Lecture 8: *Queuing Network Simulation Tool_JMT*

Prof. Hee Yong Youn
College of Software
Sungkyunkwan University
Suwon, Korea
youn7147@skku.edu

Overview

- **Java Modeling Tools (JMT) is a suite of applications developed by**
  - Politecnico di Milano (Dipartimento di Elettronica e Informazione), Italy
  - Released under GPL license, 2008
- **The project aims at offering a complete framework for**
  - Performance evaluation
  - System tuning
  - Capacity planning
  - Workload characterization study
- **Main features of JMT**
  - MVA technique
  - Modeling a queuing network
  - Markov chain

Installation Process

Software Requirements

- **Download all the components of the Java Modeling Tools (JMT) from the link below.**

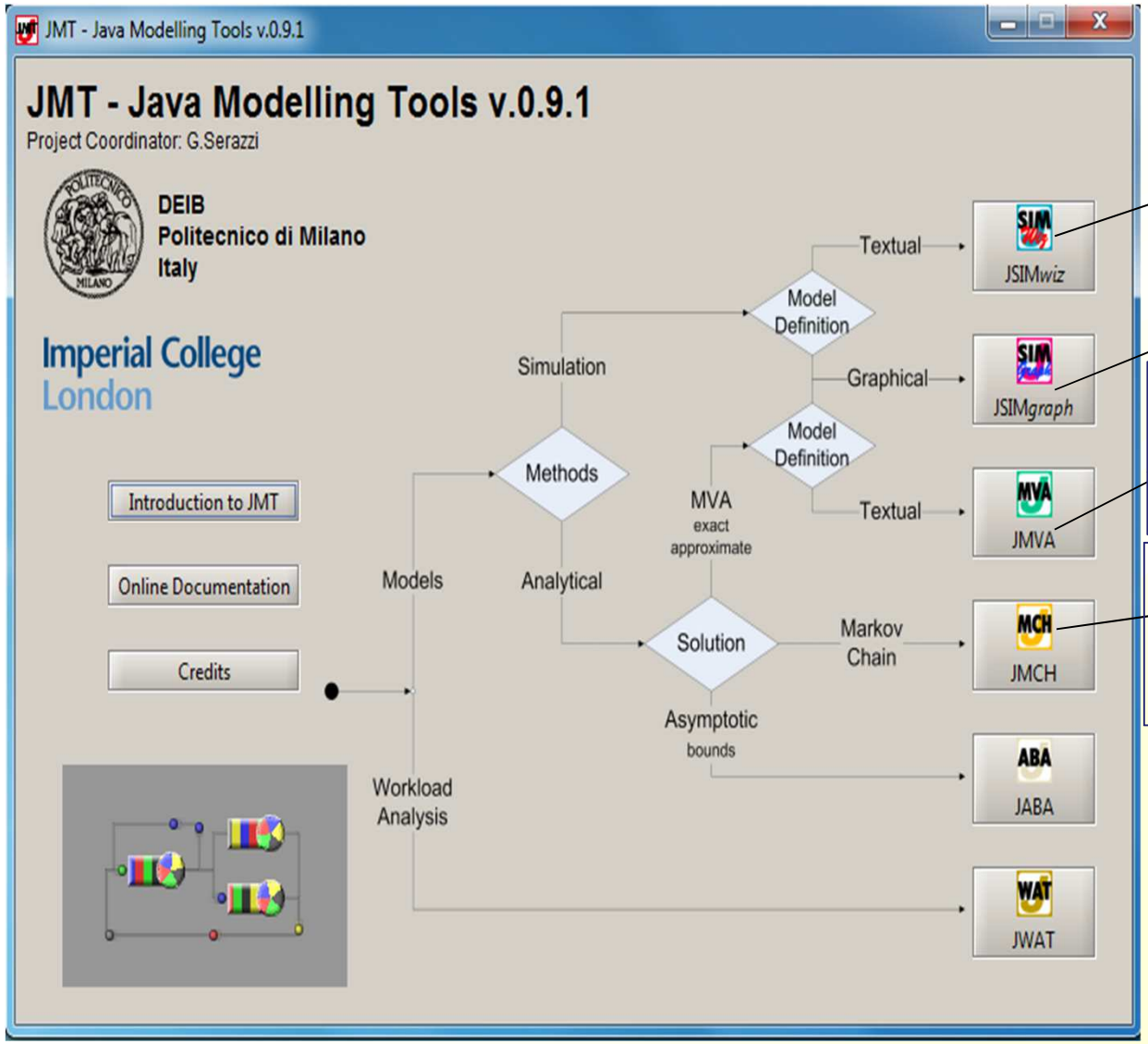
  JMT tool: http://jmt.sourceforge.net/Download.html or http://sourceforge.net/projects/jmt/files/jmt/JMT-0.9.1/JMT-installer-0.9.1.jar/download

- **JMT is platform-independent and requires only the Java Runtime Environment (version 1.6 or later).** The Java Runtime Environment can be downloaded from the link below.

  Java runtime environment: http://www.oracle.com/technetwork/java/javase/downloads/index.html

- **Install JMT and Java Runtime Environment on the PC.**

The Starting Screen of JMT Suite



wizard-based interface for the discrete-event simulator JSIM

A graphical user-friendly interface for the same simulator engine JSIM

For exact and approximate analysis of single-class or multiclass product form queuing networks, processing open, closed or mixed workloads.

It applies a simulation technique to solve a single station model with finite (M/M/1/k) or infinite queue (M/M/1)

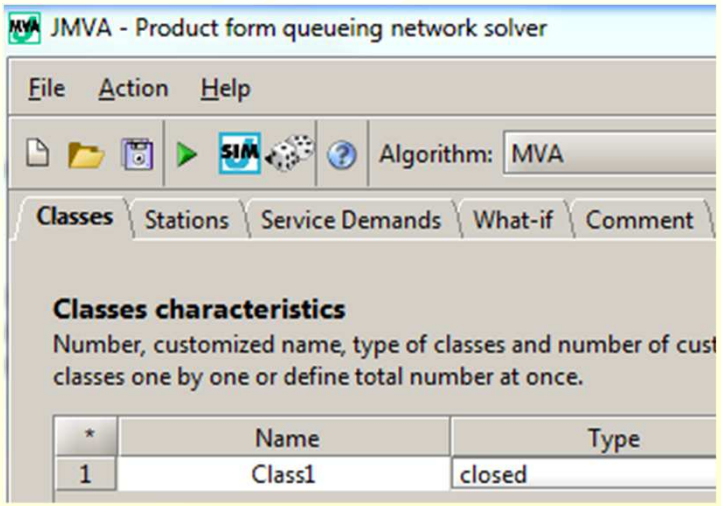
Core Algorithms - jMVA

- **Mean value analysis (MVA) algorithm**
  - Fast solution of product-form queueing networks
  - Open models: efficient solution in all cases
  - Closed models: efficient for models with up to 4-5 classes

- **Product-form queueing networks solvable by MVA**
  - PS/FCFS/LCFS/IS scheduling
  - Identical mean service time for multiclass FCFS
  - Mixed models (open + closed), load-dependent
  - Service at a queue does not depend on the state of other queues
  - No blocking, finite buffers, priorities

Model Definition

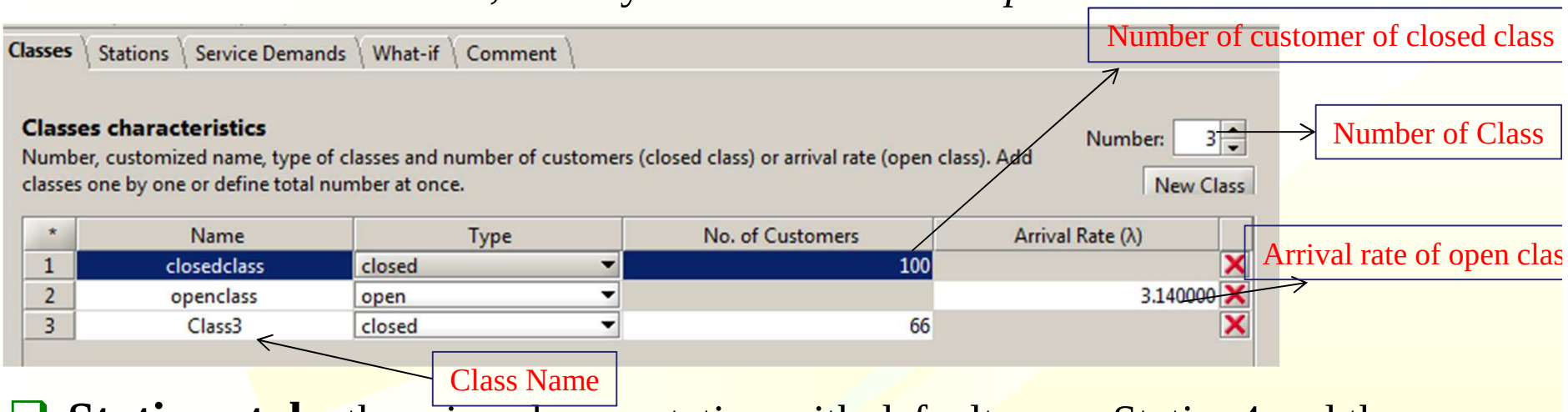
- **Parameters of a model:** The following parameters must be defined
  - Classes with their workload intensities (number of customer, N, for closed classes and arrival rate, λ, for open classes)
  - Stations (service centers)
  - Service demands (or service times and visits)
  - Optional short comment



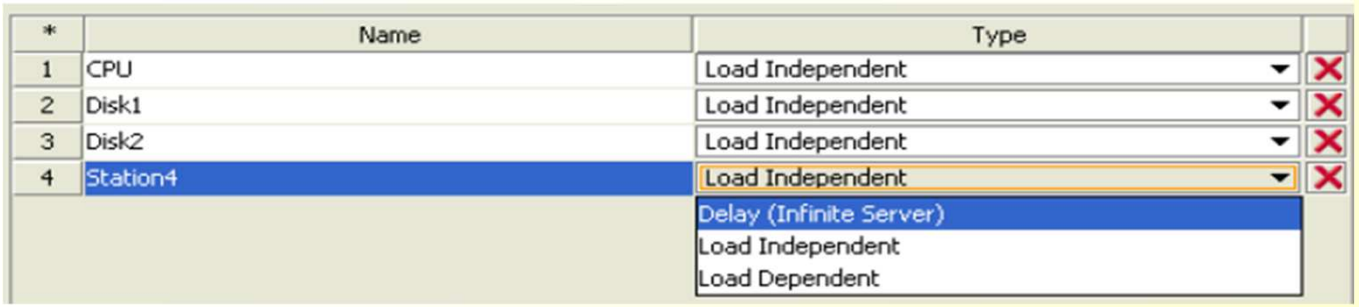
- **Performance indices (for each station)**
  - Stations (service centers)
  - Queue length
  - Throughput
  - Residence time
  - Utilization
  - System power (throughput/response time for *each class*)

Defining a Class and Stations

- **Class example:** There are three classes of customers, two closed and one open. The third class has the default name *Class3* while the other two classes have customized names, namely *ClosedClass* and *OpenClass*

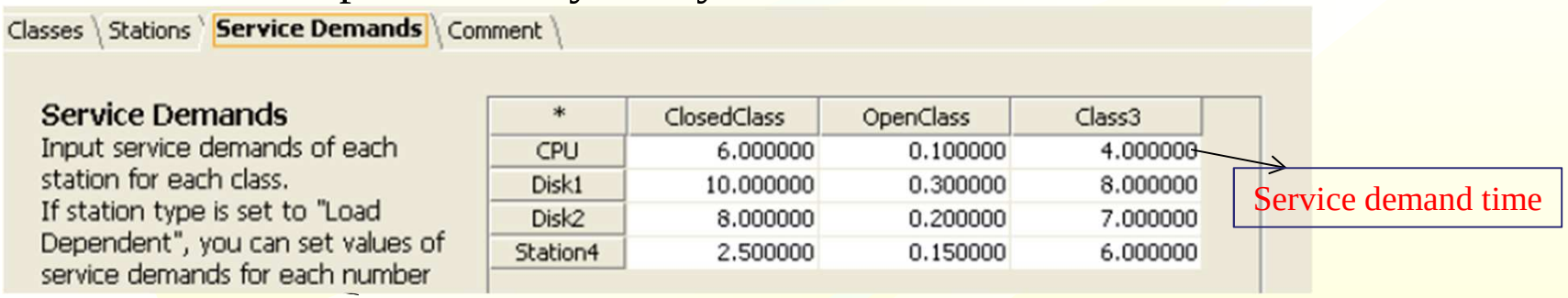


- **Stations tab:** there is only one station with default name Station4 and there are three stations with customized names: CPU, Disk1 and Disk2. A station type can be Load Independent, Load Dependent or Delay.



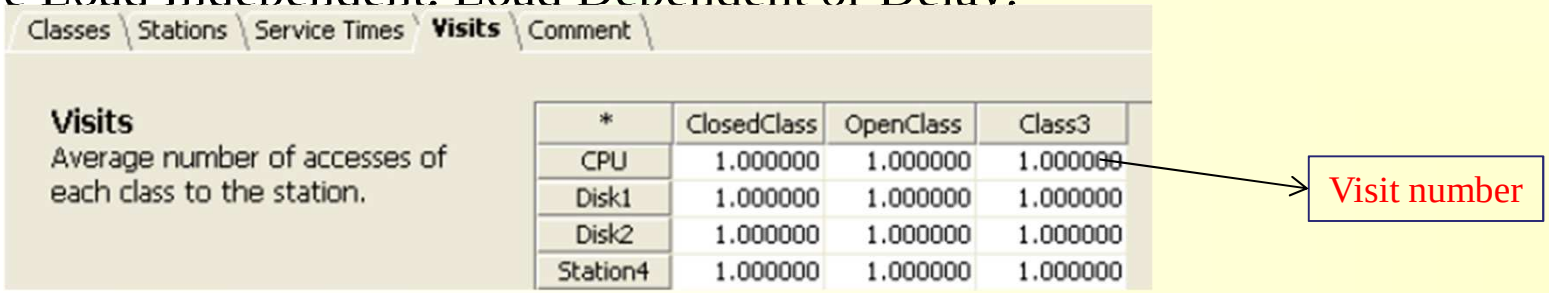
Defining Service Demands Tab

- **Service Demands Tab:** Each job of type *ClosedClass* requires an average service demand time of 6 sec to *CPU*, 10 sec to *Disk1*, 8 sec to *Disk2* and 2.5 sec to *Station4*. On the other hand, a job of type *OpenClass* requires on average 0.1s of *CPU* time, 0.3 sec of *Disk1* time, 0.2 sec of *Disk2* time and 0.15 sec of *Station4* time to be processed by the system.



*	ClosedClass	OpenClass	Class3
CPU	6.000000	0.100000	4.000000
Disk1	10.000000	0.300000	8.000000
Disk2	8.000000	0.200000	7.000000
Station4	2.500000	0.150000	6.000000

- **Visits Tabs:** there is only one station with default name Station4 and there are three stations with customized names: CPU, Disk1 and Disk2. A station type can be Load Independent, Load Dependent or Delay.



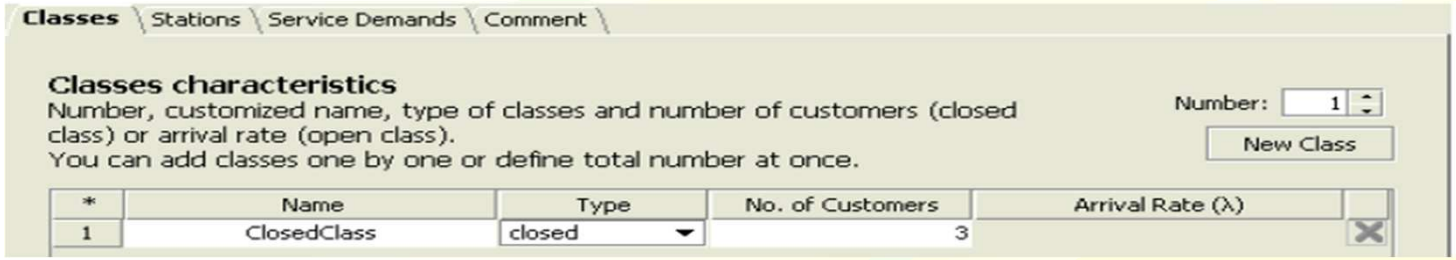
*	ClosedClass	OpenClass	Class3
CPU	1.000000	1.000000	1.000000
Disk1	1.000000	1.000000	1.000000
Disk2	1.000000	1.000000	1.000000
Station4	1.000000	1.000000	1.000000

Example of JMVA Technique

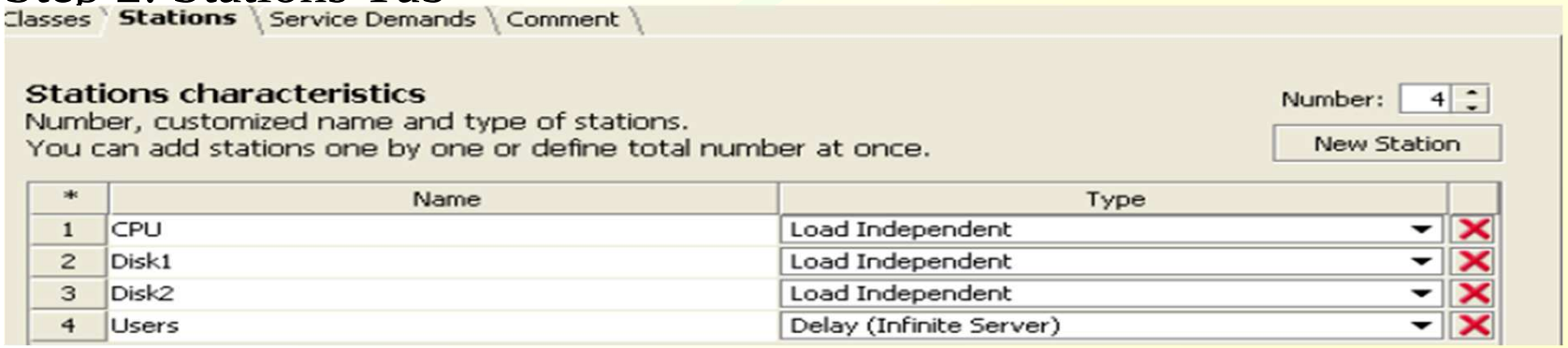
A single class closed model with three load independent stations and a delay service center. Number of population in closed class, N=3. and the service time and visit for station are given below:

	CPU	Disk1	Disk2	Users
Service Times [s]	0.006	0.038	0.030	16.000
Visits	101.000	60.000	40.000	1.000

Step 1: Define Class



Step 2: Stations Tab



Example of JMVA Technique

Step 3: Service time.

Classes | Stations | **Service Times** | Visits | Comment

Service Times
Input service times of each station for each class.
If station type is set to "Load Dependent", you can set values of service times for each number of

*	ClosedClass
CPU	0.006000
Disk1	0.038000
Disk2	0.030000
Users	16.000000

Step 4: Visits

Classes | Stations | Service Times | **Visits** | Comment

Visits
Average number of accesses of each class to the station.

*	ClosedClass
CPU	101.000000
Disk1	60.000000
Disk2	40.000000
Users	1.000000

Result of the Model

Throughput | Queue length | Residence Times | Utilization | Synopsis

Throughput
Throughput for each class at each station.

*	*Aggregate*	ClosedClass
Aggregate	0.143961	0.143961
CPU	14.540104	14.540104
Disk1	8.637686	8.637686
Disk2	5.758457	5.758457
Users	0.143961	0.143961

For the single-class model, all results in the column *Aggregate* are equal to the results in the *ClosedClass* column.

JSIM*graph* (Simulation - Graphical)

Main Features

- **Arrival rate:** for open classes of customers generated by source stations
- **Station service time***:* can be generated according to the following distributions
  - Burst (General, Modulated Markov Poisson Process MMPP2), Constant, Erlang, Exponential, Gamma, Hyperexponential, Normal, Pareto, Poisson, Student-T, Uniform
- **Queuing discipline:** supports First Come First Served, FCFS with priority, Last Come First Served, LCFS with priority
- **Routing** of the customers in the network: the path followed by the requests among the resources can be described either probabilistically or according to the following strategies:
  - Fastest service, Least utilization, Load Dependent routing, Random, Round robin, Join the Shortest Queue, Shortest response time.

Other features

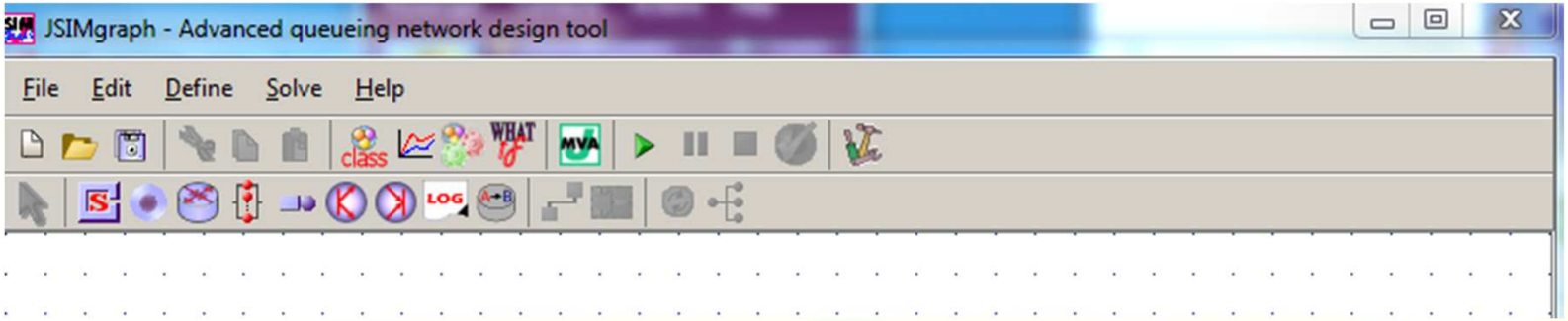
- Load dependent service time strategies
- Fork-and-join stations to model parallelism
- Simulation of complex traffic pattern and service times (e.g., burst)
- Blocking regions (in which the number of customer is limited)
- What-if analysis (with various control parameters)

Defining A New Model

- To define a new model the following steps have to be performed
  1. Draw the network (click and drop)
  2. Define *Customer Classes* and select the *Reference Station* for each class
  3. Set the parameters for each object
  4. Select the performance indices to be collected and evaluated
  5. If needed, insert one or more *Finite Capacity Regions* (FCR)
  6. Choose or change the simulation parameters
  7. Enable *What-If Analysis* and set its parameters, if required
  8. Start the simulation
  9. If an error is detected, click it. Then a window will be opened which allows immediate fix.

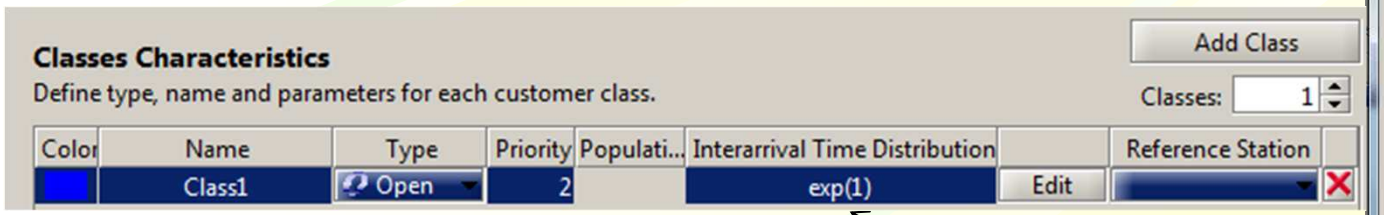
Defining A New Model (Cont'd)

Home view of JSIm

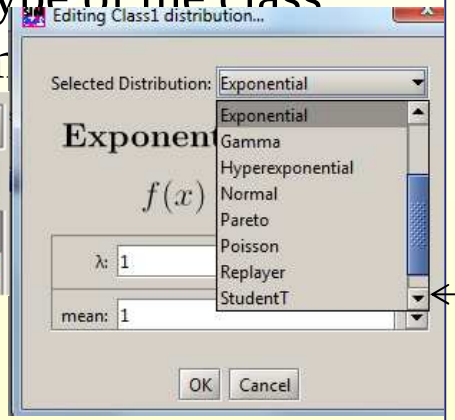


Defining Open Classes

- After adding a class and set its name and priority, the type of the class needs to be selected. Go to define tab and select custom



Exponential Distribution with λ =1



Editing Interval time distribution

Defining Open Classes

- The population size (also referred to as *N*) is the parameter that characterizes a closed class.

- A Reference Station for the class *must* be selected from the Reference Station menu

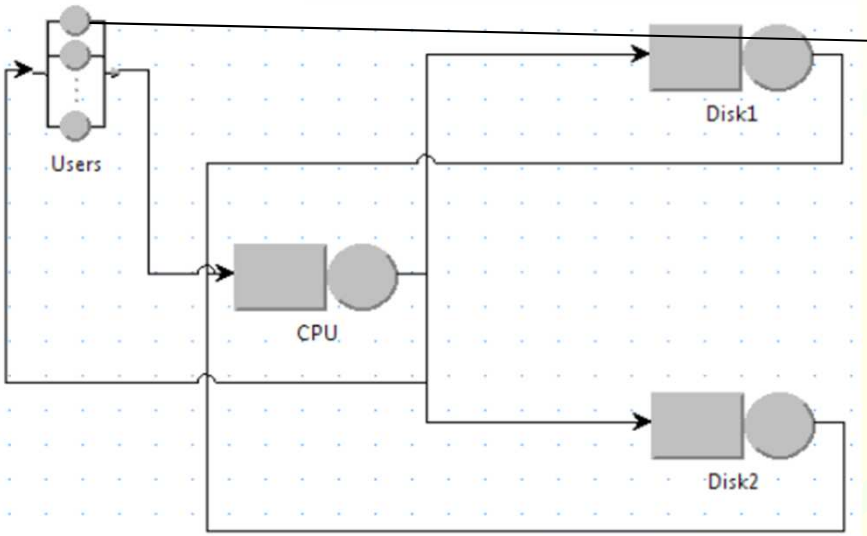
Performance Indices

- Number of Customers (of a station)
- Queue Time (of a station)
- Residence Time
- Response Time (of a station)
- Response Time per Sink
- Utilization (of a station)
- Throughput (of a station)
- Throughput per Sink
- System Throughput [X] (of the entire system)
- System Response Time [R](of the entire system)
- System Number of Customers (of the entire system)
- System Power (at system and per-class levels):

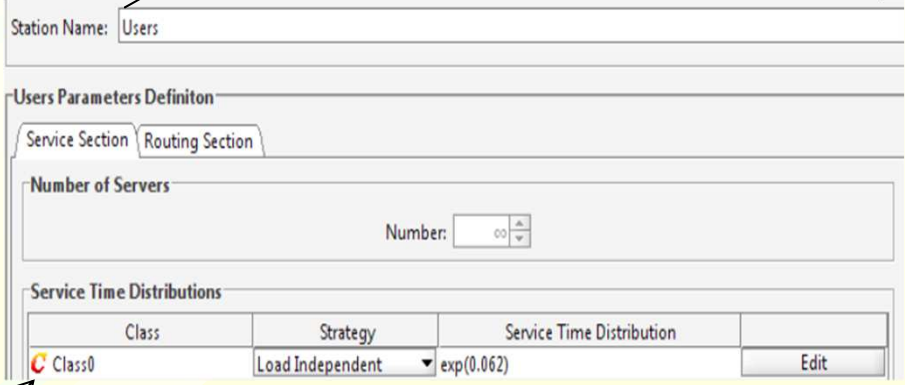
Example:

- **The previous example:** The users delay station represents user's think time (Z = 16s) between the interactions with the system.

Step 1: **Draw the graph**



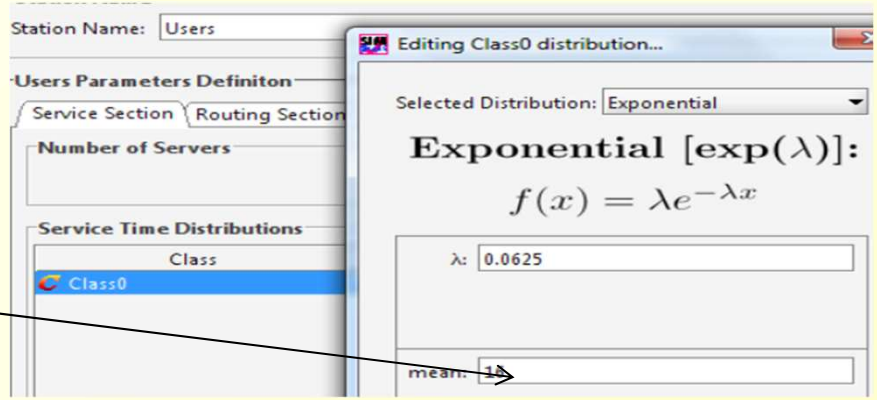
Step 2: **Define customer class and edit station properties.**



The Users properties window

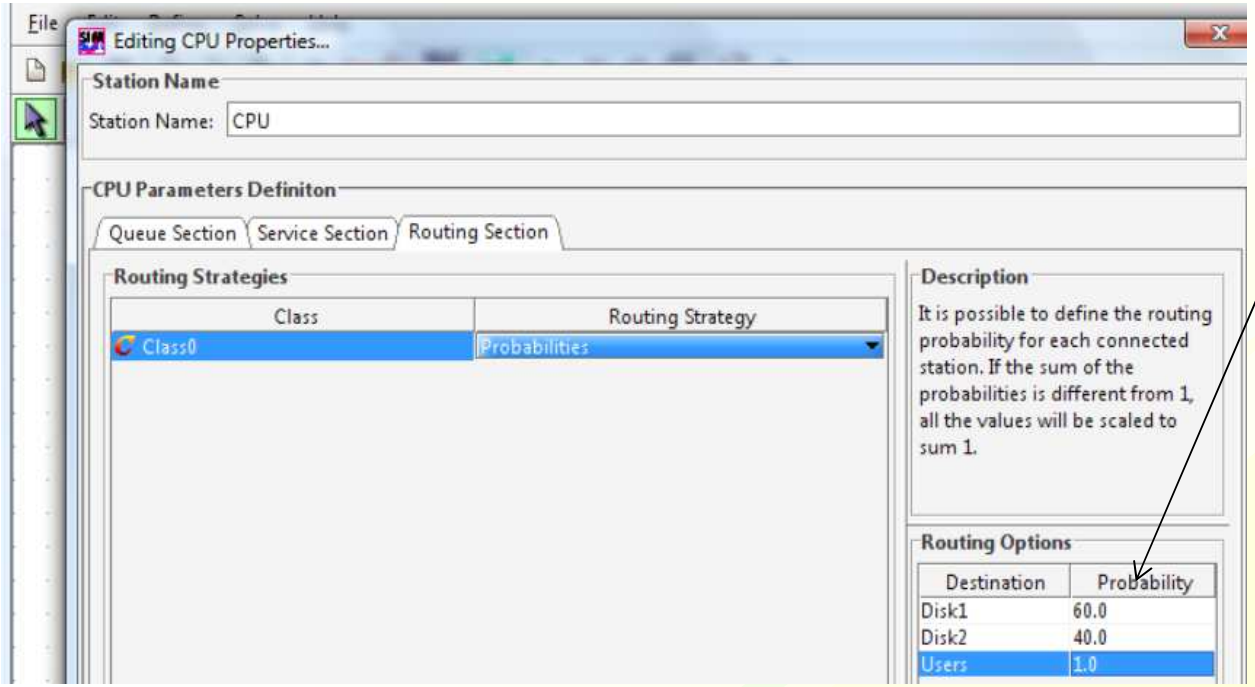
Set the value of all class for user

Set the user delay as mean value, 16



Routing Section

- As there are 3 outgoing links from *CPU*, each job's probability to choose one of the paths needs to be set. This is done by setting the probabilities of *Disk1*, *Disk2* and *Users* as 60, 40 and 1 respectively.

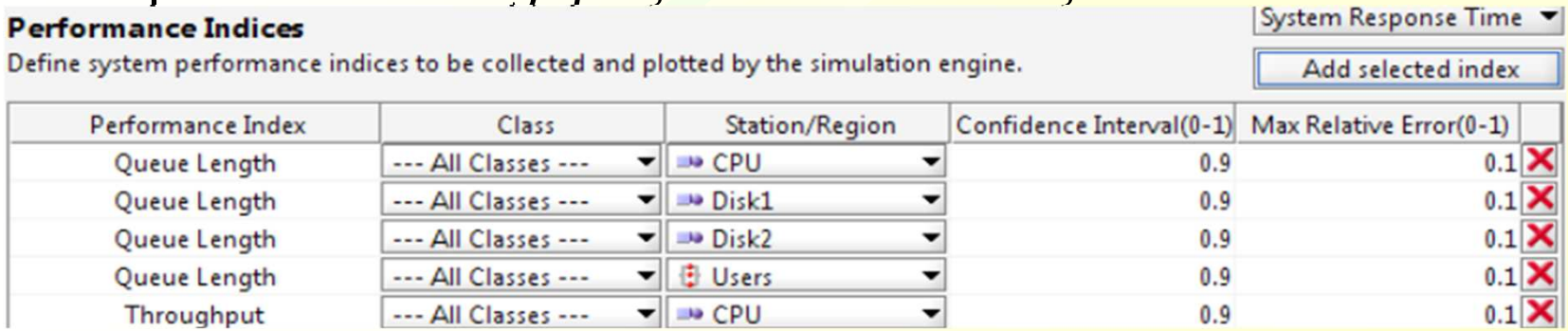


Setting the probabilities (or the visits) for the output of the *CPU* station

- Need to set the routing strategy for all stations.

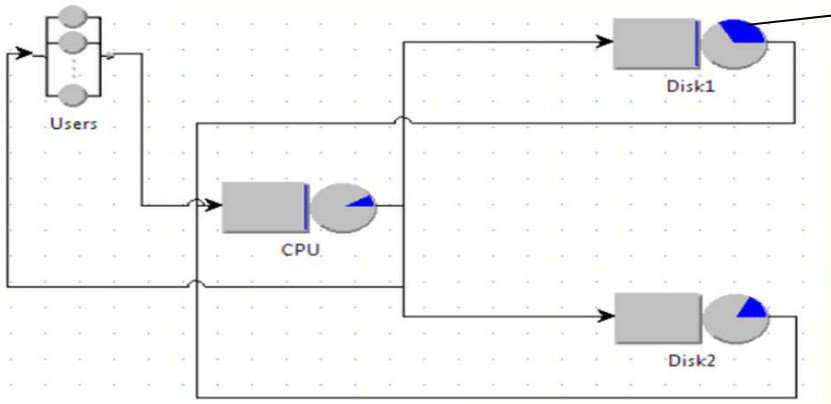
Define the Performance Indices

- In Example 1 the following indices are studied
  - *Queue length [jobs]* (for CPU, Disk1, Disk2, Users and the global System)
  - *Throughput [jobs/sec]* (for CPU, Disk1, Disk2, Users and the System)
  - *Residence time [sec]* (for CPU, Disk1, Disk2, Users and the System)
  - *Utilization* (for CPU, Disk1, Disk2 and Users).
- The following operations are required
  - Select Queue Length: Click on 'Add selected index' for four times. Change State/Region of these Indices to *CPU*, *Disk1*, *Disk2* and *Users*.
  - Repeat for *Throughput*, *Residence time* , *Utilization*.



Run the Simulation

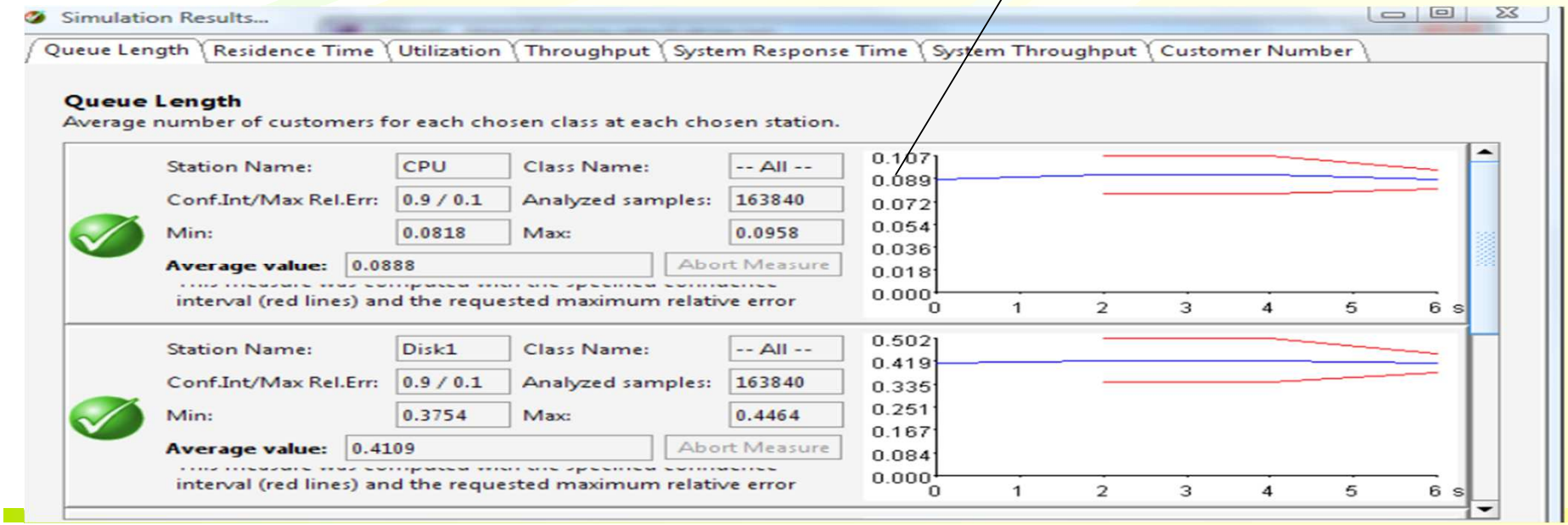
Press ▶ from the toolbar:



Color define the layout of the model with the percentage of utilization and queue length

Average number of customer each second

Simulation Result

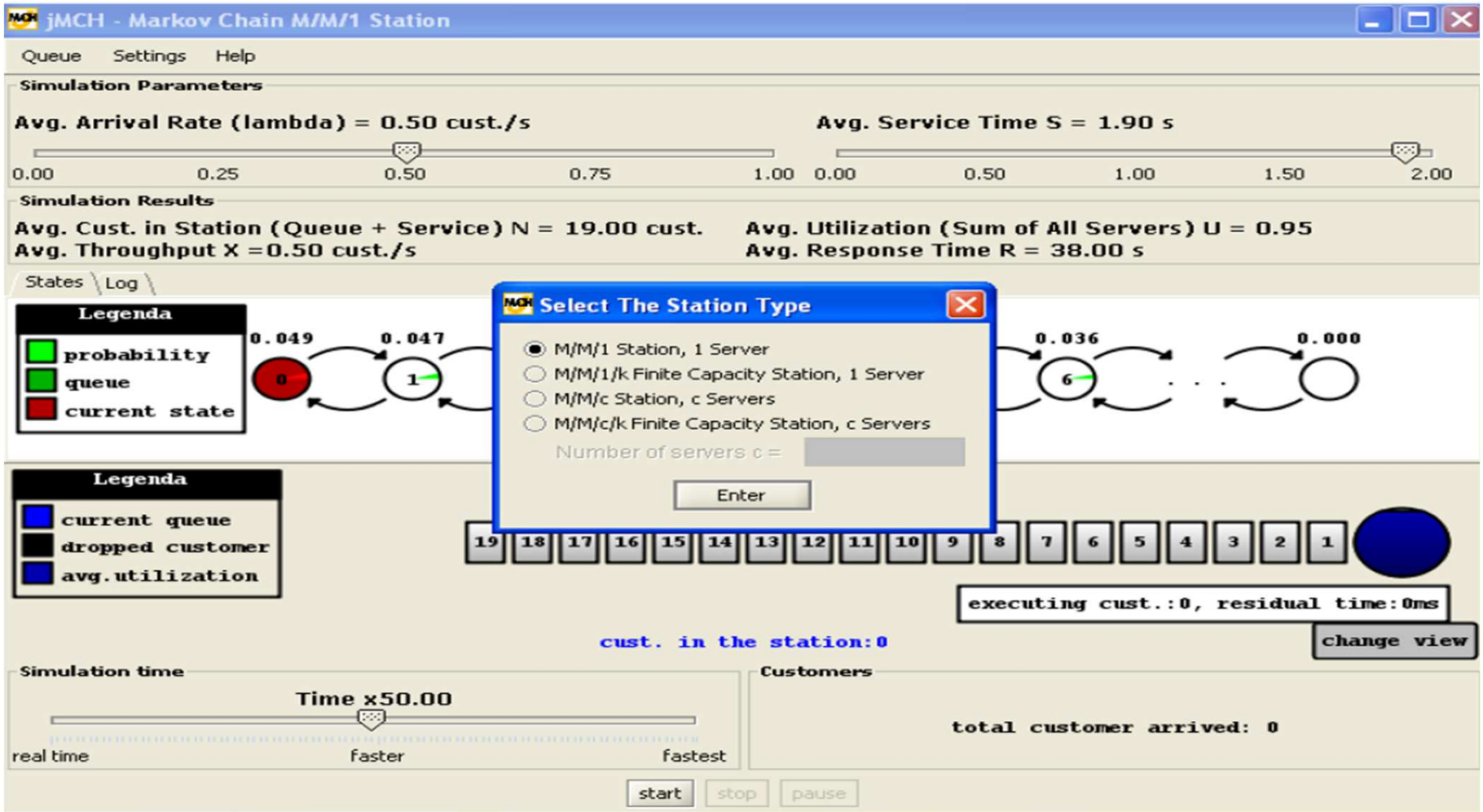


JMCH (Markov Chain)

- JMCH provides a graphical representation of the states of the Markov Chain corresponding to a single station model
- The type of station can be modified later by using the menu item 'Queue -Change station type'
  - *M*/*M*/1: 1 server and infinite queue size
  - *M*/*M*/1/*k*: 1 server and finite queue size, *k*
  - *M*/*M*/*c*: *c* servers (homogeneous) and infinite queue size
  - *M*/*M*/*c*/*k*: *c* servers (homogeneous) and finite queue size, *k*
- It is possible to change the arrival rate λ and service time *S* of the station at simulation run time
- If the model is of limited capacity, the size, *k*, of the queue can also be changed dynamically

Start the JMCH Solver

- JMCH start screen



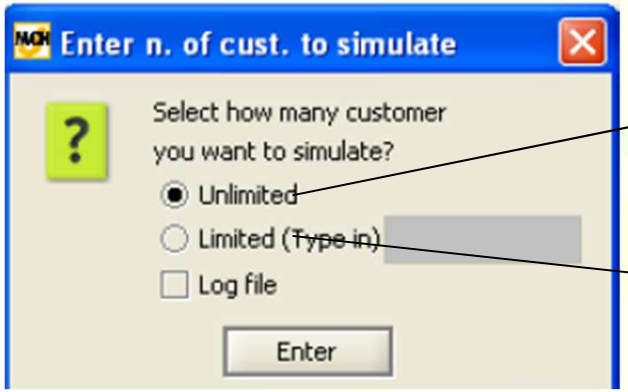
Select a Station Type and Simulate

The queue size *k* can be set dynamically while the simulation is running. The minimum value of *k* is 2.

The number *c* of the servers should be set with this dialog box, and it cannot be changed dynamically while the simulation is running.

The number *c* of the servers should be set with this dialog box, and it cannot be changed dynamically while the simulation is running. The queue size *k* can be set dynamically while the simulation is running.

- To start the simulation, user needs to 'Press Start' button, on pressing 'Start the dialog'

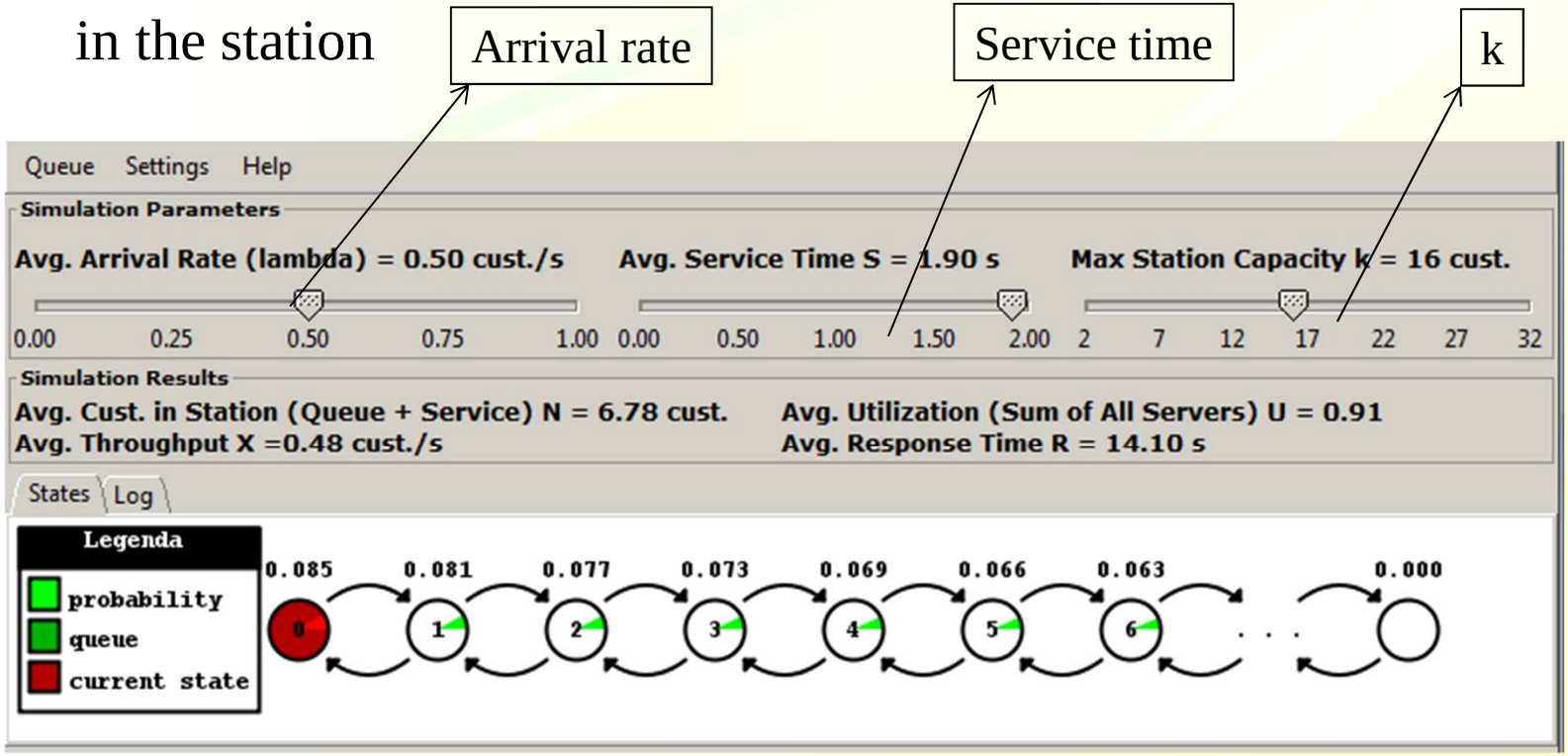


If 'Unlimited' is selected, the simulation must be stopped manually. Otherwise, it runs forever.

If 'Limited' is selected, the simulation runs for the given number of customers.

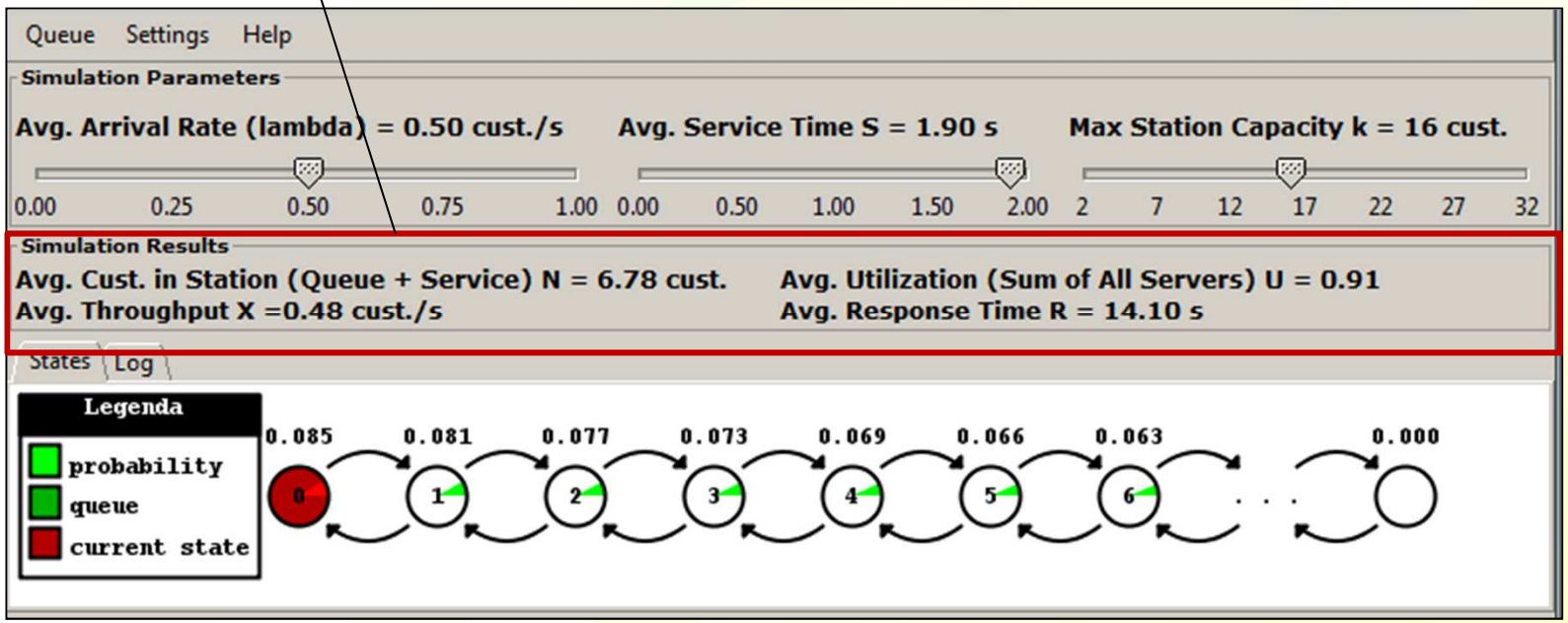
Input Parameters

- λ: Average arrival rate, the distribution of inter arrival times is exponential with mean $1/\lambda$
- S: Average service time for each customer, the values are exponentially distributed
- k: Maximum station size: the maximum number of customers allowed in the station



Performance Indicies

1. *Mean number of customers* in the station
2. *Throughput*
3. *Utilization*
4. *Mean response time*
5. *Probability of the states* of the Markov Chain



Conclusion

- **Analysis with Java Modelling Tools (http://jmt.sf.net)**
  - Queueing network simulation
  - Bottleneck identification
  - Workload analysis
  - Mean value analysis

- **Manual can be found on online from the link below**
  - http://jmt.sourceforge.net/JSIMg.html