TOMSPIN - A Tool for Modelling with Stochastic Petri Nets

E. M. Thurner, Ch. Wincheringer
Siemens AG, ZFE T SN 1
D - 81370 München

Petri nets have become a wide-spread method to model the behaviour of computing systems. One drawback of standard Petri nets however is the absence of constructs for the notion of time and conflict-probabilities. Generalized Stochastic Petri Nets (GSPNs) have been developed to deal with these constructs.

The program TOMSPIN (Tool for Modelling with Stochastic Petri Nets) supports the full descriptive power of GSPNs. It has been developed to model and to evaluate complex computing systems, such as multiprocessor machines and communication systems. In this paper, the structure of TOMSPIN is pointed out, and some examples for its usage in commercial environment are given.

Key words: Modelling, performance, realiability, Generalized Stochastic Petri Nets, TOMSPIN

Revision: Standard Petri nets

Petri nets were introduced by C.A. Petri in 1962 (see [Petri]). Petri nets have turned out to be an intuitive approach to represent the structure of computing systems as well as their dynamic behaviour. What is even more important: Petri nets are well defined in a strict mathematical sense. Due to this, they give an exact representation of notions like
- causality,
- concurrency, and
- synchronisation.

Typical examples for the usage of standard Petri nets are:
- proving the liveness of systems, resp. to find deadlocks in them,
- proving their boundedness (safety).

These system properties can be found out by analytical evaluation. This means, that statements about the system's behaviour have no restrictions that are typical for simulation-based evaluation, i.e. the modeler needs not to care for the completeness of the "simulation" or to find out confidence intervals, because this completeness is inherently given by the analytical approach. An introduction to semantics and usage of Petri nets is given in [Reisig].

Let us consider one of the evaluation methods, namely the construction of the so-called reachability graph. The states of the reachability graph (the markings) represent the complete system state. Fig. 1 gives a very simple Petri net, in which the token is wandering cyclically, i.e. starting from the place p1 through the transition t1 to the next place p2 and so on. On the right side of fig. 1, the respective reachability graph is shown: In the so-called initial marking M0 there is only one token in p1, in the next marking M1 there is one token in p2 etc.
Generalized Stochastic Petri Nets

Unfortunately, standard Petri nets are not able to represent time within a system. To deal with this problem, *Generalized Stochastic Petri Nets (GSPNs)* were introduced by [Marsan]. In this class of nets, there are two kinds of transitions:

- **Immediate transitions**, which fire immediately after their activation.
- **Timed transitions**, which fire at an exponentially distributed random time after their activation.

While immediate transitions use the same firing rule as standard Petri nets do, timed transitions are a key to represent time in a system: To each timed transition a delay time $1/\lambda$ is assigned; The value $\lambda$ is called the *firing rate* of this transition.

Furthermore, in GSPNs *conflict probabilities* can be assigned to every transition that participates in the conflict. At last, marking-depending functions, e.g. the arc-weight or the firing rate as a function of the number of tokens in a place can be expressed.

An example for a simple GSPN and its reachability graph is given in fig. 2: The immediate transitions are depicted by thin bars, and timed transitions are depicted by thick bars. The respective transitions in the reachability graph are timed or immediate. Due to the notion of time, GSPNs have the most of the evaluation capabilities of standard Petri nets and can be used in addition to get performance measures of a computing system. The evaluation process for GSPNs consists of the following steps:

---

*Fig. 1: A simple Petri net and its reachability graph*

*Fig. 2: A simple GSPN and its reachability graph*