PROXEL-BASED SIMULATION

S. Lazarova-Molnar, G. Horton

Inst. Simulation und Graphik, Fakultät für Informatik, Otto-von-Guericke-Universität

Universitätsplatz 2, D-39106 Magdeburg, Germany

sanja@isg.cs.uni-magdeburg.de

Abstract: Proxels ("probability elements") were recently introduced as a new algorithmic approach to analysing discrete-state stochastic models such as are represented by stochastic Petri nets or queueing systems. Usual approaches for this type of analysis are discrete-event simulation and solving partial differential equations derived from continuous-time supplementary variables. Both of these techniques have advantages and disadvantages. The proxel-based method provides a new way to simulate stochastic models and is designed to avoid the biggest disadvantages of each of the two existing approaches.

The proxel-based method is deterministic, as opposed to discrete-event simulation, which uses a random number generator to imitate the random variables in the model. It thus avoids the problems associated with Monte-Carlo simulation such as the need for many replications in the presence of rare events. This paper focuses on the description of the analysis of stochastic models using the proxel-based method. Simulation experiments are used to demonstrate the behaviour of the method, and its advantages and disadvantages are discussed.

Keywords: proxel-based simulation, discrete stochastic models, supplementary variables, stochastic Petri nets.

1 Goals of the paper

The goal of this paper is to introduce the proxel-based method and emphasize its main characteristics. The method is a new approach to analysing stochastic models. It is intuitive and easy to understand because it traces all of the possible developments of the model in a natural way and correspondingly distributes probability. Our idea is to prove this feature of the method as well as show what the other its advantages are over the traditional approaches. We presume basic knowledge of theory of Petri nets.

2 Proxel-based Method

Proxel-based method is a numerical approach for analysing stochastic models which implements the method of supplementary variables in a way that the partial differential equations are completely avoided. This is achievable by employing the instantaneous rate function which determines the rate with which one event can happen based on the time that the event was scheduled to happen. Prior to every proxel-based simulation, the size of the time step must be determined. This should be arranged in coordination with the distribution functions, such that the time step is small enough to capture the rates of the possible state changes.

Proxel is a unit that depicts every state of the system in a complete way. This means that each proxel contains sufficient information to generate its successor proxels, or in other words to be able to determine how the model could behave from there on. The information that was found to be necessary and each proxel contains is the following:

- Discrete state of the model,
- Age intensities of the possible state changes, which state for how long each of the possible state changes has been waiting to happen; they are necessary to determine the probability that each of the state changes could happen,
- Global simulation time, which gives the absolute time point from the beginning of simulation,
- Route, which describes the sequence of states via which the model has reached the actual state, and
- Probability, that the system is in that discrete state having the concrete age intensities and having been reached via the sequence of states stated in the *route*

We also define the term *state* as the *vector composed of the discrete state of the system and the age intensities of the relevant state changes*.

As a consequence of all said above, proxels have the following format:

Proxel = (State, Time, Route, Probability), where

State = (Discrete State, Age Intensity Vector)

Since the proxel stores the necessary information to describe any situation of the model, the initial proxel that describes the initial position of the model has the following form:

Initial Proxel = (Initial State, 0, ∅, 1.0), where

Initial State = (Initial Discrete State, $\vec{0}$*)*

Once the initial proxel is generated, the simulator generates its successors based on the possible state changes and accordingly updates the age intensity vector and calculates its probabilities. In the proxels which represent different discrete state from the initial one, the appropriate age intensity is reset. The age intensities of the proxel that represents the model staying in the initial discrete state in the next time step, are incremented by the size of the time step. In this case the possible state changes are *aging* which usually increases the probability with which they might happen.

Once the second generation of proxels is computed, the first one can be entirely thrown away. The completeness of the information in the proxels approves this step. The next (third) generation is computed in the same way as the second one, except that the successor-creation process is now performed for more than one proxel. This procedure is carried out repeatedly until the predetermined end of the simulation time is reached. The structure that represents the generations of proxels as well as their relations (in terms of successors and predecessors) will be referred to as *proxel tree*. During the calculation of each generation of proxels, there may occur many proxels which represent the same *state*. In the case where such proxel is generated (not for the first time), it is not stored as new proxel, but instead the definition of the existing one is extended. Its route now is extended to the union of all the routes (until now) that lead to that state; the probability is set to the sum of the corresponding proxels' probabilities. The proxels that have null (or negligible) – probabilities are not stored. This leads to the conclusion that usually at some stage the width of the proxel tree reaches it maximum. This is so because most of the distribution functions have limited range of non-negligible values, i.e. if a state change is distributed according to a function other than exponential, usually it has a limited time within which it can happen with a non-negligible probability. The limit in the proxel-based method is determined dynamically which we believe is an essential advantage of the approach compared to the standard deterministic approaches (eg. Markov chains).

We will use a very simple example to demonstrate how the proxel-based simulation works.

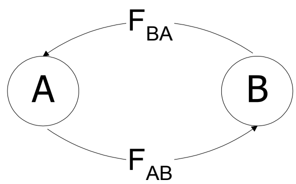

Fig. 1: Example state diagram

Let *A* be the initial discrete state of this simple model and *dt* the size of the time step. There is only one age intensity that should be memorised in each proxel. In *A* it is the age intensity of the state change distributed according to F_{AB}, and in B it is the one distributed according to F_{BA}. The initial proxel for this model is the following:

((A, 0), 0, ∅, 1.0)

In the next time step the model can either stay in the discrete state *A* or change to state *B*, in which case the age intensity is reset and now it tracks the age of the state change associated with the distribution function F_{BA}. The proxels that are generated are the following:

((A, dt), dt, ((A, 0)), 1.0- probability) and ((B, 0), dt, ((A, 0)), probability)

The probability is calculated from the instantaneous rate function ($\mu(\tau)$) with the age intensity (τ) as a parameter, multiplied by the size of the time step:

$$probability = \mu(0)*dt$$

In general instantaneous rate function can be computed as:

$$\mu(\tau) = \frac{f(\tau)}{1 - F(\tau)}$$

where f is the density function and F is the cumulative distribution function.

Now, let us observe the both generated proxels and generate their successors. From the first one, ((A, dt), dt, ∅, 1.0- *probability*), the following two can be computed:

*((A, 2dt), 2dt, ((A, 0), (A, dt)), *) and ((B, 0), 2dt, ((A, 0), (A, dt)), *).*

The second proxel from the second generation, ((B, 0), dt, ((A, 0)), *probability*), determines the following successors:

*((A, 0), 2dt, ((A, 0), (B, 0)), *) and ((B, dt), 2dt, ((A, 0), (B, 0)), *).*

The probabilities are omitted for reasons of simplicity. The are, however, computed in the same manner as previously.

It can be realised that we make one very important assumption in our approach. That is, we suppose that the model does at most one state change within time of dt. The assumption is ultimately correct when $dt \to 0$, which means that the smaller the time step, the higher accuracy can be achieved.

3 Features of the Method

The proxel-based method was recently introduced. As it is a new approach it took some time and experimenting until some of its important features became visible. The goal of this section is to summarise those features and based on them, define the class of problems for which it is useful.

3.1 No Requirement for Limitation of the State-Space

Real-life processes do not always have the feature of having limited discrete state space. Petri nets are able to represent this class of models and that is why we use them for illustration of one important feature of the proxel-based simulation, namely the ability to operate on the Petri net directly, without having to create the reachability graph. This means that there is no requirement for bounding of the state space to analyse the model. We will illustrate by an example what this attribute means. The most representative and at the same time simplest example for the above described situation is an endless queue. The Petri net for the queue is shown in Fig. 2.

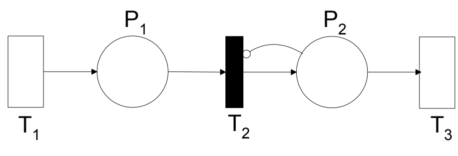

Fig. 2: Petri net for an unlimited queue with one server

Instead of building the reachability graph, the proxel-based method would operate directly on the Petri net, i.e. creating for each marking the set of reachable markings on the fly.

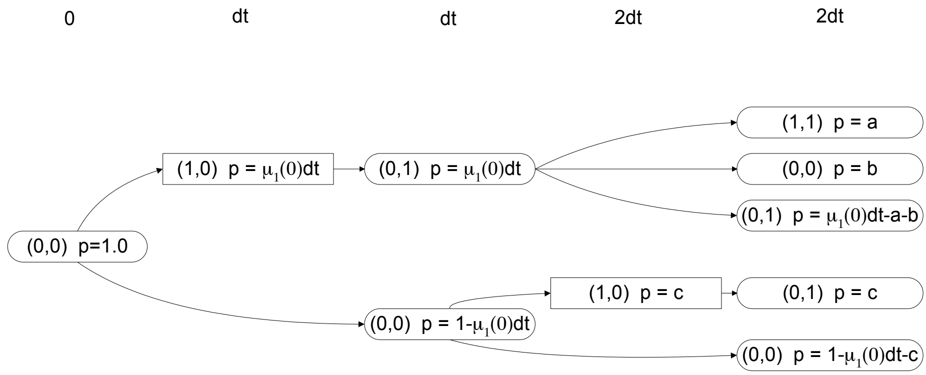

Fig. 3: Proxel-based illustration of the initial part of the state space

This means that the reachability graph is dynamically created as the proxel-based simulation advances in time. This feature makes one important (and problematic) class of models to become analysable in a deterministic manner, i.e. the class of unbounded Petri nets. In Fig. 3, the initial levels of the corresponding proxel tree are shown, the age intensity variable is omitted for simplicity of the figure. The rectangle-marked markings represent vanishing markings.

3.2 Supplementary Variables not Limited to Age Intensities

As we already mentioned, *state* in proxel-based terminology is the vector made up of the discrete state of the model (in Petri nets' terminology - marking) and the age intensities of the relevant transitions for that marking. Relevant transitions are all concurrently enabled transitions which have enabling memory policy, as well as the age memory policy transitions whose age intensities need to be remembered. Those additional components to the state of the model are so-called *supplementary variables*. The advantage of the proxel-based method is that the supplementary variable can also be of discrete type and can be used for tracking any quantity relevant for the model analysis. Those additional variables can also be parameters of the instantaneous rate function which determine the probability of a state change happening. Typical example for this situation is a model of a machine failure where the failure is also a function of the number of times that the machine has failed previously. As it was already explained in the previous subsection, the number of failures does not need to be limited.

In the following we would like to illustrate this feature by an example. The example model is the following: a machine which fails where the probability of failing is also dependant on the number of times that it has failed before, besides the age intensity of the transition that models the failure. The Petri net for this example is shown in Fig. 4.

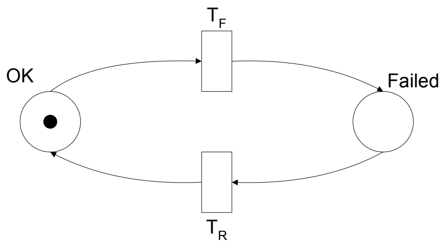

Fig. 4: Petri net representation of the machine failure model

For this model we could also add a place in the Petri net that would count the number of failures. A typical deterministic approach would imply bounds on both, the number of failures and the age intensities. The proxel-based method is free of those limitations and the bounds are computed dynamically based on the probabilities for the state changes happening. When the probability for a state change to happen becomes negligible, i.e. bellow a certain threshold (in the ideal case equal to zero), that means that the dynamic bound of that parameter has been reached. A proxel *Px* for this example would have the following format:

Px = (Marking, Age Intensity, #Failures, Time, Route, Probability).

3.3 Smooth and Transient Solutions, Extrapolation Possibility

The solutions that the proxel-based method provides are complete in discretised time. They are transient and smooth for all discrete states of the model. After some experiments were performed, it was realised that the solution values obtained using different sizes of the time step were converging in a regular fashion towards the *true* solution. This fact was employed and using polynomial extrapolation (eg. Lagrange's) of more solution points (obtained with different time steps) a better approximations of the solutions were provided in less computation time. This, until now, gave very good results in the practical application of the proxel-based method.

In order to show this we would use the model in Fig. 4, where $T_F \sim$ Exponential(0.001) and $T_R \sim$ Uniform(10,30). Lets observe the solution value at time T = 25 for the discrete state OK for different sizes of *dt*. In Fig. 5 the monotonous convergence the solution values can be noticed (in this case even linear). By performing polynomial extrapolation, a more accurate solution can be provided which is closer to the true solution value. This is shown in Fig. 5, where the y-coordinate of the point where the extrapolated line cuts the y-axis is the solution value when $dt \rightarrow 0$. This is a very important feature of the method which suggests that it is not necessary to perform simulations with tiny time steps, which are very expensive both in computation time and memory, in order to get more accurate results. The accuracy can be increased by increasing the number of solution values for different time steps.

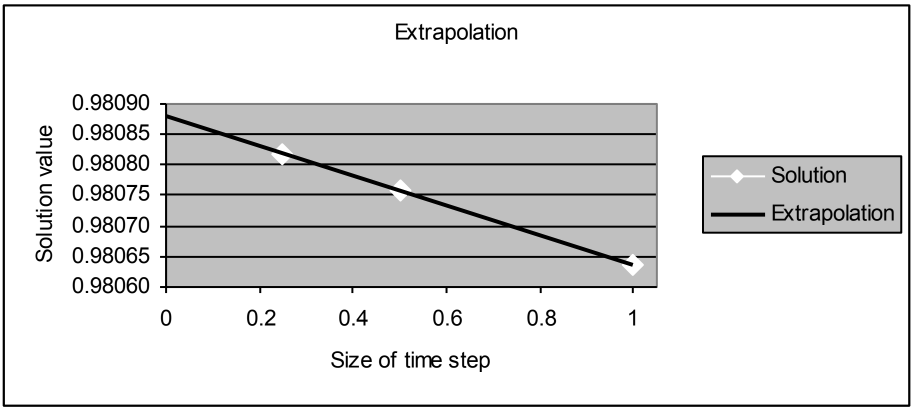

Fig. 5: Extrapolation of the solution values obtained with different sizes of the time step

Controlling the accuracy and automating the choice of the size of the time step is the direction in which our future research plans are progressing.

4 Summary and Outlook

The proxel-based method is a very young approach for analysing stochastic models. Therefore the number of open issues and further research ideas is large.

Its usefulness was recently proven on a model for warranty analysis which is used by DaimlerChrysler corporation. This was one of the first real applications of the method. For this model, the discrete-event simulation which was used until now resulted in computation times of magnitude of 20-30 hours. The proxel-based method on the opposite was computing results with comparable accuracy within seconds. This experiments gave “green light” to further applications of the method in cooperation with DaimlerChrysler.

Another class of problems for which the proxel-based method was shown to be very efficient are the fault trees. We built a tool that performs proxel-based analysis of fault trees and the results were very encouraging. There, the basic events are represented as small stochastic models and the analysis results in complete and transient solutions.

From all of the above said, it can be realised that there are already classes of problems for which the proxel-based method was much more efficient than the traditional approaches. However, one of the biggest advantages of the approach is that until now we have not found restrictions in its applicability. This means that a big number of classes of stochastic models can be analysed. The efficiency is still an issue for some of them and is a subject of future research. However, the implementation of a generalized proxel-based solver is already in progress.

5 References

1. German, R., Performance Analysis of Communication Systems. Modeling with Non-Markovian Stochastic Petri Nets, John Wiley & Sons, Ltd, 2000.
2. Horton, G., “A Splitting Method for Stiff Markov Chains”,17. Symposium Simulationstechnik (ASIM 2003), SCS European Publishing House 2003.
3. Horton, G., “A New Paradigm for the Numerical Simulation of Stochastic Petri Nets with General Firing Times”, European Simulation Symposium, Dresden, October 2002. SCS European Publishing House, 2002.
4. Lazarova-Molnar, S. and Horton G., “Proxel-Based Simulation for Fault Tree Analysis”, 17th Symposium Simulationstechnik (ASIM 2003), SCS European Publishing House 2003.
5. Lazarova-Molnar, S. and Horton G., “Proxel-Based Simulation of Stochastic Petri Nets Containing Immediate Transitions”, On-Site Proceedings of the Satellite Workshop of ICALP 2003, Eindhoven, Netherlands. Forschungsbericht Universität Dortmund. Dortmund 2003.
6. Lazarova-Molnar, S. and Horton G., “An Experimental Study of the Behaviour of the Proxel-Based Simulation Algorithm”, Simulation und Visualisierung 2003. SCS European Publishing House 2003.
7. Lazarova-Molnar, S. and Horton G., “Proxel-based Simulation of Stochastic Petri Nets”, Simulation und Visualisierung 2004.
8. Lazarova-Molnar, S. and Horton G., “Proxel-Based Simulation for Warranty Analysis”, European Simulation Multiconference, 2004.