

This form is a summary description of the model entitled “BusinessProcesses” 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.

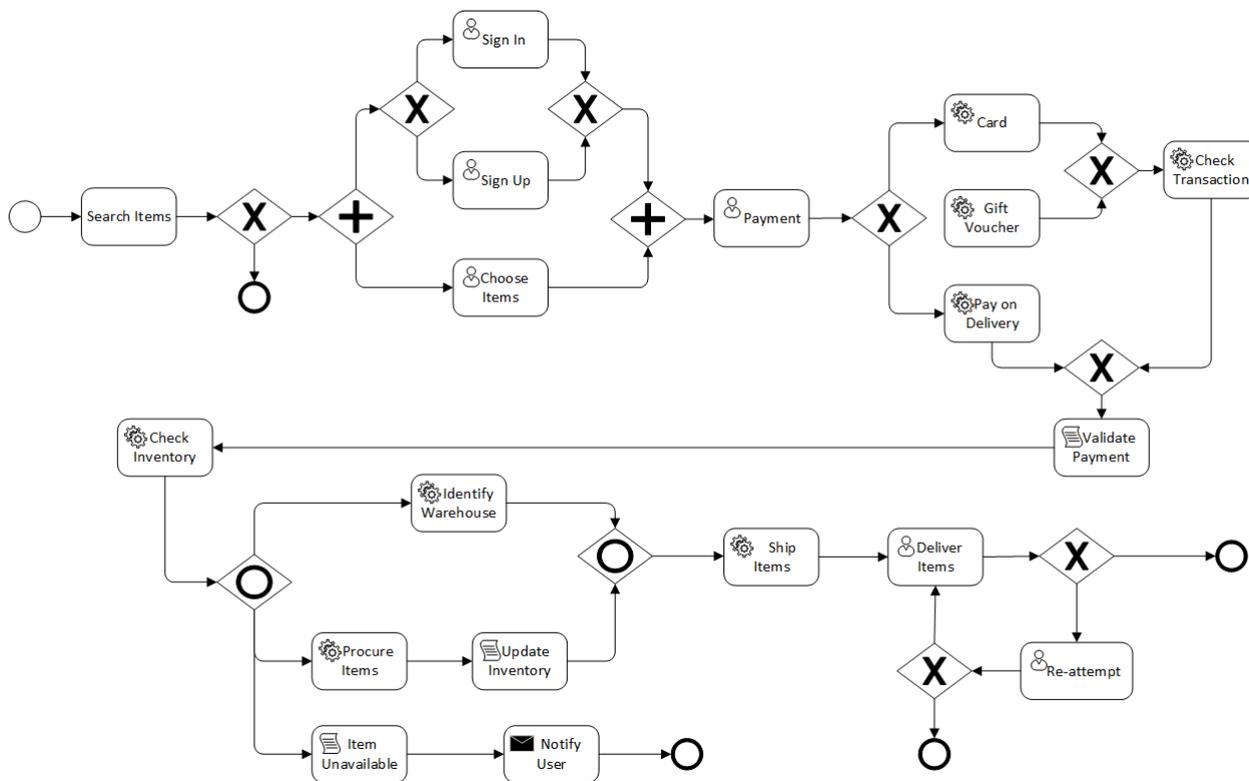
Description

A business process consists in a set of tasks and activities to perform a certain business goal. The Business Process Model and Notation (BPMN) is a widely adopted standard with good tooling support for modelling business processes. Broadly, a process model consists of an initial event, end events, gateways, tasks, and flows. Convergence and divergence of flows are defined using different kinds of gateways.

The NUPN models presented here were obtained as part of our efforts to build a formal verification framework for BPMN [1]. We developed a collection of business processes (specified in BPMN), which represent specific activities carried out by various organizations (e.g., required steps for opening a bank account or publishing a book).

Each of these BPMN models was then translated to the LNT language, a modern successor of LOTOS, using our approach for an automatic model-to-model transformation of BPMN models to LNT with preservation of the execution semantics [2]. Each LNT specification was translated to LOTOS, and then to an interpreted Petri net using the CADP toolbox. From each LOTOS specification, a P/T net was obtained by stripping out all data-related information (variables, types, assignments, guards, etc.) from the interpreted Petri net, leading to a NUPN (Nested-Unit Petri Net) model translated to PNML using the CÆSAR.BDD tool.

We only kept the NUPNs generated before applying all the structural and data-flow optimizations of the CÆSAR compiler for LOTOS, because such optimizations appear to be quite effective for business processes and lead to NUPNs that are often too small for the Model Checking Contest. In the collection of NUPNs generated this way, we carefully selected 20 models of high complexity, which we sorted by the increasing number of places.



Sample business process for online shopping

References

- [1] Ajay Krishna, Pascal Poizat, and Gwen Salaün. *VBPMN: Automated Verification of BPMN Processes*. Proc. of the 13th International Conference on Integrated Formal Methods (iFM 2017), Torino, Italy, LNCS 10510, Springer, September 2017.
- [2] Pascal Poizat, Gwen Salaün, and Ajay Krishna. *Checking Business Process Evolution*. Proc. of the International Conference on Formal Aspects of Component Software (FACS 2016), Besançon, France, LNCS 10231, Springer, October 2016.

Scaling parameter

| Parameter name | Parameter description | Chosen parameter values |
|----------------|------------------------------|-------------------------|
| N | N is the number of threads | from 1 to 20 |

Size of the model

| Parameter | Number of places | Number of transitions | Number of arcs | Number of units | HWB code |
|-----------|------------------|-----------------------|----------------|-----------------|-----------|
| $N = 1$ | 200 | 178 | 487 | 43 | 12-22-87 |
| $N = 2$ | 262 | 219 | 576 | 63 | 18-32-127 |
| $N = 3$ | 274 | 237 | 632 | 61 | 18-31-128 |
| $N = 4$ | 288 | 283 | 916 | 65 | 13-40-125 |
| $N = 5$ | 368 | 319 | 854 | 81 | 24-41-168 |
| $N = 6$ | 376 | 297 | 830 | 111 | 22-56-199 |
| $N = 7$ | 386 | 360 | 1110 | 91 | 19-53-175 |
| $N = 8$ | 393 | 344 | 911 | 87 | 26-44-185 |
| $N = 9$ | 403 | 374 | 1148 | 95 | 20-55-183 |
| $N = 10$ | 518 | 471 | 1415 | 123 | 27-69-238 |
| $N = 11$ | 567 | 508 | 1512 | 137 | 30-76-265 |
| $N = 12$ | 586 | 525 | 1560 | 141 | 31-78-272 |
| $N = 13$ | 593 | 536 | 1582 | 139 | 32-77-271 |
| $N = 14$ | 624 | 564 | 1657 | 145 | 34-80-284 |
| $N = 15$ | 630 | 566 | 1666 | 147 | 34-81-287 |
| $N = 16$ | 638 | 578 | 1694 | 147 | 35-81-290 |
| $N = 17$ | 650 | 580 | 1709 | 155 | 35-85-300 |
| $N = 18$ | 717 | 642 | 1860 | 165 | 40-89-329 |
| $N = 19$ | 772 | 685 | 1981 | 181 | 43-98-357 |
| $N = 20$ | 782 | 697 | 2011 | 181 | 43-98-358 |

Structural properties

- ordinary** — all arcs have multiplicity one yes
- simple free choice** — all transitions sharing a common input place have no other input place no ^(a)
- extended free choice** — all transitions sharing a common input place have the same input places no ^(b)
- state machine** — every transition has exactly one input place and exactly one output place no ^(c)
- marked graph** — every place has exactly one input transition and exactly one output transition no ^(d)
- connected** — there is an undirected path between every two nodes (places or transitions) ? ^(e)
- strongly connected** — there is a directed path between every two nodes (places or transitions) no ^(f)
- source place(s)** — one or more places have no input transitions yes ^(g)

^(a) stated by [CÆSAR.BDD](#) version 3.3 on all 20 instances (20 values of N).

^(b) stated by [CÆSAR.BDD](#) version 3.3 on all 20 instances (20 values of N).

^(c) stated by [CÆSAR.BDD](#) version 3.3 on all 20 instances (20 values of N).

^(d) stated by [CÆSAR.BDD](#) version 3.3 on all 20 instances (20 values of N).

^(e) stated by [CÆSAR.BDD](#) version 3.3 to be true on 19 instance(s) out of 20, and false on the remaining 1 instance(s).

^(f) stated by [CÆSAR.BDD](#) version 3.3 on all 20 instances (20 values of N).

^(g) stated by [CÆSAR.BDD](#) version 3.3 on all 20 instances (20 values of N).

- sink place(s)** — one or more places have no output transitions yes ^(h)
- source transition(s)** — one or more transitions have no input places no ⁽ⁱ⁾
- sink transitions(s)** — one or more transitions have no output places no ^(j)
- loop-free** — no transition has an input place that is also an output place yes ^(k)
- conservative** — for each transition, the number of input arcs equals the number of output arcs no ^(l)
- subconservative** — for each transition, the number of input arcs equals or exceeds the number of output arcs no ^(m)
- nested units** — places are structured into hierarchically nested sequential units ⁽ⁿ⁾ yes

Behavioural properties

- safe** — in every reachable marking, there is no more than one token on a place yes ^(o)
- dead place(s)** — one or more places have no token in any reachable marking ? ^(p)
- 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 |
|-----------|------------------------------------|------------------------------|---------------------------------|-----------------------------------|
| $N = 1$ | $2.4161e+10$ ^(u) | ? | 1 | $\in [20, 22]$ ^(v) |
| $N = 2$ | $3.14315e+12$ ^(w) | ? | 1 | $\in [30, 32]$ ^(x) |
| $N = 3$ | $1.4794e+13$ ^(y) | ? | 1 | 31 |
| $N = 4$ | $\geq 9.97774e+10$ ^(z) | ? | 1 ^(aa) | $\in [21, 40]$ ^(ab) |
| $N = 5$ | $2.38391e+17$ ^(ac) | ? | 1 | 41 |
| $N = 6$ | $\geq 1.99438e+15$ ^(ad) | ? | 1 ^(ae) | $\in [42, 56]$ ^(af) |
| $N = 7$ | $\geq 3.18893e+14$ ^(ag) | ? | 1 ^(ah) | $\in [34, 53]$ ^(ai) |
| $N = 8$ | $4.94e+17$ ^(aj) | ? | 1 | 44 |
| $N = 9$ | $\geq 5.84345e+14$ ^(ak) | ? | 1 ^(al) | $\in [36, 55]$ ^(am) |
| $N = 10$ | $\geq 1.7007e+18$ ^(an) | ? | 1 ^(ao) | $\in [48, 69]$ ^(ap) |
| $N = 11$ | $\geq 2.87495e+22$ ^(aq) | ? | 1 ^(ar) | $\in [55, 76]$ ^(as) |
| $N = 12$ | $\geq 1.74721e+21$ ^(at) | ? | 1 ^(au) | $\in [57, 78]$ ^(av) |
| $N = 13$ | $\geq 8.64912e+21$ ^(aw) | ? | 1 ^(ax) | $\in [58, 77]$ ^(ay) |
| $N = 14$ | $\geq 2.90171e+22$ ^(az) | ? | 1 ^(ba) | $\in [61, 80]$ ^(bb) |
| $N = 15$ | $\geq 3.42608e+22$ ^(bc) | ? | 1 ^(bd) | $\in [62, 81]$ ^(be) |
| $N = 16$ | $\geq 6.16854e+22$ ^(bf) | ? | 1 ^(bg) | $\in [62, 81]$ ^(bh) |
| $N = 17$ | $\geq 1.46288e+23$ ^(bi) | ? | 1 ^(bj) | $\in [64, 85]$ ^(bk) |
| $N = 18$ | $\geq 4.13695e+23$ ^(bl) | ? | 1 ^(bm) | $\in [73, 89]$ ^(bn) |
| $N = 19$ | $\geq 3.4029e+25$ ^(bo) | ? | 1 ^(bp) | $\in [79, 98]$ ^(bq) |
| $N = 20$ | $\geq 5.5722e+24$ ^(br) | ? | 1 ^(bs) | $\in [79, 98]$ ^(bt) |

^(h) stated by [CÆSAR.BDD](#) version 3.3 on all 20 instances (20 values of N).

⁽ⁱ⁾ stated by [CÆSAR.BDD](#) version 3.3 on all 20 instances (20 values of N).

^(j) stated by [CÆSAR.BDD](#) version 3.3 on all 20 instances (20 values of N).

^(k) stated by [CÆSAR.BDD](#) version 3.3 on all 20 instances (20 values of N).

^(l) stated by [CÆSAR.BDD](#) version 3.3 on all 20 instances (20 values of N).

^(m) stated by [CÆSAR.BDD](#) version 3.3 on all 20 instances (20 values of N).

⁽ⁿ⁾ the definition of Nested-Unit Petri Nets (NUPN) is available from <http://mcc.lip6.fr/nupn.php>

^(o) safe by construction – stated by the [CÆSAR](#) compiler.

^(p) stated by [CÆSAR.BDD](#) version 3.3 to be true on 17 instance(s) out of 20, and unknown on the remaining 3 instance(s).

^(q) stated by [CÆSAR.BDD](#) version 3.3 to be true on 17 instance(s) out of 20, and unknown on the remaining 3 instance(s).

^(r) stated by [CÆSAR.BDD](#) version 3.3 to be true on 5 instance(s) out of 20, and unknown on the remaining 15 instance(s).

^(s) stated by [CÆSAR.BDD](#) version 3.3 to be false on 5 instance(s) out of 20, and unknown on the remaining 15 instance(s).

^(t) stated by [CÆSAR.BDD](#) version 3.3 to be false on 17 instance(s) out of 20, and unknown on the remaining 3 instance(s).

^(u) stated by [CÆSAR.BDD](#) version 3.3.

-
- (v) upper bound given by the number of leaf units.
 - (w) stated by [CÆSAR.BDD](#) version 3.3.
 - (x) upper bound given by the number of leaf units.
 - (y) stated by [CÆSAR.BDD](#) version 3.3.
 - (z) stated by [CÆSAR.BDD](#) version 3.3.
 - (aa) stated by the [CÆSAR](#) compiler.
 - (ab) upper bound given by the number of leaf units.
 - (ac) stated by [CÆSAR.BDD](#) version 3.3.
 - (ad) stated by [CÆSAR.BDD](#) version 3.3.
 - (ae) stated by the [CÆSAR](#) compiler.
 - (af) upper bound given by the number of leaf units.
 - (ag) stated by [CÆSAR.BDD](#) version 3.3.
 - (ah) stated by the [CÆSAR](#) compiler.
 - (ai) upper bound given by the number of leaf units.
 - (aj) stated by [CÆSAR.BDD](#) version 3.3.
 - (ak) stated by [CÆSAR.BDD](#) version 3.3.
 - (al) stated by the [CÆSAR](#) compiler.
 - (am) upper bound given by the number of leaf units.
 - (an) stated by [CÆSAR.BDD](#) version 3.3.
 - (ao) stated by the [CÆSAR](#) compiler.
 - (ap) upper bound given by the number of leaf units.
 - (aq) stated by [CÆSAR.BDD](#) version 3.3.
 - (ar) stated by the [CÆSAR](#) compiler.
 - (as) upper bound given by the number of leaf units.
 - (at) stated by [CÆSAR.BDD](#) version 3.3.
 - (au) stated by the [CÆSAR](#) compiler.
 - (av) upper bound given by the number of leaf units.
 - (aw) stated by [CÆSAR.BDD](#) version 3.3.
 - (ax) stated by the [CÆSAR](#) compiler.
 - (ay) upper bound given by the number of leaf units.
 - (az) stated by [CÆSAR.BDD](#) version 3.3.
 - (ba) stated by the [CÆSAR](#) compiler.
 - (bb) upper bound given by the number of leaf units.
 - (bc) stated by [CÆSAR.BDD](#) version 3.3.
 - (bd) stated by the [CÆSAR](#) compiler.
 - (be) upper bound given by the number of leaf units.
 - (bf) stated by [CÆSAR.BDD](#) version 3.3.
 - (bg) stated by the [CÆSAR](#) compiler.
 - (bh) upper bound given by the number of leaf units.
 - (bi) stated by [CÆSAR.BDD](#) version 3.3.
 - (bj) stated by the [CÆSAR](#) compiler.
 - (bk) upper bound given by the number of leaf units.
 - (bl) stated by [CÆSAR.BDD](#) version 3.3.
 - (bm) stated by the [CÆSAR](#) compiler.
 - (bn) upper bound given by the number of leaf units.
 - (bo) stated by [CÆSAR.BDD](#) version 3.3.
 - (bp) stated by the [CÆSAR](#) compiler.
 - (bq) upper bound given by the number of leaf units.
 - (br) stated by [CÆSAR.BDD](#) version 3.3.
 - (bs) stated by the [CÆSAR](#) compiler.
 - (bt) upper bound given by the number of leaf units.