# Support for CPN4DCR Proof

## 1 Coloured Petri Net

A Petri net [5] is a formal model with mathematics-based execution semantics. It is a directed bipartite graph with two types of nodes: places (drawn as circles) and transitions (drawn as rectangles). Despite its efficiency in modelling and analysing systems, a basic Petri net falls short when the system is too complex, especially when representation of data is required. To overcome such limitations, extensions to basic Petri nets were proposed, equipping the tokens with colours or types [1], [6] and hence allowing them to hold values. A large Petri net model can therefore be represented in a much more compact and manageable manner using a *Coloured Petri net.*

A Coloured Petri Net (CP-net or CPN) [2] combines the capabilities of Petri nets, from which its graphical notation is derived, with those of CPN ML, a functional programming language based on Standard ML [3], to define data types. The formal definition of a Coloured Petri net is given in Definition 1 and the main concepts needed to define its dynamics are given in Definition 2.

**Definition 1 (Coloured Petri net).** *A* Coloured Petri Net *is a nine-tuple* $CPN = (P, T, A, \Sigma, V, C, G, E, I)$*, where:*

1. $P$ *is a finite set of* places.
2. $T$ *is a finite set of* transitions *such that* $P \cap T = \emptyset$.
3. $A \subseteq (P \times T) \cup (T \times P)$ *is a set of directed* arcs.
4. $\Sigma$ *is a finite set of non-empty* colour sets.
5. $V$ *is a finite set of* typed variables *such that* $Type[v] \in \Sigma$ *for all variables* $v \in V$.
6. $C : P \to \Sigma$ *is a* colour set function *that assigns a colour set to each place.*
7. $G : T \to EXPR_V$*, where* $EXPR_V$ *is the set of expressions provided by CPN ML with variables in* $V$*, is a* guard function *that assigns a guard to each transition* $t$ *such that* $Type[G(t)] = Bool$.
8. $E : A \to EXPR_V$ *is an* arc expression function *that assigns an arc expression to each arc* $a$ *such that* $Type[E(a)] = C(p)_{MS}$*, where* $p$ *is the place connected to the arc* $a$ *(i.e., the type of the arc expression is a multiset type over the colour set of the connected place).*
9. $I : P \to EXPR_\emptyset$ *is an* initialisation function *that assigns an initialisation expression to each place* $p$ *such that* $Type[I(p)] = C(p)_{MS}$.

**Definition 2 (CPN concepts).** *For a Coloured Petri Net* $CPN = (P, T, A, \Sigma, V, C, G, E, I)$*, we define the following concepts:*

1. $\bullet p$ *and* $p\bullet$ *respectively denote the sets of input and output transitions of a place* $p$.

2. $\bullet t$ *and* $t\bullet$ *respectively denote the sets of input and output places of a transition* $t$.
3. *A* marking *is a function* $M$ *that maps each place* $p \in P$ *into a multiset of tokens* $M(p) \in C(p)_{MS}$.
4. *The* initial marking $M_0$ *is defined by* $M_0(p) = I(p)\langle\rangle$ *for all* $p \in P$.
5. *The* variables of a transition $t$ *are denoted by* $Var(t) \subseteq V$ *and consist of the free variables appearing in its guard and in the arc expressions of its connected arcs.*
6. *A* binding *of a transition* $t$ *is a function* $b$ *that maps each variable* $v \in Var(t)$ *into a value* $b(v) \in Type[v]$. *It is written as* $\langle var_1 = val_1, ..., var_n = val_n\rangle$. *The set of all bindings for a transition* $t$ *is denoted* $B(t)$.
7. *A* binding element *is a pair* $(t, b)$ *such that* $t \in T$ *and* $b \in B(t)$. *The set of all binding elements* $BE(t)$ *for a transition* $t$ *is defined by* $BE(t) = \{(t, b)|b \in B(t)\}$. *The set of all binding elements in a CPN model is denoted* $BE$.
8. *A* step $Y \in BE_{MS}$ *is a non-empty, finite multiset of binding elements.*

A transition is said to be *enabled* if a binding of the variables appearing in the surrounding arc inscriptions exists such that the inscription on each input arc evaluates to a multiset of token colours that is present on the corresponding input place. *Firing* a transition consists in removing (resp. adding), from each input place (resp. to each output place), the multiset of tokens corresponding to the input (resp. output) arc inscription. For more details on the CPN formalism and the formal definition of its semantics, we refer readers to [2].

**Definition 3 (Enabling and occurrence of a binding element).** *A binding element* $(t, b) \in BE$ *is enabled in a marking* $M$ *if and only if the following two properties are satisfied:*

1. $G(t)\langle b\rangle$.
2. $\forall p \in P : E(p, t)\langle b\rangle \ll= M(p)$.
   *When* $(t, b)$ *is enabled in* $M$, *it may occur, leading to the marking* $M'$ *defined by:*
3. $\forall p \in P : M'(p) = (M(p) - -E(p, t)\langle b\rangle) + +E(t, p)\langle b\rangle$.

**Definition 4 (Enabling and occurrence of steps).** *A step* $Y \in BE_{MS}$ *is enabled in a marking* $M$ *if and only if the following two properties are satisfied:*

1. $\forall (t, b) \in Y : G(t)\langle b\rangle$.
2. $\forall p \in P :^{++}_{MS} \sum_{(t,b)\in Y} E(p, t)\langle b\rangle \ll= M(p)$
   *When* $Y$ *is enabled in* $M$, *it may occur, leading to the marking* $M'$ *defined by:*
3. $\forall p \in P : M'(p) = (M(p) - -^{++}_{MS} \sum_{(t,b)\in Y} E(p, t)\langle b\rangle) + +^{++}_{MS} \sum_{(t,b)\in Y} E(t, p)\langle b\rangle$.

**Definition 5 (Occurrence sequences and reachability).** *A finite occurrence sequence of length* $n \geq 0$ *is an alternating sequence of markings and steps, written as*

$$M_1 \xrightarrow{Y_1} M_2 \xrightarrow{Y_2} M_3 ... M_n \xrightarrow{Y_n} M_{n+1}$$

*such that* $M_i \xrightarrow{Y_i} M_{i+1}$ *for all* $1 \leq i \leq n$. *All markings in the sequence are said to be reachable from* $M_1$. *This implies that an arbitrary marking* $M$ *is reachable from itself by the trivial occurrence sequence of length 0.*

*Analogously, an infinite occurrence sequence is a sequence of markings and steps*

$$M_1 \xrightarrow{Y_1} M_2 \xrightarrow{Y_2} M_3 \xrightarrow{Y_3} ...$$

*such that* $M_i \xrightarrow{Y_i} M_{i+1}$ *for all* $i \geq 1$. *The set of markings reachable from a marking* $M$ *is denoted* $\mathscr{R}(M)$. *The set of reachable markings is* $\mathscr{R}(M_0)$, *i.e., the set of markings reachable from the initial marking* $M_0$.

**Theorem 1.** *Let* $Y$ *be a step and* $M$ *and* $M'$ *be markings such that* $M \xrightarrow{Y} M'$. *Let* $Y1$ *and* $Y2$ *be steps such that*

$$Y = Y1 + +Y2$$

*Then there exists a marking* $M''$ *such that*

$$M \xrightarrow{Y_1} M'' \xrightarrow{Y_2} M'$$

## 2 Dynamic Condition Response Graph

**Definition 6.** *A dynamic condition response graph is a tuple* $G = (E, M, Act, \rightarrow\bullet, \bullet\rightarrow, \rightarrow+, \rightarrow\%, \rightarrow\diamond, l)$ *where*

1. $E$ *is the set of events, ranged over by* $e$
2. $M \in \mathcal{M}(G) =_{def} \mathcal{P}(E) \times \mathcal{P}(E) \times \mathcal{P}(E)$ *is the marking and* $\mathcal{M}(G)$ *is the set of all markings*
3. $Act$ *is the set of actions*
4. $\rightarrow\bullet \subseteq E \times E$ *is the condition relation*
5. $\bullet\rightarrow \subseteq E \times E$ *is the response relation*
6. $\rightarrow+, \rightarrow\% \subseteq E \times E$ *is the dynamic include relation and exclude relation satisfying that* $\forall e \in E.e \rightarrow+ \cap e \rightarrow\% = \emptyset$
7. $\rightarrow\diamond \subset E \times E$ *is the milestone relation*
8. $l : E \rightarrow Act$ *is a labelling function mapping every event to an action.*

*The marking (2)* $M = (Ex, Re, In) \in \mathcal{M}(G)$ *is a triplet of event sets where the first component represents the set of events that have previously been executed (*$Ex$*), the second component represents the set of events that are pending responses required to be executed or excluded (*$Re$*), and third components represents the set of events that are currently included (*$In$*). The idea conveyed by the dynamic inclusion/exclusion relations (6)* $\rightarrow+$ *and* $\rightarrow\%$ *is that only the currently included events are considered in evaluating the constraints. In other words, if an event* $b$ *is a condition for an event* $a$, *but it is excluded from the graph then it no longer restricts the execution of the event* $a$. *Moreover, if event* $b$ *is the response for an event* $a$ *but it is excluded from the graph, then it is no longer*

*required to happen for the flow to be acceptable. The inclusion relation* $e \rightarrow\!\!+\ e'$ *means that, whenever* $e$ *is executed,* $e'$ *becomes included in the graph if it is not already. The exclusion relation* $e \rightarrow\!\!\%e'$ *means that when* $e$ *is executed,* $e'$ *becomes excluded from the graph if it is not already. The milestone relation (7) is similar to the condition relation in that it is a blocking one. The difference is that it based on the events in the pending response set. In other words, if an event* $b$ *is a milestone of an event* $a$ *(*$b \rightarrow\!\!\diamond\ a$*), then the event* $a$ *cannot be executed as long as the event* $b$ *is in the set of pending responses (*$Re$*). It is worth mentioning that, like a condition relation, a milestone relation is only blocking when the event in question is included in the graph. For more details on DCR Graphs and the formal definition of their semantics which establishes the dynamics of the graphs, we refer the readers to [4].*

**Definition 7 (Enabled event).** *For a dynamic condition response graph* $G = (E, M, Act, \rightarrow\!\!\bullet, \bullet\!\!\rightarrow, \rightarrow\!\!+, \rightarrow\!\!\%, \rightarrow\!\!\diamond, l)$ *with marking* $M = \{Ex; Re; In\}$*, we define that an event* $e \in E$ *is enabled, written as* $M \vdash_G e$ *if*

1. $e \in In$
2. $(\rightarrow\!\!\bullet e \cap In) \in Ex$
3. $(\rightarrow\!\!\diamond e \cap In) \in E \setminus Re$

**Definition 8 (Event execution effect).** *For a dynamic condition response graph* $G = (E, M, Act, \rightarrow\!\!\bullet, \bullet\!\!\rightarrow, \rightarrow\!\!+, \rightarrow\!\!\%, \rightarrow\!\!\diamond, l)$ *with marking* $M = \{Ex; Re; In\}$ *and with an enabled event* $M \vdash_G e$*, the result of executing the event* $e$ *will be a dynamic condition response graph* $G = (E, M', Act, \rightarrow\!\!\bullet, \bullet\!\!\rightarrow, \rightarrow\!\!+, \rightarrow\!\!\%, \rightarrow\!\!\diamond, l)$*, where* $M' = M \oplus_G e = \{Ex'; Re'; In'\}$ *such that*

1. $Ex' = Ex \cup \{e\}$
2. $Re' = (Re \setminus \{e\}) \cup e\!\!\bullet\!\!\rightarrow$
3. $In' = (In \cup e \rightarrow\!\!+) \setminus e \rightarrow\!\!\%$

**Definition 9 (Graph execution).** *For a Dynamic Condition Response Graph* $G = (E, M, Act, \rightarrow\!\!\bullet, \bullet\!\!\rightarrow, \rightarrow\!\!+, \rightarrow\!\!\%, \rightarrow\!\!\diamond, l)$ *we define an execution of* $G$ *to be a (finite or infinite) sequence of tuples* $\{(M_i; e_i; a_i; M_i')\}_{i \in [k]}$ *each consisting of a marking, an event, a label and another marking (the result of executing the event) such that*

1. $M = M_0$
2. $\forall i \in [k].a_i \in l(e_i)$
3. $\forall i \in [k].M_i \vdash_G e_i$
4. $\forall i \in [k].M_i' = M_i \oplus_G e_i$
5. $\forall i \in [k-1].M'i = M_{i+1}$.

*Further, we say the execution (or a run) is accepting if* $\forall i \in [k].(\forall e \in In_i \cap Re_i.\exists j \geq i.e_j = e \vee e \notin In_j')$*, where* $M_i = (Ex_i; In_i; Re_i)$ *and* $M_j' = (Ex_j'; In_j'; Re_j')$.

*Finally we say that a marking* $M'$ *is reachable in* $G$ *(from the marking* $M$*) if there exists a finite execution ending in* $M'$ *and let* $\mathcal{M}_{M \rightarrow\!\!^*}(G)$ *denote the set of all reachable markings from* $M$.

## 3 CPN4DCR Model

The user may want to define the behavior of smart contracts by specifying a set of constraints. This can be captured using a DCR Graph (see Definition 6). In order to be able to integrate such a representation in our CPN hierarchical model, we propose a CPN model for DCR.

**Definition 10 (CPN4DCR).** *Given a dynamic condition response graph* $G = (E, M, Act, \rightarrow\bullet, \bullet\rightarrow, \pm, l)$*, a corresponding CPN model* $CPN = (P, T, A, \Sigma, V, C, G, E, I)$ *is defined such that:*

- $P = \{S\}$
- $T = \{t_i, \forall i \in [1, n]\}$*, with* $n = |E|$ *the number of events in* $G$
- $A = \{(t_i, S), \forall i \in T\} \cup \{(S, t_i), \forall i \in T\}$
- $\Sigma = \{C_E, (C_E \times C_E \times C_E)\}$*, where* $C_E$ *is a colour defined to represent a set of events. Here we define* $C_E$ *as an integer type* ($C_E = rangeINT$) *where each event* $e_i \in E$ *is represented in* $C_E$ *by its index.*
- $V = \{Ex, Re, In, Ex', Re', In'\}$*, with* $Type[v] = C_E, \forall v \in V$
- $C = \{S \rightarrow (C_E \times C_E \times C_E)\}$
- $G = \{t_i \rightarrow guard_i, \forall i \in [1, n]\}$*, with* $n = |E|$
- $E = \{a \rightarrow < Ex, Re, In >, \forall a \in A \cap (P \cup T)\} \cup \{a \rightarrow < Ex', Re', In' >, \forall a \in A \cap (T \cup P)\}$ *with*
  - $Ex' = Ex \cup e_i$
  - $Re' = (Re \setminus e_i) \cup e\bullet\rightarrow$
  - $In' = (In \cup e_i \rightarrow +) \setminus e \rightarrow \%$
- $I = \{S \rightarrow < S_1, S_2, S_3 >\}$ *with* $< S_1, S_2, S_3 >$ *corresponding to the initial marking* $M$ *of* $G$

*For each transition* $t_i$ *in the CPN model representing an event* $e_i$ *in the DCR graph, we further precise that:*

- $guard_i$ *is the conjunction of the conditions defining the enabling of the corresponding event* $e_i$*:*
  - $i \in In$
  - $(\rightarrow\bullet i \cap In) \in Ex$
  - $(\rightarrow\diamond i \cap In) \in E \setminus Re$
- *the expression* $< Ex', Re', In' >$ *on its output arc is defined such that:*
  - $Ex' = Ex \cup i$
  - $Re' = (Re \setminus i) \cup i\bullet\rightarrow$
  - $In' = (In \cup i \rightarrow +) \setminus i \rightarrow \%$

## 4 Theorem Proof

**Definition 11 (Marking Equivalence).** *A marking* $M^G =< Ex, Re, In >$ *of a DCR graph* $G$ *is said to be equivalent to a marking* $M^C =< S \rightarrow < S_1, S_2, S_3 >>$ *of a CPN model* $C$ *iff*

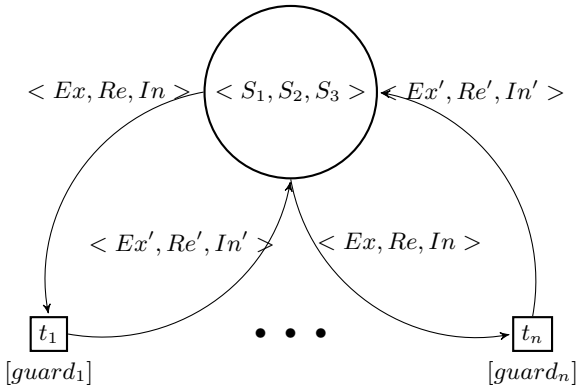


Fig. 1: CPN model for a DCR Graph

- $\forall e_i \in Ex$ *(respectively* $Re$ *and* $In$*)*, $\exists i \in S_1$ *(respectively* $S_2$ *and* $S_3$*), and*
- $\forall i \in S_1$ *(respectively* $S_2$ *and* $S_3$*)*, $\exists e_i \in Ex$ *(respectively* $Re$ *and* $In$*)*

*We note* $M^G \equiv M^C$.

**Definition 12 (Execution Sequence Equivalence).** *An execution sequence of length* $k$, $\sigma_k^G =< e_i, ..., e_j >$ *of a DCR graph* $G$ *is said to be equivalent to an execution sequence of length* $k$, $\sigma_k^C =< t_i, ..., t_j >$ *of a CPN model* $C$ *iff*

- $(M_1^G \equiv M_1^C \wedge M_1^G \xrightarrow{\sigma_k^G} M_2^G \wedge M_1^C \xrightarrow{\sigma_k^C} M_2^C) \implies M_2^G \equiv M_2^C$

*We note* $\sigma_k^G \equiv \sigma_k^C$.

**Theorem 2.** *Let* $G$ *be a DCR graph and* $C$ *the corresponding CPN model generated by following definition 10, then* $G$ *and* $C$ *are semantically equivalent.*

*Proof.* Let $G$ be a DCR graph and $C$ the corresponding CPN model generated by following definition 10. In order to prove that $G$ and $C$ are semantically equivalent we need to prove that

1. $\forall \sigma_k^G =< e_1, ..., e_k >, \exists \sigma_k^C =< t_1, ..., t_k >$, and
2. $\forall \sigma_k^C =< t_1, ..., t_k >, \exists \sigma_k^G =< e_1, ..., e_k >$

such that $\sigma_k^G \equiv \sigma_k^C$, $\forall k \in [1, m]$ with $m$ the length of the longest execution sequence.
We start by proving *(1)*:

- Let $P(n)$ be the statement: $\forall \sigma_n^G =< e_1, ..., e_n >, \exists \sigma_n^C =< t_1, ..., t_n >$ such that $\sigma_n^G \equiv \sigma_n^C$.
- $P(1) : \forall \sigma_1^G =< e_1 >, \exists \sigma_1^C =< t_1 >$ such that $\sigma_1^G \equiv \sigma_1^C$. This can be derived from Definition 10. In fact, the initial marking of $C$ ($M_0^C$) being defined as equivalent to that of $G$ ($M_0^G$), and the guard of each transition $t_i \in T$

being defined as to correspond to the enabling conditions of the relative event $e_i \in E$, we can deduce that the set of fireable transitions $(M_0^C \rightarrow)$ corresponds to the set of enabled events $(M_0^G \rightarrow)$. Additionally, the marking $M_i^C$ obtained by firing $t_i$ is equivalent to that obtained by executing $e_i$ $(M_i^C \equiv M_i^G)$ since the elements of $M_i^C$ are defined as to correspond to the effect of the execution of $e_i$ in $G$.

– Assume that $P(k) : \forall \sigma_k^G =< e_1, ..., e_k >, \exists \sigma_k^C =< t_1, ..., t_k >$ such that $\sigma_k^G \equiv \sigma_k^C$ is true for some $k \in [2, m-1]$. We will prove that $P(k+1) : \forall \sigma_{k+1}^G =< e_1, ..., e_{k+1} >, \exists \sigma_{k+1}^C =< t_1, ..., t_{k+1} >$ such that $\sigma_{k+1}^G \equiv \sigma_{k+1}^C$ is true.

$$\sigma_{k+1}^G \equiv \sigma_{k+1}^C \implies \exists e_{k+1} \in E, t_{k+1} \in T \text{ such that } \sigma_k^G \cdot e_{k+1} \equiv \sigma_k^C \cdot t_{k+1} \quad (1)$$

$$\sigma_k^G \equiv \sigma_k^C \iff (M_0^G \xrightarrow{\sigma_k^G} M_k^G \wedge M_0^C \xrightarrow{\sigma_k^C} M_k^C \wedge M_k^G \equiv M_k^C) \quad (2)$$

Analogously to the reasoning in the previous point, we can deduce that:

$$\begin{aligned} &\forall e_{k+1} \in E \text{ such that } M_k^G \xrightarrow{e_{k+1}} M_{k+1}^G, \\ &\exists t_{k+1} \in T \text{ such that } (M_k^C \xrightarrow{t_{k+1}} M_{k+1}^C \wedge M_{k+1}^G \equiv M_{k+1}^C) \end{aligned} \quad (3)$$

And therefore:

$$\forall \sigma_{k+1}^G =< e_1, ..., e_{k+1} >, \exists \sigma_{k+1}^C =< t_1, ..., t_{k+1} > \text{ such that } \sigma_{k+1}^G \equiv \sigma_{k+1}^C \quad (4)$$

The second part *(2)* is provable following a similar reasoning.

## References


1. Jensen, K.: Coloured petri nets: A high level language for system design and analysis. In: International Conference on Application and Theory of Petri Nets. pp. 342–416. Springer (1989)
2. Jensen, K., Kristensen, L.M.: Coloured Petri Nets: Modelling and Validation of Concurrent Systems. Springer Publishing Company, Incorporated, 1st edn. (2009)
3. Milner, R., Tofte, M., Harper, R.: Definition of standard ML. MIT Press (1990)
4. Mukkamala, R.R.: A Formal Model For Declarative Workflows Dynamic Condition Response Graphs. Ph.D. thesis (06 2012)
5. Murata, T.: Petri nets: Properties, analysis and applications. Proceedings of the IEEE **77**(4), 541–580 (1989)
6. Van Hee, K., Verkoulen, P.: Integration of a data model and high-level petri nets. In: Proceedings of the 12th International Conference on Applications and Theory of Petri Nets, Gjern. pp. 410–431 (1991)