SimNet, A Tool for Simulation with Application to Project Network Analysis

Fredrick von Schoultz and Aimo Törn
Åbo Akademi University
Computer Science Department
SF-20520 ÅBO, Finland

***Abstract*— Simulation Nets (SN) are Petri Nets extended for convenient modelling of discrete event simulation problems. The extensions included are zero testing, firing time for transitions, or-logic and net hierarchies. The tool SimNet accepts a text equivalence of SNs and performs the simulation implied by the SN model. Validation is facilitated by providing trace, animation, and statistics collection. An application to project network analysis is described.**

Key words: Petri Nets, discrete event simulation, Simula, Simulation Nets, project networks

I. Introduction

Simulation Nets or as they then were called, Simulation Graphs, were introduced in 1981 as a simulation modelling tool based on extended Petri Nets [Törn 1981]. The first computer tool, Simnex, capable of reading a single Simulation Net and performing the implied simulation was presented in 1985 [Törn 1985]. Simnex included inhibitor arc extension and time and provided trace facilities and automatic statistics collection and output of these for places and transitions from which the simulation results could be deduced. In 1986 an enhanced tool allowing hierarchies of nets and interactive creation and editing of the nets was presented [Törn 1986]. A book on Simulation Nets was published in 1991 [Törn 1991].

All of the tools are written in the object oriented language Simula. The latest version, SimNet, is described here. The tool is available for VAX/VMS and for PCs. The PC version includes animation facilities.

The new features that has been added to SimNet [von Schoultz 1993] after 1991 when [Törn 1991] was published are been presented and illustrated by modelling a project. Some of the original commands have been renamed. This has been done in order to give the functions more descriptive names.

II. Simulation Nets

Simulation nets have been used to model production systems, hospitals, and traffic control systems. Simulation nets can also be a tool for planning projects. By simulating the progress of the project, the project manager can take a peek at the possible future.

Using historical information from previous or completed parts of the project and the prevailing events in the project environment, projections can be made about the future of the project. Outputs of simulation can alert management to impending or potential problems. The advance alert will facilitate better management decisions. [Badiru 1991]

Projects are often represented by PERT (Program Evaluation and Review Technique) networks. In order for Simulation nets to be a useful tool when modelling and simulating projects there needs to be an easy way of transforming the PERT networks info Simulation nets.

A modification of the project used as an example in [Badiru 1991] will be used to demonstrate the use of Simulation Nets and SimNet.

A. Transformation of PERT networks into Simulation nets

Activities in a PERT network have similar properties as transitions in Simulation nets. Both take a certain amount of time to execute and they need certain conditions to be satisfied in order to start. In a PERT network the starting conditions of activities are that the predecessors have finished (or reached an certain point) and that there are the necessary resources available. In Simulation nets the conditions are represented by places containing tokens. Both the state of the predecessors and the resource allocation can be described using places and tokens.

When transforming a PERT network into a Simulation net the following steps have to be taken:

- Replace every activity with a transition. Associate a distribution to the transition that describes the duration of the activity.
- Replace every arrow with a place. Connect the proceeding transition to the place with an arc which direction is from the transition to the place. Connect the place to the succeeding transition with an arc which direction is from the place to the transition.

An example of such a transformation is given in Figures 1 and 2.

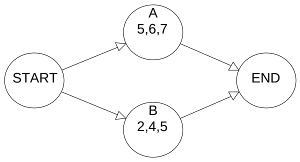


Figure 1: A small PERT network

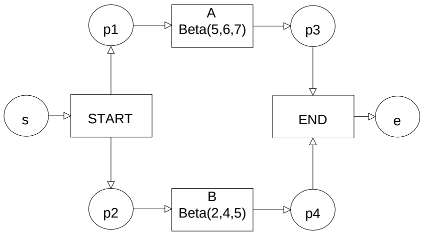


Figure 2: The PERT network from Figure 1 transformed into a Simulation net

Several probability density functions have been proposed for the modelling of nondeterministic activity times. These include the normal distribution, uniform distribution, logonormal distribution and beta distribution. The beta distribution is by far the most used for representing the probabilistic nature of activity times. [Badiru 1991b] Three time estimates, a (optimistic time), m (most likely time) and b (pessimistic time), are often used in PERT analysis to describe the activity time. [Moder 1970] A beta distribution can easily be fitted for these three time estimates. [Badiru 1991b]

Resources can be modelled by adding a place for each resource type. Each of these places are then assigned as many tokens as there are available resources of that particular type. These places are connected to the transitions that use the resources so that the number of resources needed is a starting condition for the transition. When the transition finishes the resources are placed back into the resource pool if the resources are of a reusable type.

B. Problem Setting

A modification of the Project Network Example used in [Badiru 1991] is used to demonstrate Simulation nets and SimNet. The PERT network describing the project is shown in Figure 3.

Further we know that activity G is sometimes delayed due to a breakdown of the production line. From previous projects we know that the breakdown happens on the average one time out of twenty. When such a breakdown happens it takes five time units to repair, i.e. the activity time will be five time units longer.

In the example network there is only one kind of resources. The resource is reusable. The resource requirements for the different activities in the sample project are shown in Table 1. During the repair of

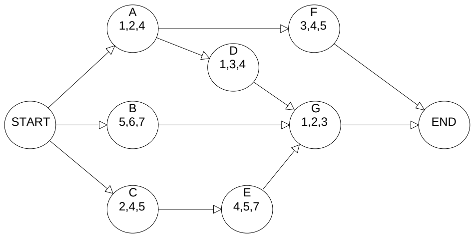


Figure 3: Project Network Example

Activity	Resource Units Required
A	3
B	5
C	4
D	2
E	4
F	2
G	6

Table 1: Resource constraints for the activities in the sample project.

the possible breakdown of the production line during activity G no resources of this type are needed.

C. Model Specification

The PERT network is transformed into a simulation net according to the rules given in the previous section. Beta distributions are used to model the duration of the activities, these are fitted from the a, b and m values given in the PERT network. The resource is modelled by one place initially containing the total number of available resources.

The possibility of the production line breaking down during activity G is modelled using or-out arcs. The or-out arcs introduce the possibility to easily model probabilistic activities. The simulation net that describes the project network is shown if Figure 4. The result will be the average over 100 realisations of the project.

From the resource constraints shown in Table 1 we see that the minimum of resources needed for the project to be completed is six. It is of interest to see how adding additional resources beyond this will influence the duration of the project. This can easily be done by running the model with a different number of tokens initially in the resource place R.

C.1. Or-out arcs

Probabilities can be expressed using or-out arcs. The or-out arc in Figure 5a will distribute the tokens between the GSEND queue and the BROKEN queue in the proportion of 19:1. The proportions by which the tokens are to be distributed are written next to the or-out arc. If the proportions are omitted they are assumed to be 1:1:..:1. The number of tokens to be deposited are written next to each branch of the arc. If the number of tokens to be deposited is

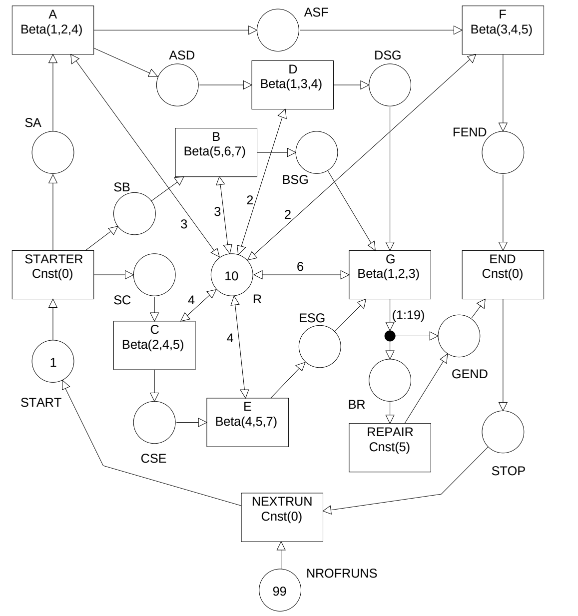


Figure 4: The Simulation net that describes the project network.

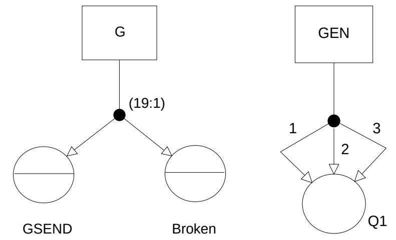


Figure 5: a) An or-out arc used to variate the path taken b)An or-out arc used to variate the number of tokens to deposit

omitted it is assumed to be one.

The or-out arc can also be used to variate the number of tokens to be deposited. In Figure 5b 1, 2 or 3 tokens are deposited with equal probability. In this case the or-out arcs branches are connected to the same place. The branches of the or-out arc are assigned different capacities.

III. SimNet representation of Simulation Nets

Simulation Nets can be described using a textual notation which SimNet can load and execute. The command `loadnet` in SimNet is used to load nets.

In the file that describes the net all blank lines are disregarded. All lines that start with `!` are treated as comments. Comments and blank lines can be inserted anywhere in the file for better readability.

The first line in the file that is not a comment nor a blank line has to contain either the identifier `Net:` followed by the name of the net or the identifier `Lib` which indicates that the net described in the file is a library module that can be used in several places in the model.

Each line that starts with `Trans:` describes a transition. Transitions are declared using the following format:

```
Trans:  <transition> [<distribution> <parameters>]
```

If the time function with its parameters is omitted the time is assumed to be zero `CNST 0`.

The transitions in-places and out-places are declared on the following lines. The identifier `Inpl:` has to proceed the declaration of the first in-place. All the places declared on the following lines will be treated as in-places to the transition. To start declaring out-places for the transition the identifier `Outpl:` is inserted in a similar way. All characters between `Inpl/Outpl` and `:` are disregarded. In-places and out-places are declared using the following format:

```
Inpl:  <place> [nr of tokens to consume] [init cont]

Outpl:  <place> [nr of tokens to deposit] [init cont]
```

The number of tokens to consume/deposit and the initial content of the place are optional. If no initial content is specified and the place has not been declared earlier, the initial content will be set to 0. If the number of tokens to consume/deposit is not specified, it is assumed to be 1. Notice that it is not possible to specify the initial content without specifying the number of tokens to consume/deposit.

To express that a group of places are connected to the transition with an or-out arc the identifier `Or-out:` is used. The declaration of or-outs are made using the following format:

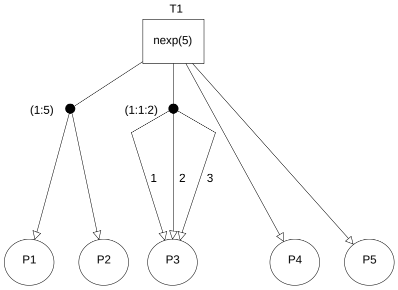
```
Or-out:  <proportion> <place> [nr to dep] [init cont]
             ...
         <proportion> <place> [nr to dep] [init cont]
```

If a transition is connected to its out-places with several or-out arcs the identifier `Or-out:` must be inserted before the declaration of each or-out arc. In a similar way the the identifier `Outpl:` must be inserted when declaring the first ordinary out-place after an or-out. The declaration of a transition with multiple or-outs and multiple out-places is shown in Figure 6. The SimNet code of the project network can be seen in Figure 7.

A. Expansion Information

If the model is large, it is convenient to build it from sever smaller nets each describing a certain part of the whole model. The nets are coupled together before the model is executed. The expansion information can be stored in a file that can be loaded with the command `loadexp` after loading the nets.

Both places and transitions can easily be expanded. When a place is expanded two places in different




```
!Multiple or-outs

Net: TestNet

TRANS: T1                 NEXP   5.00

     OR-out:              1          P1           1     0
                          5          P2           1     0

     OR-out:              1          P3           1     0
                          1          P3           2     0
                          2          P3           3     0

     Outpl & Nr to dep & Ini:        P4           1     0
                                     P5           1     0
```

Figure 6: Declaration of transition with multiple or-out arcs

```
!Pert network

Net: PROJ

TRANS: STARTER              CNST 0
     Inpl & Nr to fire & Ini:         START        1     0

     Outpl & Nr to dep & Ini:         SA           1     0
                                      SB           1     0
                                      SC           1     0

TRANS: A                    BETA 1 2 4
     Inpl & Nr to fire & Ini:         SA           1     0
                                      R            3     10

     Outpl & Nr to dep & Ini:         ASD          1     0
                                      ASF          1     0
                                      R            3     10

TRANS: B                    BETA 5 6 7
     Inpl & Nr to fire & Ini:         SB           1     0
                                      R            5     10

     Outpl & Nr to dep & Ini:         BSG          1     0
                                      R            5     10

TRANS: C                    BETA 2 4 5
     Inpl & Nr to fire & Ini:         SC           1     0
                                      R            4     10

     Outpl & Nr to dep & Ini:         CSE          1     0
                                      R            4     10

TRANS: D                    BETA 1 3 4
     Inpl & Nr to fire & Ini:         ASD          1     0
                                      R            2     10

     Outpl & Nr to dep & Ini:         DSG          1     0
                                      R            2     10

TRANS: E                    BETA 4 5 7
     Inpl & Nr to fire & Ini:         CSE          1     0
                                      R            4     10

     Outpl & Nr to dep & Ini:         ESG          1     0
                                      R            4     10

TRANS: F                    BETA 3 4 5
     Inpl & Nr to fire & Ini:         ASF          1     0
                                      R            2     10

     Outpl & Nr to dep & Ini:         FSEND        1     0
                                      R            2     10

TRANS: G                   BETA 1 2 3
     Inpl & Nr to fire & Ini:         BSG          1     0
                                      DSG          1     0
                                      ESG          1     0
                                      R            6     10

     Outpl & Nr to dep & Ini:         R            6     10

     OR-out:              19          GSEND        1     0
                           1          BROKEN       1     0

TRANS: REPAIR               CNST 5
     Inpl & Nr to fire & Ini:         BROKEN       1     0

     Outpl & Nr to dep & Ini:         GSEND        1     0


TRANS: END                  CNST 0
     Inpl & Nr to fire & Ini:         FSEND        1     0
                                      GSEND        1     0

     Outpl & Nr to dep & Ini:         STOP         1     0

TRANS: NEXTRUN              CNST 0
     Inpl & Nr to fire & Ini:         STOP         1     0
                                      NROFRUNS     1     100

     Outpl & Nr to dep & Ini:         START        1     1
```

Figure 7: The SimNet listing of the project network.

nets are made into one place that connects the two nets. This can be very useful if we want to model two or more projects with a shared resource. Both models can be designed separately and then be coupled together before the simulation is started. The expansion of a place `R` that exists in both `FIRSTNET` and `SECONDNET` is described using the following format:

```
Place R InNet FIRSTNET SameAs R InNet SECONDNET
```

When a transition is expanded the transition is replaced with a subnet. The expansion of a transition `T1` into the net `NET` is described using the following format:

```
Trans T1 ExpWithNet NET
```

IV. Validation and Simulation Results

Simulation Nets are theoretically based and therefore formal techniques such as safeness, liveness and extended reachability analysis can be used for analysing and validating the model. The hierarchical design of SN allows separate validation of the subnets. Several different solutions can be tested easily by replacing a subnet with another and therefore SNs are well suited for rapid prototyping. Submodels can be analyzed separately with e.g. the extended reachability tree technique [Törn 1991]. As the whole model is the sum of its components the aggregated model can be analyzed using the same methods.

When executing the model with SimNet statistics about the different components in the model is automatically collected. There is also a possibility to get a trace report about the activities and events in model. Both the statistics and the trace report can be saved on file for further validation of the simulation results. Also, if the user wants to monitor the activities of the different components in the model during the execution the animation facility can be used.

A. Statistics and Trace

The statistics collection is turned on using the `stat` command. The user can specify which places, transitions and nets are to be included in the statistics report. Also the time to start the collecting of the statistics can be specified. The use of statistics are further explained in [Törn 1991].

The trace function is turned on using the `trace` command. The user can specify for which nets a trace report is required. The user can also specify some critical places to be included in the trace. The content of these places is displayed together with the information concerning the firing of transitions. The use of trace is further explained in [Törn 1991].

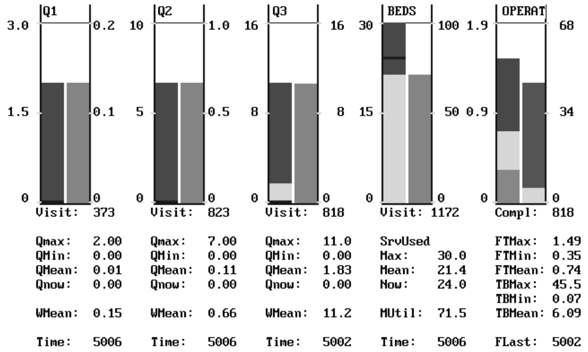


Figure 8: Snapshot of the animation

B. Animation

The animation facility is available only on the PC version of SimNet. The animation facility allows the user to follow the dynamic interaction of the different components in the model during the simulation.

Animation will help the user to convince himself that the model is working correctly. This is important when using the simulation results as a base for decision making. Also some results about the dynamics of the system can be obtained by following the animation output.

When animating Simulation Nets three types of entities are of interest; transitions, places that act as queues and places that act as resources. These three categories of entities have different properties that are of interest for the user during the simulation.

SimNet allows at most five entities to be animated at the same time. The entities to be animated can be of any of the three types. Statistics are collected automatically for the animated entities. If the user wants to animate more than five entities he can rerun the simulation with the same seed value and different entities animated. It is in doubt if a capability of animating more than five entities at the same time on the screen would increase the usefulness of the tool. Animating more than five entities at one time would split the users attention between too many entities.

A snapshot of the animation during the simulation of the Hospital Case [Törn 1991] with two arrival queues Q1, Q2 an operation queue Q3, available beds BEDS and the operation activity OPERAT is shown in Figure 8.

When transitions are animated information about the firing time (`FT`) and the time between (`TB`) the firings is displayed, see the last item in Figure 8. For both `FT` and `TB` the maximum, minimum and the mean time is displayed. The values that describe the `FT` are shown on the left half of the graph and the values that describe the `TB` on the right. The maximum is shown in red (the darkest shade of gray in the figure), the minimum in green (the middle shade of gray in the figure) and the mean level in yellow (the lightest shade of gray in the figure). All the values are also shown numerically below the graph. Additionally the time of the latest firing of the transition is shown as a numerical value.

For places that are used as queues in the model the animation focuses on the queue length and the average waiting, e.g. the number of tokens in the queue and the time that a token spends in the queue, see the three first items in Figure 8. On the left half of the graph the maximum queue length (`QMax`) is shown in red, the average queue length (`QMean`) in yellow, the minimum queue length (`QMin`) in green and the current queue length in blue (a black stripe in the figure). The average time that a token has waited in the queue (`WMean`) is shown in green on the right half of the graph. All the values are also shown numerically below the graph. Additionally the time for the last change of the the queues content is shown numerically below the graph.

When a place that serves as a resource in the model is animated information about utilization of the resource type is displayed, see the fourth item in Figure 8. On the left side of the graph the maximum number of resources used is displayed in red, the average number of resources used in yellow and the current number resources in use in blue (a black stripe in the figure). On the right half of the graph the average utilization (`MUtil`) level (0-100%) is displayed in green. All the values are also displayed numerically below the graph. Additionally the time of the last change is displayed numerically below the graph.

The animation can be viewed either stepwise or continuously. The mode is chosen when the animation is turned on. If stepwise animation is chosen the user must press some key each time the simulation is to progress one more step. If a key is held down the simulation will run continuously until the key is released. If continuous animation is chosen the user can specify a delay that is used to slow down the speed of the animation.

The animation is turned on with the command **`anim`** in SimNet. Figure 9 shows in detail how the animation is interactively turned on for a place that serves as a resource in the model. The animation will take place during two separate time intervals, the first starting at time 100 and ending at time 500 and the second starting at time 1800 and ending at time 2300.

The animation can be turned off with the command **`noanim`**. If a different set of places and transitions are to be animated during the next run the animation first has to be turned of before specifying the new places and transitions.

```
=> setanim

Stepwise or delayed animation? (s/d): s
Stepwise animaton set.

Animation of : beds

Does the place represent a queue or a server? (q/s): s
Ok, the place represents a server.

Give time to start animation: 100
Give time to stop animation: 500
Give time to start animation: 1800
Give time to stop animation: 2300
Give time to start animation:

Animation set for BEDS

Animation of :
Start statistics at time : 0

=>
```

Figure 9: The animation is turned on

C. Simulation Results

The experiments for the project network are started with the minimal number (6) of resources. Experiments are then conducted for resources numbering up to 15. It was found that having more than 13 resources available does not improve the completion time of the project. The maximal number of resources in use at the same time was 13.

From the simulation results shown in Table 2 it can be seen that adding a resource has different impact on both the mean completion time for the project and also on the utilization level of the resources, depending on how many resources there are initially. By increasing the number of resources from six to seven the mean completion time of the project is decreased by two time units. Adding one more resource thus having a total of eight resources does shorten the mean completion time of the project. Having nine resources available decreases the mean completion time of the project with more than three time units compared with the mean completion time with both seven and eight resources. Adding a ninth resource also gives a better utilization level of the resources compared to the utilization of eight resources. If one has ten resources available the tenth resource can be used in another project because the results show that the model can not use ten resources att the same time. Increasing the resources to eleven does not either give a significant improvement of completion time. The best mean completion time is reached with twelve or more resources. For twelve resources the utilization level of the resources is higher than for eight, ten and eleven resources.

As can be concluded by this Simulation nets and SimNet are very usable for analysing a project network.

Nr of resources available	Resources max used at one time	Resources Utilization %	Mean completion time
6	6	84.06	19.14
7	7	81.15	16.98
8	8	71.00	16.98
9	9	78.81	13.61
10	9	70.94	13.61
11	11	68.75	13.15
12	12	73.32	10.97
13	13	67.65	10.97
14	13	62.81	10.97
15	13	58.64	10.97

Table 2: Results from experiments with 6-15 resources, average over 1000 evaluations.

V. Technical notes

The PC-version of SimNet can be obtained by anonymous ftp from `tor.abo.fi`. In order to run PC-SimNet, a PC with the following configuration is needed:

- a 286, 386 or 486 CPU
- a 287, 387 or 487 FPU
- a VGA display
- the ANSI.SYS driver
- at least 600 Kb of free memory under DOS

To install SimNet the following steps has to be taken:

- Copy the archives to a suitable directory on your hard disk.
- Uncompress the archives using LHarc.
- Check that you have the ANSI.SYS driver installed.
- Check that you have at least 600 Kb of free memory.
- Start the program.

References

[Badiru 1991] Adedeji B. Badiru, A simulation approach to PERT network analysis, *SIMULATION* 57:4, 245-255.

[Badiru 1991b] Adedeji B. Badiru, Starc 2.0: An improved PERT network simulation tool, *Computers ind. Engng* Vol 20, No 3, 389-400.

[Moder 1970] Joseph J. Moder, Cecil R. Phillips, *Project management with CPM and PERT*, Van Nostrand Reinhold Company, 360 pp.

[von Schoultz 1993] F. von Schoultz, Master Theses, *PC-SimNet simulering med animering*, Åbo Akademi University, Department of Computer Sciense.

[Törn 1981] A. Törn, Simulation Graphs: A general tool for modelling simulation designs, *SIMULATION* 37:6, 187-194.

[Törn 1985] A. Törn, Simulation Nets, a simulation modelling and validation tool, *SIMULATION* 45:2, 71-75.

[Törn 1986] A. Törn, Simulation Nets - an application of Petri Nets to system simulation, In: Ruschizka, M (ed.), *Transactions on Scientific Computing - 85*, Vol. 2, Computer Systems: Performance and Simulation, Elsevier, 213-218.

[Törn 1991] A. Törn, *Simulation Modelling*, Åbo Akademi University, Reports on Computer Science & Mathematics, Ser. B, No 12, 140 pp.