

AUTHENTICATED PETRI NETS

Motivation

Petri net based technologies are gaining increasing attention, especially in application fields with high dependability demands, like security- and/or safety-oriented applications. Sophisticated analysis techniques, implemented in reliable tools, allow the computer-based verification of technical systems, provided their essential behavior has been modeled and the required properties have been specified, both very carefully.

To get confidence in the model, many hours are spent for its validation and verification. The run times of the related analysis algorithms may easily surmount several days. Additionally, the automatically produced analysis protocols have to be checked by "natural intelligence". The general outcome of such a usually lengthy procedure is a certification stamp, i.e. the statement that the model satisfies the given list of specified requirements.

This might be true at the moment of that declaration. Later on, during all further applications of the model for "what so ever", it is tacitly assumed that the "older" statement is still valid. But actually, taking the current practice, there is no real reason to trust in long-term confidence of the certification stamp a model once got. Over its lifetime, each model encounters many occasions where it might be changed.

There are basically two types of possible destructions a certified model should be prepared for:

- mistakenly done destruction:
There is a lengthy list of possible sources for unintended changes, which can be classified into hardware faults due to wear out or external influences, tool bugs (i.e. software faults), and user faults. Independent of the actual source, causing the fault, the impact on the model behavior may be disastrous or just misleading.
- destruction on purpose:
Models, controlling sensible equipment - like reactive or embedded systems, may be target of criminal attacks, where an intruder tries to inject faulty behavior on purpose, hoping for internal or external serious damages.

Petri net models tend to become large in size, even for smaller examples. Therefore, it is common practice to organize them in an hierarchical way, resulting into net structures which are distributed over numerous separate pages. But even in a smaller flat model, minor changes are unlikely to be found by chance or on the very first glance.

So, do we have to repeat occasionally the validation&verification procedure to gain long-term confidence? Or are there other methods to trust a model's certification stamp for ever?