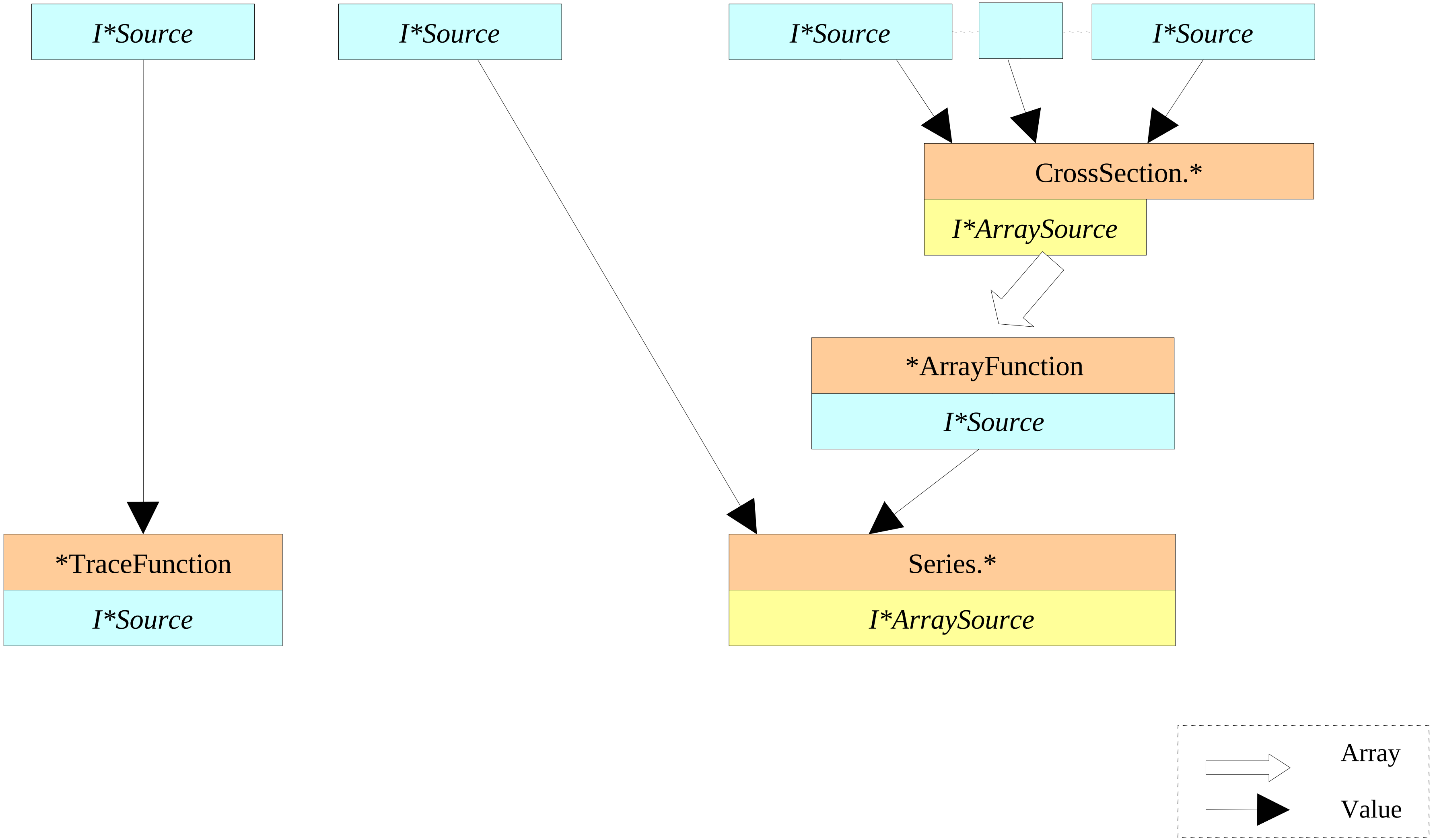


Section	Name	Value
daisy	daisyLifeTime	100
daisy	whiteDaisyReflection	0.5
daisy	blackDaisyReflection	1.5
daisy	whiteMinReproductionTem	90.0
daisy	whiteMaxReproductionTem	120.0
daisy	blackMinReproductionTem	70.0
daisy	blackMaxReproductionTem	90.0
daisy	mutationProb	0.02
model	numberOfDaisies	10
model	sunPeriod	50

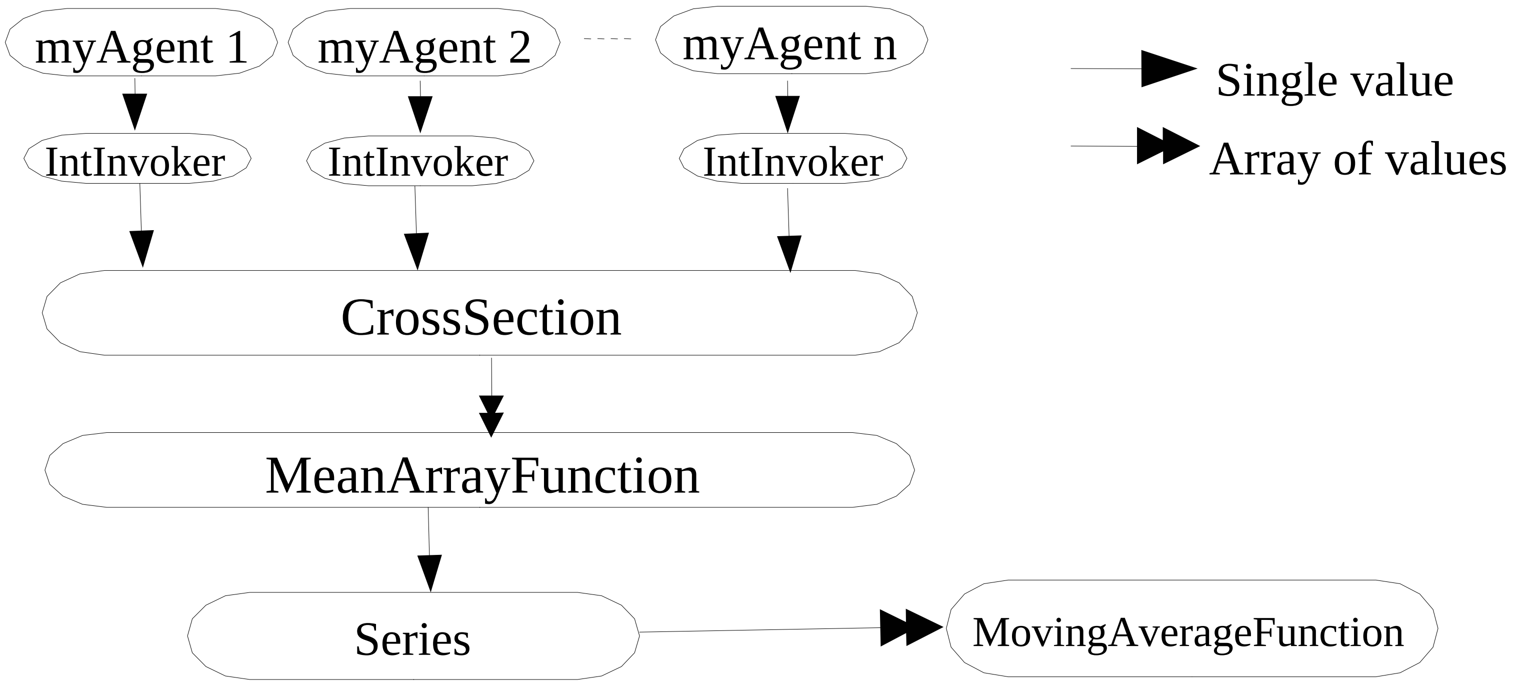
```
The daisy world is a world, whose temperature is determined by the sun.
The sun temperature oscillates over time.
In correspondence of particular temperature ranges, white and black
daisies can reproduce themself.

The white daisies reflect the sun, reducing the temperature diffusion in the world,
while the black ones absorbe the sun increasing the overall temp.
```

User's model area

Statistics: the encapsulation



Statistics: the moving average example



Statistics: a code example

```
CrossSection.Integer crossSection = new CrossSection.Integer(agentList, “income”, false);
Series.Double series = new Series.Double(new MeanFunction(crossSection));
MovingAverageArrayFunction ma = new MovingAverageArrayFunction(series, 3);

…

eventList.scheduleSimple(0, 1, ma, Sim.EVENT_UPDATE);
```

jas.statistics.db.*

Hypersonic SQL Database

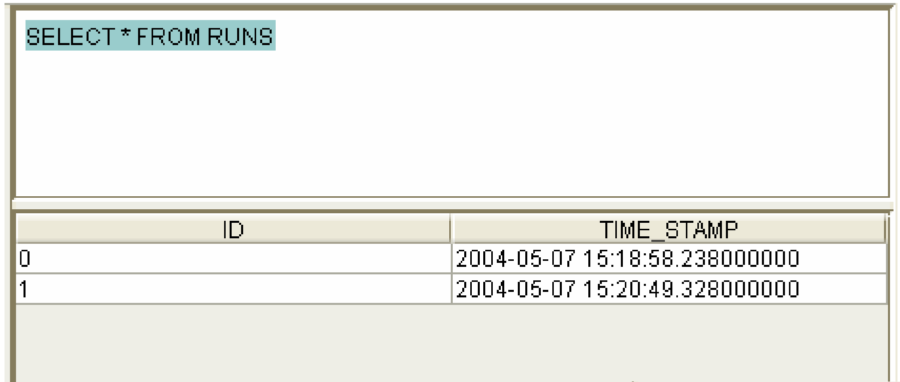
- Automatically collects simulation data
- Supports multi-run indexing
- Automatically creates primary keys

- Data can be later read with
  - Standard jdbc consumers
  - OpenOffice
  - The JAS database manager (taken from the HSQL tools)

Database: an example

```
private void buildDatabase(String databaseName)
{
    database = new Database(path + "dbout/", databaseName, true);

    CollectionTable agentIncomeTable =
            database.addCollectionTable("AgentIncome",
                                    Table.PK_SIMULATION_TIME, agents);
    agentIncomeTable.addDoubleColumn("CurrentIncome", DataAgent.INCOME);

    Table averageTable = database.addTable("Wealth", Table.PK_SIMULATION_TIME);
    averageTable.addDoubleColumn("Average", new MeanArrayFunction(agentsWealth));
    averageTable.addDoubleColumn("Min", new MinArrayFunction.Double(agentsWealth));
    averageTable.addDoubleColumn("Max", new MaxArrayFunction.Double(agentsWealth));

    database.openDatabase();
}
```

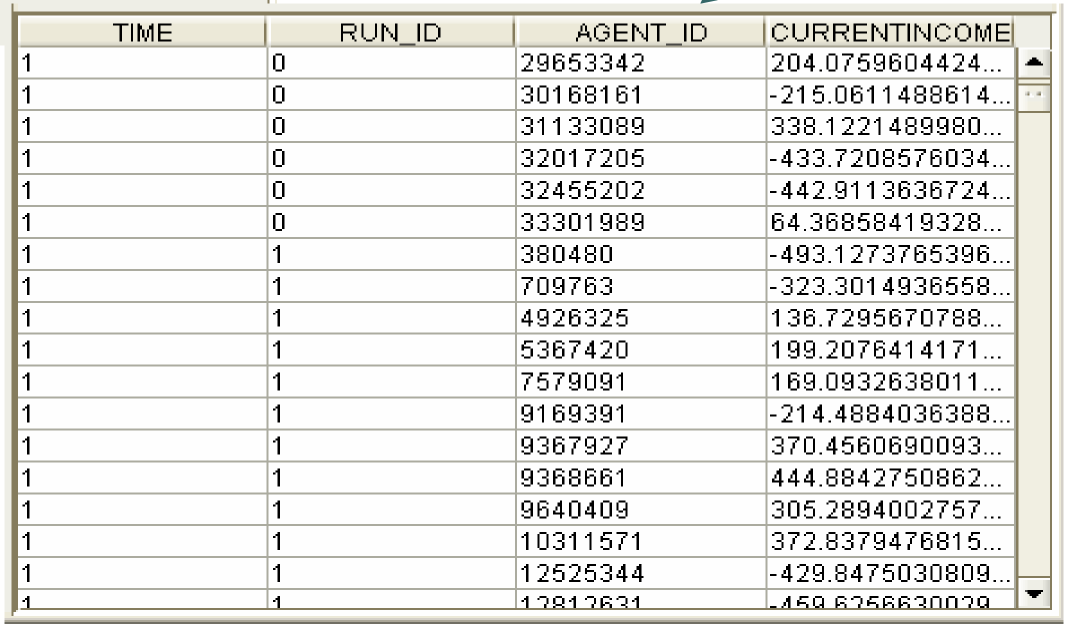
Database: output



```
SELECT * FROM RUNS
```

ID	TIME_STAMP
0	2004-05-07 15:18:58.238000000
1	2004-05-07 15:20:49.328000000

AgentIncome table



TIME	RUN_ID	AGENT_ID	CURRENTINCOME
1	0	29653342	204.0759604424...
1	0	30168161	-215.0611488614...
1	0	31133089	338.1221489980...
1	0	32017205	-433.7208576034...
1	0	32455202	-442.9113636724...
1	0	33301989	64.36858419328...
1	1	380480	-493.1273765396...
1	1	709763	-323.3014936558...
1	1	4926325	136.7295670788...
1	1	5367420	199.2076414171...
1	1	7579091	169.0932638011...
1	1	9169391	-214.4884036388...
1	1	9367927	370.4560690093...
1	1	9368661	444.8842750862...
1	1	9640409	305.2894002757...
1	1	10311571	372.8379476815...
1	1	12525344	-429.8475030809...
1	1	12812631	-459.6256630029

RUNS table

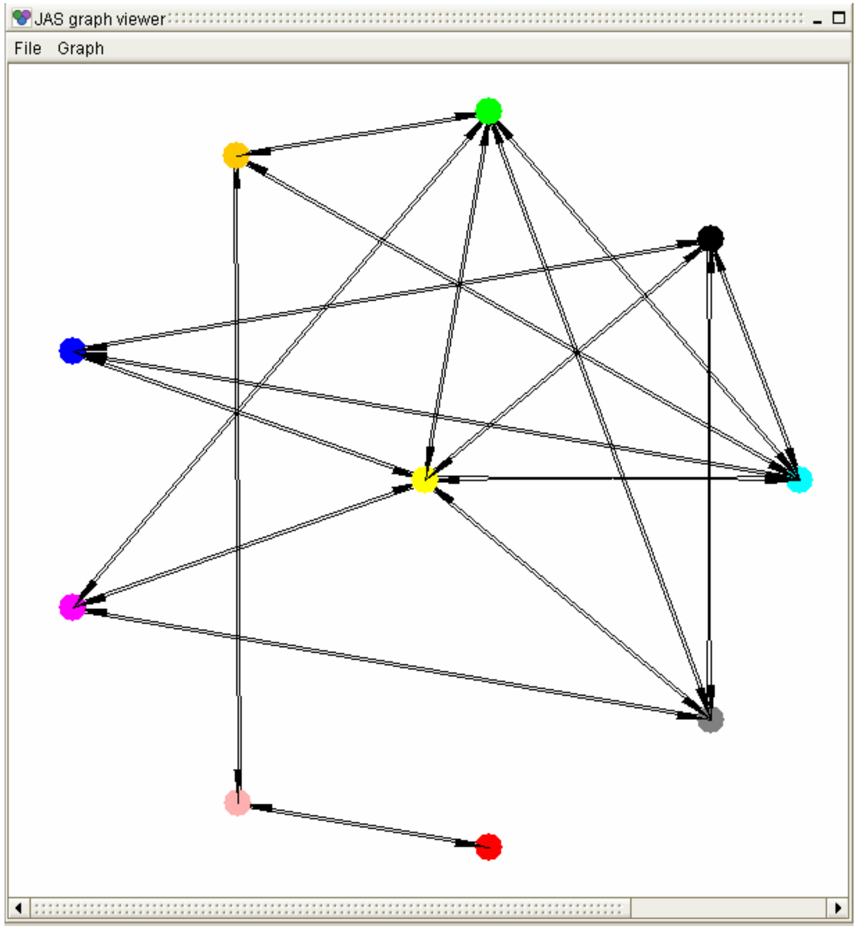
Graphs

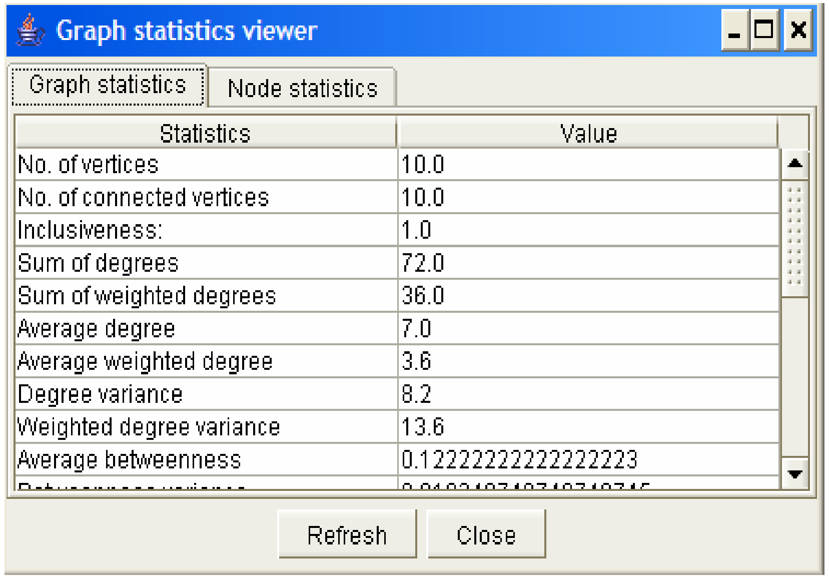
- Graphs are based on the JGraphT library
- We introduced the RelationalAgent template
- JAS provides:
  - A graph drawer
  - A graph visual editor
  - Layout managers (by R.Boero)
  - Social Network Analysis statistics (by R.Boero)
  - Support for GraphML I/O format

Graph based models

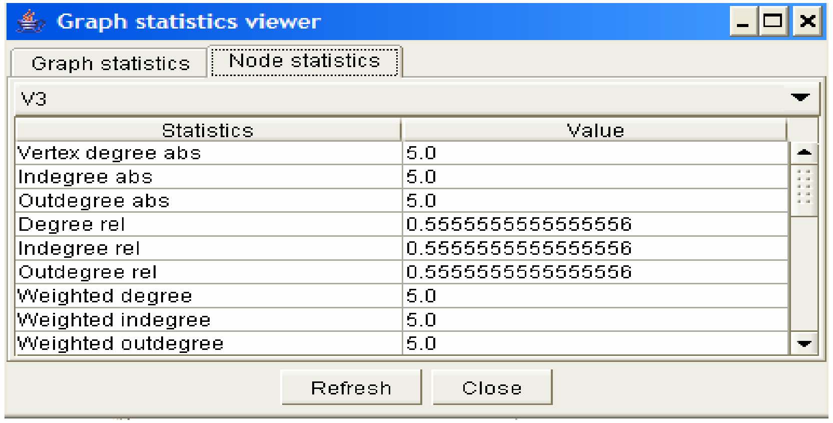
- Topological models: the agents are in relation with other agents and move in the space
- Abstract relational models: the agent localization is managed by the layer managers

Graph viewer and SNA





Statistics	Value
No. of vertices	10.0
No. of connected vertices	10.0
Inclusiveness:	1.0
Sum of degrees	72.0
Sum of weighted degrees	36.0
Average degree	7.0
Average weighted degree	3.6
Degree variance	8.2
Weighted degree variance	13.6
Average betweenness	0.12222222222222223
Betweenness variance	[illegible]



Statistics	Value
Vertex degree abs	5.0
Indegree abs	5.0
Outdegree abs	5.0
Degree rel	0.5555555555555556
Indegree rel	0.5555555555555556
Outdegree rel	0.5555555555555556
Weighted degree	5.0
Weighted indegree	5.0
Weighted outdegree	5.0

Some other new features

- Artificial neural networks can now save and load data (weights, network structure and data sets) using the XML format
- The AgentList object is able to add or remove objects during list iteration

Under construction

- A discrete-event library to easily design workflow simulations
- How-tos and examples on GA, ANN, …

Thank you

Michele Sonnessa
(sonnessa@di.unito.it)

http://jaslibrary.sourceforge.net