SEARCHING BEST PATHS TO WORST STATES

G. Florin + C. Fraize+* S. Natkin +
+ CEDRIC - Centre d'Etudes et de recherche
en Informatique CNAM
292 rue Saint-Martin 75141 Paris Cedex 03
* GEC-ALSTHOM Tour Neptune La Defense Paris

Abstract

Probabilistic validation is a new approach to deal with large state transitions systems. The user's need is to prove that, for a given period of operations, that a given assertion on the reached states is true with a sufficient level of probability.

The system to be validated is modeled by a stochastic Petri net. The analysis relies on a partial exploration of the reachability set and tries to reach as quickly as possible critical states (states in which the assertion is not verified). An exact linear program solution allows to "travel" through the graph. The main goal of this paper is to present the principles of this searching algorithm.

This method can be used in probability computations in two ways. The first one is related to acyclic graphs. A breadth or a depth first search traversal can be done without considering all the trajectories but only those leading to critical states. The second one is related to importance sampling simulation.

Keywords : safety critical systems, distributed systems, probability, simulation, stochastic Petri nets, validation.

1 Introduction

The validation of complex systems such as safety-critical or distributed systems is a fundamental goal of dependable system design. It needs a formal model (defined by a user) of the system behavior and a software tool able to prove assertions on this model.

Unfortunately exhaustive analysis of the transition systems state space is limited by the exponential growth of the model complexity. Simulation of the behavior does not give an indicator of the validation level [3].

The aim of probabilistic validation [7] is to develop a new approach based on a partial analysis of a system model. The model and the analysis method must allow to prove that assertions are verified with an acceptable probability level.

Probabilistic validation tries to use the fact that "with a very low probability in a given period" is sometimes simpler to prove than "never". For the designer or the user, a system is often sufficiently dependable if undesirable events are never observed during an operational period. An operational validation method tries to prove that "bad events" will occur with a sufficiently low probability during a given period.

Several works are related to our approach. Some authors try to relate simulation and proof [8]. Our approach is also related to approximate solution of stochastic models and in particular stochastic Petri nets [9], [1]. Principles and methods able to evaluate complex user oriented goals, called performability measures, of a system have been developed by Meyer [12]. Maxemchuk presents a computation method for protocol probabilistic validation [11].

The principles of probabilistic validation are presented in [7]. Only basic principles are recalled in the second section. The goal of this paper is to present the techniques used to limit this scanning (classified in two classes): qualitative techniques presented in section 3 and quantitative techniques defined in section 4. In the last section a particular algorithm is presented.

2 Probabilistic validation

2.1 Time dependent validation

In state transition models or even temporal logic, the time is expressed by ordering conditions between events. Operations duration and frequencies are not explicitly described.

It has been formally proved that several deterministic distributed protocols do not tolerate failures with-

TH0386-3/91/0000/0204/$01.00 © 1991 IEEE

out an explicit use of operations duration (see for instance [5]). In practice, timers and clocks are basic tools of dependable systems design. Hence models of such systems must include a real time representation.

Several authors added to state transition systems an explicit timing behavior [9]. This allows an exact validation of systems under timing constraints but the complexity of computations is often much greater than in non timed models.

Moreover numerous operation durations are not known deterministically. Hence, the timed behavior of the transition systems must be stochastically defined.

Stochastic Petri nets are a good tool for this kind of modeling. Petri nets allow to describe a system at a high level concurrency, synchronization and parallelism. The property to be verified can be expressed as an assertion on the reached markings (state assertion) or on the transitions firing sequences (trajectory assertion). The property can be expressed using temporal logic operators. The stochastic timing introduces naturally the random duration of phenomena involved in the model behavior.

2.2 Stochastic Petri nets

The concepts and the notation used in this paper are detailed in [6]. We just recall in this section some important aspects.

2.2.1 Petri nets

In this paper the underlying Petri net is a place-transition net [13]. It is denoted $R(P,T,V)$ where P is the set of places (with cardinal $|P|$), T is the set of transitions (with cardinal $|T|$), V is the set of valued arcs between places and transitions. The incidence matrices define the valuation of each arc. The backward incidence matrix (relations between places and transitions) is denoted C^-. The incidence matrix is denoted C. We denote the jth column of an incidence matrix (for instance C^-) $C^-(.,j)$.

Let M_0 be the initial marking of the Petri net. The behavior of the net is defined by:

Condition : a transition t_j is firable from a marking M_i if: $M_i \geq C^-(.,j)$.

Action : the firing of t_j from M_i leads to a new marking M_k such that $M_k = M_i + C(.,j)$.

We denote by $\mathcal{S}_k$ the set of the firable transitions in M_k $\mathcal{S}_k = \{t_j \ : \ C^-(.,j) \leq M_k\}$

Let $s = (t_{j_1}, t_{j_2}, \cdots, t_{j_n})$ a sequence of transitions fired from a marking X_0 and $X_1, X_2, \cdots, X_n$ the sequence of successive reached markings in s. The characteristic vector of a sequence s is an integer vector $\overline{s}$. The jth component of $\overline{s}$ is equal to the number of transition t_j firings in the sequence s.

According to this definition we can write the firing equation (or fundamental equation) of the net. For any marking X_n reached from X_0 : $X_n = X_0 + C.\overline{s}$.

It is important to notice that if a marking X_n is reachable from X_0, the linear system of equation in $\overline{s}$: $X_n - X_0 = C.\overline{s}$ has a positive integer solution. The reverse is not true. As the firing conditions are not taken into account in the firing equation, there are possibly integer solutions of the firing equation that do not correspond to firable sequences.

2.2.2 Markov stochastic Petri nets

A stochastic Petri net is first and foremost a timed Petri net. According to a set of hypothesis for the timed behavior, it is possible to build the trajectory space of the timed net i.e. the set Ω of the trajectories ω defined by the infinite sequence of couples $(X_n(\omega), \tau_n(\omega))$ where $X_n(\omega)$ is the *nth* marking reached from $X_0(\omega) = M_0$ and $\tau_n(\omega)$ is the date of arrival in the marking from $\tau_0(\omega) = 0$).

Informally a stochastic Petri net (SPN) is a timed Petri net with a random timed behavior. A study of the general definition of SPN can be found in [6].

In this paper we consider the class of stochastic Petri nets such that the markings at time t constitute an homogeneous Markov process with continuous time. This is the case if the probability to fire a transition t_j between t and $t + dt$ in any marking M_i is equal to $\lambda_{j,M_i}.dt + o(dt)$. The real positive number λ_{j,M_i} is called the firing rate of the transition t_j in the marking M_i.

The probability of firing a transition t_j from a marking M_i is equal to:

$$P_{j,M_i} = \frac{\lambda_{j,M_i}}{\sum_{\mathcal{S}_i} \lambda_{j,M_i}}$$

2.3 Probabilistic validation principles

For a system modeled using a stochastic Petri net, the main goal is to prove that a given property is satisfied. The property can be defined by an assertion on the marking or state space and more generally on the trajectory space.

An assertion on the state space is defined as a boolean function g on the markings of the reachability graph. For example $g(M) = (M(P1) \geq 1)$ may represent that at least one equipment is always available.

An assertion on the trajectory space expresses that a boolean functions is true on a sequence of visited states. For example, let us consider the following proposition : a system is working correctly if a given resource periodically becomes available. This property is true when in a stochastic Petri net model a given place P1 is periodically marked, that is to say $g(M) = (M(P1) \geq 1)$ is true infinitely often.

In this paper we consider assertions on the state space (not on the trajectory space). The exact validation of an assertion on the state space is to prove that a property is true for all reachable marking on all trajectory. If $X_n(P1,\omega)$ denotes the marking of the place $P1$ in the X_n marking of the trajectory ω we get for the previous assertion example:

$$\forall\omega\ ,\ \forall n\ ,\ :\ X_n(P1,\omega)\ \geq\ 1$$

We can now state one of the fundamental differences between exact and probabilistic validation. From an operational point of view it is sufficient that the property remains true over the set of non null measure trajectories.

$$Prob\{\omega\ /\ \forall n\ ,\ X_n(P1,\omega)\ \geq\ 1\}\ =\ 1$$

In probability theory the property is said to be almost sure. A property can be false and also almost sure. A system may travel through sequences of states so that a given property is not verified. For an exact validation method the property is false. But if all the trajectories built from this sequence have a null measure then the user can be confident in the system.

At last from a practical point of view it may be sufficient that the property is true with a probability greater than a given value $1-\epsilon$ because it is generally simpler to prove that an assertion is often true rather than it is always true. In this case we say that the property is epsilon sure.

$$Prob\{\omega\ /\ \forall n\ ,\ X_n(P1,\omega)\ \geq\ 1\}\ >\ 1-\epsilon$$

3 Qualitative techniques

3.1 Introduction

This section is devoted to the analysis of qualitative techniques able to allow a partial scanning of a large state graph in order to validate a state assertion. We are looking for a decision method able to direct the traversal of a large state graph, in order to reach quickly an interesting class of states.

A system which is submitted to a formal validation procedure is supposed to be correctly designed. An incorrect behavior is a very rare event. If such an event may be observed, it is generally the consequence of a complex operations sequence, which is out of the standard behavior of the system. Moreover it is often verified that a very small number of system behaviors covers most of the operational situations. This remark leads to design the validation algorithms to be efficient for models such that all state is a successful state (a state such that the assertion is true) or the state space includes a small number of failed states (a state such that the assertion is false).

Three classes of traversal strategies of the graph are considered. The two first strategies are mainly applied when the trajectory space is finite. This property is verified in the modeling of embedded non repairable systems [1].

The first method visits the reachability graph in a breadth first search traversal. For non repairable systems, we have proved that the reachability graph can be generated by layers. The evaluation of the graph and the parameters computation can be done during the generation step without storing the whole graph [1]. Nevertheless the number of layers and the number of states in each layer to be kept can be very large. So we need a method to define for each layer the subset of the most interesting states to be kept.

The second method is associated with a depth first search traversal of the graph. In this case the graph generation is driven in order to reach as quickly as possible the failed situations (steepest descent) [2].

The third method is the Monte-Carlo simulation. In many cases the probability of sequences leading to critical states is very small. It has been observed that simulation takes a large amount of computational time to obtain significant figures of critical states probabilities. In order to reduce the simulation duration we must increase the frequency of such sequences (importance sampling, variance reduction [4]). But in probabilistic validation the notion of critical event is not as clear as in reliability analysis. The structural analysis can help to choose the events the frequency of which must be increased.

For one of the preceding traversal strategies, a decision method must allow to select in each marking, among the different firable transitions, the "best" transition to fire in order to reach the failed states subset. We call decision system the set of equations the solution of which give the transitions to be fired.

3.2 Decision system

The successful states are defined by a boolean function (for example $g(M) = (M(P1) > 0) = true$) but in this study we are mainly interested in searching failed states so we assume that $\neg g$ can be written in a disjunctive form.

$$\neg g(M) = \bigvee_i \bigwedge_j (G_{ij}(M) \leq 0) \quad (1)$$

We denote by $G_{ij}(M) \leq 0$ a boolean function of the place marking and $G_i(M) \leq 0 = \bigwedge_j (G_{ij}(M) \leq 0)$ the ith minterm of the disjunctive form ($G_i(M) \leq 0$ is a set of simultaneous inequations $G_{ij}(M) \leq 0$. We study separately each of these terms. The global result is the union of the transitions subsets obtained by the separate analysis. In most cases $G_{ij}(M)$ are linear functions. This assumption is needed below to decide that a transition leads to a given subset of markings.

Let us consider a current reached marking M_k from the initial marking M_0. We denote by $\mathcal{R}_k$ any subset of $\mathcal{S}_k$ the set of firable transitions in M_k ($\mathcal{R}_k$ is associated with the transitions not already fired in a given traversal of the reachability graph). The method must choose, among the set $\mathcal{R}_k$, the subset of transitions leading to the considered subset of markings (failed states).

If a marking M such that $G_i(M) \leq 0$ is reachable from M_k, there is a characteristic vector $X \geq 0$ of a firing sequence leading from M_k to $M = M_k + C.X \geq 0$. Hence we have the following decision system for failed states:

$$\begin{cases} -C.X \leq M_k \\ G_i(M_k + CX) \leq 0 \\ X \geq 0 \\ \sum_{\mathcal{R}_k} X_j \geq 1 \end{cases} \quad (2)$$

Unfortunately the existence of a solution X for the decision system 2 does not imply that there is necessarily a firing sequence leading from M_k to M.

3.3 Trajectory qualitative elimination

If the decision system for failed states 2 is impossible, it is sure that there are no failed markings reachable from M_k,starting by a transition in $\mathcal{R}_k$. Hence the generation of these markings and their successors can be avoided. Moreover the generation can be limited to markings reachable from M_0 by sequences which characteristic vectors are less or equal to all solution of the decision system. This approach can avoid the scanning of a large number of markings.

When all equations are linear, the impossibility problem is equivalent to the search of a linear program feasible solution so the simplex method can be applied.

In the linear programming problem an objective function can be used to maximize the probability of the visited path. A good heuristic, used in importance sampling simulation, is to minimize the length of the path leading from M_0 to a failed state. When the transition rates are all in the same order of magnitude, this strategy tends to generate paths by decreasing probabilities. Hence a possible objective function is $min(\sum_j X_j)$.

The algorithm tries to build in each marking a firable sequence of transition such that its characteristic vector is equal to the optimal solution of the linear program. When a failed state is reached the sequence is obtained, the traversal strategy defines the next marking to try. If, in a marking it is impossible to fire a transition in the support of the characteristic vector, a new linear program solution must be computed. If this new linear program is impossible the generation of its consequent markings can be avoided The probability of the current trajectory is added to the probability of successful trajectories (trajectory qualitative elimination).

4 Probabilistic techniques

This section is devoted to the analysis of quantitative techniques able to direct the scanning of a large state graph.

We consider acyclic Markov stochastic Petri nets models hence all trajectories are finite and a trajectory probability is defined by the product of the probability to fire each transition in each visited marking.

4.1 Classification of visited trajectories

In the scanning of a state graph we reach markings corresponding to different situations according to the probability evaluation of the assertion.

Failure probability As soon as a failed marking is reached, not only the probability of the current trajectory but more generally the probability of all trajectories including this marking must be added to the failure probability of the assertion. According to the hypothesis there is no need to continue a further examination of the state graph.

Success probability In an acyclic Markov stochastic Petri net, all trajectories are finite. If the

assertion is still verified at the end of a trajectory, its probability must be added to the success probability.

Assume that in a given marking M_k and for a given subset of transitions $\mathcal{R}_k$, the decision sytem has no solution. If $P(M_k)$ denotes the probability of all trajectories leading to M_k then the success probability is increased by:

$$\frac{P(M_k).\sum_{j\in\mathcal{R}_k}\lambda_{j,M_k}}{\sum_{i\in\mathcal{S}_k}\lambda_{j,M_k}}$$

Probability of non classified trajectories According to a traversal strategy there are possibly some pending markings. These markings are the current end of trajectories that are not sufficiently evaluated to be classified as success or failure.

4.2 Stopping criteria

There are different possibilities to terminate the scanning before the evaluation of all trajectories.

The most interesting approach is to use a parameter defined by an user $(1-\epsilon)$ as the required level of truthfulness. At the end of the computation, we must be able to conclude that the probability for g to be true is greater than $(1-\epsilon)$.

We denote in the validation tool, $\Pi 1$ the success probability, $\Pi 2$ the failure probability, $(1-\Pi 1-\Pi 2)$ the non visited trajectories probability. The exploration is performed until either $\Pi 1 > (1-\epsilon)$ the property is ϵ sure or $\Pi 2 > \epsilon$ the property is ϵ false.

4.3 Probabilistic rules to direct the traversal of a graph

In this section we define the probabilistic rules which can be applied when the qualitative analysis does not define precisely the transition to fire in a given marking.

The first simple idea is to select among the selected transitions in a given marking the one with the maximum firing rate. As the probability to fire a transition in a given marking is equal to the product of the transition rate by the state mean sojourn time, this leads to maximize locally the path probability.

The second idea is called probabilistic elimination. At a given step of the graph traversal, there are non classified trajectory left for a future evaluation. If the probability of such a trajectory is sufficiently small we can decide to stop its evaluation. The corresponding pending marking is discarded but the probability of this trajectory must be considered as an uncertainty on the final result. The global uncertainty is a parameter δ defined by an user. This parameter avoids the omission of a too large part of the trajectory space. This approach can be very useful to reduce the complexity of the traversal.

4.4 An acyclic stochastic Petri nets probabilistic validation algorithm

4.4.1 General considerations

From the previous principles numerous algorithmic methods can be derived. A method can rely either on simulation or on analytical evaluation. In both cases the linear programming elimination method may be used. Analytical evaluation may be done computing each trajectory probability. Using a stored subgraph this evaluation may be done computing the probability to reach each marking (as in [1]). At last a depth first search or a breadth first search of the graph can be performed.

An optimal method for all possible models is out of sight. As a matter of fact a method efficiency depends on its ability to take into account several characteristics of the model: lack or existence of failed states, dynamics of firing rates, strongly or weakly synchronized systems.

The experimental analysis of different algorithms is a difficult task. Several arbitrations between all these possibilities must be performed. The corresponding algorithms must be implemented and the different solutions must be compared from several selected examples.

In the following section we present an algorithm designed to obtain an accurate evaluation for systems that either include no critical states or include a small number of failed sates.

4.4.2 Principles of a depth first search partial traversal algorithm

This method uses a depth first search traversal of the reachability graph in order to obtain an analytical solution. In such a traversal only trajectories corresponding to a given solution of the decision system are stored. It builts integer solutions of the decision system, by increasing paths length, and the corresponding sequences until either there are no more solutions or a probabilistic stopping criterion is reached.

Starting from the initial marking and the minimal integer solution $X(0)$, the reachability set of all solutions having $X(0)$ as characteristic vector. Each time a failed state is reached the probability of failure is

increased. Then all reached markings are added to a list and for each of these markings transitions already fired are forbidden. For each stored marking M_k a new solution is computed using M_k as initial marking and the remaining enabled transitions as $\mathcal{R}_k$.

If the decision system have no solution the success probability is increased. Then the marking M_k corresponding to the minimal path solution $X(n)$ is chosen as a new initial marking with $X(n)$ as characteristic vector.

This algorithm has been implemented and experimented on several test cases. The first results show that when a model has no failed states the algorithm stops immediately. Such a result can be easily explained for toy models (an example of this type is the correct version of the railroad crossing [10]). But we have also verified this property for practical examples (a simple RPC protocol and an election protocol). For models with a few number of failed states reachable by a few number of sequences the algorithm is very efficient. An assertion on an example with 3000 markings can be proved visiting 4 markings and one trajectory.

Two main points still have to be checked. The first one is the efficiency of the probabilistic stopping criteria. But the main problem is to characterize models able to be efficiently solved by the algorithm. Our experimentations are now related to the validation of a complex RPC protocol and a control system for the brakes in a train.

5 Conclusion

Techniques presented in this paper are defined to limit the analysis of a complex model. The qualitative principles rely on the fundamental equation of Petri nets. They can be applied on any Markov stochastic Petri net model to guide a graph traversal. Several probabilistic criteria can also be used to reduce the complexity of the graph traversal.

Implementation and practical experimentation on several examples are in progress.

Two important works have to be done. The use of importance sampling for probabilistic validation must be completely specified. Assertions on the firing sequences expressed using temporal logic must be introduced in the model.

References

[1] Barkaoui K., Florin G., Fraize C., Lemaire B., Natkin S. Reliability analysis of non repairable systems using stochastic Petri nets - Proc FTCS18 Tokyo June 1988

[2] Bouissou M. Automatic generation and quantification of event sequences leading to repairable system failure - 5Th congress on Reliability and Maintainability, Biarritz, France, 1988

[3] Cavalli A.R., Paul E. Exhaustive analysis and simulation for distributed systems, both sides of the same coin, Distributed computing N2 1988.

[4] Conway A.E., Goyal A. Monte carlo simulation of computer system availability reliability models - Proc. 17th IEEE International symposium on fault tolerant computing 1987

[5] Fisher M.J., Lynch N.A., Paterson M.S. Impossibility of distributed consensus with one faulty process - J. ACM 32, 1985

[6] Florin G., Fraize C., Natkin S. Stochastic Petri nets : properties applications and tools - Microelectronic and reliability, Vol 31, nu 4, pp 669-698, 1991

[7] Florin G., Fraize C., Natkin S. A new approach of formal proof: probabilistic validation - International working conference on Dependable Computing for Critical Applications, Tucson, Arizona, February 18-20, 1991

[8] Groz R. Unrestricted verification of protocol properties on a simulation using an observer approach - Bochmann G, Sarikaya B (eds) VI IFIP WG6. 1 Workshop Gray Rocks, Montreal, Canada, June 1986.

[9] Juanole G., Roux J.L. On the pertinence of the extended time Petri net model for analyzing communication activities , PNPM 89, Kyoto 1989

[10] Leveson N.G., Stolzy J.L. Safety analysis using Petri nets, IEEE TSE, vol SE13,no 3, march 1987

[11] Maxemchuck N. F., Sabnani K. Probabilistic verification of communication protocol - Distributed computing N3 1989.

[12] Meyer J.F., Sanders W.H. Performability evaluation of distributed systems using stochastic activity networks - International workshop on Petri Nets and Performance Models, Madison, Wisconsin, August 24-26, 1987

[13] Murata T. Petri nets : properties, analysis and applications - Proceedings of the IEEE, vol. 77, n. 4, 1989