Discrete-Event Simulation of Hybrid Petri Nets with General One-Shot Transitions

Daniël van de Giessen
University of Twente
P.O. Box 217, 7500AE Enschede
The Netherlands
d.vandegiessen@student.utwente.nl

ABSTRACT

Fluid critical infrastructures such as water treatment and distribution networks are considered of increasing importance for our economy and society. Previously, M. Gribaudo and A. Remke [4] introduced a hybrid Petri net formalism specifically tailored towards these kind of infrastructures. More recently efficient analysis algorithms [2] were developed and integrated into a software solution, the Fluid Survival Tool (FST) [6], which allows for analytical evaluation of Stochastic Timed Logic (STL) [3] formulas against hybrid Petri nets with a general one-shot transition (HPNG).

FST however does not allow for analysis of hybrid Petri nets with multiple stochastic variables. This paper describes an extension to FST implementing Discrete-Event Simulation (DES) allowing FST to perform transient analysis on such models.

Keywords

Petri net, HPNG, Fluid Survival Tool, Discrete-Event Simulation.

1. INTRODUCTION

HPNGs are often restricted to contain just a single stochastic variable, since the currently available algorithms and methods for analysis of HPNGs only work for Petri nets with a single stochastic variable.

The Fluid Survival Tool is a software solution which allows for modeling and analysis of HPNG models using Stochastic Timed Logic (STL). Using the algorithms developed by H. Ghasemieh [2], it efficiently checks complex and nested properties of the so-called underlying Stochastic Time Diagram (STD), an partitioning of the state-space of HPNGs, using region-based techniques for STDs. FST allows the user to enter an input STL formula, as introduced in [3], upon which it can analytically reach a verdict for a specific time or produce a probability distribution.

1.1 Problem Statement

Although FST is able to analytically evaluate HPNG models with various conditions, due to this analytical approach and the available algorithms, it can only do so for models with no more than one stochastic transition.

21th Twente Student Conference on IT, June 23st, 2014, Enschede, The Netherlands.

Needless to say, this limits its usability by requiring simplification of models to only contain a single stochastic transition. Many of the modelled real-life scenarios may contain multiple stochastic elements and these scenario's cannot be modelled directly in HPNG form, e.g. multiple systems which can break down independently. Although simplification of such models to use only one stochastic transition is possible for some scenarios, many scenario's cannot be transformed in such a way. Currently no algorithms exists to simplify such models thus requiring this to be done manually.

The goal of this research is to implement a method for determining the probability of hybrid Petri nets with multiple stochastic transitions reaching a given state. This will greatly increase the value of the FST tool for use in future research by allowing more complex models to be tested and provides a means of validation for the analytical solutions.

1.2 Related work

There exists related work specific on Discrete-Event Simulation of HPNG models. In 2013, L. Knoben, M. Otten and R. Fransen [5] developed a simulation of a HPNG model for the specific case of a sewage water treatment facility. Although their approach was useful for this specific piece of infrastructure, it provided no general method for running such simulations.

G. Ciardo, D. Nicol and K. S. Trivedi [1] addressed simulation of Fluid Stochastic Petri Nets [8], a Petri net formalism similar to HPNG, which also contain both discrete and fluid places. In the FSPN formalism however the discrete and continuous state space are mixed, resulting in a more complex approach to simulation.

1.3 Research Questions

Based on the problem statement, this research addresses the following questions:

How can Discrete-Event Simulation be applied to HPNGs with multiple stochastic transitions?

1. What are the semantics of multiple stochastic variables in a HPNG?
2. What is a maintainable and well-performing way to structure a software implementation of Discrete-Event Simulation for HPNGs?
3. How can this functionality be integrated into the (graphical) user interface of the Fluid Survival Tool?

1.4 Method of research

To answer these questions, first the key differences between HPNGs with one and multiple stochastic transitions are determined. Based on this information, an algorithm for simulation of HPNGs using the currently available algorithms

present in FST is described. The algorithm will then be integrated into FST and some simulations are run to showcase the newly implemented functionality.

This paper is structured as follows. In Section 2 we give a brief description of HPNGs and take a look at the properties significant when introducing multiple stochastic transitions. In Section 3 the simulation algorithm is detailed, and in Section 4 we look into the actual integration into the Fluid Survival Tool. Section 5 presents the results and computation times of the implemented simulation functionality and Section 6 discusses the conclusions of the work and possible future extensions.

2. HYBRID PETRI NETS

Petri nets are directed bipartite graphs used for modelling various infrastructure. They consist of two main kinds of nodes, namely places and transitions, and directed edges called arcs. Places are defined as either being discrete or fluid. Discrete places always contain an integer, i.e. the number of tokens, whereas fluid places contain a real number, which describes the amount of fluid present in the place.

Transitions are nodes which allow for changing the contents of places. Transitions may "fire" when all incoming arcs are enabled. Four kinds of transitions exists: discrete immediate, discrete deterministic, discrete stochastic and fluid. Discrete transitions move an integer number of tokens from one discrete place to another, fluid transitions have a real-numbered transfer rate at which they transfer fluid content from one fluid place to another. The three kinds of discrete transitions differ in their firing behavior: immediate transitions fire immediately once all incoming arcs are enabled, deterministic transitions fire after a predefined period of time after all their arcs were enabled, and stochastic transitions fire once with a predefined probabilistic distribution from the moment all their arcs were enabled.

Arcs are edges connecting the places and transitions. Four kinds of arcs are defined: discrete arcs, fluid arcs, inhibitor arcs and test arcs. Discrete and fluid arcs connect between respectively two discrete and two fluid places, whereas inhibitor- and test-arcs connect discrete places to any kind of transition.

In addition to the places, transitions and arcs an initial state and a set of nine parameters functions is defined, which define various bounds and conditions such as the weight and priority used to determine activation of an arc, the maximum capacity of places, et cetera. A more specific overview of these functions can be found in [4].

If a Petri net contains both discrete and fluid places/transitions/arcs, it is called a hybrid Petri net.

Introducing multiple stochastic discrete transitions into the HPNG formalism does not change its behavior. Note that, for the purposes of simulation, stochastic discrete transitions can be treated similarly to deterministic discrete transitions. Their significant difference is in the firing time which for stochastic transitions is not predefined at one constant value but a new sample for the firing time is drawn from the distribution at the moment the transition becomes enabled.

3. IMPLEMENTING SIMULATION

Discrete-Event Simulation is a form of simulation where the operation of the simulated model is divided into discrete "events". These events are always sequential and nothing happens in the time between, allowing the simulation to directly jump from one event to the next. Although events may happen at the same time in the simulated model, their execution order is defined in a deterministic manner. Compared to continuous time simulation, which splits up the entire simulation in small slices of time, DES does not have to simulate explicitly any of the time in between events thus allowing for greater efficiency in the simulation.

(Hybrid) Petri nets are suitable for this kind of simulation by handling all the discrete transition firings as events. The HPNG formalism provides in a priority for transitions, the parameter function ϕ_p^T, to resolve the order of firing transitions which occur at the same moment in time. Fluid transitions do not influence discrete transitions (since arcs from fluid places to discrete transitions do not exist in the HPNG formalism) and their correct state can be calculated whenever such a discrete event occurs by shifting their clocks and updating the fluid levels for the passage of time. A fluid place reaching its minimum or maximum fluid level does also not influence the firing of discrete transitions, although adjustments for the fluid part of the model simulation may occur whenever this happens.

In order to simulate a model with stochastic elements we run the simulation a large number of times. During each "run" of the simulation, we simulate the outcome of each stochastic transition by picking a random value according to its probability distribution. By then combining the statistics over a large number of runs, we can approach the probability of the model being in a certain state at the end of the simulation. The probability of the system being in a specific state can then be approximated by the number of simulation runs in which the model ended up in the desired state in comparison to the total number of simulation runs.

As noted in the previous section, stochastic transitions differ from deterministic transitions only in the firing time. We will make use of this property when implementing simulation by handling stochastic transitions in the same manner as deterministic transitions during a single simulation run, using the randomly picked firing time.

3.1 DES algorithm

The algorithm for simulating a HPNG is then as follows:

```
Input: The HPNG, maximum time to simulate for, Boolean condition on an atomic or continuous property, number of runs
Output: fraction of runs fulfilling the given condition.
Algorithm:
1:  Result = 0
2:  For the given number of runs
3:      Reset the HPNG to initial state
4:      Pick random values for all stochastic transitions
5:      While the given maximum time is not reached
6:              Find the next transition to fire
7:              Update the state by firing the transition
8:      Update fluid places for the difference in time between the last transition and the given maximum time
9:      If the given condition is fulfilled
10:             Result = result + 1
11: Result = result / number of runs
```

We count the number of runs after which the given condition is fulfilled. At the start of each run, we reset the HPNG to its initial state (line 4) since each run needs to be independent. We pick random values for each stochastic transition (line 5), each from their respective probabilistic distribution, using a set of randomly generated numbers. From this point forward we can treat the stochastic transitions as being deterministic with the picked firing time for the rest of this run.

The simulation itself consists of determining the next event using the already existing algorithms (line 6), firing this transition and updating the model to account for changed state due to the firing of the transition and the elapsed time since the previous event (line 7). This process is repeated until the firing time of the next transition exceeds the specific maximum simulation time (line 5). At this point, for each enabled fluid transition we update the connected fluid places for the remaining time since the last transition that was fired (line 8).

Once the simulation run is finished; we evaluate the Boolean condition and check whether it is fulfilled for the current state of the model. If so, we increase the number of successful runs (line 10).

After all runs are finished, the variable "result" contains the number of runs in which the specified condition was met. We divide this number by the total number of runs (line 11). This number is then the result of the algorithm.

Since we used random numbers in picking the firing time of all stochastic transitions, every run of the simulation forms a Bernoulli-experiment and thus the calculated number is an approximation of the probability that the model will fulfill the specified condition at the given time. The number of runs determines how precise this approximation is, more simulation runs will yield better precision. The algorithm scales linear for the number of runs.

4. INTEGRATION IN FST

The Fluid Survival Tool already contains various functionality for working with HPNG models and is built to be easily extensible. We used various of this functionality during the implementation of the algorithm described in the previous section.

FST employs the Façade pattern for providing access to all the functionality of the models and algorithms. In order to implement Discrete-Event Simulation, we extended this Façade with an additional method, similar in signature to the existing method for analytical solving HPNGs with a single stochastic transition.

The method takes a model, a fluid place name and constant value to check for a condition, a maximum time to simulate and a given number of runs. The model is initialized and the initial state is generated through the functionality already present in FST. We then loop over all stochastic transitions in the model and transform them into deterministic transitions, saving a copy of their stochastic properties in a separate data structure. At this point we enter the loop which runs for each simulation run. For each run, we loop again over all the previously stochastic transitions and generate a random number for each, according to the distribution stored earlier. We set this value as the deterministic firing time of the transition.

We use the already present functionality in FST to run the now initialized model and check if for the given place name the fluid level is equal or larger than the specified constant. If so, we increase the count of successes with one. At this point, one simulation run is completed and we continue with the next one, until the number of runs specified is reached.

Since FST generates a graph of the probabilities up to the maximum time, we repeat this entire process for each point in the graph we want to generate according to the parameters entered in the user interface. The resulting data is stored in a file and the gnuplot program is automatically called to display the graph.

4.1 User interface

The DES functionality is made available through an addition in the FST graphical user interface. Since the tool uses the Model-View-Controller pattern for the implementation of its user interface, the change requires minimum modification of the existing code.

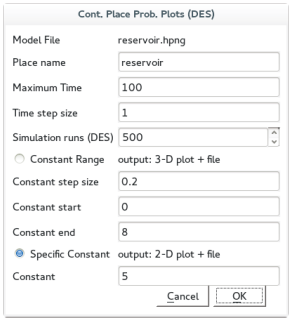

Figure 1: Interface for DES

In the menu, an extra option for accessing the Discrete-Event Simulation functionality is added. This shows the user the same dialog window as is displayed for the analytical functionality, extended with an option specific for DES: the number of runs per point in the graph. See Figure 1. This allows to easily enter the required parameters. Since the interface is almost identical to the one used for the analytical functionality of the tool it should pose no problems for existing users of the tool.

5. SIMULATION RESULTS

With Discrete-Event Simulation implemented into the Fluid Survival Tool, some simulations were run to showcase the results of this method as well provide some measurements regarding the performance of large simulations.

We will make use of the "reservoir"-model included in FST, which is described in more detail in [7]. In Figure 1 and Figure 3, the 3D graphing functionality of FST is used. The default parameters, as shown in Figure 2, were used for rendering this graph. Each graph shows the probability of a fluid place containing a certain minimum amount of fluid content on the vertical axis. The two horizontal axis display the point in time in the simulation and the minimum fluid level checked for.

Figure 2 is the result as calculated by the analytical methods already available in FST and is considered to be the exact solution to for the given parameters. The result of the simulation using 500 runs is shown in Figure 3, which is an approximation of the solution. Note that "500 runs" in this context means each single point in the graph is calculated using the described DES algorithm with 500 runs, thus to determine the total number of runs we multiply this by the specified number of points on the time and constant axis: in total the results of 2070500 runs were used for generating this graph.

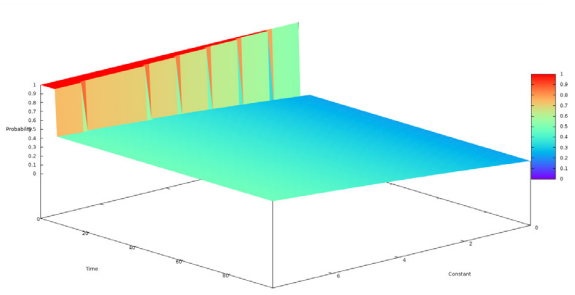

Figure 2: Reservoir result graph, analytical

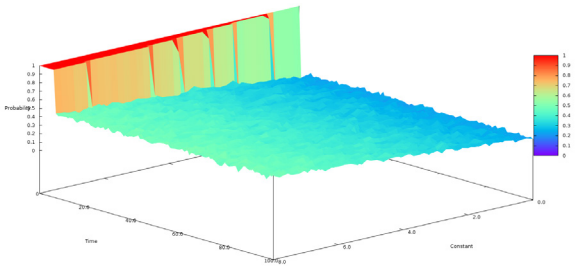

Figure 3: Reservoir result graph, DES with 500 runs per point

Table 1: Run times for calculation of single value

Number of runs	Computation time	Result value
500	8.598 ms	0.232000
5000	85.91 ms	0.219400
50000	863.862 ms	0.223000
500000	8432.7 ms	0.222496
1000000	16914.2 ms	0.220714
Analytical method	0.283 ms	0.221199

To attain some insight into the performance of the simulation algorithm some time measurements were taken. Table 1 contains the simulation time for calculating a single value (a single point in the graph), the number of runs that were performed, and the result value. As noted in Section 3, the algorithm has a linear complexity for the number of runs. This clearly shows in the measurements: for each number of runs, the simulation time scales proportionally.

The precision of the result increases with the number of simulation runs, as can be observed in the results in Table 1. The analytical result, noted in the bottom row, is approached more precise as the number of runs increases. Statistically, given that our simulations use random values, the default error margin for Bernoulli-experiments of $1/\sqrt{N}$ would apply, where N is the number of runs.

One of the primary advantages of simulation is the ability to simulate Petri nets with multiple stochastic transitions. To illustrate this, a model with two stochastic transitions was constructed, shown in Figure 4. The first transition has an exponential distribution with parameters 0.5, the second transition has a uniform distribution between 5 and 8. The center fluid place will be filled with fluid content once the first stochastic transition fires, and be drained again once the second stochastic transition fires. Thus, the center fluid place will be empty at the start of the simulation, but as time progresses the probability of it being empty will decrease exponentially. After 5 seconds, the second stochastic transition may fire, and thus the probability of the center fluid place being empty will increase again, this increase will be linear given the uniform distribution.

This model was ran through FST. Figure 5 shows the resulting 2D graph as generated with the new simulation functionality. The vertical axis displays the probability of the center fluid place

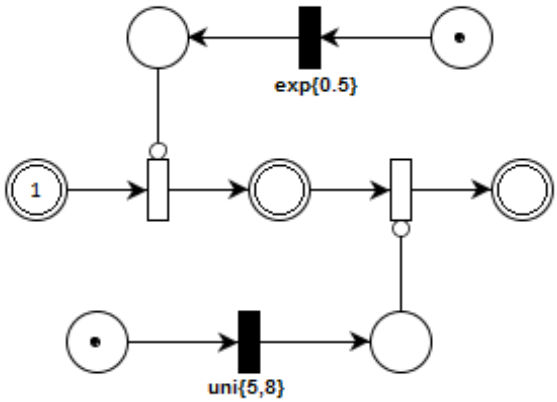

Figure 4: HPNG model with multiple stochastic transitions

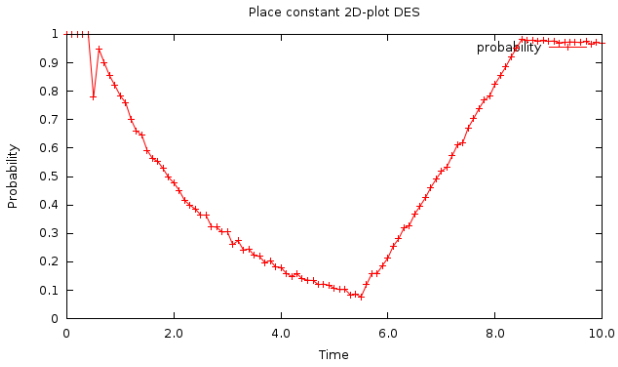

Figure 5: Multiple stochastic transitions result graph, DES with 2500 runs

being empty while the horizontal axis indicates the time passed in the simulation. This result matches our expectations about the probabilities and is a clear example of how the newly implemented functionality can be used to model and simulate systems which could not be analyzed using the previously existing methods.

6. CONCLUSION

This research focused on implementing Discrete-Event Simulation for HPNGs in the existing Fluid Survival Tool. An algorithm has been designed to simulate HPNGs and Petri nets with multiple stochastic transitions. The algorithm has a linear

complexity for the number of simulation runs, and can be extended to incorporate additional functionality such as collection of interesting measurements, additional statistics over the runtime and results.

Introducing multiple stochastic transitions does not introduce problems with the existing semantics of HPNGs. For the purposes of the simulation algorithm; we treated these stochastic transitions similarly to deterministic transitions, predetermining their firing times at the start of the simulation.

The new functionality has been incorporated into FST, allowing it to be used directly from within the tools graphical user interface. Existing interface elements were extended with the additional elements required for accessing the simulation functionality. This approach ensures the interface is consistent with the pre-existing functionality and will make it easy for existing users to start using the new functionality.

The current approach has as limitation that it currently uses the available functionality from FST for a large part of the simulation, which is not optimized for this kind of usage and thus may not yield optimal performance. Future work could rework this part of the algorithm specifically for use with DES. Such extension would also allow for further additional functionality such as gathering additional data and statistics, e.g. the fluid levels of fluid places over time, during the simulation runs. The algorithm could also be extended to allow simulation of a broader class of Petri nets, in which a stochastic transition is not limited to firing a single time.

7. REFERENCES

[1] G. Ciardo, D. Nicol and K. S. Trivedi. 1999. *Discrete-event simulation of fluid stochastic Petri nets.* In: IEEE Transactions on Software Engineering Vol. 25 Issue 2.
DOI= http://dx.doi.org/10.1109/32.761446

[2] H. Ghasemieh, A. Remke, B. Haverkort and M. Gribaudo. 2012. *Region-Based Analysis of Hybrid Petri Nets with a Single General One-Shot Transition.* In: 10th International Conference, FORMATS 2012, 18-20 Sep 2012, London, UK.
DOI= http://dx.doi.org/10.1007/978-3-642-33365-1_11
URL= http://purl.utwente.nl/publications/83495

[3] H. Ghasemieh, A. Remke and B. R. Haverkort. 2013. *Survivability evaluation of fluid critical infrastructures using hybrid Petri nets.* In: 19th IEEE Pacific Rim International Symposium on Dependable Computing, PRDC 2013, 2-4 Dec 2013, Vancouver, Canada.
URL= http://purl.utwente.nl/publications/89265

[4] M. Gribaudo and A. Remke. 2010. *Hybrid Petri nets with general one-shot transitions for dependability evaluation of fluid critical infrastructures.* In: IEEE 12th IEEE International High Assurance Systems Engineering Symposium, HASE 2010, 3-4 Nov 2010, San Jose, USA.
DOI= http://dx.doi.org/10.1109/HASE.2010.27
URL= http://purl.utwente.nl/publications/75311

[5] L. Knoben, M. Otten and R. Fransen. 2013. *De waterzuivering als Petrinet: het voorspellen van overstromingen.*
URL= http://purl.utwente.nl/essays/63890

[6] B. F. Postema. 2013. *Fluid Survival Tool: A model checker for Hybrid Petri nets*. In: 17th International GI/ITG Conference, MMB & DFT 2014, 17-19 Mar 2014, Bamberg, Germany.
DOI= http://dx.doi.org/10.1007/978-3-319-05359-2_18
URL= http://purl.utwente.nl/essays/63856

[7] B. F. Postema. 2013. *Fluid Survival Tool: A model checker for Hybrid Petri nets.* Master's thesis, University of Twente.
URL= http://purl.utwente.nl/essays/63856

[8] K. S. Trivedi and V. G. Kulkarni. 1993. *FSPNs: fluid stochastic Petri nets.* In: Proc.14th Int. Conf. On Applications and Theory of Petri Nets, Chicago, 24-31, 1993.
DOI= http://dx.doi.org/10.1007/3-540-56863-8_38