Frédéric Cristini and Silvano Dal Zilio
dalzilio@laas.fr

This form is a summary description of the model entitled "SatelliteMemory" proposed for the Model Checking Contest @ Petri Nets. Models can be given in several instances parameterized by scaling parameters. Colored nets can be accompanied by one or many equivalent, unfolded P/T nets. Models are given together with property files (possibly, one per model instance) giving a set of properties to be checked on the model.

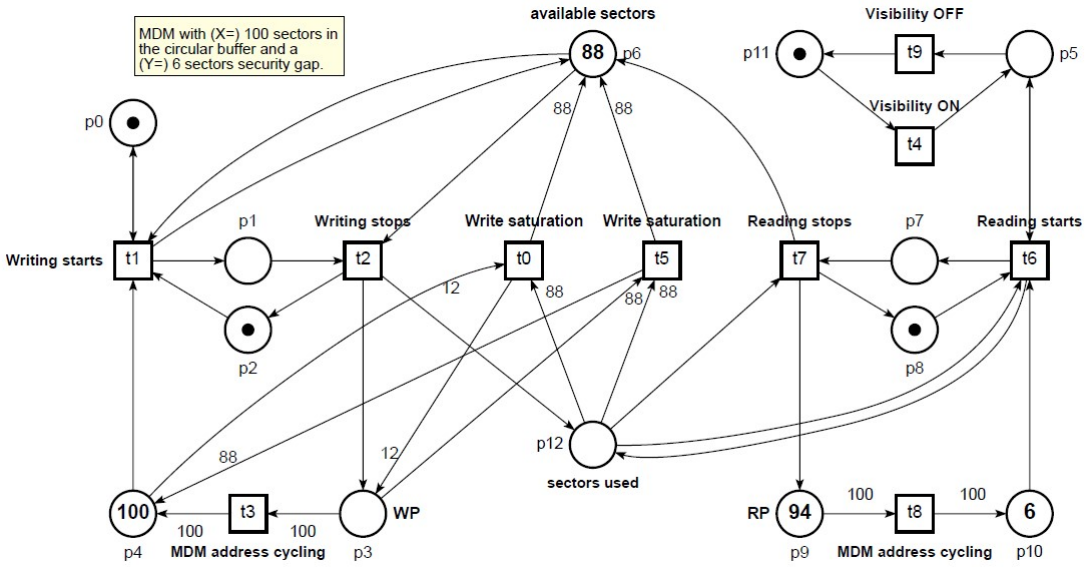
Description

This Petri net models the stylized behaviour of the mass memory management system in a micro-satellite from the Myriade product line, designed by CNES, the French Space Agency. In these satellites, the main memory device can be viewed as a circular buffer accessed through two pointers: a Writing Pointer (whose value is modelled by the marking of place `p3`), and a Reading Pointer (corresponding to place `p9`). In a real life system, the memory device typically has 16 GB of FIFO memory, divided into 65 535 separate "sectors". In our case, the capacity of the memory device is given by the value of parameter X.

We consider that the satellite can write into its memory at all time. On the opposite, we can only read (and delete value) from memory when a download station is visible. To take into account the possible loss of memory sectors during the mission duration, and to prevent possibly erroneous, a controller ensures that there is always a "security gap" (or buffer) between the sectors pointed by RP and WP. The size of this buffer is given by parameter Y. Hence the system ensure the invariant $(\mathrm{WP} - \mathrm{RP}) \geq Y$.

Finally, the system can write data even when the system is full (the write pointer is ahead of $X - 2.Y$ sectors from the read pointer). In this case, called a *write saturation* event, the writing pointer can overtake the reading pointer, still ensuring the security margin.

This model is interesting from a verification point of view. Many properties can be inferred from the study of its invariants. It also provides a nice example of models where an explicit, enumeration-based approach may be more efficient than one based on the use of decision diagrams.



Graphical representation for $X = 100$ and $Y = 6$

Frédéric Cristini and Silvano Dal Zilio
dalzilio@laas.fr

References

This model was initially described in: S. Sho, F. Cristini, J.-C. Damery. *Formal verification of the TARANIS mass memory management system using Petri nets.* Master of Science in Aeronautical and Space Systems, ISAE, 2017.

Scaling parameter

Parameter name	Parameter description	Chosen parameter values
(X, Y)	X is the number of sectors in the Mass Memory Device and Y is the number of reserved sectors (with the constraint that $X > 2 \times Y$).	$(100, 3)$, $(1000, 32)$, $(1\,500, 46)$, $(3\,000, 94)$, $(65\,535, 2\,048)$

Size of the model

Although the model is parameterized, its size does not depend on parameter values.

number of places:	13
number of transitions:	10
number of arcs:	40

Structural properties

ordinary — *all arcs have multiplicity one* ..... ✗ [a]
simple free choice — *all transitions sharing a common input place have no other input place* ..... ✗ [b]
extended free choice — *all transitions sharing a common input place have the same input places* ..... ✗ [c]
state machine — *every transition has exactly one input place and exactly one output place* ..... ✗ [d]
marked graph — *every place has exactly one input transition and exactly one output transition* ..... ✗ [e]
connected — *there is an undirected path between every two nodes (places or transitions)* ..... ✓ [f]
strongly connected — *there is a directed path between every two nodes (places or transitions)* ..... ✓ [g]
source place(s) — *one or more places have no input transitions* ..... ✗ [h]
sink place(s) — *one or more places have no output transitions* ..... ✗ [i]
source transition(s) — *one or more transitions have no input places* ..... ✗ [j]
sink transitions(s) — *one or more transitions have no output places* ..... ✗ [k]
loop-free — *no transition has an input place that is also an output place* ..... ✗ [l]
conservative — *for each transition, the number of input arcs equals the number of output arcs* ..... ✗ [m]
subconservative — *for each transition, the number of input arcs equals or exceeds the number of output arcs* ..... ✗ [n]
nested units — *places are structured into hierarchically nested sequential units* [o] ..... ✗

[a] checked by the INA tool version 2.2 on April 2020; confirmed by CÆSAR.BDD version 3.4 on all 5 instances (see the aforementioned list).
[b] the net is not ordinary; checked by the INA tool version 2.2 on April 2020.
[c] the net is not ordinary.
[d] the net is not ordinary; checked by the INA tool version 2.2 on April 2020.
[e] the net is not ordinary.
[f] checked by the INA tool version 2.2 on April 2020; confirmed by CÆSAR.BDD version 3.4 on all 5 instances (see the aforementioned list).
[g] checked by the INA tool version 2.2 on April 2020; confirmed by CÆSAR.BDD version 3.4 on all 5 instances (see the aforementioned list).
[h] checked by the INA tool version 2.2 on April 2020; stated by CÆSAR.BDD version 3.4 on all 5 instances (see the aforementioned list).
[i] checked by the INA tool version 2.2 on April 2020; stated by CÆSAR.BDD version 3.4 on all 5 instances (see the aforementioned list).
[j] checked by the INA tool version 2.2 on April 2020; stated by CÆSAR.BDD version 3.4 on all 5 instances (see the aforementioned list).
[k] checked by the INA tool version 2.2 on April 2020; stated by CÆSAR.BDD version 3.4 on all 5 instances (see the aforementioned list).
[l] 2 transitions are not loop free, e.g., transition "t1".
[m] transition `t1` is not conservative; stated by PNML2NUPN 3.2.0 on all 5 instances (see the aforementioned list); confirmed by the INA tool version 2.2 on April 2020.
[n] transition `t2` is not subconservative; stated by PNML2NUPN 3.2.0 on all 5 instances (see the aforementioned list); confirmed by the INA tool version 2.2 on April 2020.
[o] the definition of Nested-Unit Petri Nets (NUPN) is available from `http://mcc.lip6.fr/nupn.php`

Frédéric Cristini and Silvano Dal Zilio
dalzilio@laas.fr

Behavioural properties

safe — *in every reachable marking, there is no more than one token on a place* ..... ✗ (p)
dead place(s) — *one or more places have no token in any reachable marking* ..... ?
dead transition(s) — *one or more transitions cannot fire from any reachable marking* ..... ✔ (q)
deadlock — *there exists a reachable marking from which no transition can be fired* ..... ✗ (r)
reversible — *from every reachable marking, there is a transition path going back to the initial marking* ..... ✔ (s)
live — *for every transition t, from every reachable marking, one can reach a marking in which t can fire* ..... ✔ (t)

Size of the marking graphs

Parameter	Number of reachable markings	Number of transition firings	Max. number of tokens per place	Max. number of tokens per marking
$(X = 100, Y = 3)$	76 358 (u)	209 484 (v)	100 (w)	298 (x)
$(X = 1000, Y = 32)$	7 499 494 (y)	20 618 550 (z)	1 000	2 940
$(X = 1500, Y = 46)$	16 913 270 (aa)	46 503 906 (ab)	1 500	4 412
$(X = 3000, Y = 94)$	67 522 502 (ac)	185 671 698 (ad)	3 000	8 816
$(X = 65\,535, Y = 2\,048)$	?	?	65 535	192 513

Other properties

The most important property enforced by the memory controller is the "safe distance" between RP and WP. This can be expressed as invariants (here in CTL), such as AG $((\texttt{p3} > \texttt{p9}) \Rightarrow (\texttt{p3} - \texttt{p9} \geq Y))$, or a simpler, less general version (since we always have $Y \geq 1$): AG $((\texttt{p3} \geq \texttt{p9} + 1) \vee (\texttt{p9} \geq \texttt{p9} + 1))$.

(p) by construction of the model: the initial marking is not safe; checked by the INA tool version 2.2 on April 2020; confirmed by CÆSAR.BDD version 3.4 on all 5 instances (see the aforementioned list).
(q) checked with TINA version 3.5.0 on April 2020 on the first two instances.
(r) by construction of the model: there is a live cycle between places p5 and p11.
(s) confirmed by TINA version 3.5.0 on the first three instances of the problem.
(t) checked with TINA version 3.5.0 on April 2020, on the first two instances.
(u) computed by TINA version 3.5.0 on April 2020.
(v) computed by TINA version 3.5.0 on April 2020.
(w) based on invariants, the maximal number of tokens is less than parameter X, which is also the initial marking of place p4.
(x) based on invariants, the total number of tokens in a marking is a constant, equal to $3.X - 2.Y + 4$; confirmed by TINA version 3.5.0 on April 2020 on the first four instances.
(y) computed by TINA version 3.5.0 on April 2020.
(z) computed by TINA version 3.5.0 on April 2020.
(aa) computed by TINA version 3.5.0 on April 2020.
(ab) computed by TINA version 3.5.0 on April 2020.
(ac) computed by TINA version 3.5.0 on April 2020.
(ad) computed by TINA version 3.5.0 on April 2020.