Reducing the Effect of Stiffness For a State Space-Based Simulation Method Using Adaptive Time Steps

Fabian Wickborn, Graham Horton
Lehrstuhl für Simulation
Otto-von-Guericke-Universität Magdeburg
{fabian, graham}@sim-md.de

Proxel-based simulation is a state-space based method for the transient solution of discrete stochastic models with general distributions. The method is very promising for the simulation and evaluation of rare events, even in stiff models. It computes the probability of model states and events deterministically. Contrary to rare event simulation techniques like, e.g., Importance Sampling or Splitting, it requires no further knowledge from the user and is free from sample variances, allowing for an easy handling even by simulation beginners. Further on, proxel-based simulation can be accelerated by using different time steps sizes. We show how to implement different time steps and how to choose their sizes. Two small examples are given to illustrate the application.

1 Introduction

Computer simulation is a convenient and inexpensive method for the analysis of a dynamic system, be it the reliability of a planned technical innovation or the future development of an existing economy. The real or fictitious system is abstracted to form a mathematical or an algorithmic model describing the system dynamics which are relevant to the analysis. However, not all simulation studies of interest are practically feasible.

The computational effort that is needed to generate results can become an issue even for simple stochastic simulation models when some of the activities of interest only happen very slowly while others do so rapidly. This is commonly known as stiffness. The termination of slow activities marks rare events of the model. Unfortunately, many very important practical analysis applications are built around systems with stiff dynamics and rare events. For example, errors and accidents in nuclear power plants or commercial aircrafts are rare events and it is vital for planning and engineering to evaluate the reliability of these systems. Since it is too hazardous and too expensive to perform the neccessary experiments on the real system there is a strong need for feasible simulation of the corresponding models.

In this paper, we review the feasibility and the advantages of the proxel-based simulation method for rare-event simulation problems. Although the main aim of the paper is to present the method to a new audience it also presents original contributions, such as heuristics for choosing appropriate parameters for the simulation automatically.

The organisation of this paper is as follows. In section 2 we briefly review existing solutions for stiff problems. Section 3 introduces the proxel-based simulation methods and emphasises on the significance of the timestep parameter. We give heuristics on how to determine parameter values that will result in an appropriate accuracy-runtime trade-off. In Section 4 the application of the proxel-based method is illustrated with respect to two small examples, that is, the (un-)availability of items in an inventory and the availability of a jet engine. Experiments compare the efficiency and the accuracy of the proxel-based approach and plain Monte Carlo simulation. A conclusion and an outlook which are given in Sections 5 and 6, respectively, finish the paper.

2 Review of Approaches for Rare Event Simulation

2.1 Enhanced Monte Carlo methods

The most common algorithm for the solution of stochastic problems is the Monte Carlo approach, in which samples of the behaviour of the model are drawn. While this simple algorithm can be used to solve stiff problems, it often does not so in acceptable time, due to the high number of samples which are needed to gain a sufficient amount of statistical confidence in the results. In recent years, much effort has been expended in developing time-saving methods such as Importance Sampling (IS) and Splitting. All of the methods have in common that they try to reduce the runtime of the analysis by reducing the variance of the stochastic sample gained from the model. Thereby, less samples need to be drawn to compute statistically significant results, and hence the runtime is lowered.

Importance Sampling (IS) reduces the sample variance by altering the model. Stochastic distributions of slow activities are replaced with alternative distributions (a.k.a. biasing densities) that represent much faster activities. This is called "change of measure". Then simulation is performed with the altered model. The results are weighted by what is called a likelihood ratio that is defined as the ratio between the original and the alternate probability density functions. It is hoped that the biasing density will lead to a much smaller sample variance. In many important cases IS was able to boost simulation efficiency. For example, it has been successfully applied for the simulation of rare events in queueing and reliability models [1].

Unfortunately, there is no easy way to find biasing densities that actually have a smaller variance, since the optimal biasing densities depend on the model itself. Even for well-trained and highly-skilled IS experts the selection of biasing densities remains a tough challenge, more an art than an engineering task [2, 3]. Still worse, an inauspicious choice will increase the sample variance, not reduce it. For the average simulation user Importance Sampling must surely feel like a closed book.

The idea of Splitting has been around for more than five decades [4]. With Splitting, one computes the probability of a rare event by examining paths through the reachability graph that lead to the rare event more closely. The most famous implementation of the concept of Splitting is the RESTART technique. [5], which has successfully been applied to stochastic Petri nets [6]. RESTART works in this way: While performing a replication of a simulation and when reaching a state S (i.e., reaching a certain threshold for the state variables) that leads to a rare event E, save the current state of the system and perform multiple sub-replications from this state on. Of these sub-replications some will execute the rare event and some will not, resulting in a probability of the execution under the condition that the replication was started in S ($P\{E|S\}$). It is assumed that $P\{S\}$ can be determined easily since S is not reached as rarely as E. Then, the probability of E can be computed as $P\{E\} = P\{S\} \cdot P\{E|S\}$. The challenge with RESTART (and Splitting in general, as well) is the choice of suitable thresholds for S and the number of sub-replications that need to be performed starting from S. For simple stochastic model classes like stochastic Petri nets that task can be automated [6], but [3] provides a number of references that indicate this is no longer possible for larger models or models with a multi-dimensional state space.

Both Importance Sampling and Splitting provide the ground for significant improvements in simulation efficiency. But both approaches require the user to have additional know-how in stochastics and/or Monte Carlo simulation. Both methods deliver their results as random variables and require the user to perform algorithmic engineering with his advanced knowledge to find appropriate parameters (biasing densities) for the execution of the simulation. In our opinion this is a drawback since the majority of real-world simulation users does not possess the necessary skills or, even worse, do not even take the trouble to form statistically significant results, as was criticised in [7]. A method for rare event methods that does not require more know-how than common general-purpose simulation software would be beneficial.

2.2 State-space based methods

Besides Monte Carlo simulation, there are numerical methods which determine the flow of probability between the discrete states a simulation model can be in. These methods are independent of random numbers and therefore do not need to be replicated. The most important implication of this property is that the results are free of variance and deviation (thus eliminating the need of complicated variance reduction techniques). Yet the numerical methods suffer from high requirements for memory and time when the model of interest can take up many different discrete states. In addition, the analysis algorithm needs to be performed with a granularity that is inversely proportional to the fastest activities. This can result in very long computation times for stiff models where the frequencies of the events differ by many magnitudes.

For Markovian models such as Discrete-Time Markov Chains (DTMC) or Continuous-Time Markov Chains (CTMC) there exist so-called multi-level algorithms [8]. The algorithms split the model into subsets which then are analysed individually with a more appropriate granularity, thereby reducing the effect of the stiffness of the model. Unfortunately the requirement for the model to be memoryless is an unreasonable restriction for many real-life applications.

At the moment, Monte Carlo simulation enhanced by IS or splitting strategies seems to be the better choice for the analysis of stiff models. Nevertheless, the best of both worlds would be an analysis method that, one the one hand, is free of randomness and which, on the other hand, is able to maximize computational efficiency by using different granularities for events with different frequencies.

For general stochastic models an numerical solution can be made possible by the introduction of supplementary variables. First of all, non-memoryless stochastic activities can be replaced by specific Markov chain patches, so-called phase-type distributions [9]. The phase-type distributions use multiple memoryless transitions to approximate the original stochastic distribution. The parameters are gained by fitting algorithms that usually rely on optimisation techniques which minimise the discrepancy between original distribution and phase-type approximation. Phase-type distributions are available for the continuous [10] and the discrete time domain [11]. In this paper not phases but another approach for transforming non-Markovian models into memory-less representations is used which will be introduced in the following section.

3 Proxel-based simulation

3.1 Origin and Basic algorithm

Proxel-based simulation was introduced by Horton in 2002 [12] as a state space-based analysis method for general continuous-time stochastic models. It implements Cox' method of supplementary variables to translate general distributions into a memoryless representation (cf. [13]). It creates and analyzes the state space of a model on-the-fly. The proxel method performs a discretisation of simulation time and tracks all possible developments of the system behaviour in discrete time. Thereby, it traverses the reachability graph of the model and solves its underlying inhomogeneous DTMC. The method is deterministic and easy to implement because it can be thought of as an easy iteration loop that distributes the continuous flow of probability between the states for discrete steps in time. The method has shown to be applicable for the numerical simulation of Stochastic Petri Nets (SPN) [14]. It has dramatically reduced the analysis time of warranty models of automotive industry and, for this reason, was implemented in an industrial software tool for the analyses of reliability, safety, and costs [15].

Let $s \in S$ be a discrete state of the model, τ be a vector of supplementary variables which store the duration of ongoing or pre-empted activities, and t be a time point in the discretised time dimension. Then a proxel $\alpha = (p, s, \tau, t)$ is a vector consisting of a model state given by (s, τ, t) and the probability p for that state. The proxel is a computational unit and resembles the probability p of being in state s at time t and being in this state for a duration τ. The model "moves" through the time domain in discrete steps which are multiples of the time step width Δ.

In the proxel method, the probability of all potential state changes is computed deterministically by the means of the hazard rate function $h(\tau)$ of the distributions of an activity. This function computes the probability of the ending of an activity under the assumption that has not ended yet. The integration of this function yields the amount of probability that is shifted between the discrete states of the stochastic model. The method uses a first-order discretisation of the hazard rate functions. This allows us to use solutions obtained with two different values of Δ in order to extrapolate linearly to $\Delta = 0$, obtaining an even more accurate solution [12].

The assumption is made that the probability is negligible that more than one event will take place during any one time step. As a result of the proxel-based simulation, a transient solution for the state probability vector π of the model is computed for all discrete points in time. A more detailed description of the algorithm can be found in [12].

Listing 1 shows the proxel-based simulation algorithm. The listing makes use of two functions update() and succ() which determine values for the age vector τ respectively the state of new proxels. The data structure Q is implemented as an array of binary trees which is indexed by the current time step.

Algorithm 1: Proxel-based simulation

```
input: model reachability graph, initial state $s_0$, activity set $\mathbb{A}$, step size $\Delta$,
       simulation end time $t_{\max}$
create empty proxel data structure $Q$
add initial proxel $\alpha_0 = (1.0, s_0, \vec{0}, 0)$ to $Q$
while $Q$ is not empty do
    remove proxel alpha from $Q$
    add probability $\alpha.p$ to transient solution element $\pi_{\alpha.s}(\alpha.t)$
    if $\alpha.t$ is smaller than $t_{\max}$ then
        add new proxel $(\alpha.p \times (1 - \Delta \times \Sigma_t h_t(\tau))$, alpha.s, update($\alpha.\tau$, $\alpha.s$, $\emptyset$),
        $\alpha.t + \Delta)$ to $Q$
        for all activities $A \in \mathbb{A}$ that are enabled in state $\alpha.s$ do
            add new proxel$(\alpha.p \times \Delta \times h_t(\tau))$, succ($\alpha.s$, $A$), update($\alpha.\tau$, $\alpha.s$, $T$),
            $\alpha.t + \Delta)$ to $Q$
    discard proxel $\alpha$
```

The proxel method generates and tracks all possible developments of the system behaviour of the system for discrete steps over the simulation time. Rare events and the thereby reached system states are guaranteed to be considered. In the next subsection we present an approach to address the major drawback of the proxel method, that is, the exponential growth of computational effort during simulation.

3.2 Adaptive time steps

Proxel-based simulation offers the potential of the introduction of different granularities during the analysis. It can be made to use small time steps for events with high frequencies and large time steps for events with low frequencies, thereby eliminating the need to consider slow activities with the same computational effort as fast activities. In [16] a small experimental study of this idea was performed. The study showed that the use of different time steps reduces the memory and computational complexity for the analyses of models with rare events. Although the effect of stiffness can not totally be avoided it can be dramatically lessened. The use of variable time steps will lead to a slight loss of accuracy of the computed results. Since Richardson's Extrapolation method can also be applied to the thereby gained solutions, proxel-based simulation with variable time steps is able to compute results with comparable accuracy in less time. Therefore, the method is very promising for the simulation and evaluation of rare events, even in stiff models.

In the experiments that are described in [16] the different sizes of the time steps were chosen manually. Since the aim is to have a method which requires no special know-how, but in which the time steps are truly adaptive, a strategy is needed to automatically set appropriate time steps for the proxel-based simulation of a given stochastic model. One proposal for such a strategy is as follows.

1. For each individual activity A calculate the largest acceptable size Δ_A that a time step would need to have in order to deliver reasonable results for this particular activity. In this case, reasonable means that the time step is small enough to allow error elimination by applying Richardson's Extrapolation. A very loose heuristic for this individual time step is to use a size that is not exceeding the half of the mean of the distribution of the activity as it was suggested in [17]. For most applications, a step size that is at maximum a fifth of the mean has shown to be more appropriate.

2. To the best advantage are time steps for which some multiples are powers of ten as well, since thereby almost any simulation time that makes sense can be described as a multiple of the time step size. So one restricts the different Δ_A to fulfil the following property, where $f_A(\tau)$ denotes the probability densitity function for the stochastic duration of an activity A and $E(X) = \int_0^\infty \tau f_A(\tau) d\tau$ is the expected value of this random variable.

$$\Delta_A = \max\{n10^k | n \in \{1, 2, 4, 5\}, k \in \mathbb{Z}\} \leq \frac{E(X)}{10} \tag{1}$$

Example: Let $\tau \geq 0$ and let $f_A(\tau) = \lambda e^{-\lambda\tau}$. Then the duration of the activity is exponentially distributed with a mean of $\frac{1}{\lambda}$. Hence, the maximum time step for this activity would be $\Delta_A = \frac{1}{10\lambda}$. Now, let $\lambda = 1.4$. Then the maximum Δ_A that fulfils property (1) is $5 \cdot 10^{-2} = 0.05$. See [17] for more examples on how derive time step sizes for activities from the type and the parameter of the underlying probability distributions.

3. If all activities in the model have similar mean values, use the minimum of the individual time steps to perform a standard proxel-based simulation with this minimum time step $\Delta = \min\{\Delta_A\}$.

   For stochastic models with activity means (a.k.a. event rates) that differ by some magnitudes the minimum time step becomes the so-called *base step* and is just used for the fastest activity. Further on, Δ is also used for all other activities which rates are at least half as high as the rate of the fastest activity.

4. For all activities A which have rates that are less than half as high as the rate of the fastest activity, find the maximum exponent $k_A \in \mathbb{N}$ so that $\Delta'_A = 2^{k_A}\Delta \leq \Delta_A$ is still a reasonable time step for activity A and $2^{k_A} \leq \frac{t_{\max}}{2\Delta}$.

This strategy yields time steps which are good choices for proxel-based simulation since they are as small as necessary to make only small discretisation errors but which are as large as possible to minimise runtime.

4 Examples

With two example models we illustrate proxel-based simulation of rare events, how to implement different time steps, and how to choose their size. The first example is an inventory stock in which the rare event of an unfulfilled request due to an empty stock leads to high penalty fees. The second example concerns the availabilty of the engine of a commercial jet plane. Both models are given as stochastic Petri nets and with non-Markovian activities. The proxel-based simulation is compared to Monte Carlo with regard to computational efficiency and the accuracy of the results.

4.1 Inventory Stock

The inventory stock of this example has a capacity of five items. In random time intervals, one item is requested. Unless the stock is empty the request is fulfilled and the item is removed. If the number of stored items sinks to two then an order for new items is submitted which will take some time to arrive. The order refills the stock completely. While the order is awaited requests are handled as usual. The storage of the item generates costs as well as the penalty that has to be paid if a request for an item cannot be fulfilled. The penalty is very high so the stock needs to be designed in a way that this event is as rare as possible.

Figure 1 shows a Petri net representation of this simple model. The costs are accumulated in a reward variable Costs as the combination of a rate reward that models storage fees and an impulse reward for the penalty that is triggered by the firing of the transition Unable to fulfil request. In this model, the storage costs are 10 credits per item per time unit and the penalty is 10^9 credits per miss. The request intervals are randomly distributed according to a Weibull distribution $W(\alpha = 3, \beta = 10)$. Reorders take a fixed interval of 1.4 time units to arrive. event of the model is that a request for an item is filed with an empty stock so that it cannot be fulfilled. This event is rare because the state of an empty stock is reached very seldomly and is left very fast.

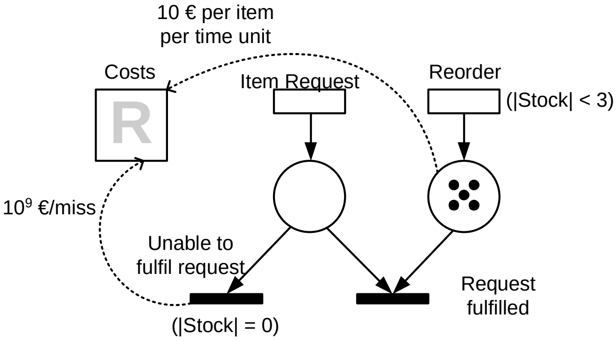

Figure 1: Petri net of Inventory Stock

For this model, due to the similar rate of the refill and request events, the heuristic of subsection 3.2 yielded the same step size for both activities. A transient solution of the probability of an empty inventory and the accumulating costs was gained with a proxel-based simulation of the model by applying Richardson's Extrapolation technique using step size $\Delta_1 = 0.1$ and $\Delta_2 = 0.05$. For the first 50 time units, the transient solution is given in Figure 1.

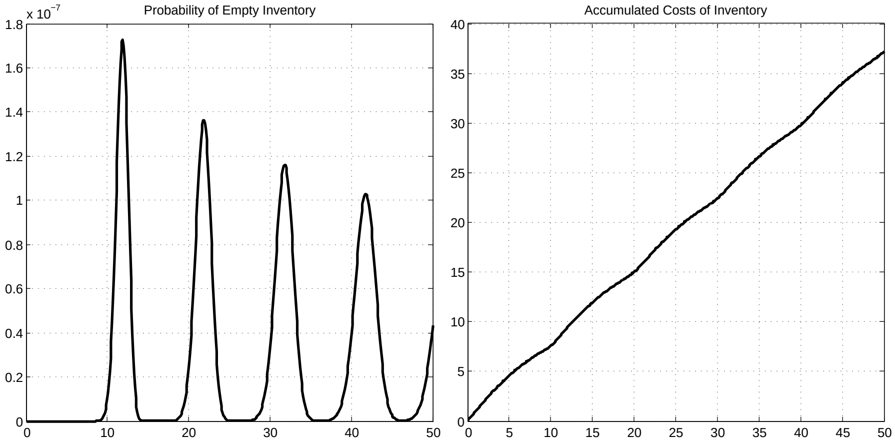

Figure 2: Probability of Empty Inventory and Accumulated Costs

On a Pentium IV computer with a clock rate of 3.07 GHz the whole computation took 14.871 seconds. A Monte Carlo simulation of this Petri net did not enter the state of an empty inventory once in 10^6 replications. A user might have drawn the conclusion that the rare event cannot happen at all, which would be wrong. The Monte Carlo simulation took about 4 hours to perform. This is a very long runtime for just 10^6 replications of such a small model. The simulation had to deal with an overhead due to of the extraction of transient values out of the simulation memory in order to gain a transient solution over time. So caution should be used when drawing conclusions from this result.

For this model proxel-based simulation was able to outperform plain Monte Carlo simulation very easily. The rare event of a request that cannot be fulfilled is considered adequately in the solution of the proxel-based simulation, while it was not in the Monte Carlo simulation. The runtime of the proxel-based was substantially lower than the runtime of Monte Carlo simulation. While this speed-up might have been achieved with Importance Sampling or Splitting, too, the yielded solution was computed deterministically and is not a random variable. Further on, there was no need to find an "change of measure" nor "thresholds" in advance as it would have been necessary with the enhanced Monte Carlo methods. This is very beneficial for optimisation of the inventory in which multiple scenarios for the storage size and costs are evaluated and compared with each other.

4.2 Aeroplane Jet Engine

The second example considers the availability of a jet engine that drives an aeroplane and the costs that are produced by failures of the engine. The model is used to derive inspection and overhaul strategies and considers both unavoidable external hazards as well as failures due to the mechanical and thermodynamic deterioration of the engine. First, the engine may fail due to wear and tear. This error happens at random number of operating hours of the engine. Second, a foreign object can damage the engine, leading to an immediate failure (e.g. bird strike). Both failures are rare events compared to the mission time of the aircraft (i.e., the flight duration) which itself is a random variable. After arrival, the aircraft stays on the ground for a fixed amount of time in which neither foreign object damage nor deterioration failure can occur.

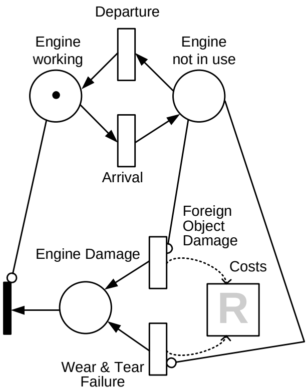

Figure 3: Petri net of Jet Engine

Figure 3 provides a Petri net representation of the jet engine model. The basic time unit of the model is one hour. The used stochastic distributions are a Normal distribution $N(\sigma = 4, \mu = 1)$ for the flight durations, Weibull distributed intervals for the a turnaround deterioration failures $W(\alpha = 1000, \beta = 3)$, and an exponential distribution $E(2 \cdot 10^{-6})$ for foreign object damage. The fixed turnaround time is 2 hours. The wearout transition Wear & Tear Failure has a race age condition, so that the operating hours are accumulated over time. A wearout failure produces costs of 10 million credits while a foreign object damage event produces costs of 20 million credits.

The heuristic of 3.2 suggested maximum time step sizes of $\Delta = 0.4$ for the transitions Arrival and Departure, $2^7\Delta$ for the wear and tear failure, and $2^9\Delta$ for the foreign object damage. A transient solution was gained with a proxel-based simulation of the model by applying Richardson's Extrapolation technique using step size $\Delta_1 = 0.4$ and $\Delta_2 = 0.2$. The runtime of this analysis was 205.614 seconds. If the same Δ had been used for all activities the runtime would have been 858.655 seconds. While a speed-up can be expected for every model, its magnitude depends on the model itself. At the moment there are no heuristics for estimating the speed-up in advance.

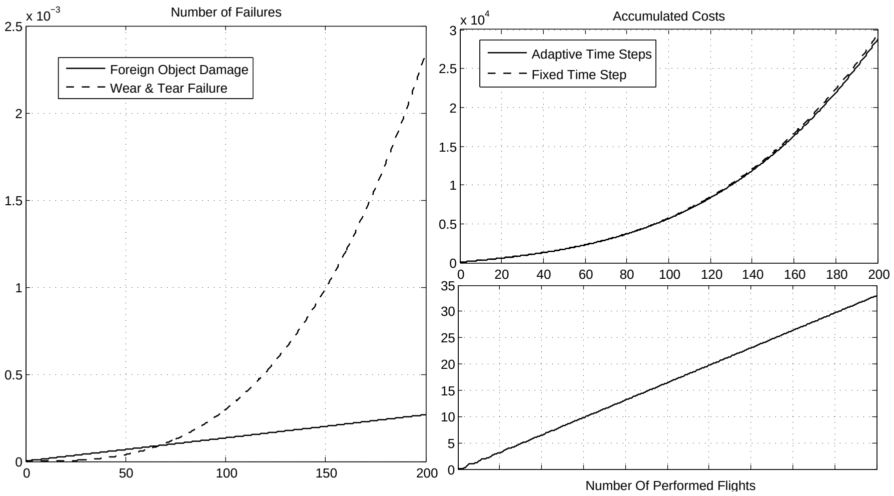

Figure 4: Number of failures and amount of costs for the jet engine

Figure 4 plots the computed number of failures, the accumulated repair costs, and the number of performed flights over time. In the cost diagram there is also the cost curve that would have been gained with a fixed time steps. As one can see, both curves differ only very slightly. This clearly justifies the use of adaptive time steps for models in which the event rates differ by magnitudes.

The solution is free from randomness since it was computed deterministically. There is no need for confidence intervals or other statistics for the sample variance. The parameters for the simulation were found based on a heuristic. A user of proxel-based simulation is no longer required to find them on his or her own so he or she only needs modelling skills. Therefore, the user needs even less know-how as with plain Monte Carlo simulation.

5 Conclusion

Proxel-based simulation is a mature alternative for the analyses of rare events in small discrete stochastic models. Its major advantages are that it is free of randomness and rare events are guaranteed to be considered, it reduces the effects of stiffness by using different discretisation parameters, its execution parameters can be found by heuristics, its execution therefore requires less know-how than ordinary discrete-event simulation since the results are not random variables but deterministic values, and it is based on a simple iteration that is easy to implement in software.

We therefore recommend the use of proxel-based simulation for the state space-based analysis of the stochastic models with rare events. Since with it models can be analysed in less time and with less memory needed, more and larger models can be feasibly analysed with state space-based simulation for the first time. Thereby, the demands on simulation users are reduced.

6 Outlook

Part of our current work is the development of a software tool chain that incorporate modelling in form of stochastic Petri nets with general firing times and proxel-based simulation of the models. Advanced modelling features will include impulse and rate rewards. This software will be used for testing further enhancements in state-space based simulation such as the combination of of discrete phase-type distributions as well as for teaching simulation and modelling. The tools will be ready for deployment in late autumn 2006.

For Monte Carlo simulation it is a rule of thumb that the rarer the events in the model, the more computation time is needed to make sure the event is represented accurately enough in the results. The opposite seems to be true for proxel-based simulation. The rarer the event, the more computation time could be saved compared to a similar model containing only non-rare events. This is due to the fact that the rarer the event, the slower the corresponding activity is and the larger the corresponding time step may be. In the future we will investigate this matter.

References

[1] Philip Heidelberger. Fast simulation of rare events in queueing and reliability models. *ACM Trans. Model. Comput. Simul.*, 5(1):43–85, 1995.

[2] S. Asmussen and R. Y. Rubinstein. Steady state rare event simulation in queueing models and its complexity properties. In *Advances in Queueing: Theory, Methods and Open Problems*, pages 429–461. CRC Press, 1995.

[3] Werner Sandmann. *Simulation seltener Ereignisse mittels Importance Sampling unter besonderer Berücksichtigung Markovscher Modelle.* PhD thesis, Rheinische Friedrich-Wilhelms-Universität, Bonn, 2004.

[4] H. Kahn and A.W. Marshall. Methods of reducing sample size in monte carlo computations. *Journal of the Operations Research Society*, 1(5):263–278, 1953.

[5] Manuel Villén-Altamirano and José Villén-Altamirano. Restart: a straightforward method for fast simulation of rare events. In *WSC '94: Proceedings of the 26th con-*

ference on Winter simulation, pages 282–289, San Diego, CA, USA, 1994. Society for Computer Simulation International.

[6] C. Kelling. A framework for rare event simulation of stochastic petri nets using "restart". In *1996 Winter Simulation Conference*, pages 317–324, Coronado, California, USA, December 1996.

[7] K. Pawlikowski, H.-D.J. Jeong, and J.-S. Ruth Lee. On credibility of simulation studies of telecommunication networks. *IEEE Communications Magazine*, 40(1):132–139, January 2002.

[8] Claudia Isensee and Graham Horton. A multi-level method for the steady state solution of discrete-time markov chains. In *Proceedings of the 2nd Balkan conference in informatics*, pages 413–420, Ohrid, Macedonia, November 2005.

[9] Marcel F. Neuts. *Matrix-Geometric Solutions in Stochastic Models: An Algorithmic Approach.* Johns Hopkins Series in the Mathematical Sciences. The John Hopkins University Press, 1981.

[10] Marita Olsson. The empht-programme. Technical report, Department of Mathematics, Chalmers University of Technology, and Göteborg University, Sweden, 1998.

[11] Claudia Isensee and Graham Horton. Approximation of discrete phase-type distributions. In *ANSS '05: Proceedings of the 38th annual Symposium on Simulation*, pages 99–106, Washington, DC, USA, 2005. IEEE Computer Society.

[12] Graham Horton. A new paradigm for the numerical simulation of stochastic petri nets with general firing times. In *Proceedings of the European Simulation Symposium 2002*, Dresden, October 2002. SCS European Publishing House.

[13] D. R. Cox. The analysis of non-markovian stochastic processes by the inclusion of supplementary variables. In *Proceedings Cambridge Philosophical Society*, volume 51(3), pages 433–441, 1955.

[14] Sanja Lazarova-Molnar and Graham Horton. Proxel-based simulation of stochastic petri nets containing immediate transitions. In *On-Site Proceedings of the Satellite Workshop of ICALP 2003 in Eindhoven, Netherlands*, Dortmund, Germany, 2003. Forschungsbericht Universität Dortmund.

[15] Fabian Wickborn, Graham Horton, Stefan Heller, and Felix Engelhard. A general-purpose proxel simulator for an industrial software tool. In *18. Symposium Simulationstechnik (ASIM 2005).* SCS European Publishing House, 2005.

[16] Fabian Wickborn and Graham Horton. Feasible state space simulation: Variable time steps for the proxel method. In *Proceedings of the 2nd Balkan conference in informatics*, pages 446–453, Ohrid, Macedonia, November 2005.

[17] Sanja Lazarova-Molnar. *The Proxel-Based Method: Formalisation, Analysis and Applications.* PhD thesis, Otto-von-Guericke-Universität, Magdeburg, 2005.