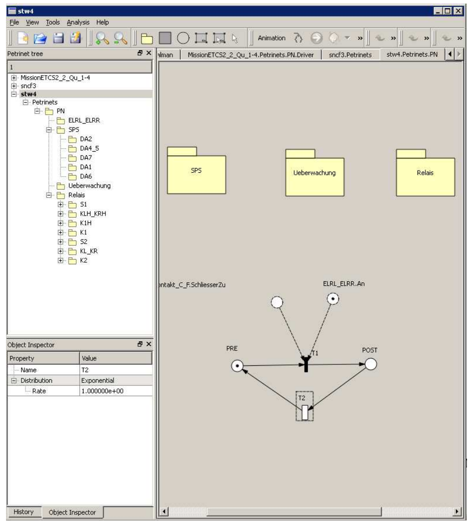
Features

Formal Modelling using hierarchical Petrinets

Real world systems are too complex to be modelled using a single sheet Petrinet. The support of hierarchical modelling by shared places and transitions is an easy to use approach implemented in ·-Tool.



stw4

File View Tools Analysis Help

Petrinet tree

- MissionETCS2_2_Qu_1-4
- sncf3
- stw4
  - Petrinets
    - PN
      - ELRL_ELRR
      - SPS
        - DA2
        - DA4_5
        - DA7
        - DA1
        - DA6
      - Ueberwachung
      - Relais
        - S1
        - KLH_KRH
        - K1H
        - K1
        - S2
        - KL_KR
        - K2

Object Inspector

Property	Value
Name	T2
Distribution	Exponential
Rate	1.000000e+00

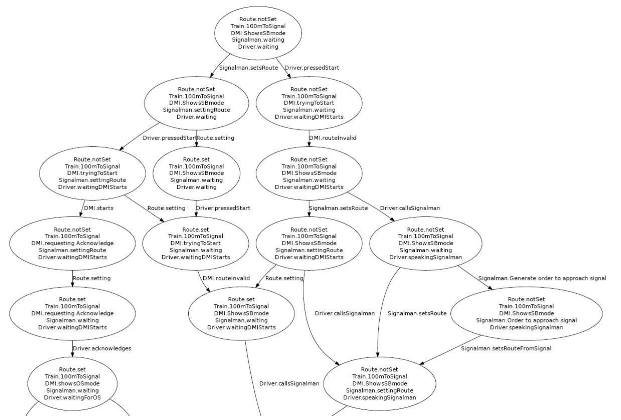
Debugging techniques: Animation

Modelling with Petrinets is a kind of a high level programming: defining of states, transitions and conditions for a complex system is an error prone task: e.g. using an inhibitor arc instead of test arc is easy to overlook. Therefore it is essential to have debugging support by the modelling tool. π-Tool provides

- Marking game – animation of transition firing
- Highlighting of active transitions
- Enforcement firing of a desired transition
- Deadlock-detection
- Calculation of most probable firing sequence to any transition

Reachability analysis

π-Tool is able to calculate all reachable system states (reachability graph), as long as it fits into the memory of PC. It is used for the calculation of the reachability of transitions, deadlocks, Markov Chains, as a mathematical prove of unreachability of some states and for process visualisation.


Route.notSet Train.100mToSignal DMI.ShowsSBmode Signalman.waiting Driver.waiting
Signalman.setsRoute
Driver.pressedStart
Route.notSet Train.100mToSignal DMI.ShowsSBmode Signalman.settingRoute Driver.waiting
Route.notSet Train.100mToSignal DMI.tryingToStart Signalman.waiting Driver.waitingDMIStarts
Driver.pressedStart
Route.setting
DMI.routeInvalid
Route.notSet Train.100mToSignal DMI.tryingToStart Signalman.settingRoute Driver.waitingDMIStarts
Route.set Train.100mToSignal DMI.ShowsSBmode Signalman.waiting Driver.waiting
Route.notSet Train.100mToSignal DMI.ShowsSBmode Signalman.waiting Driver.waitingDMIStarts
DMI.starts
Route.setting
Driver.pressedStart
Signalman.setsRoute
Driver.callsSignalman
Route.notSet Train.100mToSignal DMI.requesting Acknowledge Signalman.settingRoute Driver.waitingDMIStarts
Route.set Train.100mToSignal DMI.tryingToStart Signalman.waiting Driver.waitingDMIStarts
Route.notSet Train.100mToSignal DMI.ShowsSBmode Signalman.settingRoute Driver.waitingDMIStarts
Route.notSet Train.100mToSignal DMI.ShowsSBmode Signalman.waiting Driver.speakingSignalman
Route.setting
DMI.routeInvalid
Route.setting
Signalman.Generate order to approach signal
Route.set Train.100mToSignal DMI.requesting Acknowledge Signalman.waiting Driver.waitingDMIStarts
Route.set Train.100mToSignal DMI.ShowsSBmode Signalman.waiting Driver.waitingDMIStarts
Driver.callsSignalman
Signalman.setsRoute
Route.notSet Train.100mToSignal DMI.ShowsSBmode Signalman.Order to approach signal Driver.speakingSignalman
Driver.acknowledges
Signalman.setsRouteFromSignal
Route.set Train.100mToSignal DMI.showsOSmode Signalman.waiting Driver.waitingForOS
Driver.callsSignalman
Route.notSet Train.100mToSignal DMI.ShowsSBmode Signalman.settingRoute Driver.speakingSignalman

Visualisation of reachability graph as a process specification

Modelling of stochastic Systems

Besides Boolean conditions, time conditions can be assigned to transitions, which indicate how long they must be active before they fire. This time can be deterministic or stochastic according to a specified distribution function. π-Tool supports

- Negative exponential distribution
- Normal distribution
- Gamma distribution
- Weibull distribution
- Uniform distribution
- LogNormal distribution

As long as the model contains just immediate and negative exponential distributions, a Markov-Chain can be generated basing on reachable states.

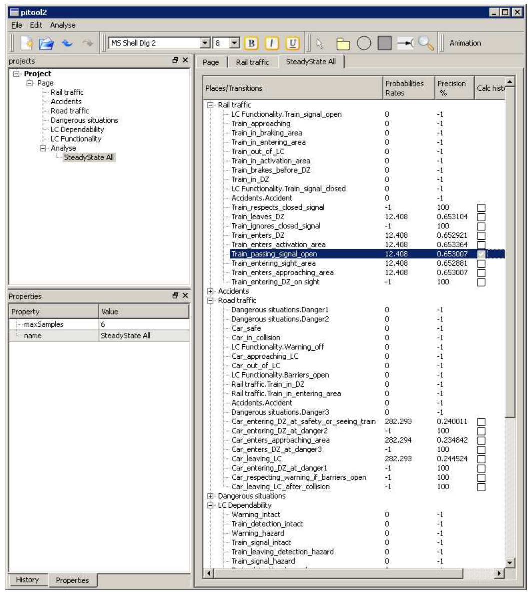
Markov-Chains generation and evaluation

Markov chain is a well known description mean for modelling of stochastic systems. In contrast to a stochastic Petrinet it models global system states, e. g. a system with 9 objects each with 3 states would be modelled using 27 places in a Petrinet or 19683 places in a Markov chain. π-Tool generates such a Markov-Chain from the Petrinet automatically.

The most important results for RAMS-Analysis are the firing rates of transitions in a steady state. Depending on the meaning of a transition it can be interpreted as throughput for performance analysis or e. g. hazard rate for safety analysis. π-Tool provides a calculation of steady state firing rates using analytical solution with a relative precision up to 10^{-18} which is sufficient for analysis of SIL4-Systems.

Monte-Carlo-Simulation

Not all transitions in real world systems follow memoryless negative exponential distributions. Diagnosis systems, maintenance work, timetable based operation and age based failure are examples for deterministic, Weibull or normal distributed random numbers. Assuming negative exponential distribution for these parameters would generally produce an error of several orders of magnitude. Such systems can be analysed using Monte-Carlo simulation. π-Tool provides an efficient c++-based implementation of the Monte-Carlo simulation with an estimation of current error of firing rate.



pitool2

File Edit Analyse

projects

- Project
  - Page
    - Rail traffic
    - Accidents
    - Road traffic
    - Dangerous situations
    - LC Dependability
    - LC Functionality
    - Analyse
      - SteadyState All

Properties

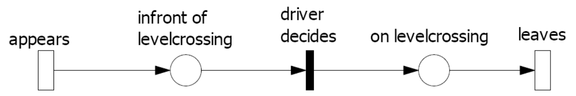
Property	Value
maxSamples	6
name	SteadyState All

Places/Transitions	Probabilities Rates	Precision %	Calc hist
Rail traffic			
LC Functionality.Train_signal_open	0	-1	
Train_approaching	0	-1	
Train_in_braking_area	0	-1	
Train_in_entering_area	0	-1	
Train_out_of_LC	0	-1	
Train_in_activation_area	0	-1	
Train_brakes_before_DZ	0	-1	
Train_in_DZ	0	-1	
LC Functionality.Train_signal_closed	0	-1	
Accidents.Accident	0	-1	
Train_respects_closed_signal	-1	100	☐
Train_leaves_DZ	12.408	0.653104	☐
Train_ignores_closed_signal	-1	100	☐
Train_enters_DZ	12.408	0.652921	☐
Train_enters_activation_area	12.408	0.653364	☐
Train_passing_signal_open	12.408	0.653007	✓
Train_entering_sight_area	12.408	0.652881	☐
Train_enters_approaching_area	12.408	0.653007	☐
Train_entering_DZ_on sight	-1	100	☐
Accidents			
Road traffic			
Dangerous situations.Danger1	0	-1	
Dangerous situations.Danger2	0	-1	
Car_safe	0	-1	
Car_in_collision	0	-1	
LC Functionality.Warning_off	0	-1	
Car_approaching_LC	0	-1	
Car_out_of_LC	0	-1	
LC Functionality.Barriers_open	0	-1	
Rail traffic.Train_in_DZ	0	-1	
Rail traffic.Train_in_entering_area	0	-1	
Accidents.Accident	0	-1	
Dangerous situations.Danger3	0	-1	
Car_entering_DZ_at_safety_or_seeing_train	282.293	0.240011	☐
Car_entering_DZ_at_danger2	-1	100	☐
Car_enters_approaching_area	282.294	0.234842	☐
Car_enters_DZ_at_danger3	-1	100	☐
Car_leaving_LC	282.293	0.244524	☐
Car_entering_DZ_at_danger1	-1	100	☐
Car_respecting_warning_if_barriers_open	-1	100	☐
Car_leaving_LC_after_collision	-1	100	☐
Dangerous situations			
LC Dependability			
Warning_intact	0	-1	
Train_detection_intact	0	-1	
Warning_hazard	0	-1	
Train_signal_intact	0	-1	
Train_leaving_detection_hazard	0	-1	
Train_signal_hazard	0	-1	

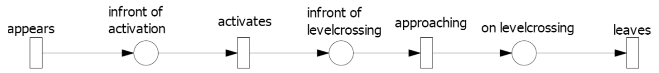
Another application area for Monte-Carlo simulation is the analysis of complex systems. Modelling is a time consuming and hence an expensive task. As the size of Markov-Chain depends on a number of global system states there is a high risk at the end of the modelling to get a state space which is bigger than the available memory on PC. It could happen just by adding an additional object with e. g. 5 states. It would increase the number of global states by factor 5 making the Markov chain too big for calculation. The Monte-Carlo simulation is independent from number of reachable states and can be used in such situations.

Modelling example

As an example a level crossing will be modeled. There are three kinds of objects in the system: cars, trains and warning lights. Model for car traffic is quite simple:

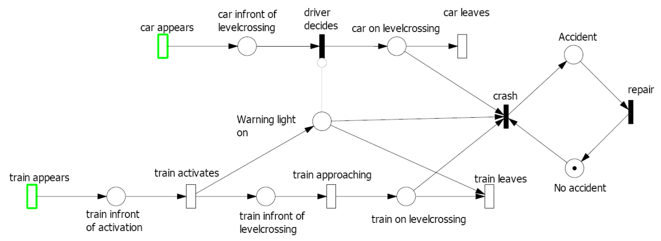


The train model is similar:



The train and car models "interact" with each other by means of warning light: when the train passes the activation sensor, the warning light is activated and when the train leaves the warning light is deactivated.

The car driver takes in account the state of the warning light for the decision to enter or not the level crossing.



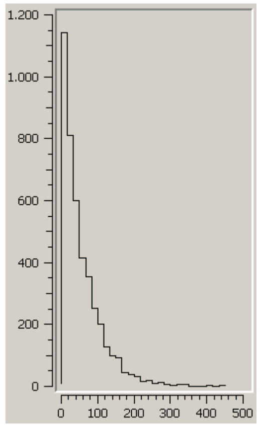
When train and car are both on the levelcrossing there is a crash. The "white" transitions are negative exponential distributed because the running time and the length of the train and cars are stochastic. The simulation in π-Tool gives following firing rates for a defined set of parameter:

Transition	SS rates	sim rates
PN.appears	96.6735	96.6973
PN.driver decides	96.6735	96.6973
PN.leaves	96.6553	96.6795
PN.activates	1.95855	1.95867
PN.appears	1.95855	1.95866
PN.approaching	1.95855	1.95867
PN.leaves	1.94041	1.94082
PN.crash	0.0181384	0.0178611
PN.repair	0.0181384	0.0178611

SS-Rates : Steady state rates calculated with Markov Chain

Sim rates: Simulated rates calculated with Monte-Carlo simulation.

For every transition a histogram can be calculated.



Histogram from Monte-Carlo-Simulation

Customers

Institute for Traffic Safety and Automation Engineering

Federal office of transport, Switzerland

Requirements

- 100 MB harddrive disk
- 1024x768 monitor resolution
- 1 GB RAM
- 1 GHz processor

Contact

Hermann-Blenk-Str. 22

38108 Braunschweig

ph	+49 163 5768550
fx	+49 531 3544416
em	info@iqst.de
w3	www.iqst.de

Institute for Quality, Safety and Transportation

Institute for Quality, Safety and Transportation

Petrinet modeling and analysis tool

π-Tool