HUMAN CAPITAL
NATIONAL COHESION STRATEGY

Wrocław University of Technology

EUROPEAN UNION
EUROPEAN
SOCIAL FUND

THE DEVELOPMENT OF THE POTENTIAL AND ACADEMIC PROGRAMMES OF WROCŁAW UNIVERSITY OF TECHNOLOGY

Internet Engineering

Tomasz Babczyński

Information Systems Analysis

part 2 – the Queuing Systems

Project co-financed by European Union within European Social Fund

Instruction to all the laboratory assignments of the queuing systems

The theory of queues in performance evaluation

The aim of this part of course is to learn how to construct queuing model of the real problem and how to analyse it. The leading example of the following little tutorial is not technical but it is familiar to everyone who have ever sat in a queue in any clinic. The practice of building queuing models of real-life problems will be presented as well as the methods of solving them and some tools supporting us in this work.

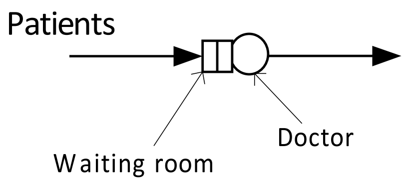
Real system specification:

Let us assume we are going to open an allergic clinic. We want it to contain some consulting rooms, treatment rooms, the registration, and of course waiting rooms. We will be looking for answers to the questions about the number of doctors needed in our clinic, the number of chairs needed in waiting room, etc.

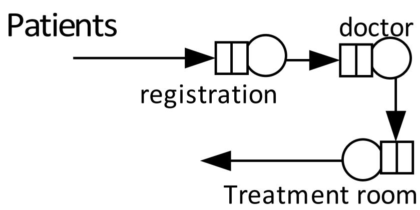
First, to warm, we build a model for a very simple case – one doctor in one consulting room, no treatment rooms, no registration. The model will contain only one "service place" with a queue.

The queuing theory model of the system:

In the simple version of the problem we have one service node symbolizing a doctor.



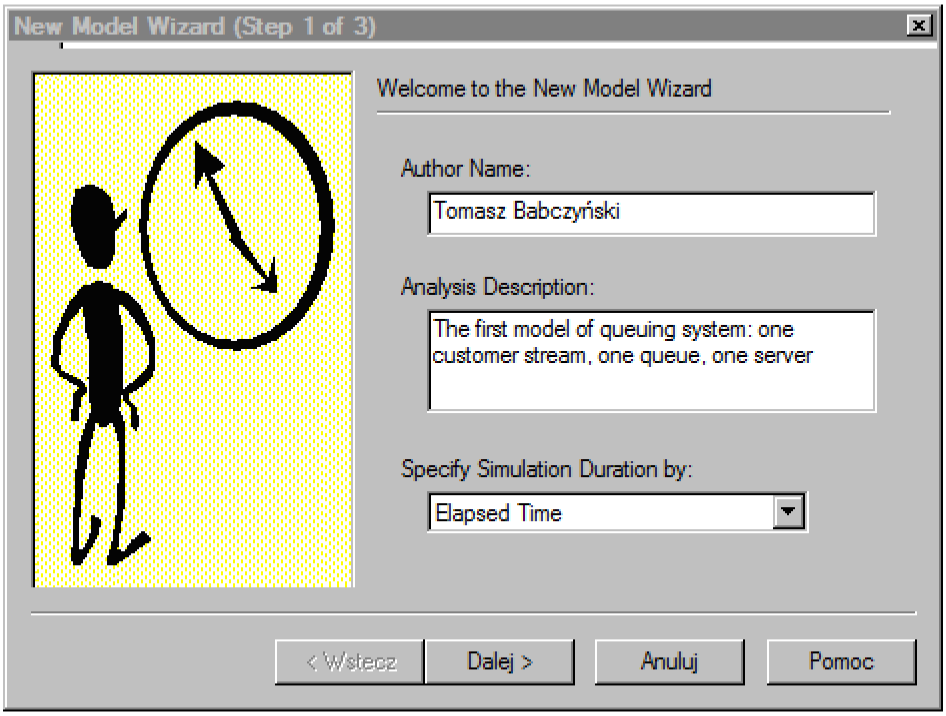
The following model is for more complex case where patients have to register, then go to the consulting room and finally to the treatment room everywhere waiting in queues.



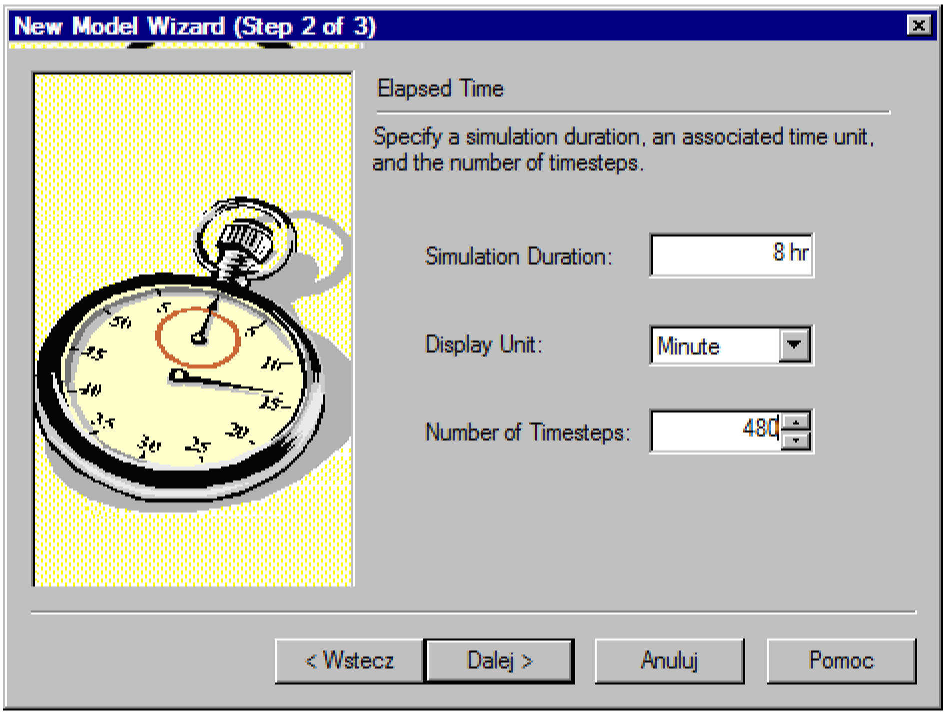
Analysis by simulation:

First, let us try to simulate the model. We will use the discrete event simulator GoldSim.

After launching the program, the "new model wizard" appears. At the first stage of running the wizard it is very encouraged to write down the description of the model and the name of the author (or authors).

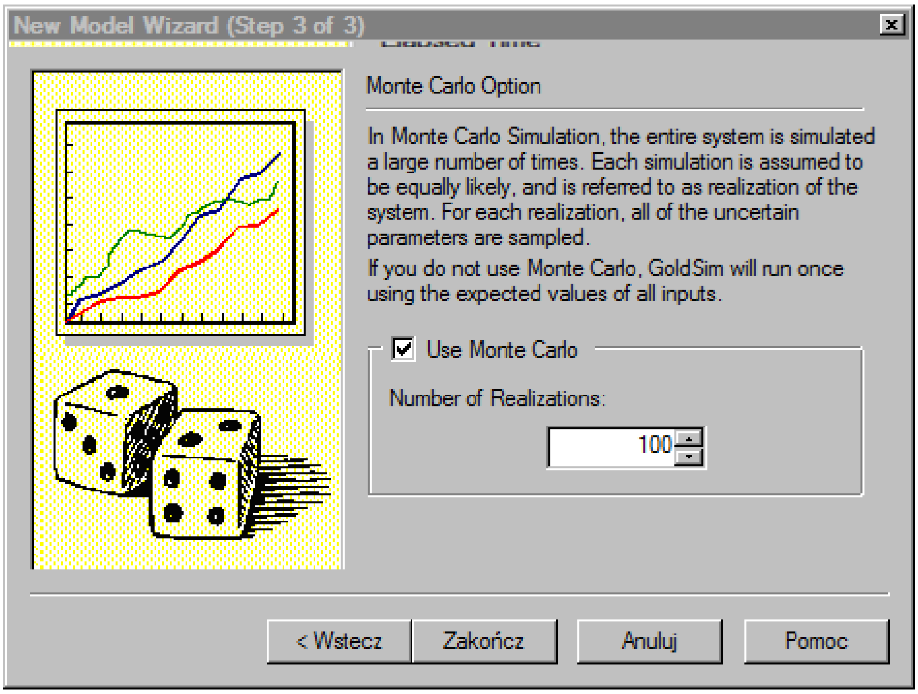


The second stage is for specifying the time of simulation. According to the method of specifying of the time, chosen at the previous dialog, one of the two dialogs appears. The time can be specified by defining its duration or giving the starting and the ending dates. In the our example.

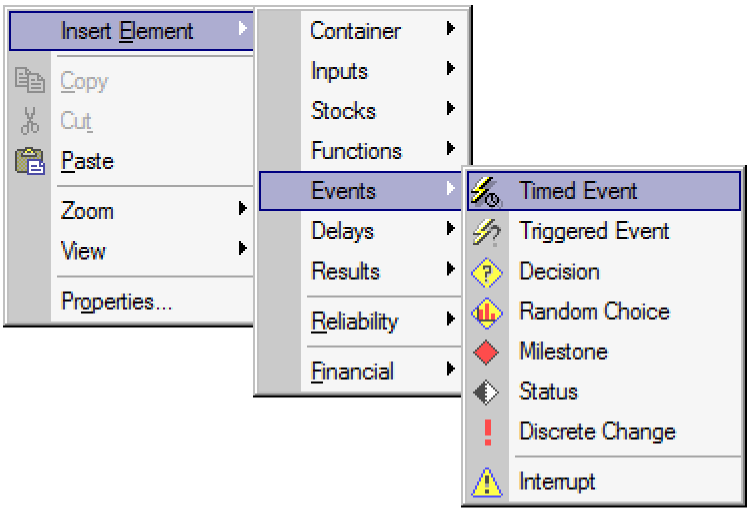


The first method is chosen. We are to specify the units in which the time will be measured, the duration of the simulation process and its granularity. In the example "hr" is the shortcut for the "hour".

Third stage of the wizard is to decide whether one simulation or a number of runs with random parameters will be performed. Using the GoldSim simulator for simulating of queuing systems, the Monte Carlo simulations should be used. The number of realisations depends on the accuracy of the results we want to obtain. For the preliminary simulation of the small model the one hundred is good enough.

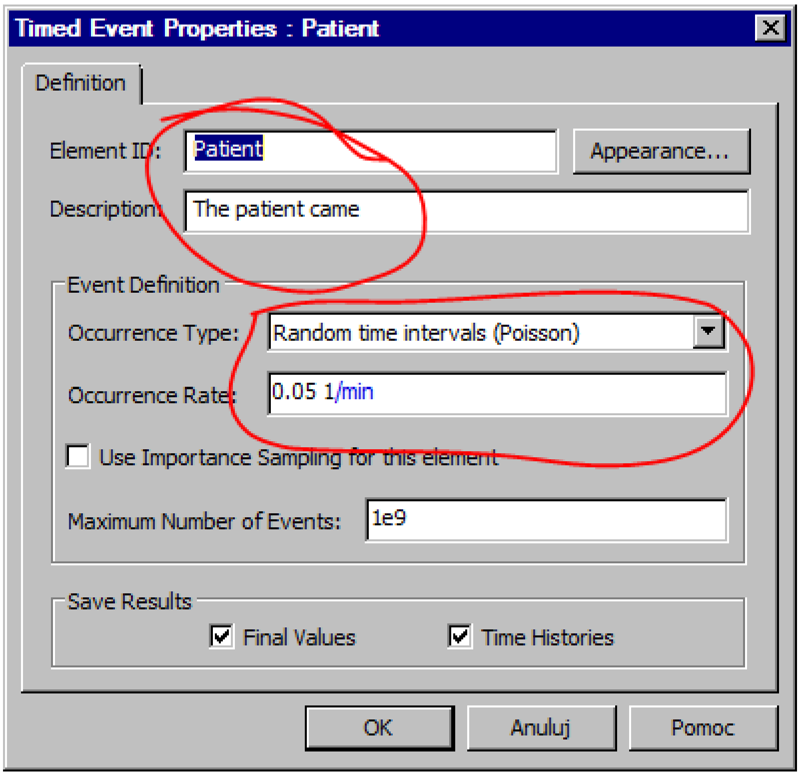


After defining the fundamentals of the model with the help of the wizard we can start to define the model of the queuing system. We start from the defining the incoming stream. The coming of the patient will be simulated using the timed event element which can be found in the context menu (under the right mouse button) of the main screen of the simulator.

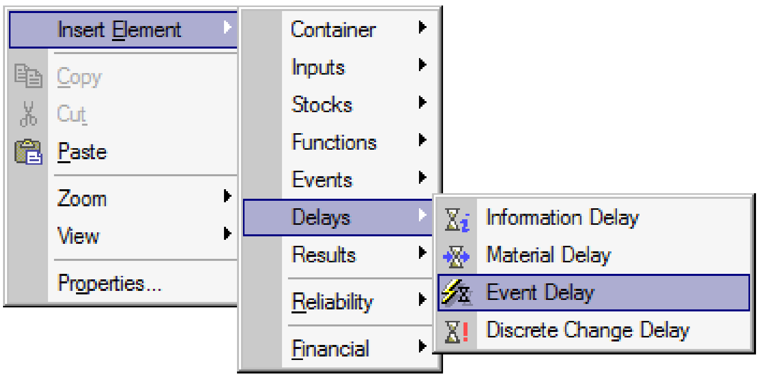


The timed event has a number of parameters which should be defined. First, it is very, very encouraged to set element ID and the description of the element. The ID can not contain spaces and it will be visible on the screen and on the printout. The description will be visible after choosing the element. For the large model the discipline in naming elements would be very helpful.

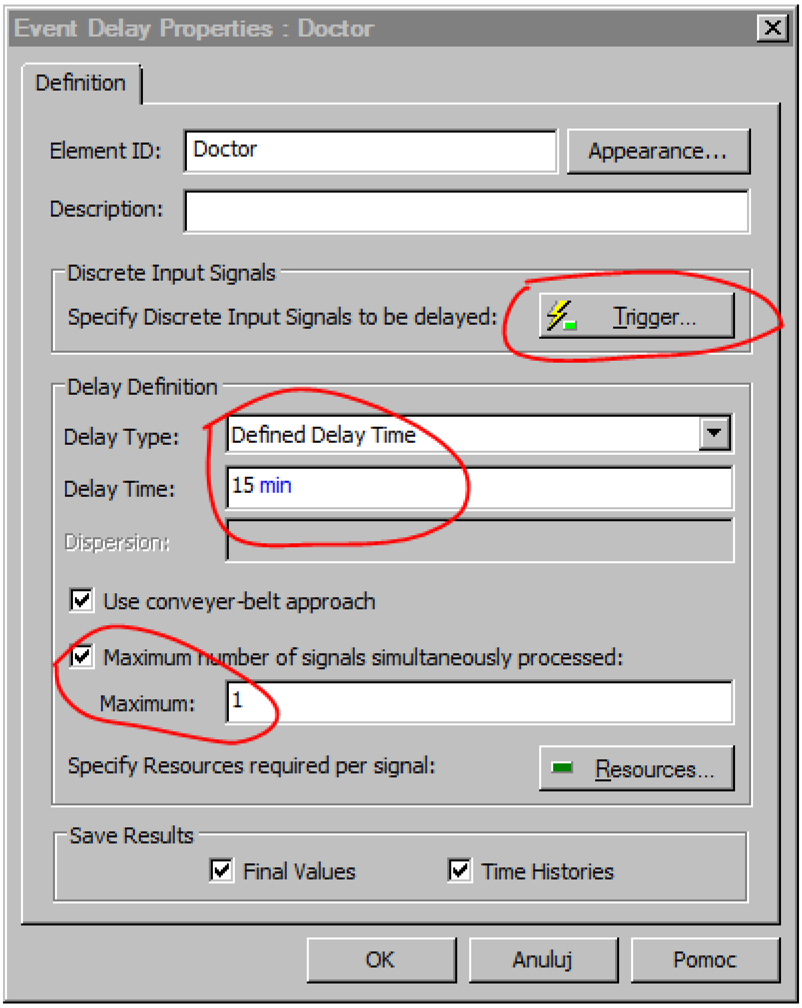
The next thing we define for the timed event is its definition. The event can be triggered once or periodically. In the second case, the repetition can be defined using a random variable as will be shown later. When the event should be repeated in regular intervals or it should form a Poisson stream, the specification can be simplified. Instead of the random variable, only the parameter can be specified. In our example we choose the most common in queuing systems theory method of repetition i.e. Poisson arrival. The parameter is the arrival rate. In our example it is 0.05/min what gives 1 patient per 20 minutes (average of course).



Now we define the doctor with the queue in front of his room. In the context menu we find the "Event Delay" element. This element is intended by the authors of the simulator, among others, for modelling of queues.



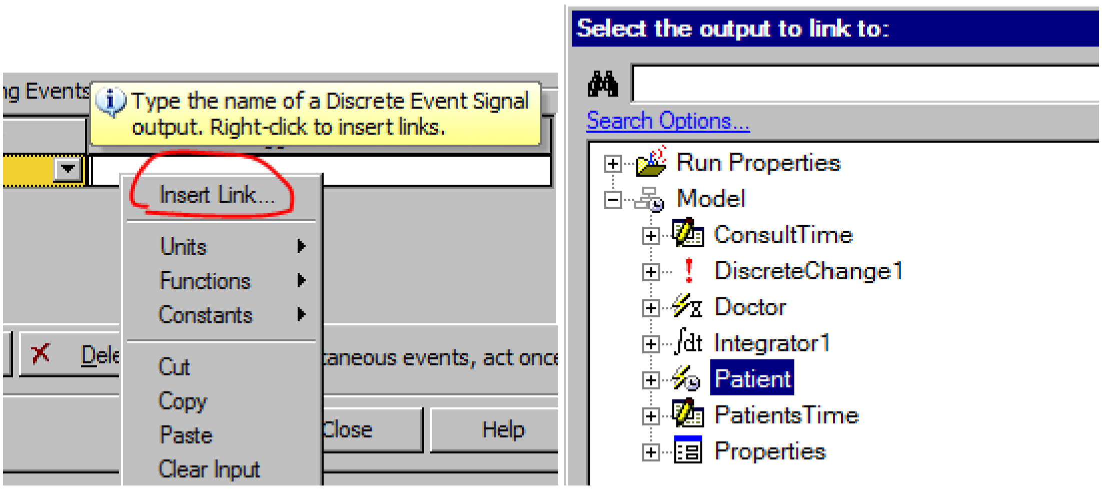
The properties of the event delay element contains already described fields "ID" and "Description". More important now are the "Trigger" button and parameters marked with red on the picture below.

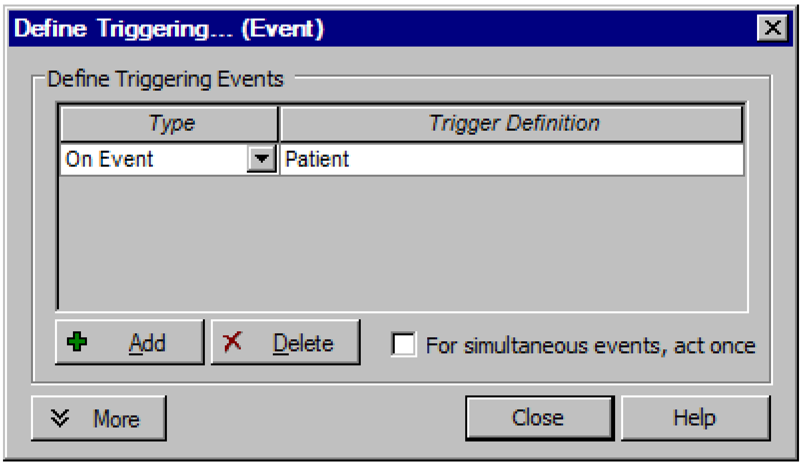


There are five types of events that can be added as triggers:

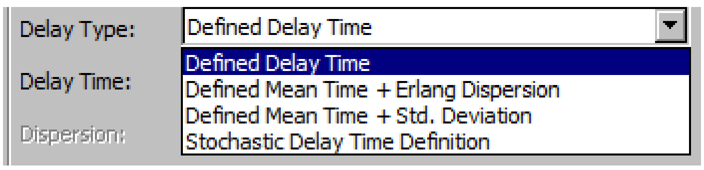
- **On Event**: The Trigger Definition must be a discrete event or discrete change signal from another element. The element is triggered whenever the signal is received.
- **On Changed**: The Trigger Definition can be any continuous output (it cannot be an expression or a discrete signal). The element is triggered whenever the value of Trigger Definition changes.
- **On True/On False**: The Trigger Definition can be any condition output or conditional expression. The element is triggered whenever the Trigger Definition becomes True/False.
- **Auto Trigger**: The using of that kind of triggering is addressed to conditional containers only and it is useless for our purposes.

The delay can be triggered by a number of events but in our simple example only one is used. We can type a name of the desired event (e.g. "Patient") or select it using the "Insert Link..." command.

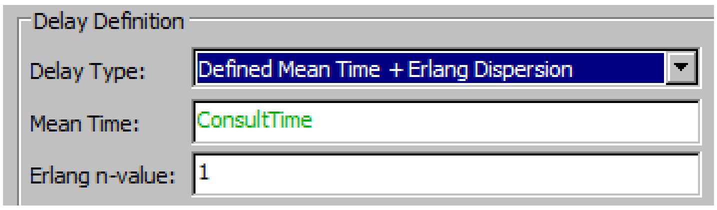




Now, when we have the triggering event defined, we may choose the type of the delay. There are four possibilities.

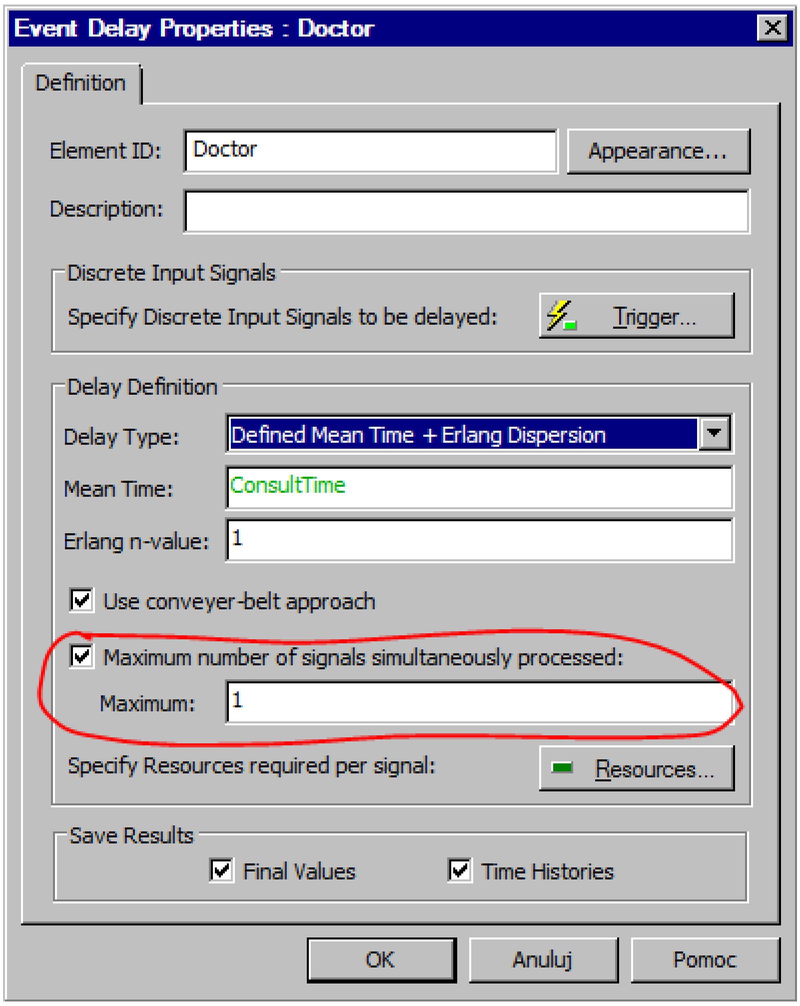


The first one is a simple constant delay. We define only the delay time e.g. 15 min. The next two possibilities introduce a dispersion of the delay time with Erlang or Standard distribution respectively. The widely used in the theory of queues exponential distribution is the Erlang distribution with "n" parameter equal to 1.

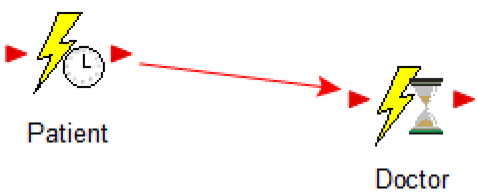


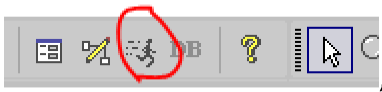
In the above window instead of the absolute value of the mean time, the link to the data element is used. What for? It will be explained later.

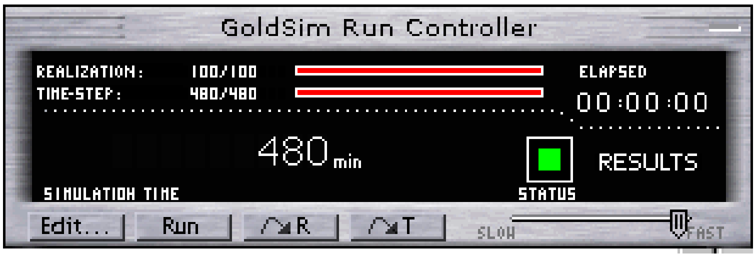
Let us recall the full dialog box because we can not forget about one parameter. If we want to model a queue, we must check the box marked at the following picture. In the queuing theory terminology it is the number of servers (in our example – doctors). Unchecking the box we say that the number is infinite and so there is no queue needed.



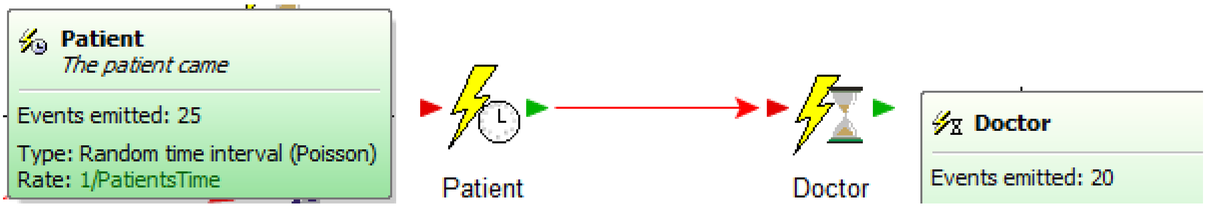
Now, we have our first simulation model defined. It looks as follows.



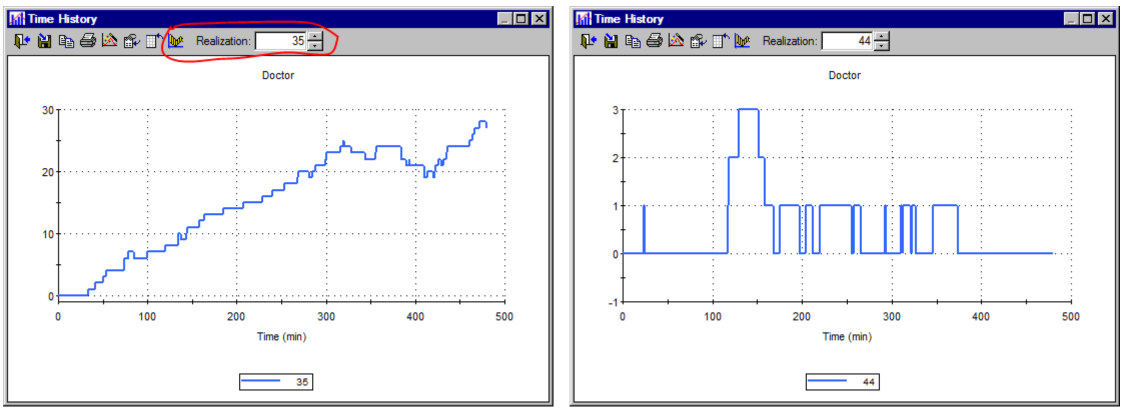
Let us run it using the toolbar button: , from the menu, or using the keyboard shortcut "F5". The following window will appear and we can click on the "Run" button.



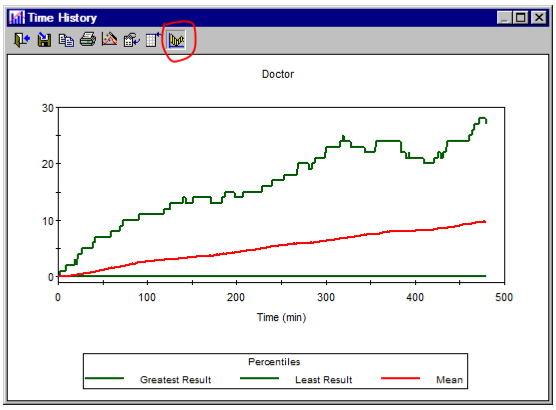
After some seconds the simulation finishes and we can look at the results. Note that the arrows outgoing from the elements are green now what means that the result data is available for the outputs. Placing the mouse cursor over the arrows we can get some values, clicking on the arrows we can get the history of the outputs. Here we can see that in the last simulation 25 patients came but only 20 were served.

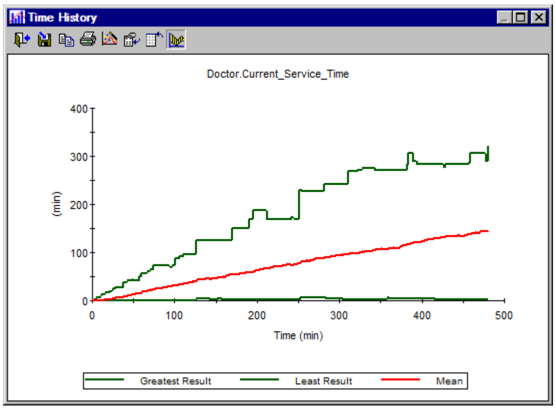


The historical data from the simulation can be viewed individually for each realization or statistically aggregated. The left chart shows the 35th realization of the simulation. That was a hard day for the doctor and for his patients (up to 28 patients in queue). The day simulated as the 44th realization was much better. As it can be seen the single realizations differ very much.

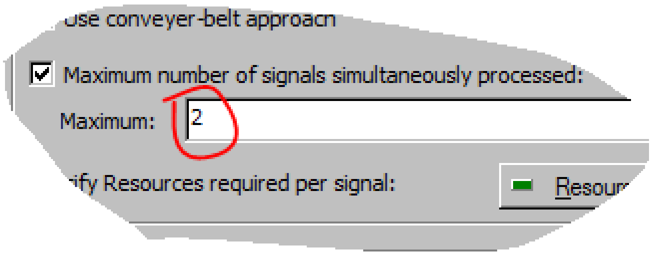


The following charts show the results of all 100 simulations statistically processed. The marked button should be pressed. The mean value, the upper and lower band, the median, and the percentiles can be displayed. On the charts below the mean value (red) and upper and lower bands (dark green) are presented. The left chart shows the queue length while the right one shows the service time. It is easy to see that the parameters are badly chosen. The queue length may grow to the length near 30 and the service time grows to even 5 hours. Unacceptable!

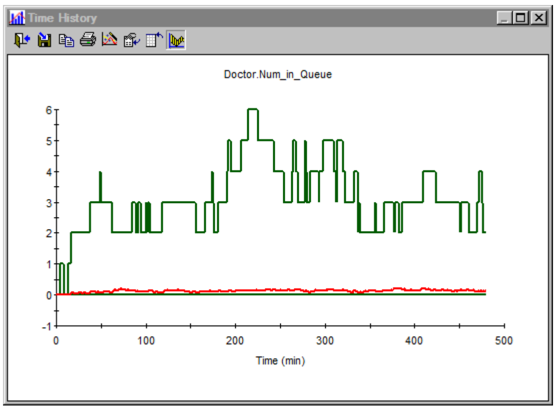


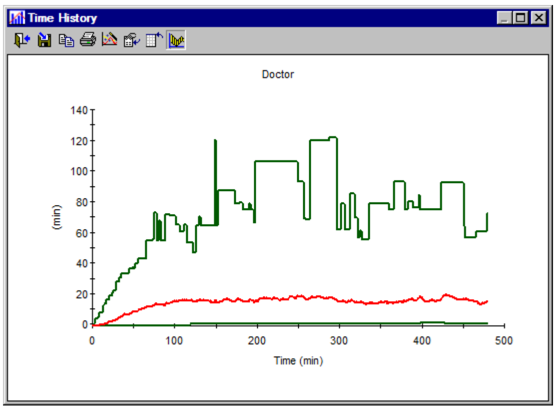


Let us try to improve the effectiveness of our clinic by employing a second doctor. In the model this fact is expressed by changing the one value in properties of the delay element:

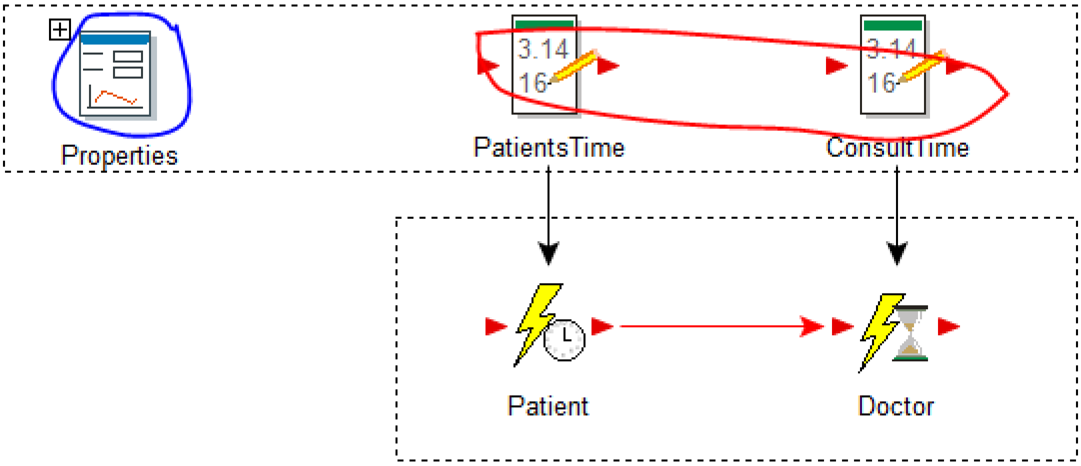


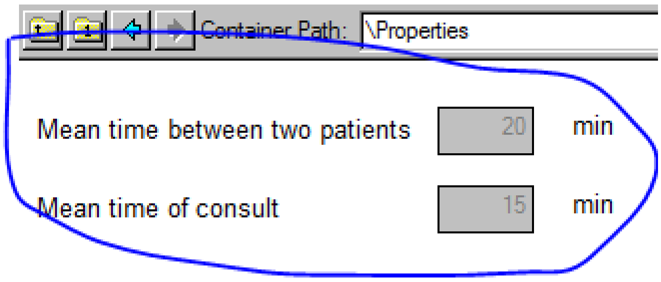
Now, after the simulation, we can see that the employing of a doctor helped. The maximal queue length is now only 6, the maximal waiting time ab. 2 hours. The mean values are even better – the queue length ab. 0, the mean service time less than 20 minutes. These values are acceptable. We can buy 6 chairs and open the clinic.



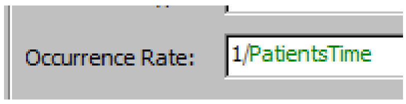


It was easy. Now, let us involve in a harder case. We need to add the registration and a treatment room (or rooms). But first, some promised words about data elements and links. Below is the model enriched by two data elements (marked red) and a Properties dashboard (marked blue) shown expanded on the next picture. The edition boxes are connected with the values of the appropriate data elements. It is obvious which ones are bound together.





The properties of "Patient" element needs to have rate instead of mean time which is defined in the dashboard. The rate is the reciprocal of the time, so it is given it the following way:

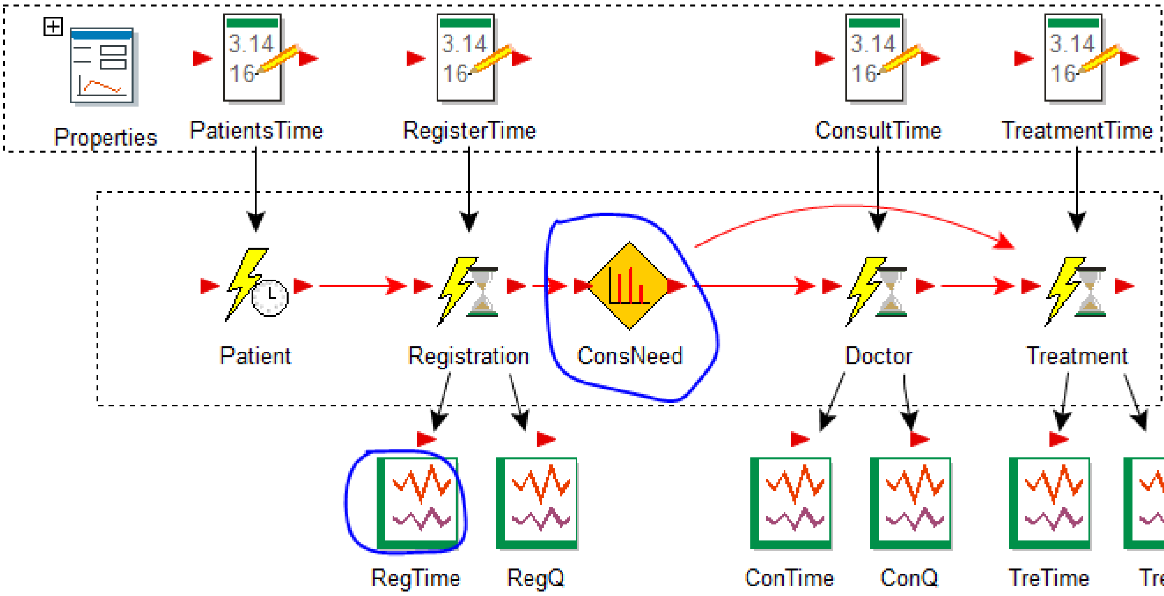


The using of dashboard makes it possible to have all important parameters in one place and easily watch it or change.

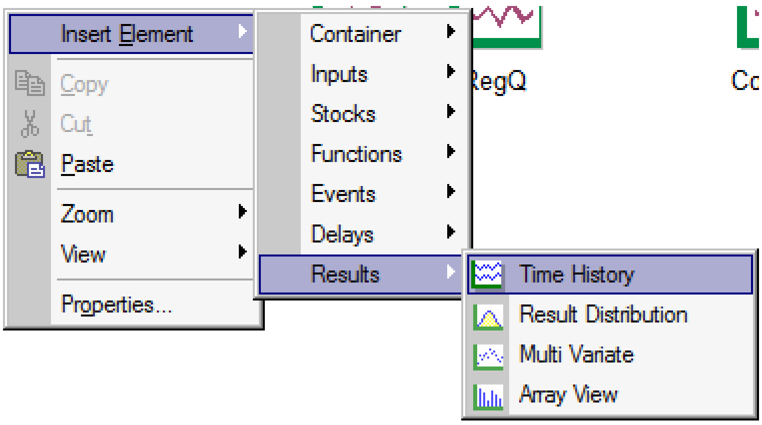
The full model of the clinic is as follow. First the parameters:

Mean time between two patients	20	min
Mean time of registration	5	min
Mean time of consult	15	min
Mean time of treatment	2	min

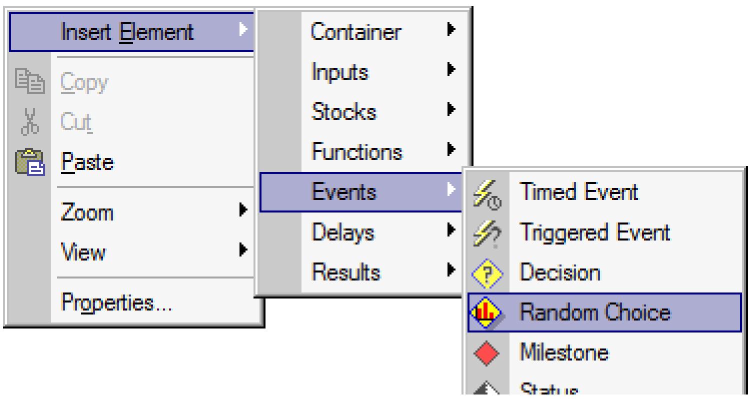
Again, we have only one doctor.



Two elements in the model, marked with blue, need additional explanation. The first of them is the preformatted chart element named "Time history" used for easier presentation purposes.



The second element is the "Random Choice" of event. It is used here for modelling the fact that some patients come only for vaccination and the rest of them needs the doctor consultation before that.

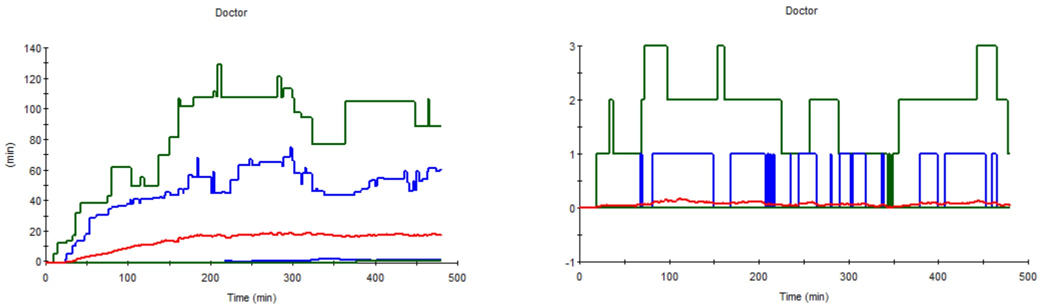


The element has in its properties the table of probabilities. In our example it directs the patient to the doctor with the probability 0.3 and to the treatment room with probability 0.7.

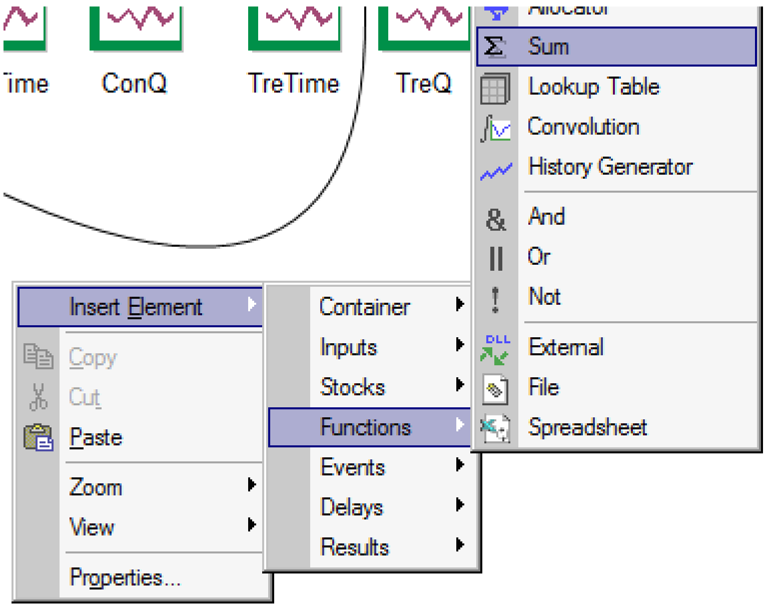
Probability / Event Table

	Probability	Output Event
1	0.3	toDoctor
2	$1-\sum p_i$	toTreatment

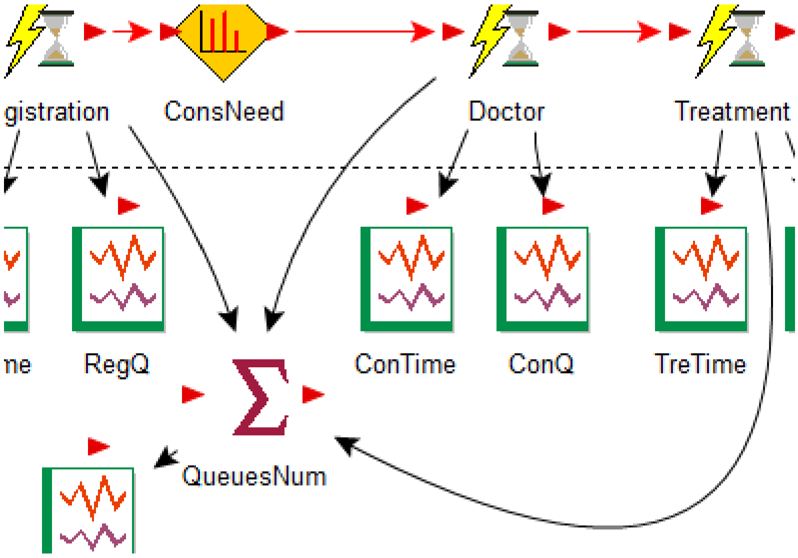
We can provide the simulation now. The results will be kept in predefined charts so it will be sufficient to double click on them to have seen the results. The blue lines show the value of the 95 percentile.



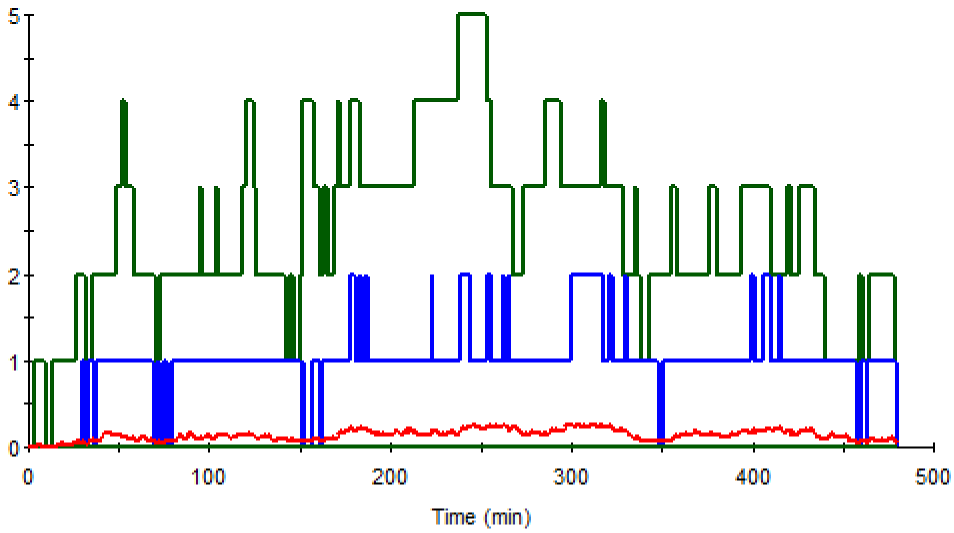
Finally we want to answer the question how many patients wait in our clinic simultaneously. We must to sum the lengths of all queues at all time steps. We use, of course, the "sum" function.



The element will sum for us all queues lengths during all simulation steps sending the results to predefined chart element.

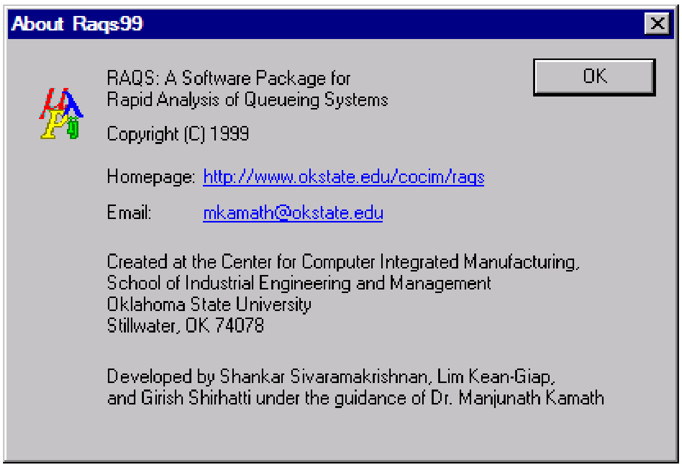


The result is as follows. The lower, red line is for the mean queues length, the upper, green line is for the greatest values, and the middle, blue line is the 95% percentile. It means that during the simulation, in the most cases (95% of them) the queues were not longer than two and never were longer than 5. But remember – it is only the result of the simulation.

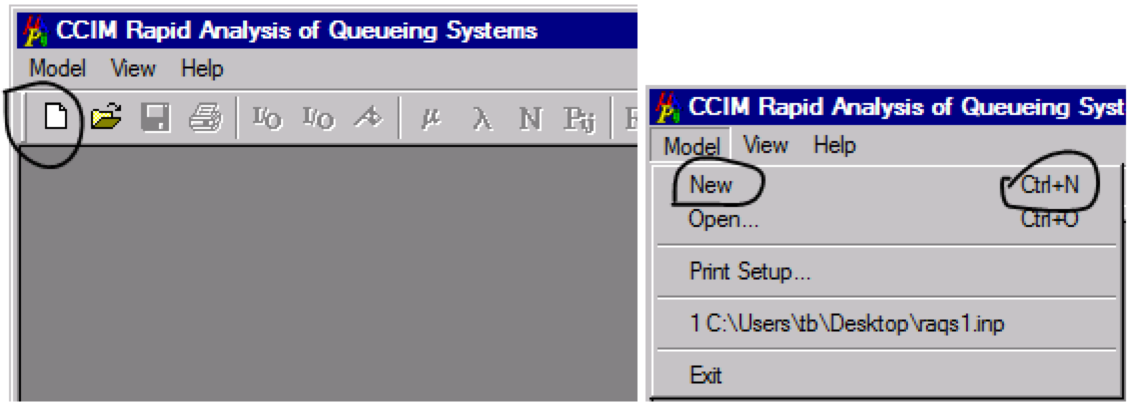


Theoretical analysis:

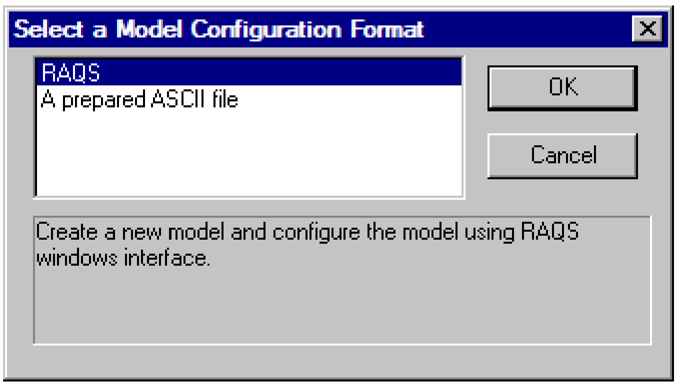
For the theoretical analysis of queuing systems we use a rather simple program RAQS. The program is available for free from the address http://www.okstate.edu/cocim/raqs/. It is rather old program in fact but it is still useful.



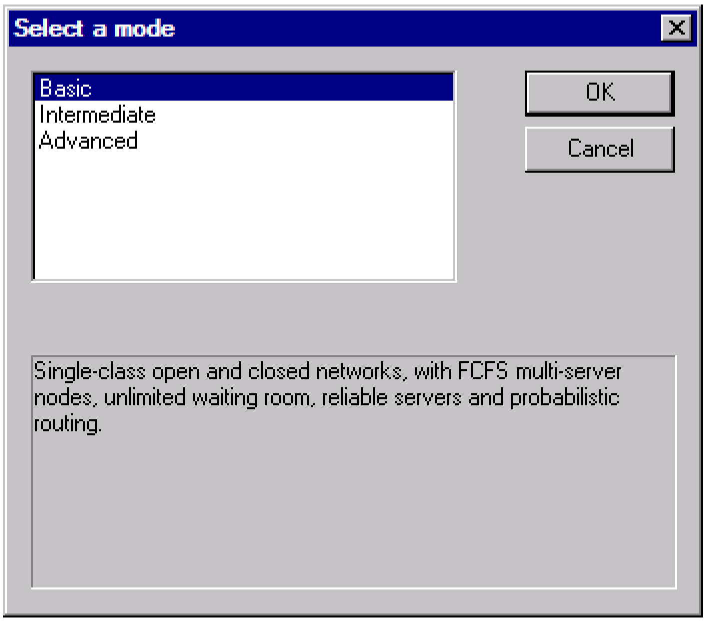
After running the program we have to build a model. We start it with the new model command which can be found in the menu or at the toolbar as presented on the following picture.



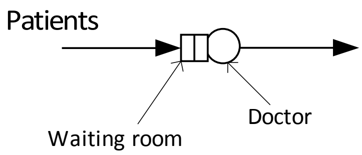
The model building wizard is starting. First, it gives us the choice to build the model from scratch or to get is from a prepared ASCII file. Unfortunately the format of the file is undocumented although it looks easy to understand. Now let us choose the building from scratch – the RAQS option.

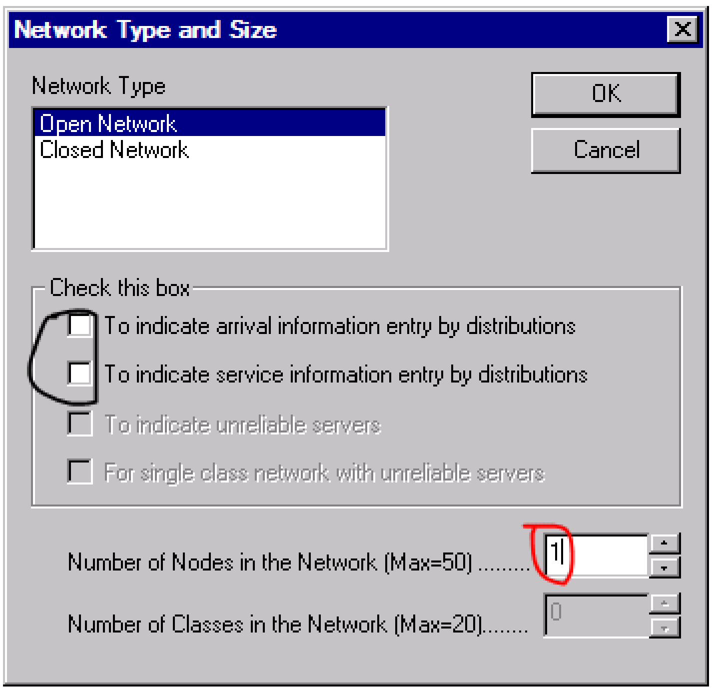


Next step is to decide how much complex will be the model. The first example is very simple so the basic mode will be adequate.

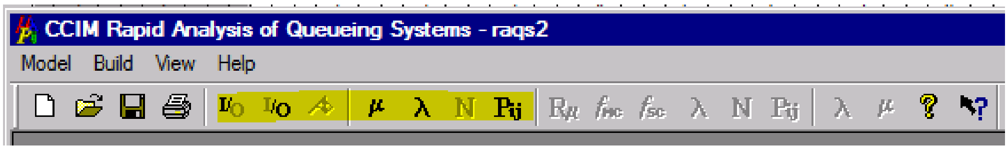


Let us recall the model:

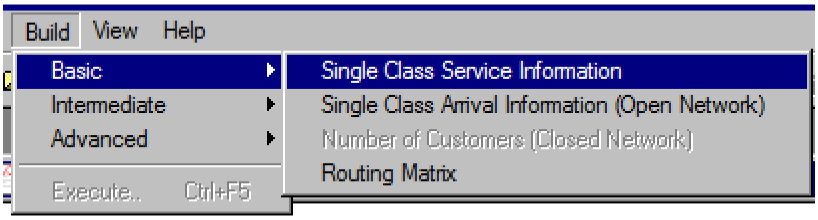




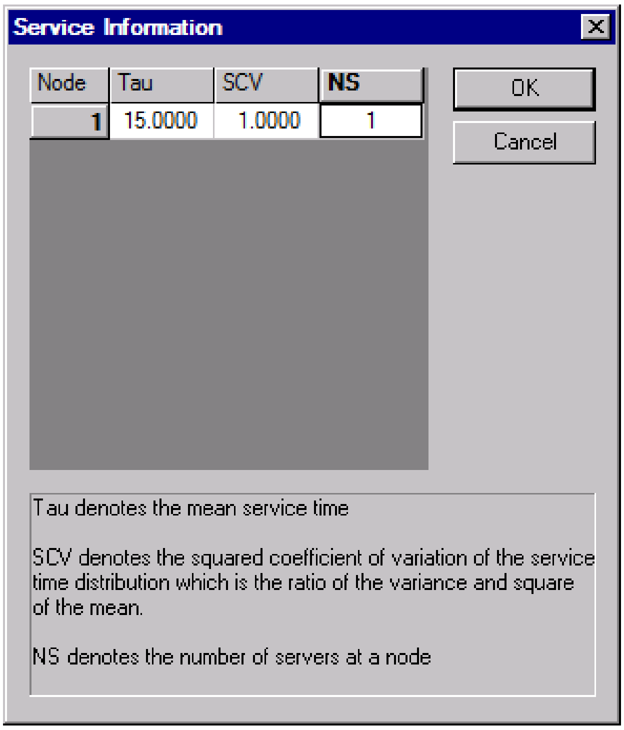
It is simple open network, so we choose it in the following dialogue window. We will use exponentially distributed times for incomings and service times. It is basic distribution in the queuing theory so we need not to check the boxes indicating that the distributions will be special. The number of nodes in the system is of course 1. After we clicked OK in this dialogue window – we have the model built but the parameters still remains undefined. The set of (marked here) buttons on the toolbar makes it possible to define the parameters of service time, arrival time and transitions matrix (in the first example unneeded).



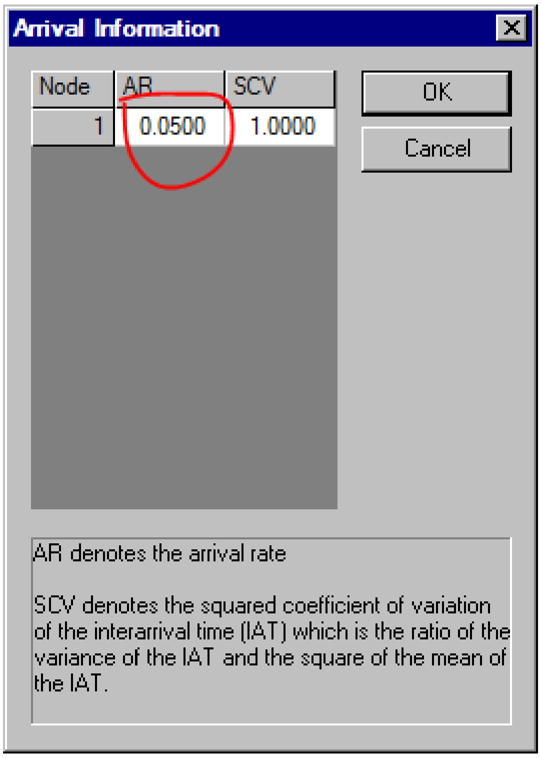
Alternatively we can call the same dialogue boxes from the menu:



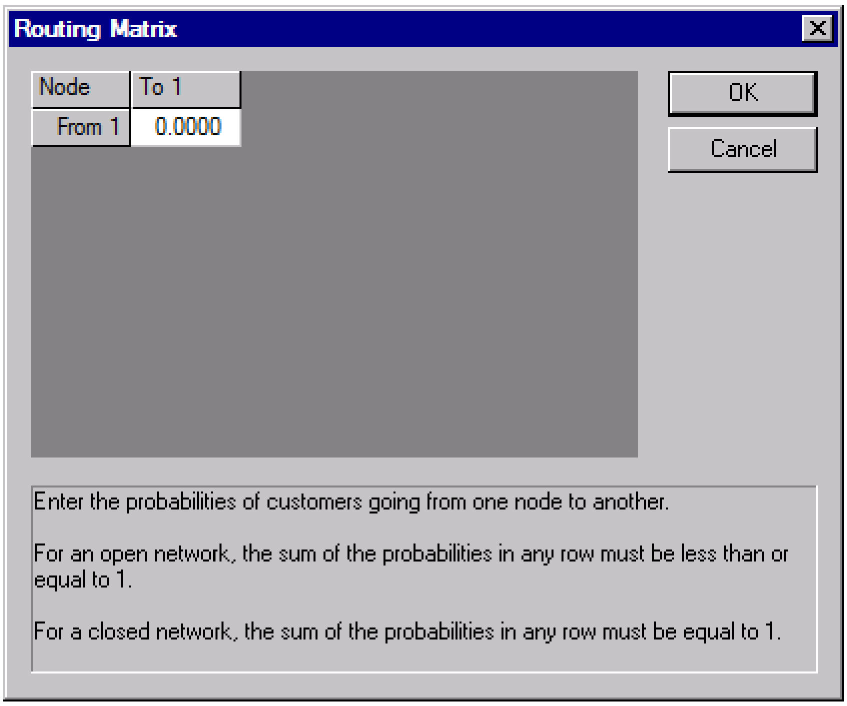
The service information is characterized by the mean time of service denoted as *Tau*. In our example it is 15. The data in the program are dimensionless so we should have in mind that the units we use are minutes. the second parameter denoted as *SCV* is the *Squared Coefficient of Variation* – the parameter describing the random distribution. Numerically it is the variation of the random variable divided by the square of its expected value. The exponential distribution has the SCV=1. The third parameter is the number of servers, doctors in our example. In the first version of the experiment it is equal to 1.



In the similar way we can define the arrival parameters. For the arrival we do not set the mean time between arrivals but the arrival rate. The *AR* is equal to the one divided by the mean time between arrivals. As in the simulation we set it to the 1/20. The *SCV* as in the service time is equal to 1 because we want to have the exponential distribution i.e. the Poisson arrival.



The routing matrix is trivial in our example because we have only one node.

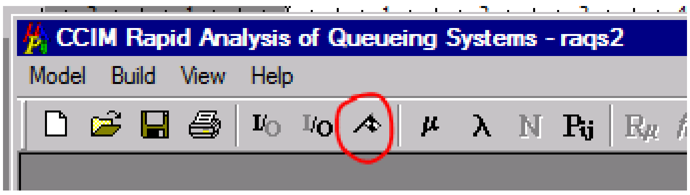


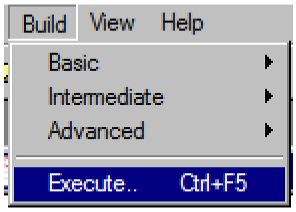
The window with the input information gives us the summary of all parameters put in the system.

```
Input information
This Model has been developed in the Basic Mode
Type of Network - Open Network
Number of nodes = 1

Node  Number       Arrival   Arrival  Mean Serv  Service time
 #    of Servers   Rate      SCV       Time       SCV
1     1            0.050     1.00     15.000     1.00
```

Now we can run the computation. We can choose the race flag from the toolbar or the Execute command from the menu.





As we can remember the results of the simulations gave us the unacceptably great values of the consult time and the queue length. Those values were the extreme values obtained during the simulation. Now we have only the mean values calculated. The situation does not look dramatically. The mean time spent in the clinic is about 60 minutes, average number of patients - 3.

```
Output Report
This Model has been developed in the Basic Mode
Type of Network - Open Network

Network Measures
Average Number in the Network =    3.000
Average time spent in the Network    =   60.000

Node Measures
Node Util    AvTIQ    VarTIQ  AvNIQ    AvTAN    VarTAN   AvNAN    VarNAN
1    0.750  45.000   3375.000 2.250   60.000   3600.000 3.000    12.000


Util - the Utilization at a Node
AvTIQ, VarTIQ - Mean and Variance of the waiting time in queue at a node
AvNIQ - Mean queue length at a node
AvTAN, VarTAN - Mean and Variance of time spent at a node
AvNAN, VarNAN - Mean and Variance of the number of customers at a node
```

But let us look closer at the variation values. They are big what means informally speaking that it is probable that the actual value of the random variable is far from the mean value. Let us analyse deeper the mean and the variance of time spent at a node, the **AvTAN** and **VarTAN** values. For simplicity and because we do not know what the distribution actually is, we assume that the distribution is normal. For such a distribution the majority of the data values – 99.7% lays in the range of the mean value ±3σ, where σ is the standard deviation of the random variable equal to the square root of the variation. The square root of the **VarTAN**=60, so the time the patient have to spent in the clinic may be even 60+3*60 minutes, i.e. 4 hours.

Now we provide the second calculation – with two doctors. We have to change only the *NS* value in the service information. The results now are as follows:

```
Network Measures
Average Number in the Network =    0.873
Average time spent in the Network   =   17.455

Node Measures
Node Util   AvTIQ   VarTIQ  AvNIQ   AvTAN   VarTAN  AvNAN   VarNAN
1    0.375  2.455   26.107  0.123   17.455  251.107 0.873   0.704
```

Now the average time in the clinic is about 17.5 minutes and the majority of the patients waits no longer than 17.455+3*sqrt(251.107)=65 minutes.

Now we build the model with the registration and the treatment room.

```
Input information
This Model has been developed in the Basic Mode
Type of Network - Open Network
Number of nodes = 3

Node   Number        Arrival   Arrival  Mean Serv  Service time
 #     of Servers    Rate      SCV       Time        SCV
1      1             0.050     1.00     5.000      1.00
2      1             0.000     1.00     15.000     1.00
3      1             0.000     1.00     2.000      1.00
```

The node 1 states for the registration, node 2 – the doctor, node 3 – the treatment room. The mean service times are the same as during the simulation – 5, 15 and 2 minutes, exponentially distributed. The route table is as follows:

Routing Matrix

Node	To 1	To 2	To 3
From 1	0.0000	0.3000	0.7000
From 2	0.0000	0.0000	1.0000
From 3	0.0000	0.0000	0.0000

As in the simulation the 30% registered patients go to the doctor and the rest directly into the treatment room. The results are as follows:

```
Network Measures
Average Number in the Network =    0.735
Average time spent in the Network   =   14.695

Node Measures
Node Util   AvTIQ   VarTIQ  AvNIQ   AvTAN   VarTAN  AvNAN   VarNAN
1    0.250  1.667   19.444  0.083   6.667   44.444  0.333   0.444
2    0.225  4.355   149.610 0.065   19.355  374.610 0.290   0.375
3    0.100  0.222   0.938   0.011   2.222   4.938   0.111   0.123


Util - the Utilization at a Node
AvTIQ, VarTIQ - Mean and Variance of the waiting time in queue at a node
AvNIQ - Mean queue length at a node
AvTAN, VarTAN - Mean and Variance of time spent at a node
AvNAN, VarNAN - Mean and Variance of the number of customers at a node
```

The similar calculation as the above shows that the time spent in the node 2 – the consult room is typically 19 minutes and in the most cases not longer than 80 minutes.

Exercises

Remark:

System properties with its parameters will be given individually for every student.

Hence, the following exercises are defined without them.

Exercises:

Exercise 1

You have learnt the theory of queues in performance evaluation, and how to do the analysis.

Perform a similar analysis for changed parameters (given individually by the teacher).

Exercise 2

For a client-server service (given individually by the teacher) do the following:

- build a test application to obtain the chosen efficiency metrics,
- build a simulation model of the service using queues and obtain the same metrics by the simulation,
- build a queuing system model and obtain the metrics using the theoretical analysis.

This exercise will be repeated for some various services, protocols and scenarios.