SEN9110 Simulation Masterclass

14. Simulation Languages (2)

Prof.dr.ir Alexander Verbraeck
a.verbraeck@tudelft.nl

Brightspace: SEN9110

Agenda

- Simulation Language demonstrations and comparison [2]

- Simulation Environments
  - AnyLogic
  - DSOL
  - Simio
  - Salabim

- Simulation Comparison

  Paper: T.W. Tewoldeberhan, A. Verbraeck and V. Hlupic. Implementing a discrete-event simulation software selection methodology for supporting decision making at Accenture. Journal of the Operational Research Society (2010) 61, 1446-1458.

- Questions about earlier lectures

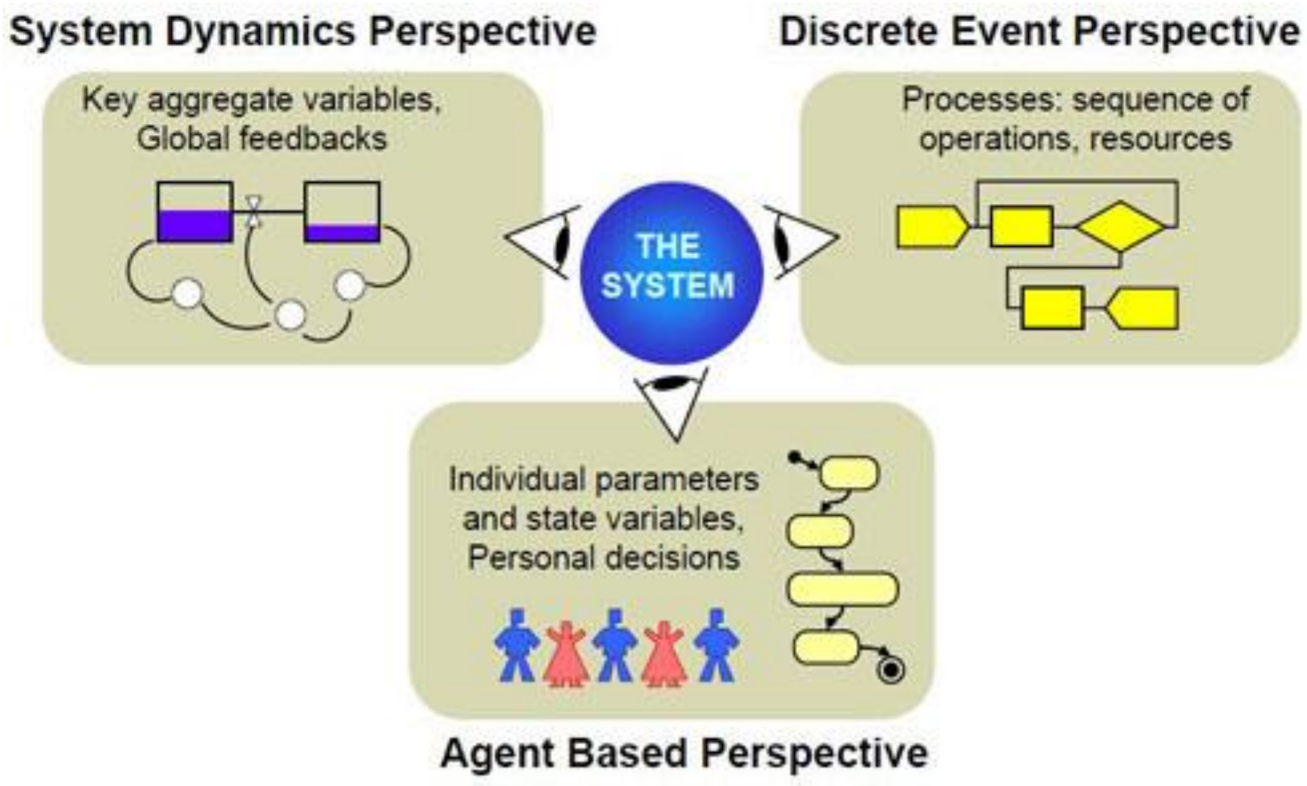
7.

AnyLogic

AnyLogic

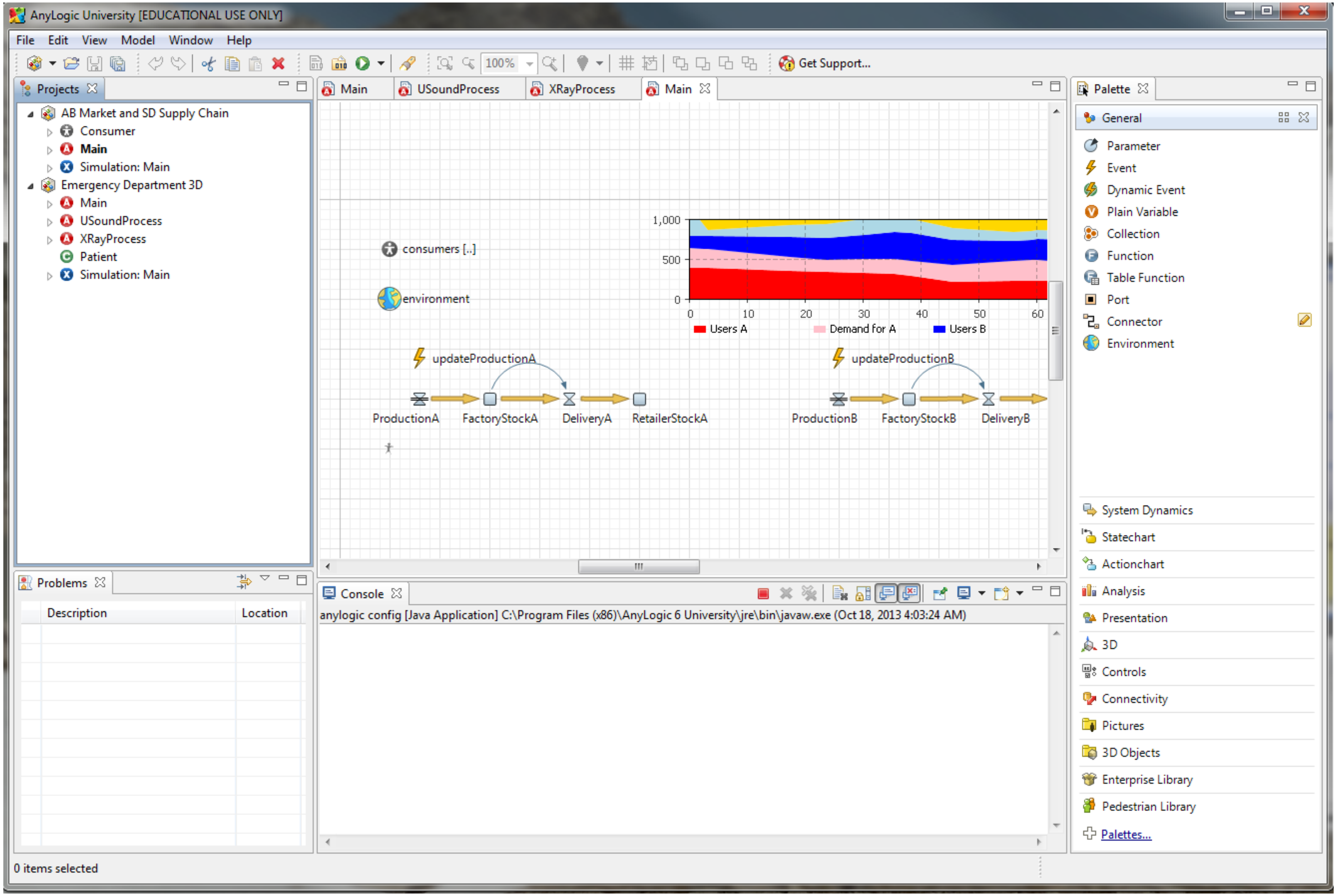
- Java-based, adapted Eclipse development environment
- DES + SD + Agents
- Mixed models possible
- Discrete time and continuous time
- Different libraries of components available
- 2D and 3D animation, optimization

AnyLogic


System Dynamics Perspective
Key aggregate variables, Global feedbacks
Discrete Event Perspective
Processes: sequence of operations, resources
THE SYSTEM
Individual parameters and state variables, Personal decisions
Agent Based Perspective

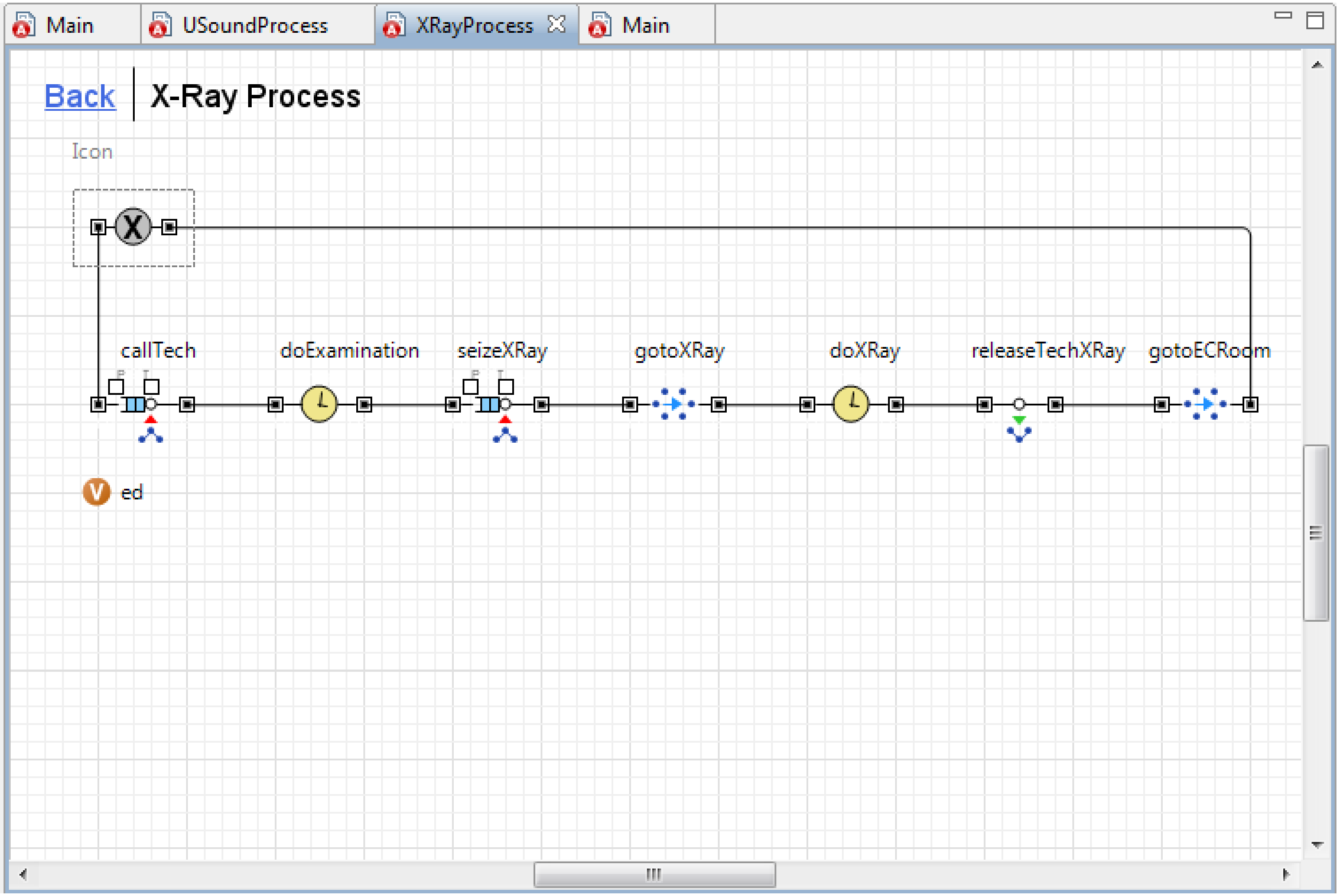
Source: wikipedia

AnyLogic: model development

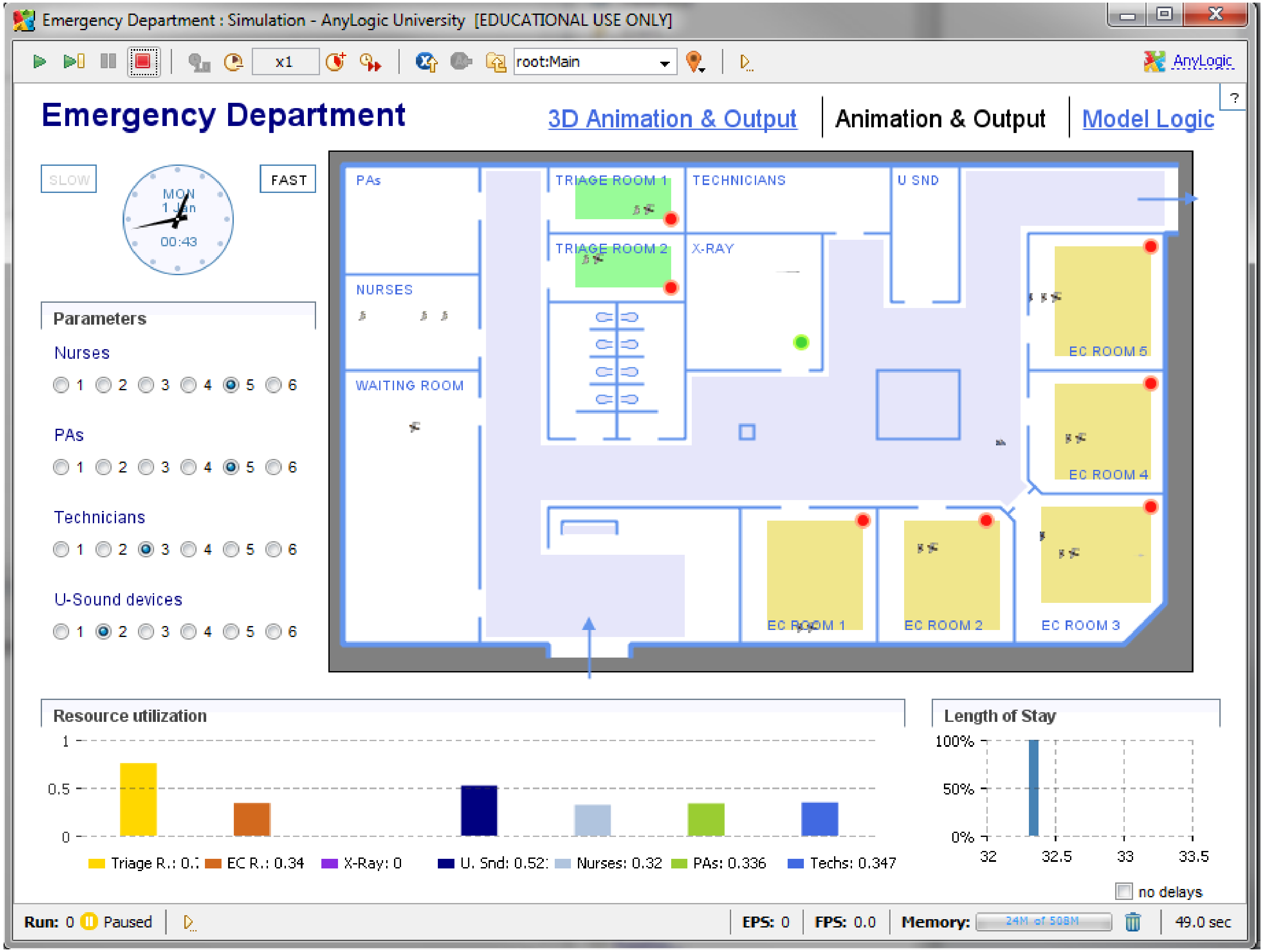


```
anylogic config [Java Application] C:\Program Files (x86)\AnyLogic 6 University\jre\bin\javaw.exe (Oct 18, 2013 4:03:24 AM)
```

AnyLogic: DES model development



AnyLogic: Running model



Emergency Department : Simulation - AnyLogic University [EDUCATIONAL USE ONLY]

Emergency Department

Parameters

Nurses

○ 1 ○ 2 ○ 3 ○ 4 ● 5 ○ 6

PAs

○ 1 ○ 2 ○ 3 ○ 4 ● 5 ○ 6

Technicians

○ 1 ○ 2 ● 3 ○ 4 ○ 5 ○ 6

U-Sound devices

○ 1 ● 2 ○ 3 ○ 4 ○ 5 ○ 6

□ no delays

Run: 0 Paused | EPS: 0 | FPS: 0.0 | Memory: 24M of 508M | 49.0 sec

8.

DSOL

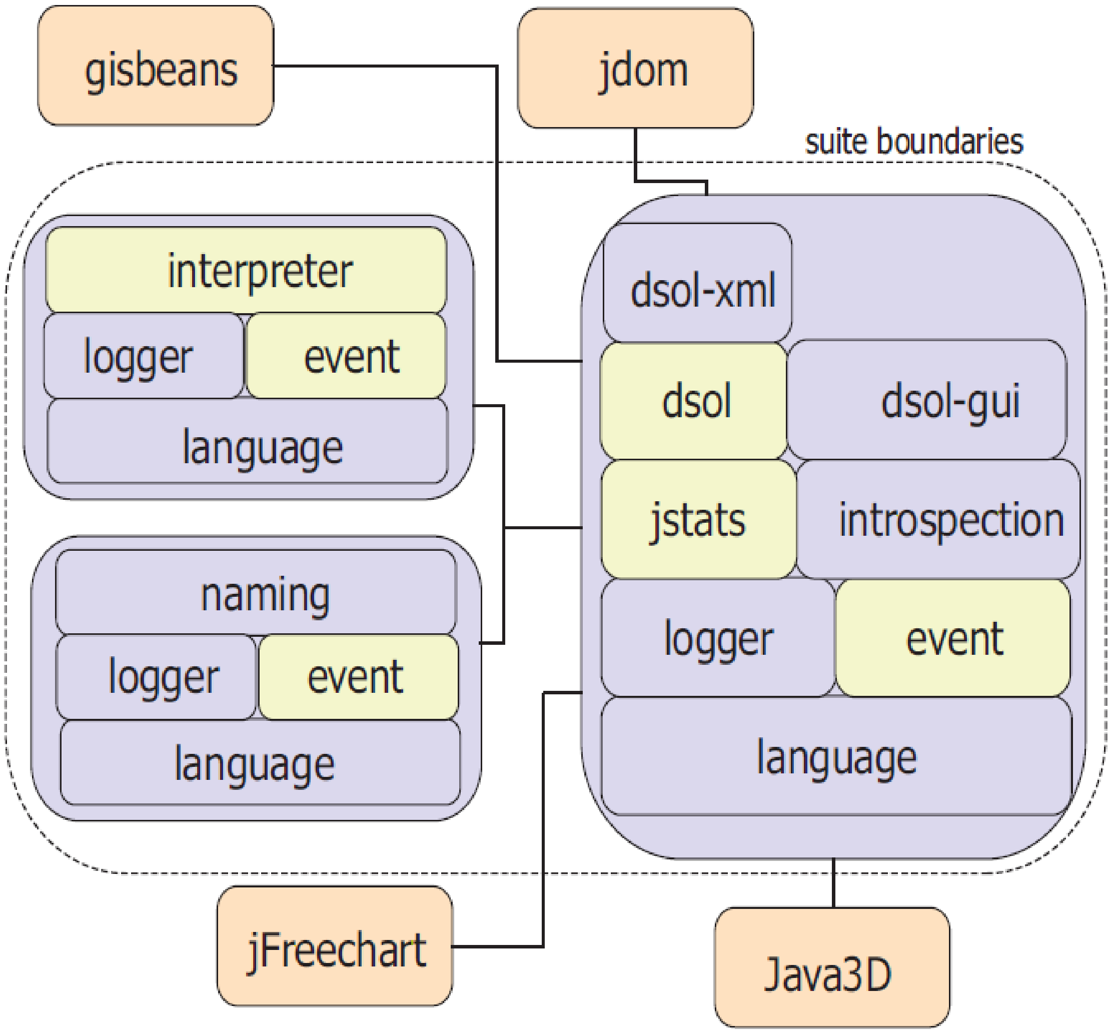
DSOL

- Programming environment in Java with libraries
- Event scheduling, continuous modeling, process interaction
- Explicit DES, DEVS, DESS, DEV&DESS, DSDEVS, etc.
- Mixed model possible
- Animation, optimization, statistics included
- Embedding and extension possible: open, public domain software

The 3 main requirements for DSOL

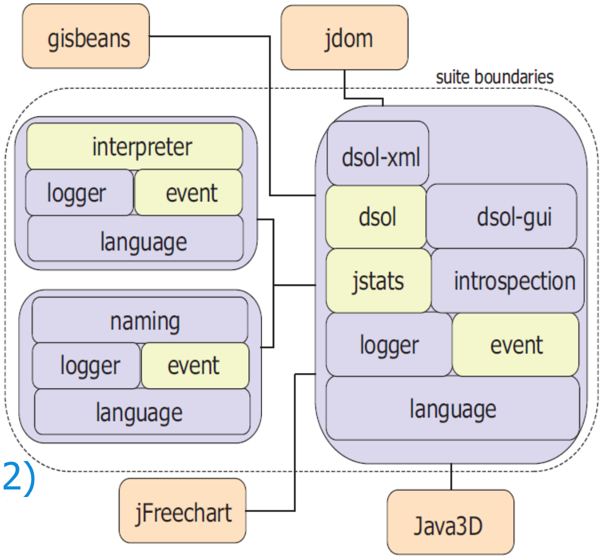
- **Distributing** the framework for modeling and simulation.
- Providing enough **formalisms** for the construction of models
- Implementing in a **service oriented** architecture.

Services of DSOL



DSOL service

- The DSOL service is the core simulation service.
- The core service provides a set of interfaces and classes for simulation. It provides a set of interfaces and classes for simulation.
- This service contains discrete and continuous formalisms, the specification of the DSOL experiment, continuous and discrete distributions, statistics and classes supporting 2-dimensional and 3-dimensional animation.

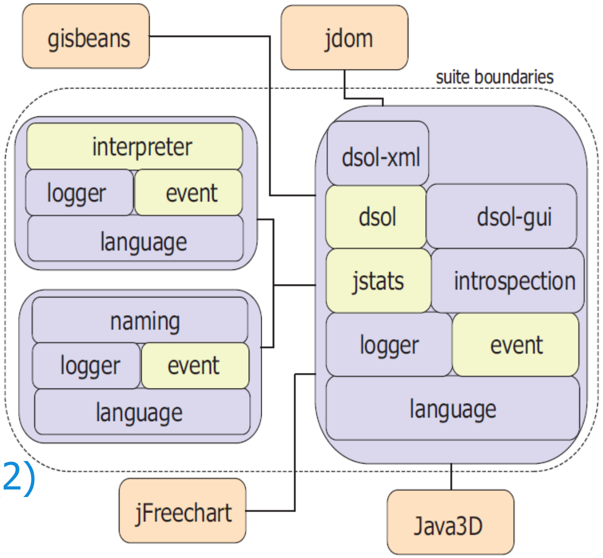


DSOL-GUI service

- The collection of classes which provide a web enabled graphical user interface for DSOL.

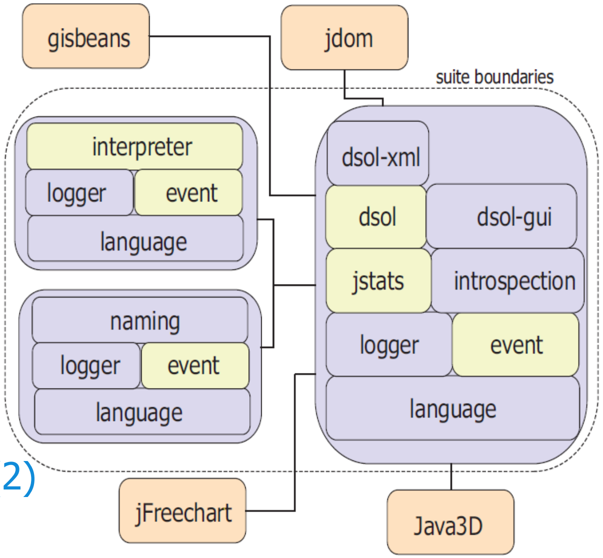
- We emphasize the importance of designing a good user interface to separate environments for model development and for model execution, but this is not part of DSOL.

- A reference implementation of a web enabled user interface is presented with this service.



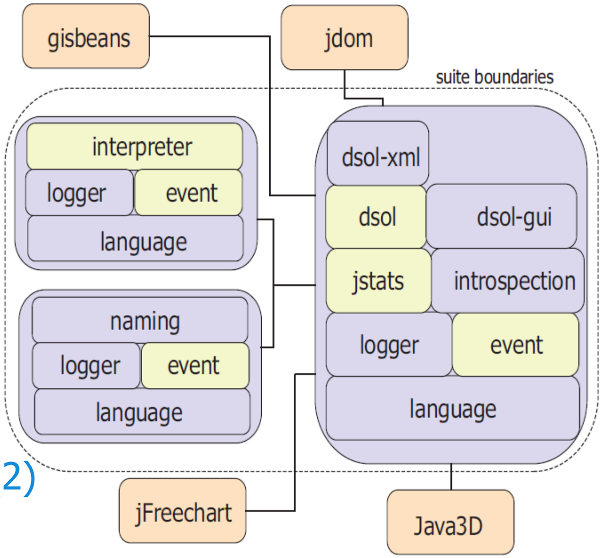
DSOL-XML service

- Over the last decade, XML has become the lingua franca for the configuration of applications.
- XML is namely a platform independent, human readable language.
- The DSOL-XML service provides parsers for the DSOL experimental frame and as such enables users to specify an experimental frame in XML.
- The value of this service is that it enables the specification of experiments without having knowledge of the Java programming language.



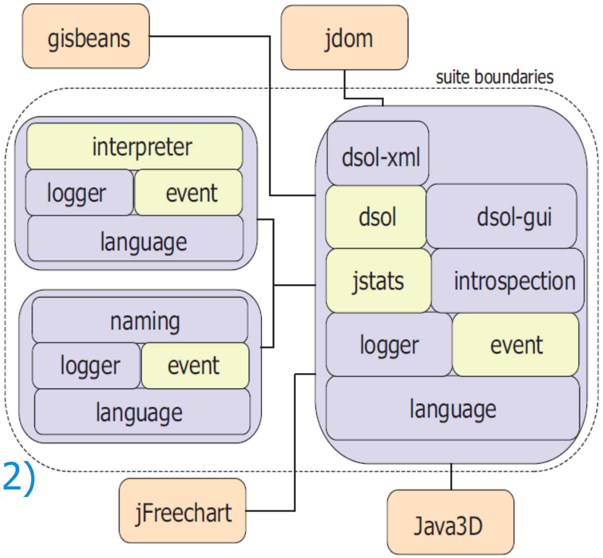
Interpreter service

- The specification of the process interaction formalism should not be based on Java threads.
- To specify the process interaction formalism in Java we have developed the interpreter service: a Java virtual machine implemented in Java.



JSTATS Service

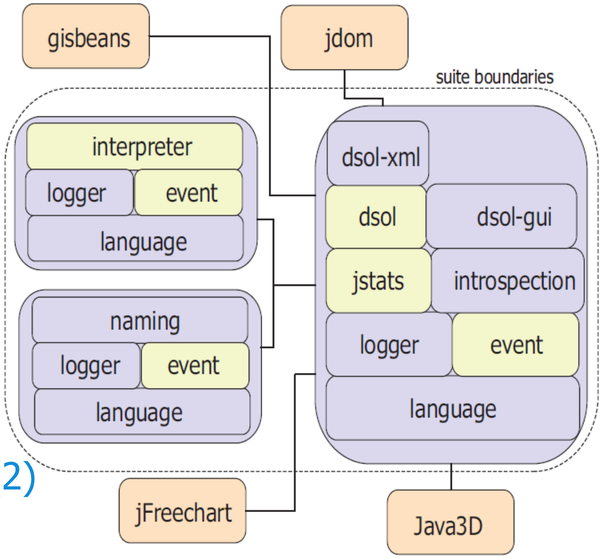
- This service provides a set of continuous and discrete distribution functions and links DSOL to external mathematical and chart libraries.



Naming service

- The naming service provides Yellow Page functionality to the DSOL suite.

- The naming service provides this functionality both to simulation model objects and to those objects constituting the DSOL suite.

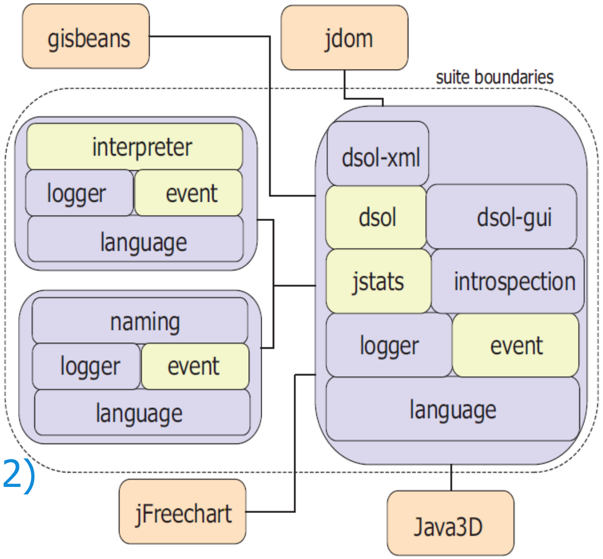


Introspection service

- This service provides an introspection service to users; the service enables users to open a simulation model object and to introspect, i.e. to see and change, attribute values through a graphical user interface.

- The value of the introspection service is that it provides the ability to drill down into simulation objects.

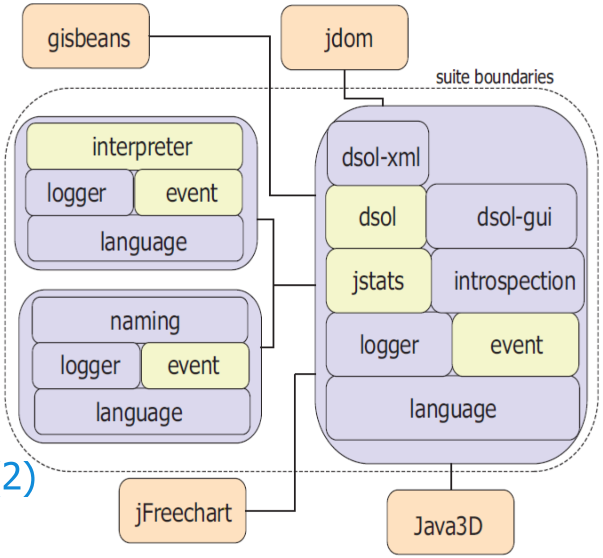
- This service aims to improve operational insight in the output of experimentation.



Event service

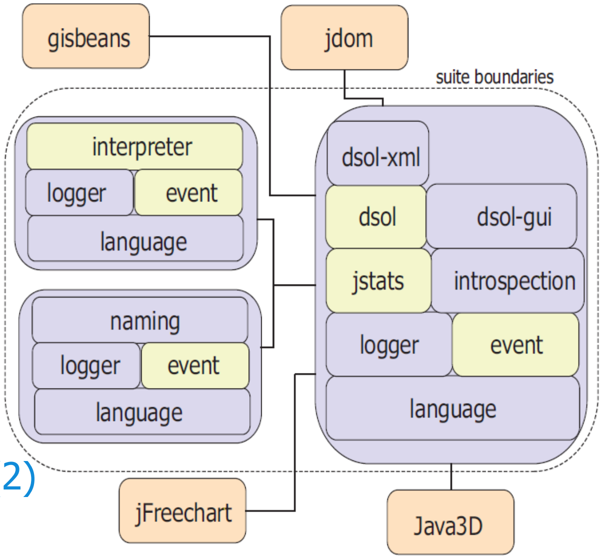
- This service provides a distributed asynchronous event mechanism.

- The value of the event service is that it enables loosely coupled relations between objects in the suite.

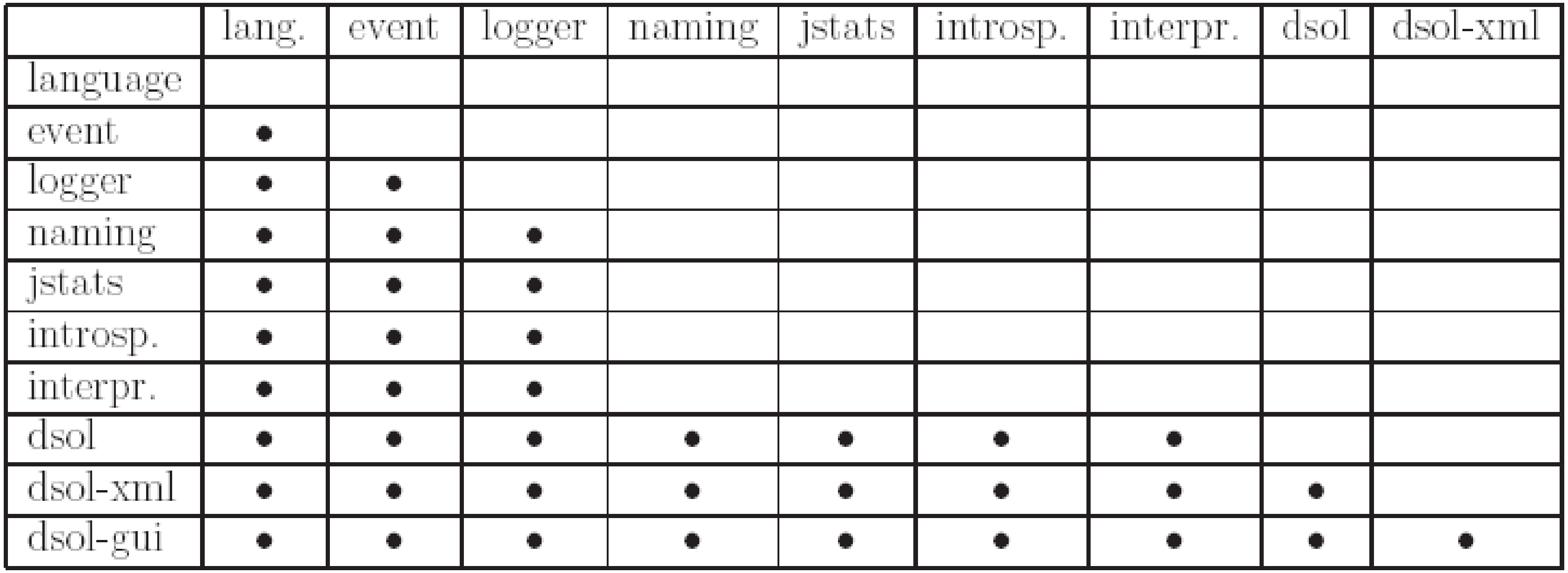


Logger service

- The DSOL simulation suite contains a logger service which is based on Java's logging mechanism (Sun Microsystems, 2001a).
- The value of this service is that output, debug information, warnings, etc. produced by objects in the suite are captured and, after they are filtered and formatted, presented to subscribed listeners.
- DSOL's Logger service provides a set of filters and formatters to provide distributed logging.

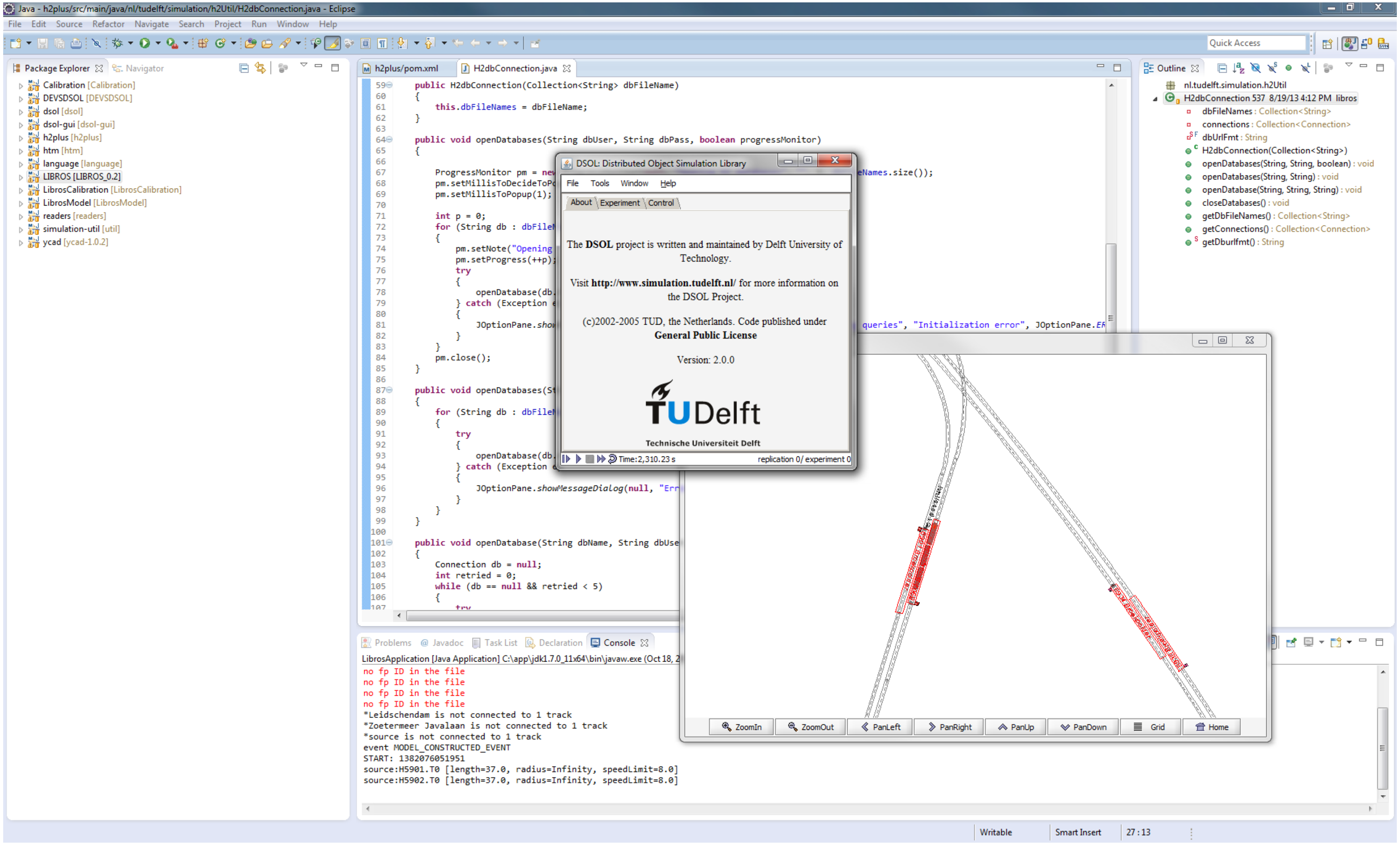


Interdependencies between the services



	lang.	event	logger	naming	jstats	introsp.	interpr.	dsol	dsol-xml
language									
event	●								
logger	●	●							
naming	●	●	●						
jstats	●	●	●						
introsp.	●	●	●						
interpr.	●	●	●						
dsol	●	●	●	●	●	●	●		
dsol-xml	●	●	●	●	●	●	●	●	
dsol-gui	●	●	●	●	●	●	●	●	●

DSOL: programming environment (Java)



Package Explorer:

```
Calibration [Calibration]
DEVSDSOL [DEVSDSOL]
dsol [dsol]
dsol-gui [dsol-gui]
h2plus [h2plus]
htm [htm]
language [language]
LIBROS [LIBROS_0.2]
LibrosCalibration [LibrosCalibration]
LibrosModel [LibrosModel]
readers [readers]
simulation-util [util]
ycad [ycad-1.0.2]
```

```java
    public H2dbConnection(Collection<String> dbFileName)
    {
        this.dbFileNames = dbFileName;
    }

    public void openDatabases(String dbUser, String dbPass, boolean progressMonitor)
    {

        ProgressMonitor pm = new                                              eNames.size());
        pm.setMillisToDecideToPo
        pm.setMillisToPopup(1);

        int p = 0;
        for (String db : dbFileN
        {
            pm.setNote("Opening
            pm.setProgress(++p);
            try
            {
                openDatabase(db.
            } catch (Exception e
            {
                JOptionPane.show                                               queries", "Initialization error", JOptionPane.ER
            }
        }
        pm.close();
    }

    public void openDatabases(St
    {
        for (String db : dbFileN
        {
            try
            {
                openDatabase(db.
            } catch (Exception e
            {
                JOptionPane.showMessageDialog(null, "Err
            }
        }
    }

    public void openDatabase(String dbName, String dbUse
    {
        Connection db = null;
        int retried = 0;
        while (db == null && retried < 5)
        {
            try
```

Outline:

```
nl.tudelft.simulation.h2Util
H2dbConnection 537  8/19/13 4:12 PM  libros
    dbFileNames : Collection<String>
    connections : Collection<Connection>
    dbUrlFmt : String
    H2dbConnection(Collection<String>)
    openDatabases(String, String, boolean) : void
    openDatabases(String, String) : void
    openDatabase(String, String, String) : void
    closeDatabases() : void
    getDbFileNames() : Collection<String>
    getConnections() : Collection<Connection>
    getDburlfmt() : String
```

The **DSOL** project is written and maintained by Delft University of Technology.

Visit **http://www.simulation.tudelft.nl/** for more information on the DSOL Project.

(c)2002-2005 TUD, the Netherlands. Code published under **General Public License**

Version: 2.0.0

Time:2,310.23 s   replication 0/ experiment 0

```
LibrosApplication [Java Application] C:\app\jdk1.7.0_11x64\bin\javaw.exe (Oct 18, 2
no fp ID in the file
no fp ID in the file
no fp ID in the file
no fp ID in the file
*Leidschendam is not connected to 1 track
*Zoetermeer Javalaan is not connected to 1 track
*source is not connected to 1 track
event MODEL_CONSTRUCTED_EVENT
START: 1382076051951
source:H5901.T0 [length=37.0, radius=Infinity, speedLimit=8.0]
source:H5902.T0 [length=37.0, radius=Infinity, speedLimit=8.0]
```

Conclusions

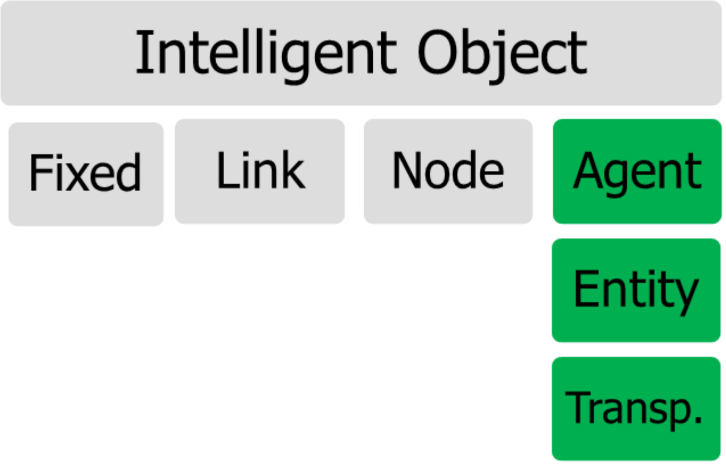
- DSOL is a Java based distributed application.
- DSOL is an object-oriented simulation framework for distributed modeling.
- DSOL supports several formalisms among which DES, DESS, DEVS, DEV&DESS.
- DSOL is open source and published under BSD on github

9.

Simio

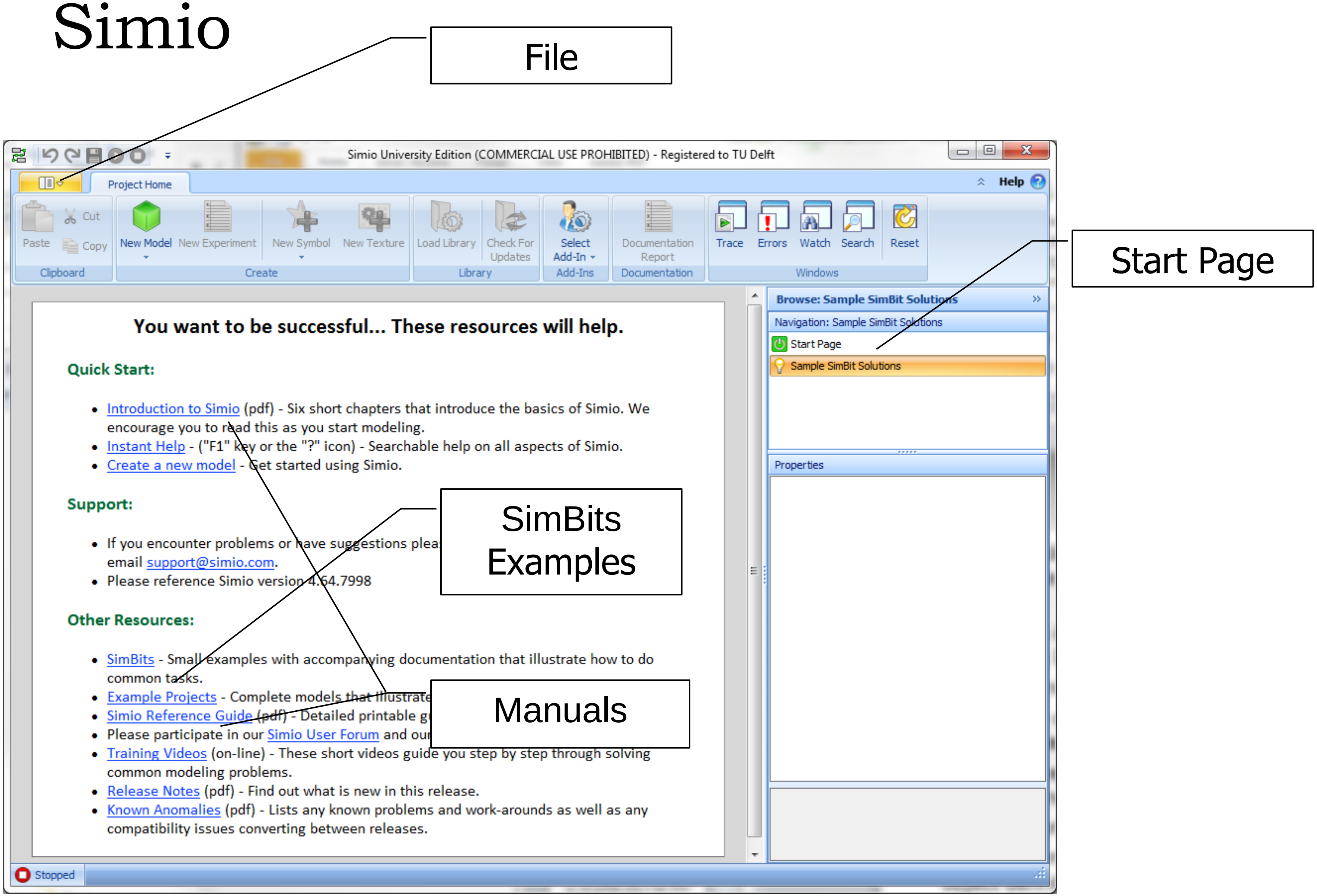
Simio

- Object oriented (intelligent objects -> agents)
- Mix object (simple) and process (flexible) paradigms within the same model.
- Every object is a model; every model is an object



- Strong animation; 3D objects can be imported from the Google 3D Warehouse

Simio



You want to be successful... These resources will help.

Quick Start:

- Introduction to Simio (pdf) - Six short chapters that introduce the basics of Simio. We encourage you to read this as you start modeling.
- Instant Help - ("F1" key or the "?" icon) - Searchable help on all aspects of Simio.
- Create a new model - Get started using Simio.

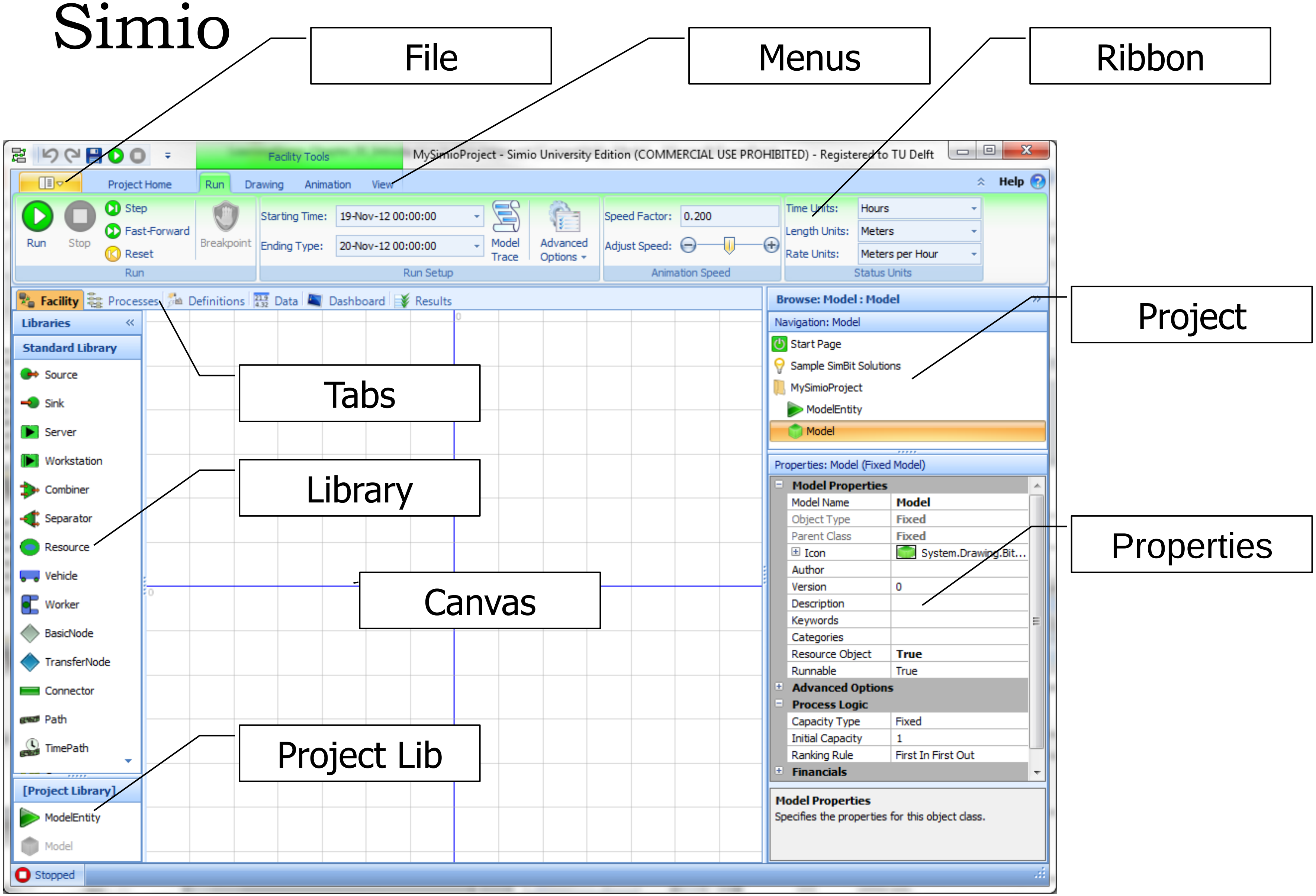
Support:

- If you encounter problems or have suggestions plea[...] email support@simio.com.
- Please reference Simio version 4.64.7998

Other Resources:

- SimBits - Small examples with accompanying documentation that illustrate how to do common tasks.
- Example Projects - Complete models that illustrate
- Simio Reference Guide (pdf) - Detailed printable g
- Please participate in our Simio User Forum and our
- Training Videos (on-line) - These short videos guide you step by step through solving common modeling problems.
- Release Notes (pdf) - Find out what is new in this release.
- Known Anomalies (pdf) - Lists any known problems and work-arounds as well as any compatibility issues converting between releases.

Simio



File

Menus

Ribbon

Project

Tabs

Library

Canvas

Project Lib

Properties

Starting Time: 19-Nov-12 00:00:00

Ending Type: 20-Nov-12 00:00:00

Speed Factor: 0.200

Adjust Speed:

Time Units: Hours

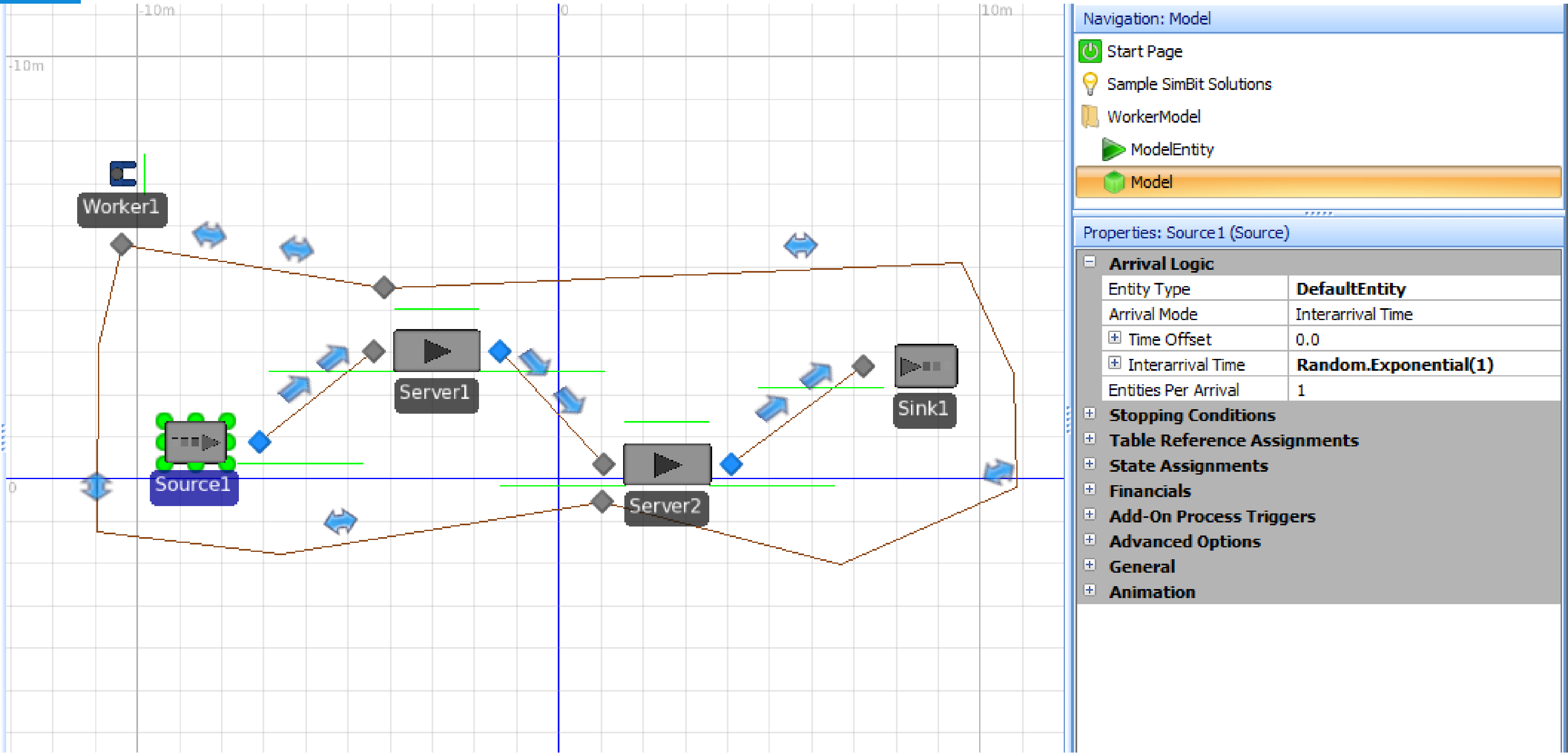
Length Units: Meters

Rate Units: Meters per Hour

Model Properties	
Model Name	**Model**
Object Type	Fixed
Parent Class	Fixed
Icon	System.Drawing.Bit…
Author	
Version	0
Description	
Keywords	
Categories	
Resource Object	**True**
Runnable	True
Advanced Options	
Process Logic	
Capacity Type	Fixed
Initial Capacity	1
Ranking Rule	First In First Out
Financials	

Model Properties
Specifies the properties for this object class.

Model with objects



Properties: Source1 (Source)

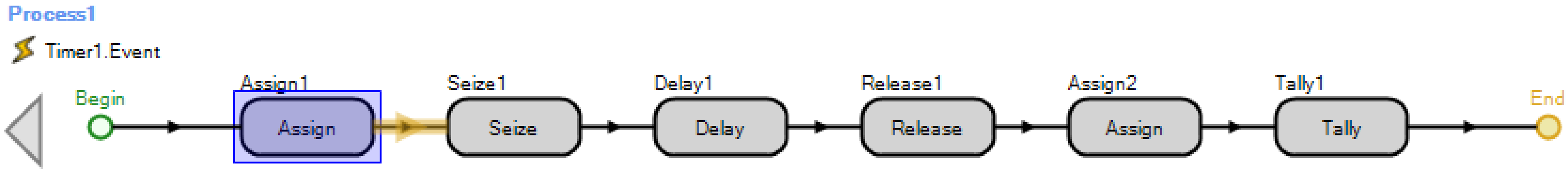
⊟ Arrival Logic

Entity Type	**DefaultEntity**
Arrival Mode	Interarrival Time
⊞ Time Offset	0.0
⊞ Interarrival Time	**Random.Exponential(1)**
Entities Per Arrival	1

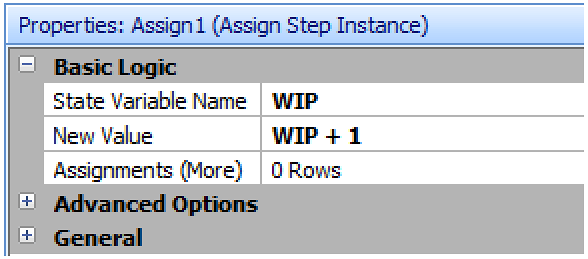
⊞ Stopping Conditions
⊞ Table Reference Assignments
⊞ State Assignments
⊞ Financials
⊞ Add-On Process Triggers
⊞ Advanced Options
⊞ General
⊞ Animation

Detailed process definitions

- Process description



- Logic for the different steps



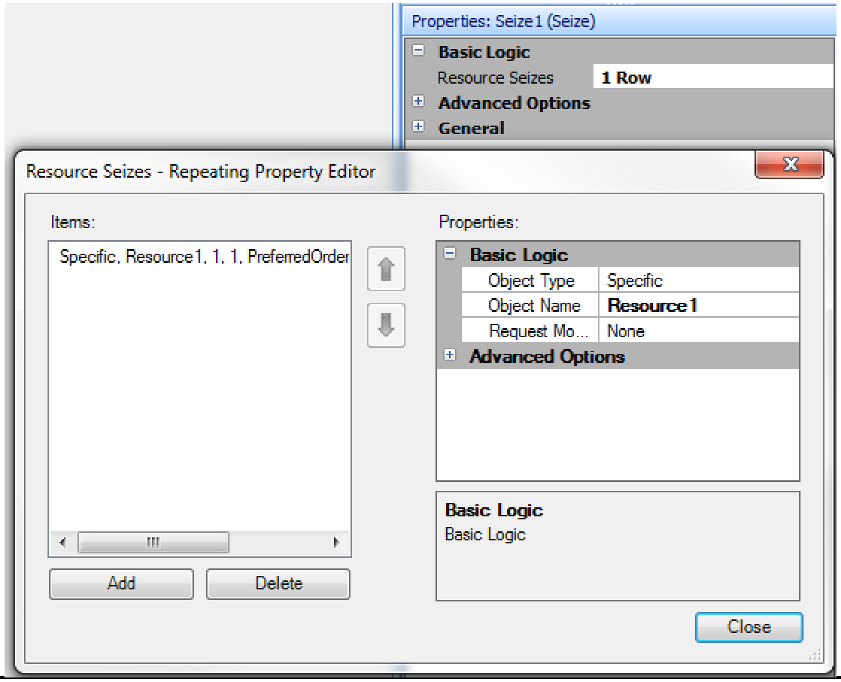
Properties: Assign1 (Assign Step Instance)

Basic Logic

State Variable Name	**WIP**
New Value	**WIP + 1**
Assignments (More)	0 Rows

Advanced Options

General



Properties: Seize1 (Seize)

Basic Logic

Resource Seizes	**1 Row**

Advanced Options

General

Resource Seizes - Repeating Property Editor

Items:

Specific, Resource1, 1, 1, PreferredOrder

Properties:

Basic Logic

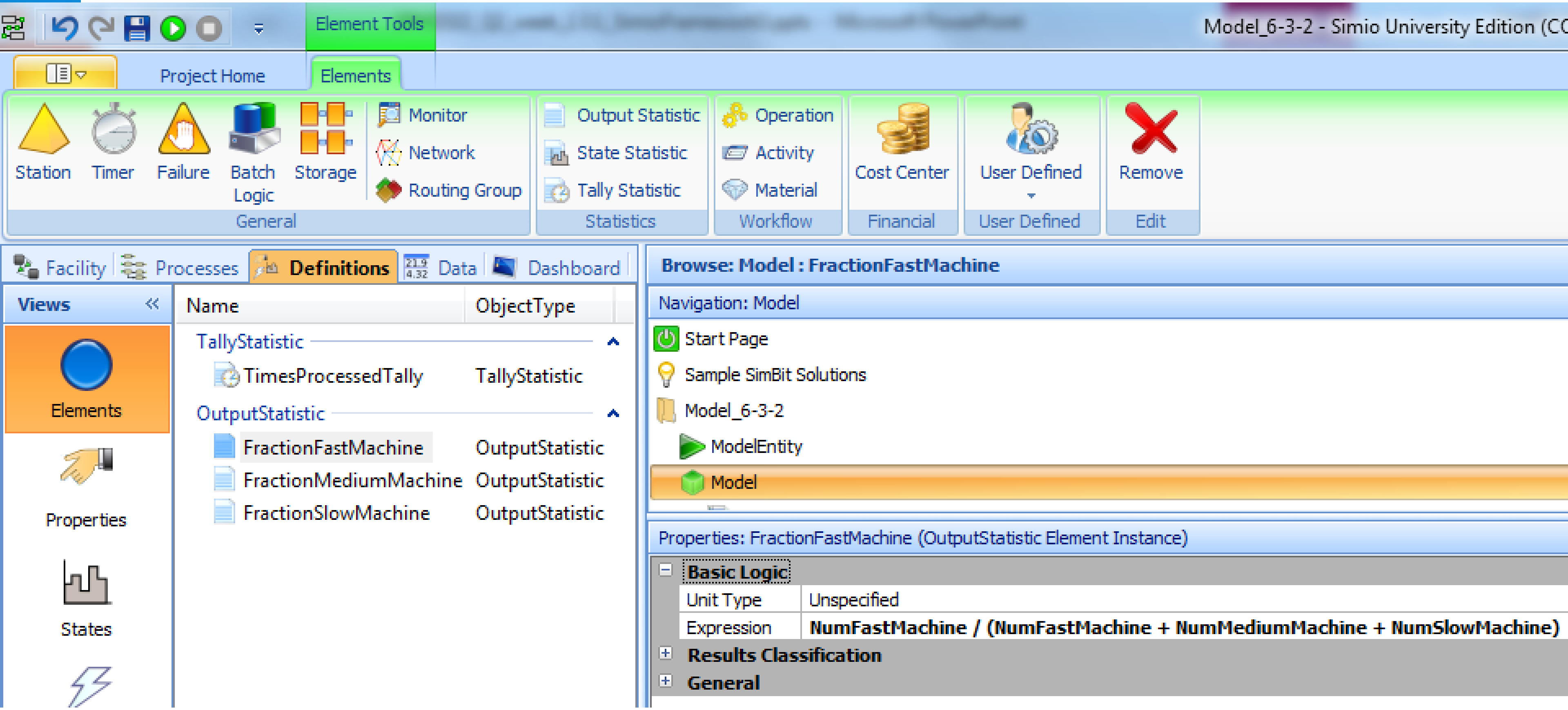
Object Type	Specific
Object Name	**Resource1**
Request Mo...	None

Advanced Options

Basic Logic

Basic Logic

Modern interface



Name	ObjectType
TallyStatistic	
TimesProcessedTally	TallyStatistic
OutputStatistic	
FractionFastMachine	OutputStatistic
FractionMediumMachine	OutputStatistic
FractionSlowMachine	OutputStatistic

Browse: Model : FractionFastMachine

Navigation: Model

- Start Page
- Sample SimBit Solutions
- Model_6-3-2
  - ModelEntity
  - Model

Properties: FractionFastMachine (OutputStatistic Element Instance)

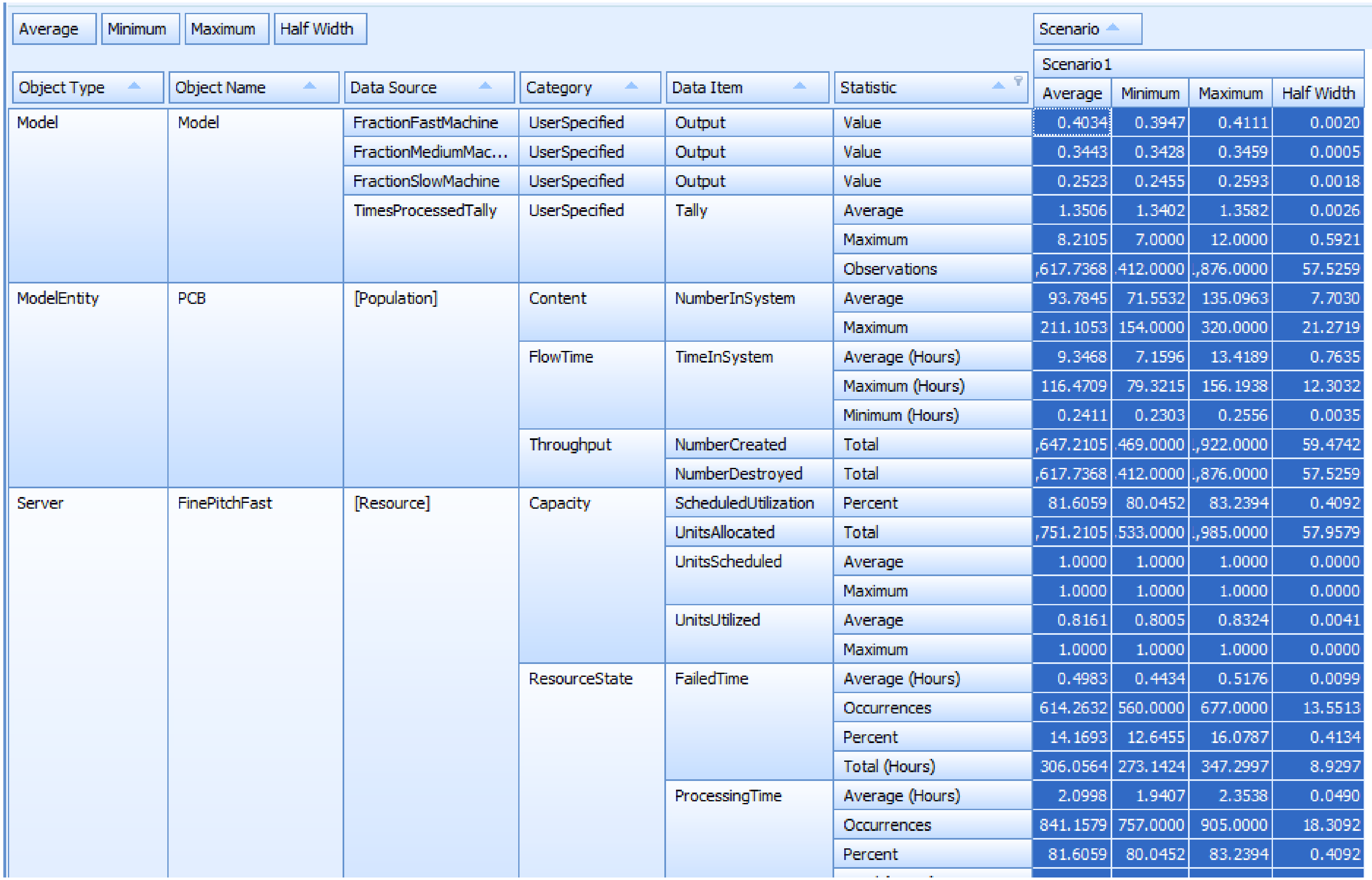
⊟ Basic Logic

Unit Type	Unspecified
Expression	NumFastMachine / (NumFastMachine + NumMediumMachine + NumSlowMachine)

⊞ Results Classification

⊞ General

Strong experimentation and results

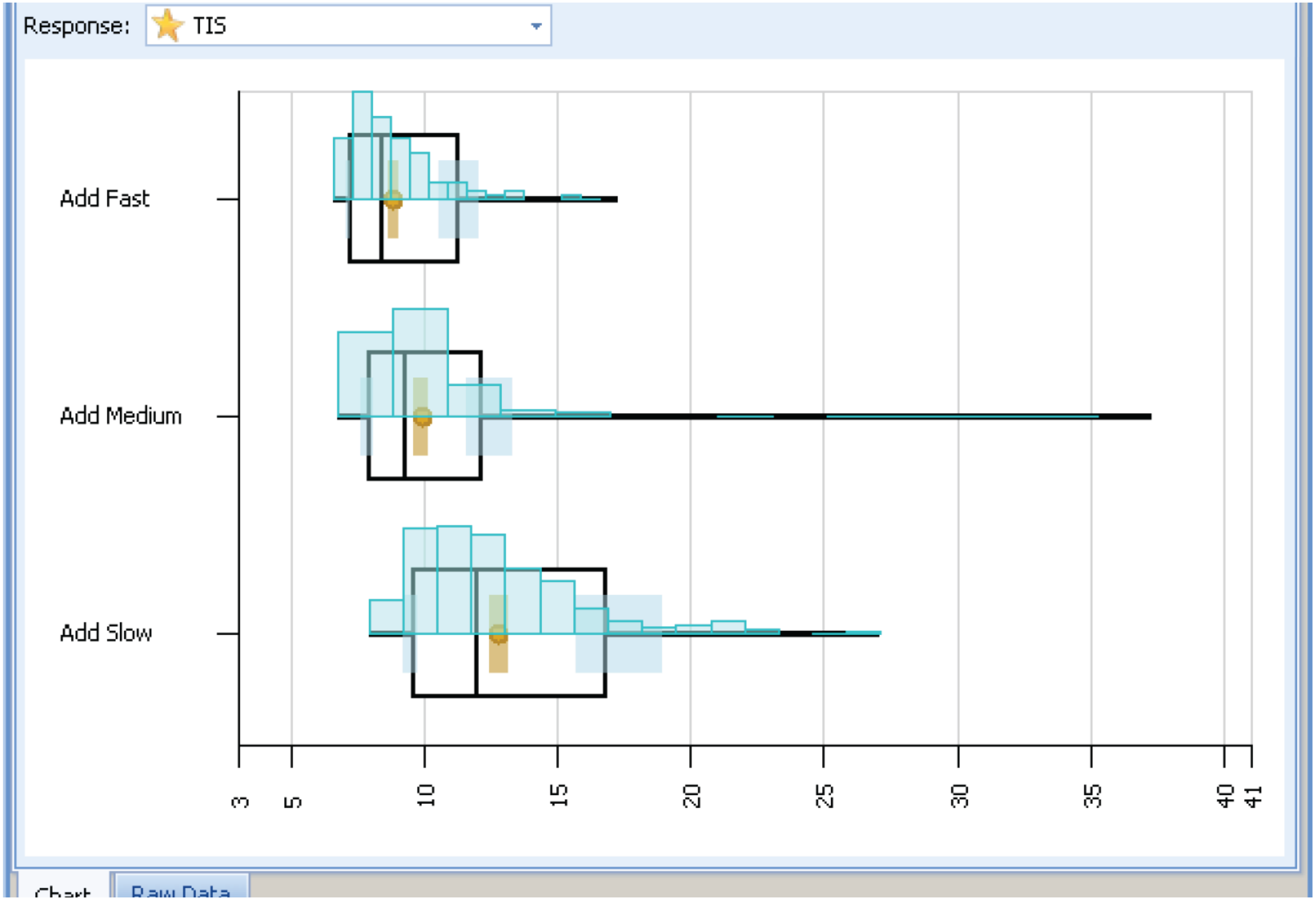


Average | Minimum | Maximum | Half Width

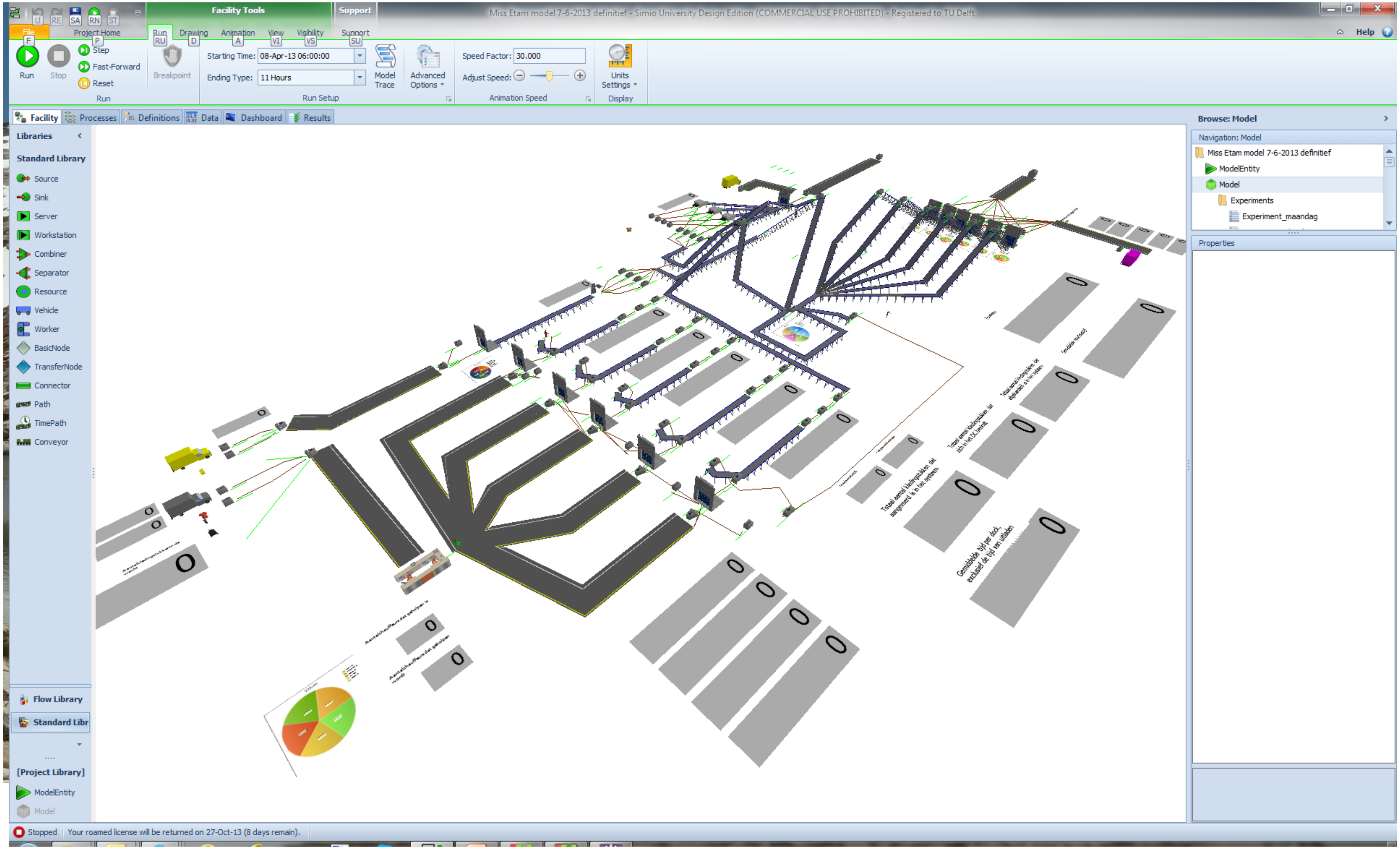
Scenario

Object Type	Object Name	Data Source	Category	Data Item	Statistic	Scenario1			
						Average	Minimum	Maximum	Half Width
Model	Model	FractionFastMachine	UserSpecified	Output	Value	0.4034	0.3947	0.4111	0.0020
		FractionMediumMac...	UserSpecified	Output	Value	0.3443	0.3428	0.3459	0.0005
		FractionSlowMachine	UserSpecified	Output	Value	0.2523	0.2455	0.2593	0.0018
		TimesProcessedTally	UserSpecified	Tally	Average	1.3506	1.3402	1.3582	0.0026
					Maximum	8.2105	7.0000	12.0000	0.5921
					Observations	,617.7368	.412.0000	.,876.0000	57.5259
ModelEntity	PCB	[Population]	Content	NumberInSystem	Average	93.7845	71.5532	135.0963	7.7030
					Maximum	211.1053	154.0000	320.0000	21.2719
			FlowTime	TimeInSystem	Average (Hours)	9.3468	7.1596	13.4189	0.7635
					Maximum (Hours)	116.4709	79.3215	156.1938	12.3032
					Minimum (Hours)	0.2411	0.2303	0.2556	0.0035
			Throughput	NumberCreated	Total	,647.2105	.469.0000	.,922.0000	59.4742
				NumberDestroyed	Total	,617.7368	.412.0000	.,876.0000	57.5259
Server	FinePitchFast	[Resource]	Capacity	ScheduledUtilization	Percent	81.6059	80.0452	83.2394	0.4092
				UnitsAllocated	Total	,751.2105	.533.0000	.,985.0000	57.9579
				UnitsScheduled	Average	1.0000	1.0000	1.0000	0.0000
					Maximum	1.0000	1.0000	1.0000	0.0000
				UnitsUtilized	Average	0.8161	0.8005	0.8324	0.0041
					Maximum	1.0000	1.0000	1.0000	0.0000
			ResourceState	FailedTime	Average (Hours)	0.4983	0.4434	0.5176	0.0099
					Occurrences	614.2632	560.0000	677.0000	13.5513
					Percent	14.1693	12.6455	16.0787	0.4134
					Total (Hours)	306.0564	273.1424	347.2997	8.9297
				ProcessingTime	Average (Hours)	2.0998	1.9407	2.3538	0.0490
					Occurrences	841.1579	757.0000	905.0000	18.3092
					Percent	81.6059	80.0452	83.2394	0.4092

Strong experimentation and results



Extended models possible



Miss Etam model 7-6-2013 definitief - Simio University Design Edition (COMMERCIAL USE PROHIBITED) - Registered to TU Delft

Starting Time: 08-Apr-13 06:00:00

Ending Type: 11 Hours

Speed Factor: 30.000

Stopped Your roamed license will be returned on 27-Oct-13 (8 days remain).

10.

Salabim

Salabim

- Like DSOL, based on a 3GL, but in this case Python
- Relative young package

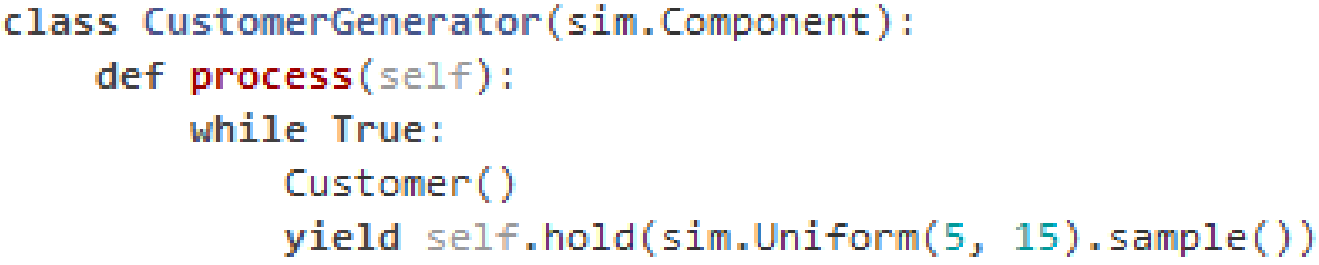
- Python is seen as an ideal 'prototyping' package
- But no good simulation package available (SimPy is very basic)

- Process interaction (locality of object) as the basis
- Just queueing systems

- Still very basic, but showing potential for the future

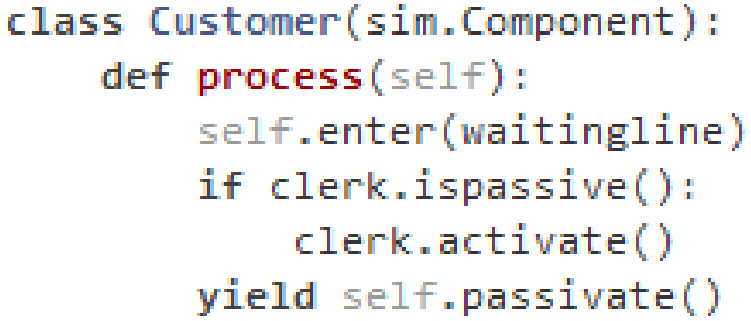
Basic concepts (1)

Generation of entities:



```
class CustomerGenerator(sim.Component):
    def process(self):
        while True:
            Customer()
            yield self.hold(sim.Uniform(5, 15).sample())
```

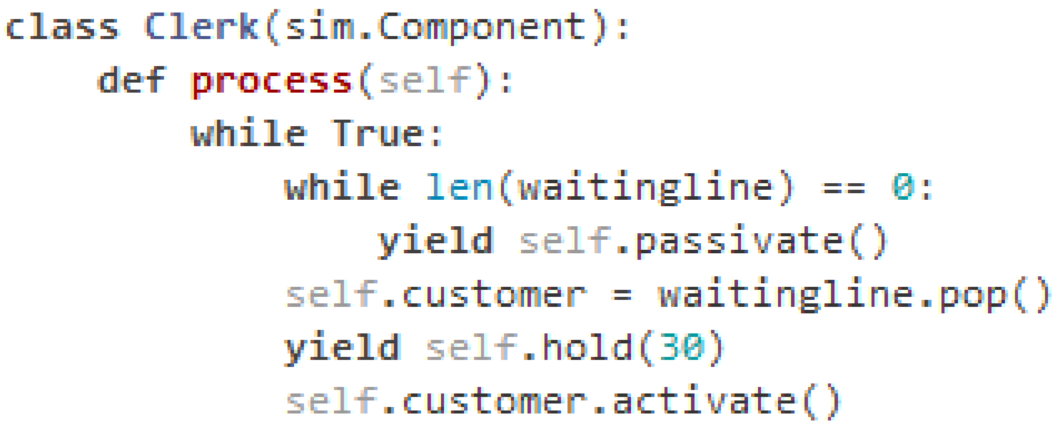
Customer process:



```
class Customer(sim.Component):
    def process(self):
        self.enter(waitingline)
        if clerk.ispassive():
            clerk.activate()
        yield self.passivate()
```

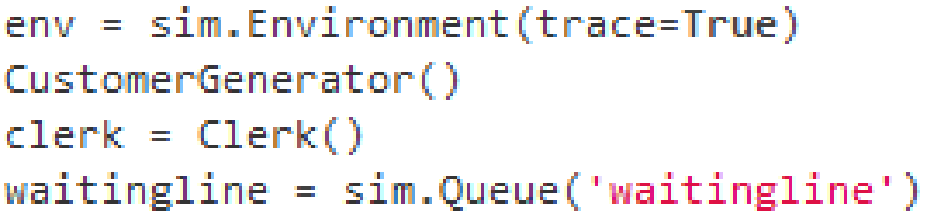
Basic concepts (2)

Resource (server) process:



```
class Clerk(sim.Component):
    def process(self):
        while True:
            while len(waitingline) == 0:
                yield self.passivate()
            self.customer = waitingline.pop()
            yield self.hold(30)
            self.customer.activate()
```

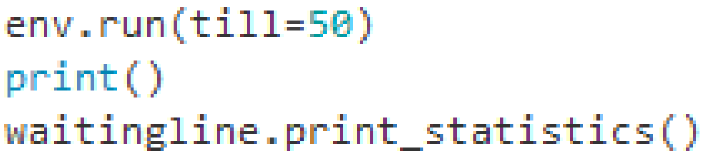
Main process:



```
env = sim.Environment(trace=True)
CustomerGenerator()
clerk = Clerk()
waitingline = sim.Queue('waitingline')
```

Basic concepts (3)

Experimental design:



```
env.run(till=50)
print()
waitingline.print_statistics()
```

So all very bare and basic...

Animation is present, but fully DIY

The interesting part is the fact that it uses process interaction

- synchronization between processes?

11.

Simulation Software Comparison and Selection

Comparison (1)

	Arena	Plant Sim	DSOL/pydsol
Locality	Time (Flow)	Time (Flow)	Time
Formalisms	Discrete	Discrete	Discrete, Cont, Agent
Hierarchy	Yes	Yes	Partly, programmed
Inheritance	No	Yes	Yes
Distribution	No	Yes	Yes, programmed
User coding	No (VBA)	Yes, script	100% Java/Python
Programming	No	Partly, script	Yes
2D animation	Easy	Easy	Yes, programmed
3D animation	Yes, difficult	Yes	Partly, libraries
Interaction	Yes	Yes	Yes, programmed
Optimization	OptQuest	Yes	Yes, libraries
Price	$$	$$$	Free
Vendor	Rockwell Software	Siemens	None

Comparison (2)

	AutoMod	ED	AnyLogic	Simio
Locality	Time	Time (Flow)	Time (Flow)	Time (Flow)
Formalisms	Discrete	Discrete	DES, SD, Agent	Discrete
Hierarchy	No	Yes	Yes	Yes
Inheritance	No	Yes	Yes	Yes
Distribution	No	Yes	Possible	No
User coding	Script	4D-script	Yes, Java	Yes, Process
Programming	No	No	Yes, Java	No
2D animation	Easy	Easy	Easy	Easy
3D animation	Easy	Easy	Yes	Easy, Google 3D
Interaction	Yes, libraries	Yes	Yes, Java	Yes, some
Optimization	AutoStat	OptQuest	OptQuest	OptQuest
Price	$$	$$	$$	$$
Vendor	Brooks	InControl	AnyLogic	Simio

Comparison (3)

	ExtendSim	Salabim	SimPy
Locality	Time (Flow)	Object	Time
Formalisms	Discrete Rate/DES	Discrete	Discrete
Hierarchy	Yes	No	No
Inheritance	No	By user	By user
Distribution	No	No	No
User coding	No	Full python	Full python
Programming	No	Full python	Full python
2D animation	Yes	Very basic	No
3D animation	Partly	No	No
Interaction	Yes, many	No, python	No, python
Optimization	Built-in	No, libraries	No, libraries
Price	$$	Free	Free
Vendor	Imagine That Inc.	None	None

Comparison (4)

	Jaamsim	MESA	NetLogo
Orientation	Time (Flow)	State	State
Formalisms	Discrete	Agent	Agent
Hierarchy	Yes	No	No
Inheritance	Partly	Programmed	Yes
Distribution	No	No	No
User coding	Yes, Java	Full python	Yes, NetLogo
Programming	Yes, Java	Full python	Java Extensions API
2D animation	Yes	Basic	Yes
3D animation	Yes	No	Yes, NetLogo 3D
Interaction	Yes	No, python	Yes, UI
Optimization	No, libraries	No, libraries	No, external
Price	Free	Free	Free
Vendor	None	None	None

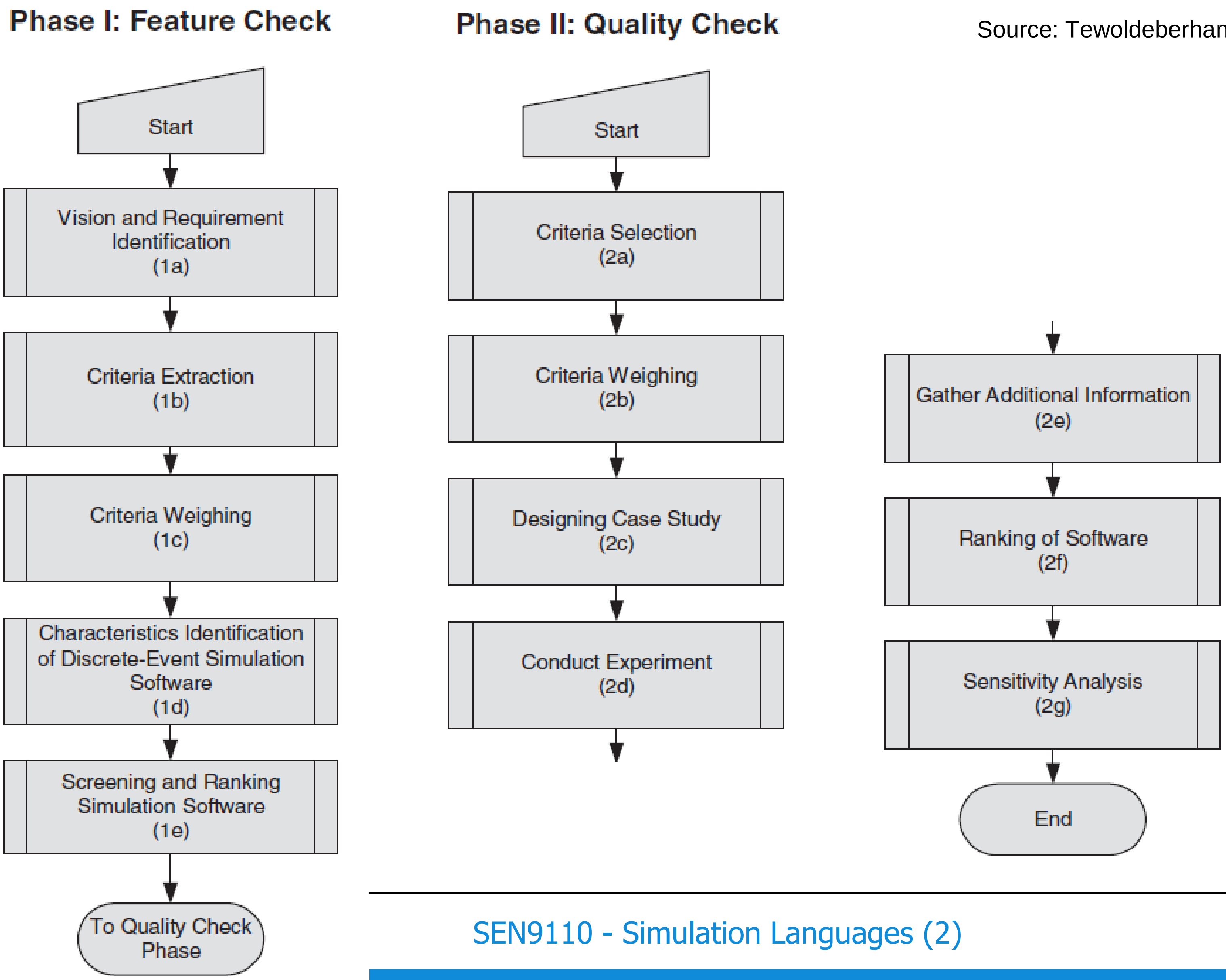
Simulation comparisons

- https://www.informs.org/ORMS-Today/Public-Articles/October-Volume-38-Number-5/Software-Survey-Simulation-Back-to-the-future

- http://www.orms-today.org/surveys/Simulation/Simulation.html

- And look on http://www.informs-sim.org for simulation software tutorials at the Winter Simulation Conferences

Simulation software selection


Phase I: Feature Check
Start
Vision and Requirement Identification (1a)
Criteria Extraction (1b)
Criteria Weighing (1c)
Characteristics Identification of Discrete-Event Simulation Software (1d)
Screening and Ranking Simulation Software (1e)
To Quality Check Phase
Phase II: Quality Check
Start
Criteria Selection (2a)
Criteria Weighing (2b)
Designing Case Study (2c)
Conduct Experiment (2d)
Gather Additional Information (2e)
Ranking of Software (2f)
Sensitivity Analysis (2g)
End

Source: Tewoldeberhan et al, 2010

Simulation software selection: Example criteria

Source: Tewoldeberhan et al, 2010

Model Development and Input

- Graphical model building
- Merging models
- Conditional routing
- Statistical distribution
- Queuing policies
- Reuse of user defined modules
- Built-in functions
- Link to other languages
- Coding tools and utilities
- Input from text files
- Input from database
- Input from spreadsheets
- Automatic data collection
- Batch input mode
- Interactive input mode
- Random number generators
- Program generator

Animation

- Integration of animation
- Library of icons
- Screen layout
- Concurrent animation mode
- Animation on/off feature
- 3D animation
- Animation development feature

Output

- Standard report generation
- Report customization
- Integration with statistical packages
- Integration with other simulation packages
- Exporting data to database
- Exporting data to spreadsheets
- Exporting data to text files or word processors
- Optimization
- Output analysis feature
- Business graphics

Simulation software selection: Example result

Source: Tewoldeberhan et al, 2010

Criteria	Weight	Package A	Package B	Package C	Package D	Package E
Vendor	5.6	3.00	2.00	2.67	3.00	2.33
Model development & input	9.5	2.71	2.57	2.71	2.00	2.43
Execution	7.6	2.00	2.33	2.33	2.00	2.00
Animation	6.3	2.67	2.33	1.33	2.67	1.00
Testing & efficiency	7.6	2.38	2.38	2.50	2.00	1.75
Output	6.6	2.33	1.67	2.33	2.00	2.67
Experimental design	5.9	3.00	2.00	2.00	3.00	2.00
User	5.6	2.00	1.50	2.50	2.00	3.00
Total		136.9	117.3	127.1	125.1	117.1
Rank		1	4	2	3	4

12.

Any Questions for the Term Papers?