

Petri Net File Format Syntax Specification

This document defines the syntax of the interpreted Petri Net file format (RIPN) integrated with a shared BeliefStore. It supports:

- Facts (with/without parameters)
- Integer and real variables
- Durative and discrete actions
- Timers (start, stop, pause, continue)
- Logical and arithmetic conditions using MVEL syntax
- Wildcard matching of facts (using '_' in forget/conditions)
- Immediate vs. non-immediate transitions

Example with all headers:

FACTS:

VARSENT: x; y

VARREAL:

INIT: x:=1; y:=1

DISCRETE: act1();

DURATIVE:

TIMERS: temp

hola

PLACES: p0; p1; p2; p3; p4

TRANSITIONS: t0; t1; t2; t3

ARCS: p1->t1; t1->p2; p2->t2; t2->p1; p1->t3; t3->p4; p0->t0; t0->p1; p3->t3

INITMARKING: (1,0,0,1,0)

<PN>

p0: [temp.start(30)]

t1: [] if(!temp.end)

p1: [act1(); temp.pause()] if(x<5)

p2: [x:=x+1]

t3: [y:=2] if (temp.end)

Additional Examples

Wildcard forget:

forget(detected(_, 3))

Action with parameters:

turn(90)

heat(zone, level)

Remember and condition:

remember(fire(2))

if (fire(2) || fire(3))

Timer management:

t1.start(5) # Starts a timer of 5 seconds

t1.pause() # Pauses the timer

t1.continue() # Resumes the paused timer

t1.stop() # Stops the timer and activates t1.end fact

Using conditions in transitions and places:

t1: [] if (x > 3 && !alert)

p2: [remember(alert); x := x + 1] if (ready)

Timers generate a fact 't.end' automatically, usable in conditions.

Immediate transitions: not listed in <PN> (fire as soon as enabled).

Non-immediate transitions: listed in <PN> and may have conditions.

Transitions can update the BeliefStore (variables/facts) but do NOT start actions.

Places can both update the BeliefStore and launch actions.

Wildcards (_) are allowed in conditions and forget(f(...)), not in remember(...).

```
<program> ::= { <header_line> } <structure_section> <pn_section>
```

```
<header_line> ::= "FACTS:" <name_list>
                | "VARSINT:" <name_list>
                | "VARSREAL:" <name_list>
                | "INIT:" <init_statement_list>
                | "DISCRETE:" <action_signature_list>
                | "DURATIVE:" <action_signature_list>
                | "TIMERS:" <name_list>
                | "#" <comment>
```

```
<name_list> ::= <name> { ";" <name> }
```

```
<action_signature_list> ::= <action_signature> { ";" <action_signature> }
```

```
<action_signature> ::= <name> "(" [ <param_type_list> ] ")"
```

```
<param_type_list> ::= <type> { "," <type> }
```

```
<type> ::= "INT" | "REAL"
```

```
<init_statement_list> ::= <init_statement> { ";" <init_statement> }
```

```
<init_statement> ::= <var_name> ":=" <expression>
```

```
<structure_section> ::= "PLACES:" <name_list>
                        | "TRANSITIONS:" <name_list>
                        | "ARCS:" <arc_list>
                        | "INITMARKING:" "(" <marking_list> ")"
```

```
<arc_list> ::= <arc> { ";" <arc> }
```

```
<arc> ::= <place_name> "->" <transition_name>
        | <transition_name> "->" <place_name>
        | <place_name> "-o>" <transition_name>      (* inhibitor arc *)
```

```
<marking_list> ::= <number> { "," <number> }
```

```
<pn_section> ::= "<PN>" { <pn_entry> }
```

```
<pn_entry> ::= <place_or_transition> ":" "[" <statement_list> "]" [ "if" "(" <condition>
)" " ]
```

```
<statement_list> ::= <statement> { ";" <statement> }
```

```
<statement> ::= <assignment>
                | <action_call>
                | "remember(" <fact_instance> ")"
```

| "forget(" <fact_instance> ")"

<assignment> ::= <var_name> ":=" <expression>

<action_call> ::= <name> "(" [<arg_list>] ")"

<fact_instance> ::= <fact_name> ["(" <int_arg_list> ")"]

<arg_list> ::= <expression> { "," <expression> }

<int_arg_list> ::= <int> { "," <int> }

<condition> ::= <boolean_expression> (* passed to MVEL evaluator *)

<name> ::= (letter | "_") { letter | digit | "_" }

<var_name> ::= <name>

<place_name> ::= <name>

<transition_name> ::= <name>

<fact_name> ::= <name>

<expression> ::= ... (* as allowed by MVEL *)

<boolean_expression> ::= ... (* as allowed by MVEL *)

<number> ::= digit { digit }

<int> ::= <number>