

Atelier « Évaluation de Performance »

Bruno DENIS

Maître de Conférences à ENS de Cachan

- Introduction à la simulation des Réseaux de Petri colorés temporisés [Jensen 94]
- CPN Tools : un outil pour les RdP colorés temporisés
- Etude de cas avec CPN Tools
- Evaluation (des étudiants) du 1h30

Définition des RdPC

■ Définition d'un multi-set

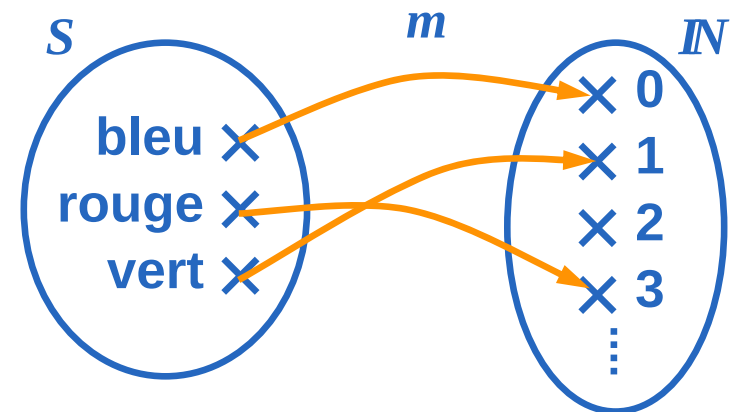
- Un multi-set, défini sur un ensemble S non vide, est une fonction $m \in [S \rightarrow \mathbb{N}]$ qui sera représentée symboliquement par la somme :

$$\sum_{s \in S} m(s) \cdot s$$

- On note S_{MS} l'ensemble de tous les multi-set définis sur S

■ Exemple de multi-set

- Le multi-set m représenté par la somme $3 \cdot \text{rouge} + 1 \cdot \text{vert}$ correspond à la fonction ci-contre



Définition des RdPC

■ Définitions : addition, multiplication par un scalaire, comparaison, cardinalité et soustraction des multi-sets

- Soit $m, m_1, m_2 \in S_{MS}$ et $n \in \mathbb{N}$

(i) $m_1 + m_2 = \sum_{s \in S} (m_1(s) + m_2(s)) \cdot s$ (addition)

(ii) $n * m = \sum_{s \in S} n * m(s)$ (multiplication par un scalaire)

(iii) $m_1 \neq m_2 \Leftrightarrow \exists s \in S : m_1(s) \neq m_2(s)$ (comparaison ; $\geq$ et $=$ sont définis de façon analogue à $\leq$)
 $m_1 \leq m_2 \Leftrightarrow \forall s \in S : m_1(s) \leq m_2(s)$

(iv) $|m| = \sum_{s \in S} m(s)$ (cardinalité)

- (v) Si $m_1 \leq m_2$ on peut définir la soustraction de deux multi-set

(vi) $m_2 - m_1 = \sum_{s \in S} (m_2(s) - m_1(s)) \cdot s$ (soustraction)

Définition des RdPC

- Exemples : addition, multiplication par un scalaire, comparaison, cardinalité et soustraction des multi-sets

- Soient $S = \{bleu, rouge, vert\}$
 $m_1 = 1`bleu + 3`rouge + 1`vert$
 $m_2 = 2`bleu$
 $m_3 = 1`bleu + 2`rouge$

$m_1 + m_2 = 3`bleu + 3`rouge + 1`vert$	(addition)
$3 * m_3 = 3`bleu + 6`rouge$	(multiplication par un scalaire)
$m_3 \leq m_1$	(comparaison)
$m_1 - m_3 = 1`rouge + 1`vert$	(soustraction)

- La syntaxe concrète retenue dans CPNTools pour l'addition de deux multisets est « ++ »

Définition des RdPC

■ « Colorset » : le type de jeton

- Colorsets de base
 - **les entiers (INT)**
 - **le type énuméré** → colset name = with id0 | id1 | ... | idn;
ex : pour $S = \{bleu, rouge, vert\}$, la déclaration est
colset S = with bleu | rouge | vert ;
- Colorsets composés
 - **les produits cartésiens** →
colset name = product name1 * name2 * ... * namen;
ex : pour le produit cartésien de S par INT, la déclaration est
colset SxINT = product S * INT ;
(rouge, 6) et (bleu, 0) sont deux exemples de jetons de type SxINT
 - **les listes** → colset name = list name0;
 - ex : pour les listes de couleurs appartenant à S, la déclaration est
colset S_List = list S ;

Définition des RdPC

■ Opérations

- **Type énuméré**

- Typename.all()

ex : avec

la fonction S.all() retourne

colset S = with bleu | rouge | vert ;

1`bleu++1`rouge++1`vert

- **Produit cartésien**

- #i retourne la ième composante

ex : #1 (bleu, 4) donne bleu

- **Liste**

- nil liste vide (notée également [])

- e::l retourne la liste l à laquelle l'élément e est ajouté à gauche

- l1^l2 concatène les deux listes

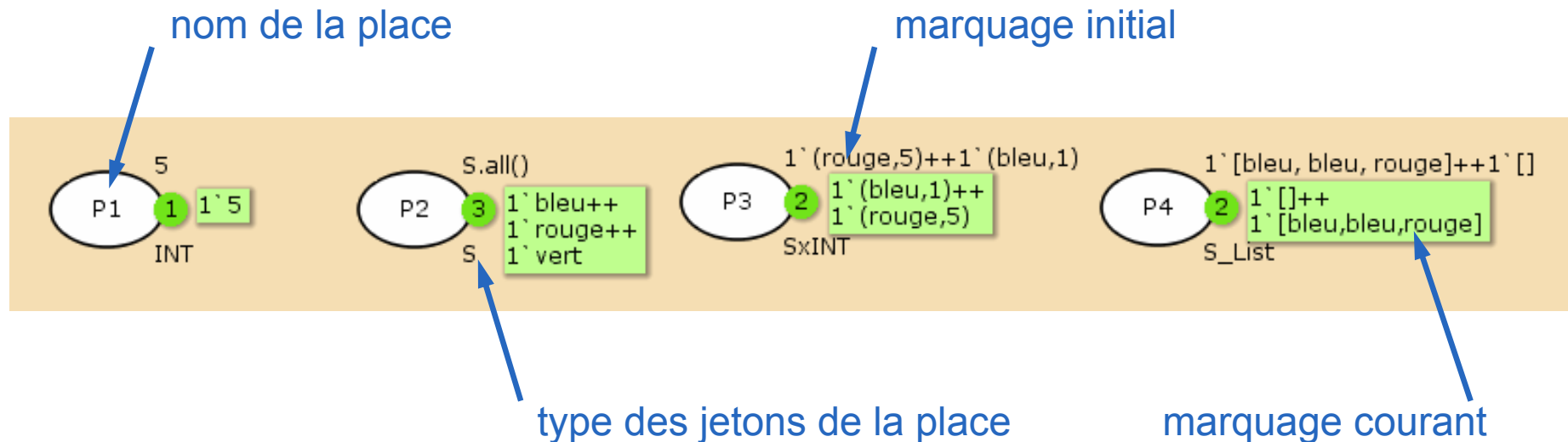
- length l1 longueur de la liste l1

Définition des RdPC

■ Les places

- soient les déclarations de variables suivantes :

```
colset S = with bleu | rouge | vert ;  
colset SxINT = product S * INT ;  
colset S_List = list S ;
```

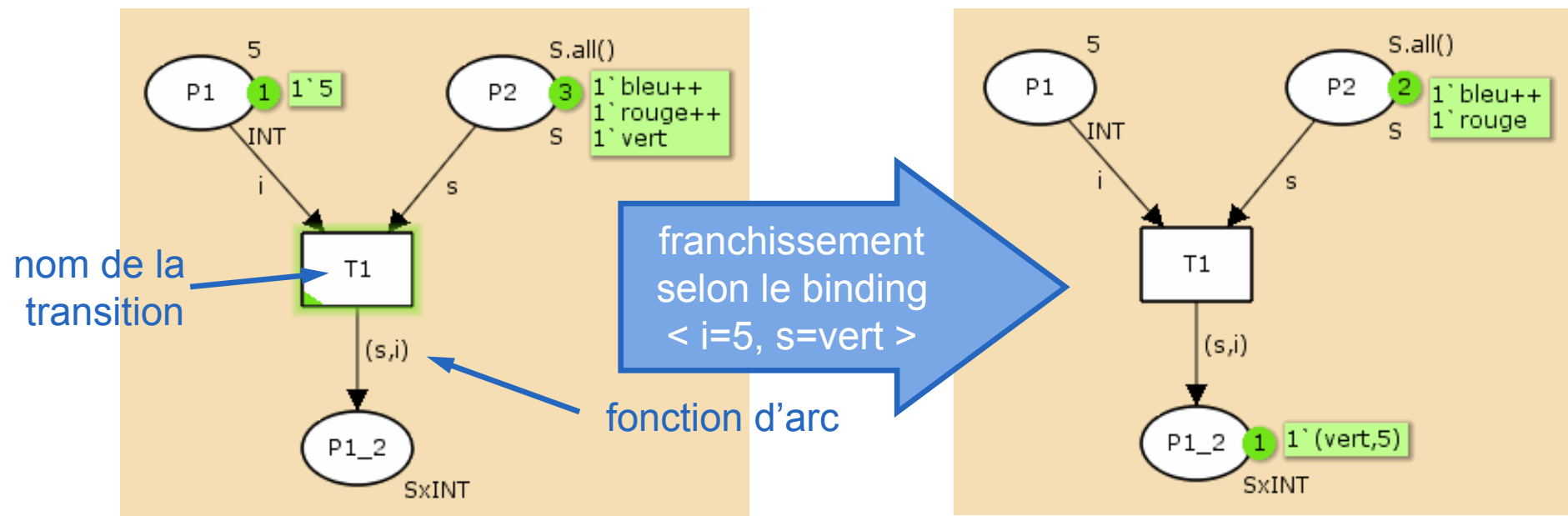


Définition des RdPC

■ Les transitions et les arcs

- soient les déclarations suivantes :

```
colset S = with bleu | rouge | vert ;  
colset SxINT = product S * INT ;  
var i : INT ;  
var s : S ;
```

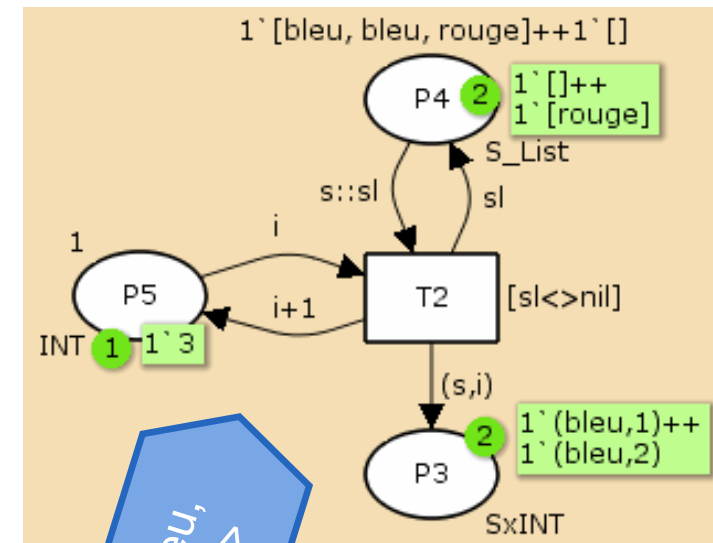


Définition des RdPC

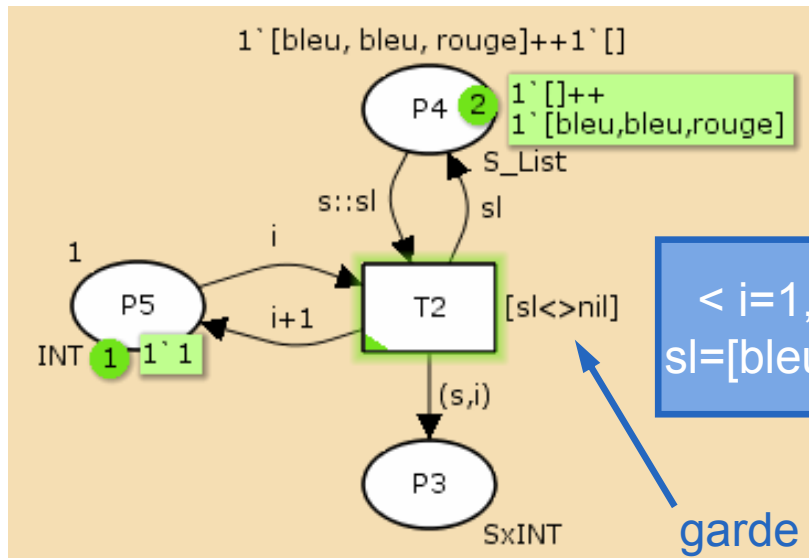
Les transitions et les arcs

- soient les déclarations suivantes :

```
colset S = with bleu | rouge | vert ;
colset SxINT = product S * INT ;
colset S_List = list S ;
var i : INT ;
var s : S;
var sl : S_List ;
```

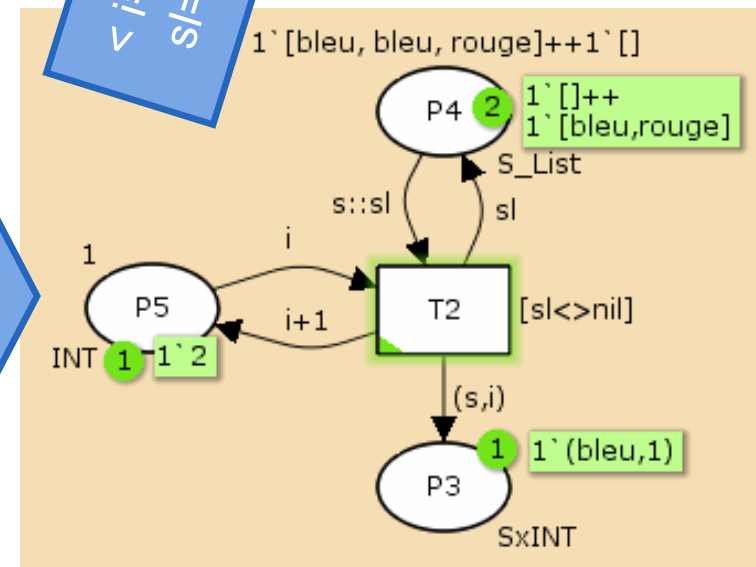


$\langle i=2, s=bleu, sl=[rouge] \rangle$



$\langle i=1, s=bleu, sl=[bleu, rouge] \rangle$

garde



Définition formelle des RdPC

- Un RdPC est un n-uplet $CPN = (\Sigma, P, T, A, N, C, G, E, I)$ avec :
 - (i) Σ un ensemble fini non vide de **types**, également appelé « **colour sets** »
 - (ii) P un ensemble fini de **places**
 - (iii) T un ensemble fini de **transitions**
 - (iv) A un ensemble fini d'**arcs** tels que $P \cap T = P \cap A = T \cap A = \emptyset$
 - (v) N une fonction définie de A dans $P \times T \cup T \times P$. Elle correspond aux **applications d'incidence avant et arrière**
 - (vi) C une fonction définie de P dans Σ . A chaque place est associé un type, c'est-à-dire un « colour set »
 - (vii) G une **fonction de garde** définie de l'ensemble des transitions T dans l'ensemble des expressions
 - (vii) E une **fonction d'expression des arcs** définie de l'ensemble des arcs A dans l'ensemble des expressions
 - (viii) I une **fonction d'initialisation** définie de P dans l'ensemble des expressions

Définition des RdPC

■ Définition d'un multi-set temporisé

- Soit R l'ensemble des valeurs du temps (un sous ensemble de $\mathbb{N}$)
- Un multi-set temporisé, défini sur un ensemble S non vide, est une fonction $m \in [S \times R \rightarrow \mathbb{N}]$ telle que :

$$tm(s) = \sum_{r \in R} tm(s, r)$$

est fini pour tout $s \in S$. La liste

$$tm[s] = [r_1, r_1, \dots, r_n]$$

est définie pour contenir les valeur du temps pour chaque $tm(s, r) \neq 0$

- Un multi-set temporisé sera représentée symboliquement par la somme :

$$\sum_{s \in S} tm(s) \cdot s @ tm[s]$$

- Exemple : $tm = 2 \cdot (bleu, 5) @ [456, 756] + 1 \cdot (rouge, 1) @ [583]$

Définition des RdPC

- Les opérateurs @, @+, and @@+ sont utilisés pour ajoutée des « time stamps » aux jetons

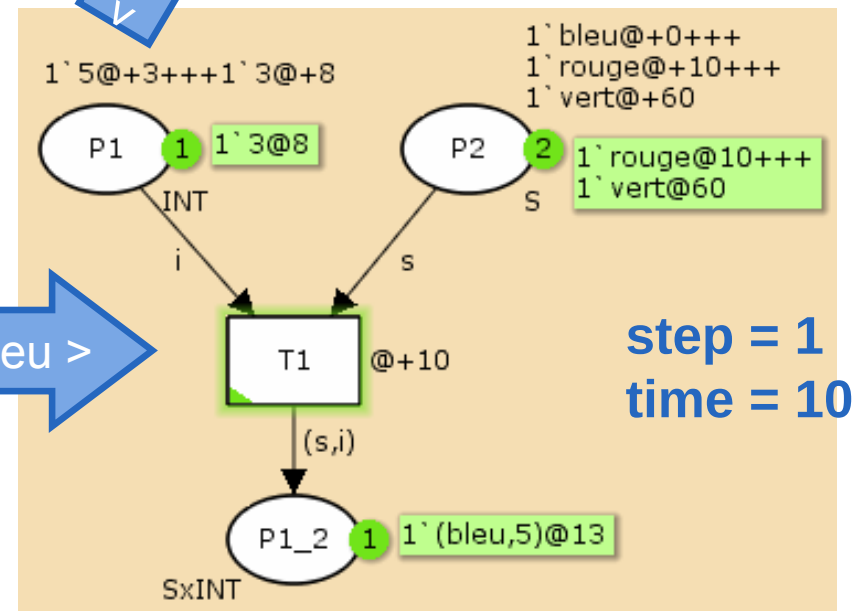
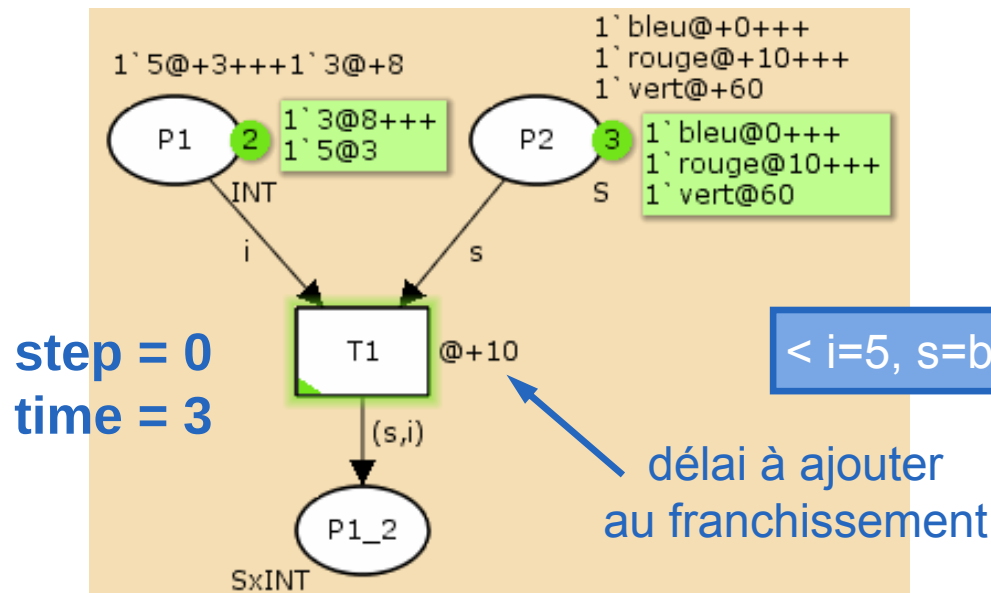
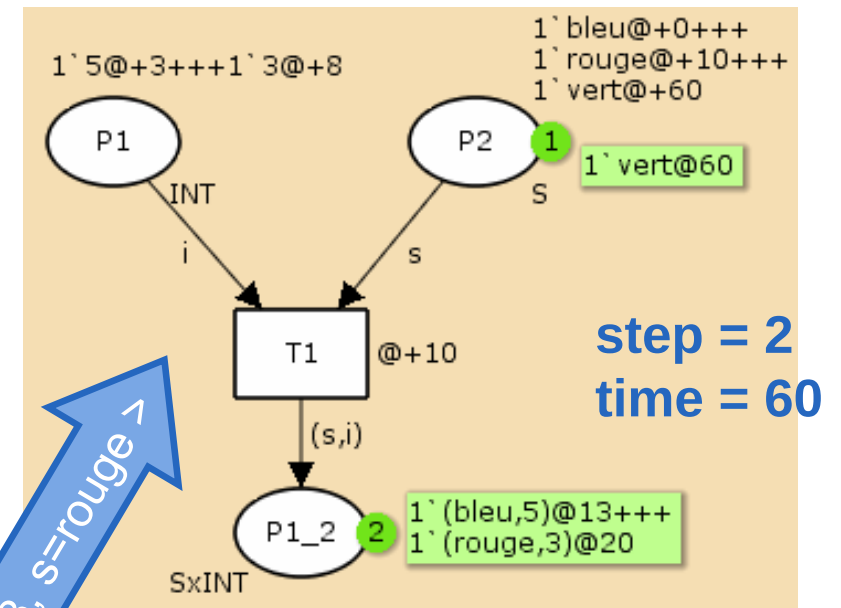
<code>c @ t</code>	attache le « time stamp » t (de type Time.time) au jeton c
<code>ms @+ i</code>	ajoute le délai i (de type entier) à chaque jeton du multi-set ms, et retourne un multi-set temporisé
<code>ms @@+ t</code>	Ajoute un délai t (de type Time.time) à chaque jeton du multi-set ms, et retourne un multi-set temporisé
<code>IntInf.fromInt i</code>	Convertit un entier en valeur de temps
<code>tms1 +++ tms2</code>	Addition de deux multi-sets temporisés

Définition des RdPC

Exemples d'évolution

- soient les déclarations suivantes :

```
colset INT = int timed;  
colset S = with bleu | rouge | vert timed;  
colset SxINT = product S * INT timed;  
var i : INT ;  
var s : S;
```



Définition formelle des RdPC temporisé

- Un RdPCT est un n-uplet $TCPN = (CPN, R, r_0)$ avec :
 - (i) CPN un RdPC
 - (ii) R l'ensemble des **valeurs du temps**
 - (iii) r_0 un élément de R , appelé **date de départ**

Bibliographie

■ Site WWW

- <http://www.daimi.au.dk/CPNTools/>

■ Définition d'un RdPC

[Jensen 94] An Introduction to the Theoretical Aspects of Coloured Petri Nets, Kurt Jensen, LNCS vol. 803, p. 230-272, 1994

■ Définition d'un RdPCT

[Jensen 95] Coloured Petri Nets – Basic Concepts, Analysis Methods and Practical Use, vol. 2, Kurt Jensen, Springer-Verlag, Berlin, 1995, 174 p.

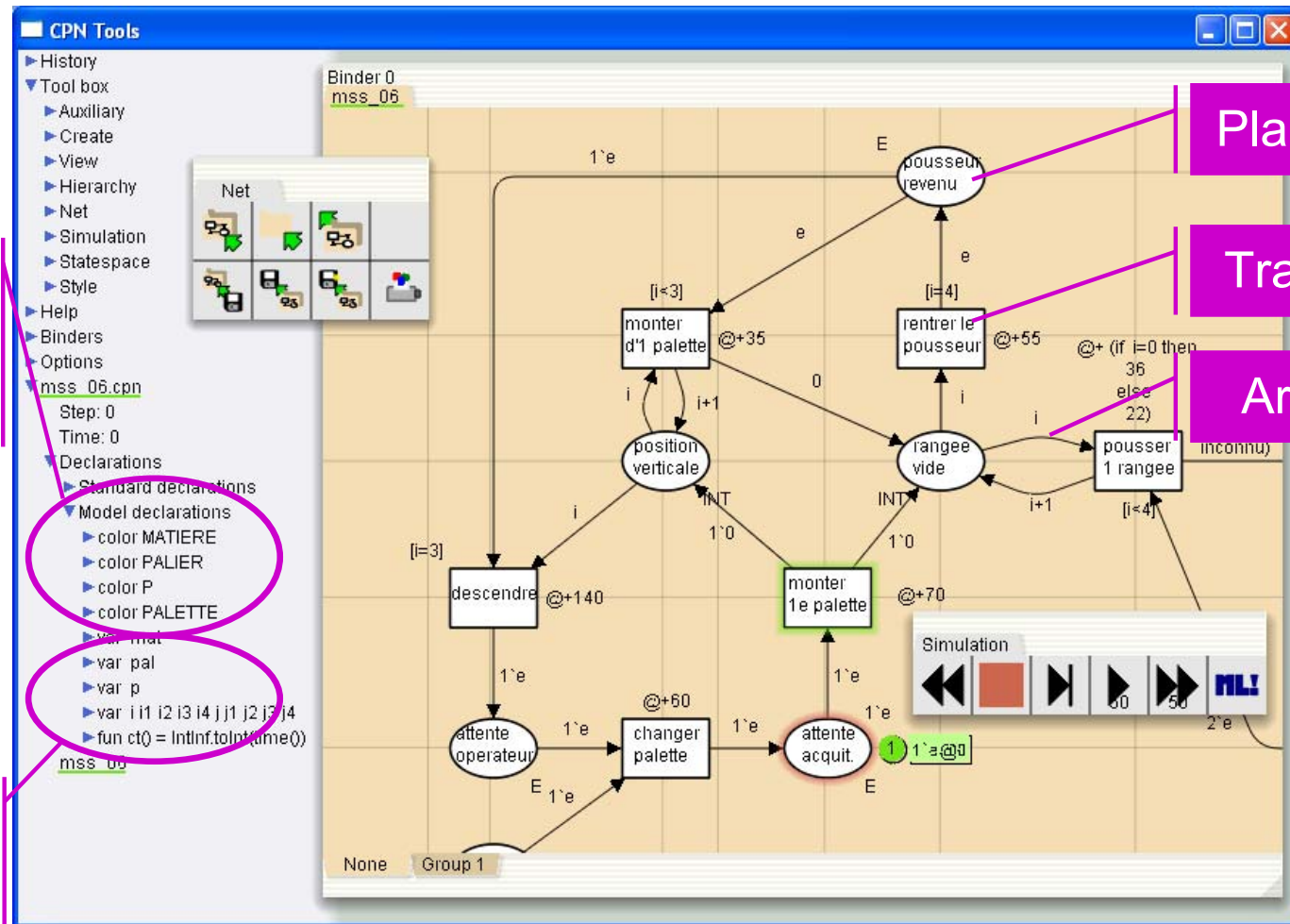
■ CPN Tools

[Beaudouin 01] CPN/Tools: A Post-WIMP Interface for Editing and Simulating Coloured Petri Nets, M. Beaudouin-Lafon, et al., Proc. of 22nd International Conf. on Applications and Theory of Petri Nets 2001 (ICATPN 2001), pages 71-80. LNCS vol. 2075, Springer, 2001

Le logiciel CPN Tools

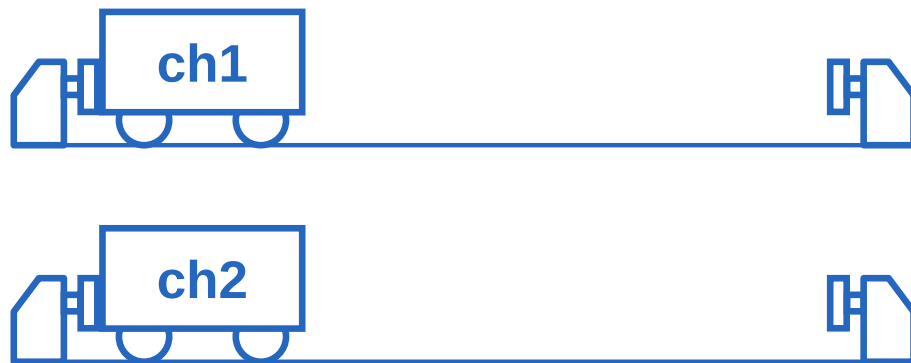
Déclarations
des colors
sets

Déclarations
des
variables

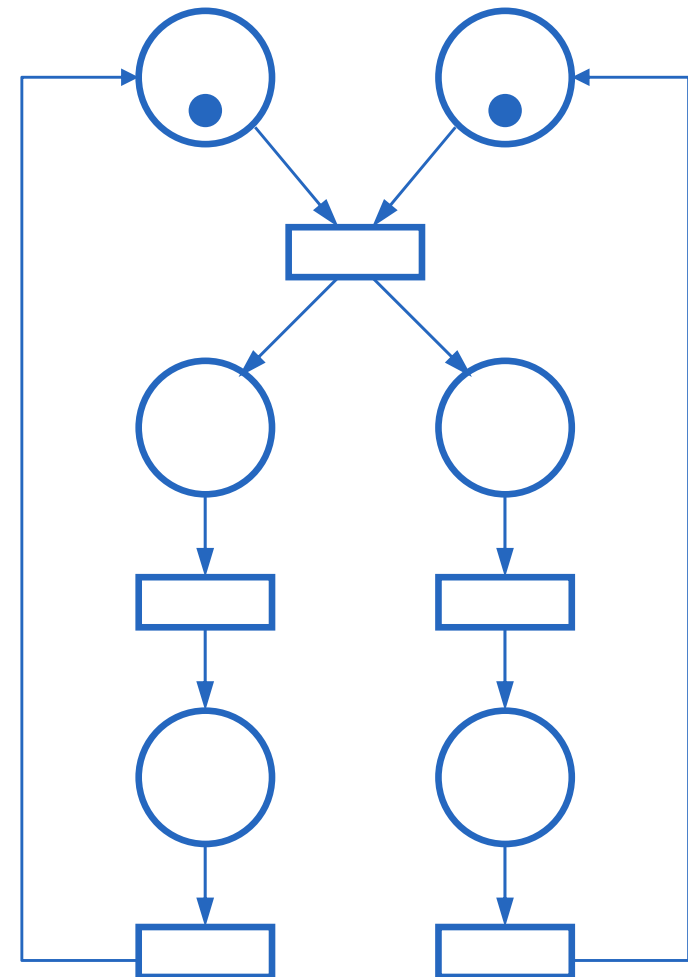


Exemple de comportement

- Deux chariots faisant un aller et retour avec départ simultané de la gauche

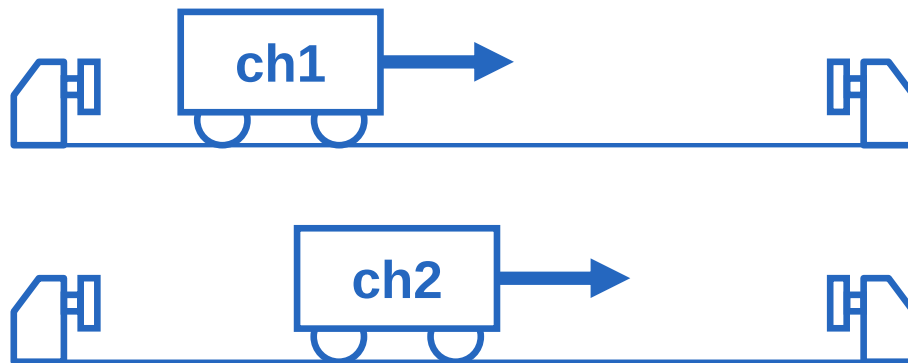


Réseau de Petri ordinaire

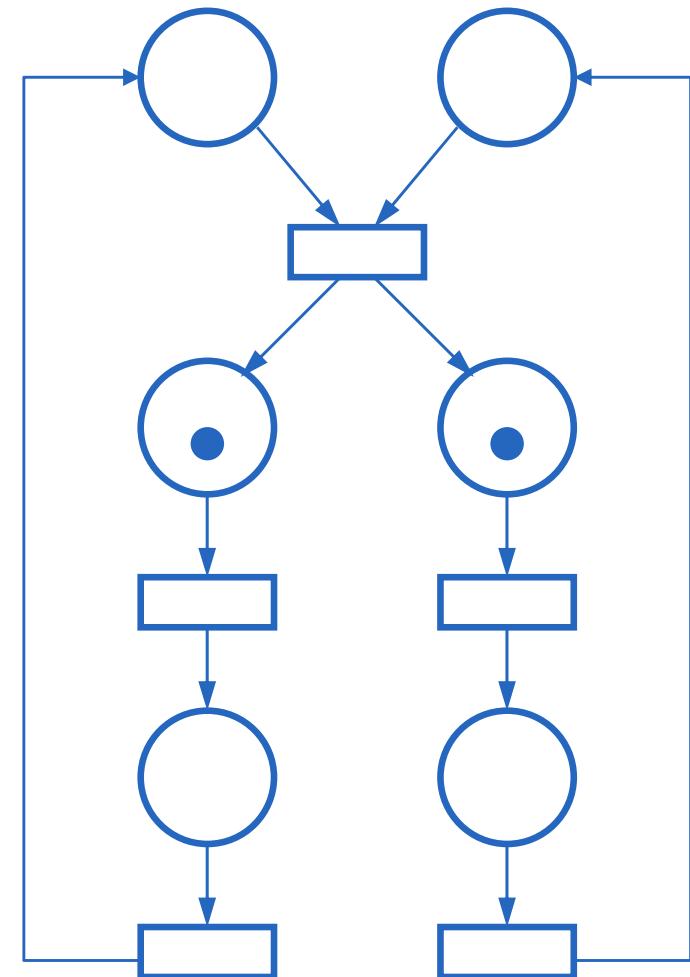


Exemple de comportement

- Deux chariots faisant un aller et retour avec départ simultané de la gauche

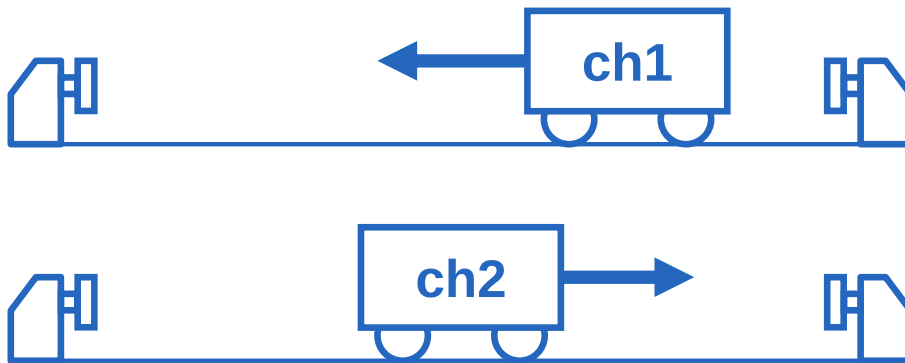


Réseau de Petri ordinaire

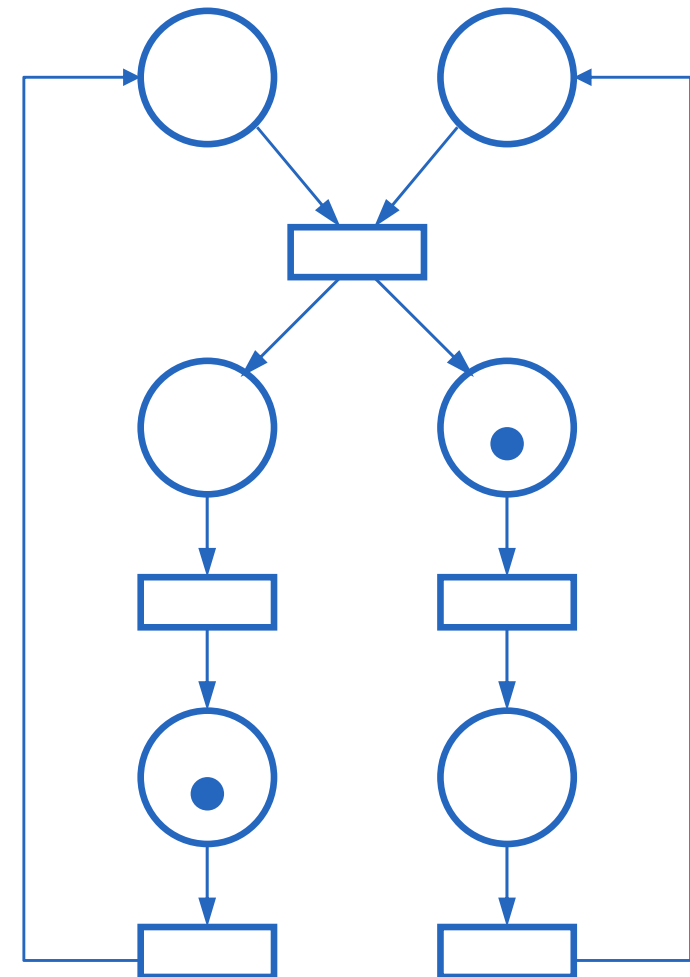


Exemple de comportement

- Deux chariots faisant un aller et retour avec départ simultané de la gauche

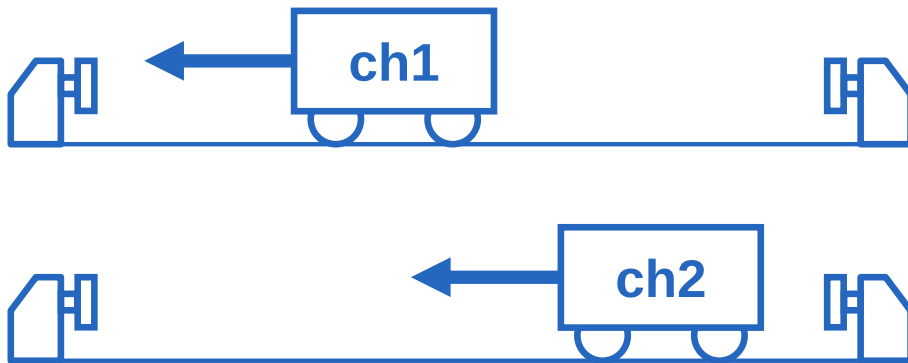


Réseau de Petri ordinaire

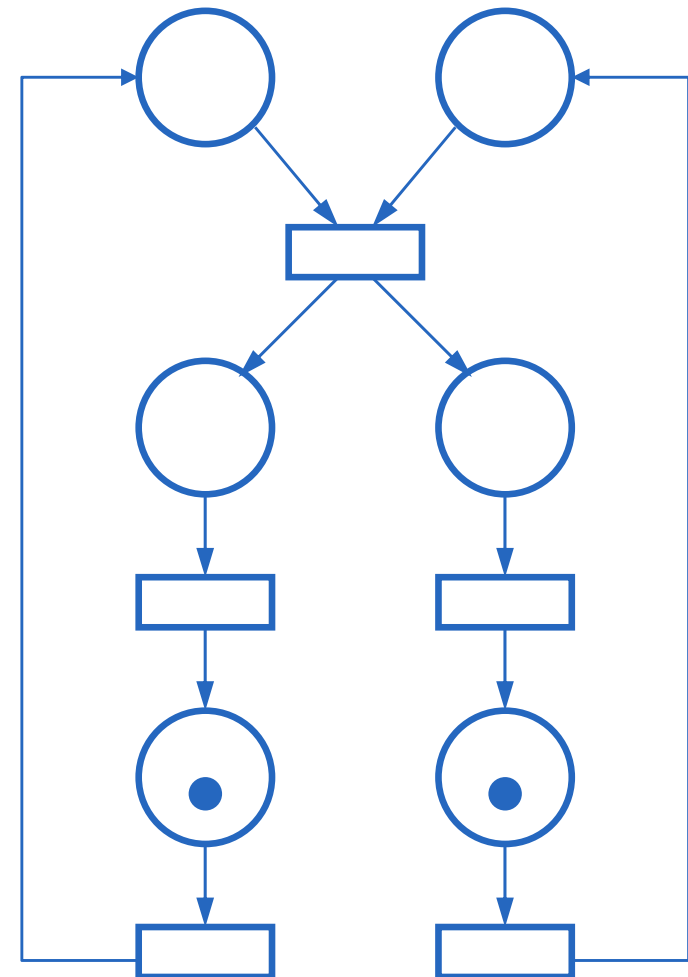


Exemple de comportement

- Deux chariots faisant un aller et retour avec départ simultané de la gauche

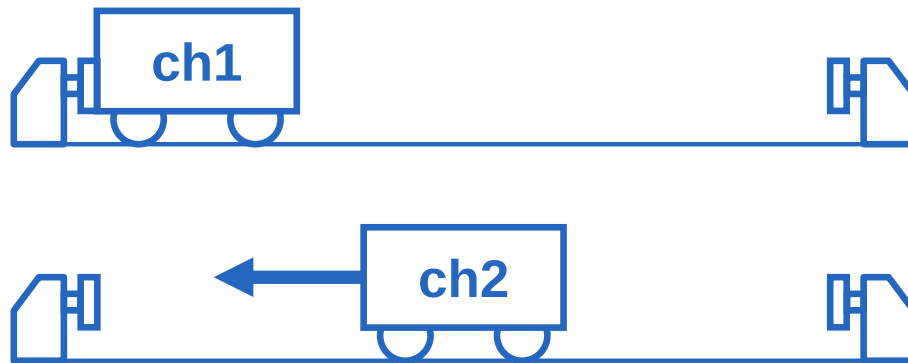


Réseau de Petri ordinaire

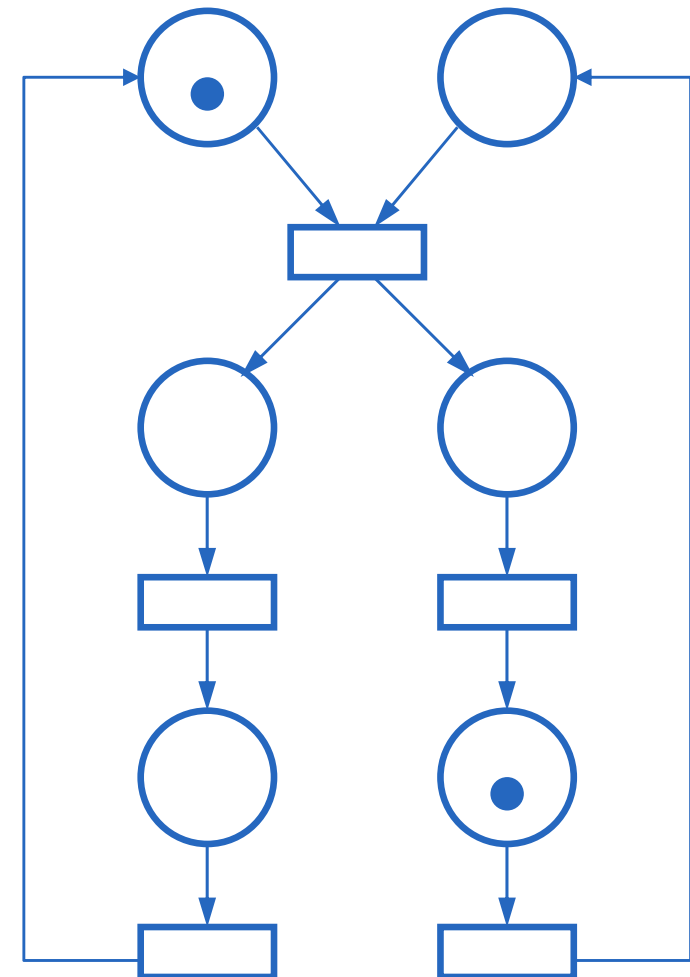


Exemple de comportement

- Deux chariots faisant un aller et retour avec départ simultané de la gauche

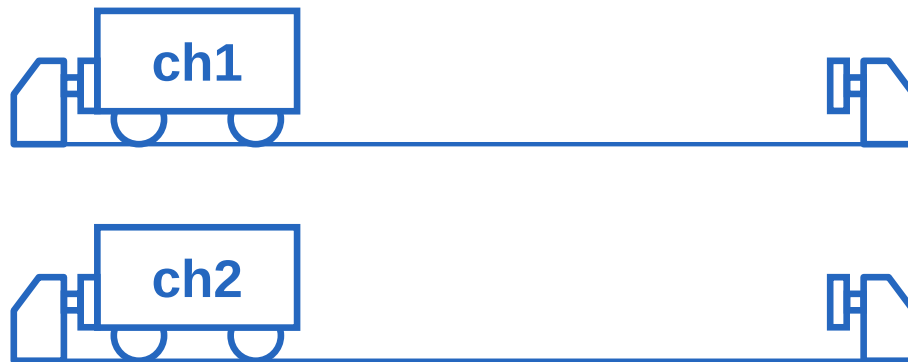


Réseau de Petri ordinaire

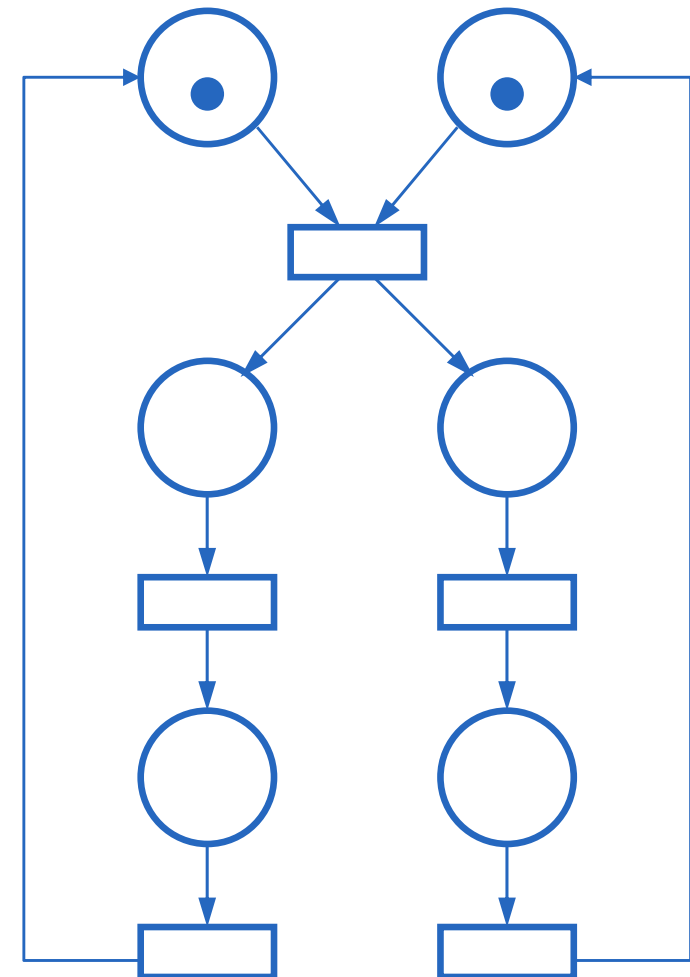


Exemple de comportement

- Deux chariots faisant un aller et retour avec départ simultané de la gauche



Réseau de Petri ordinaire



UNIT

- 10

- CPN Tools (Version 2.0.0 - March 2006)

Tool box
 Help
 Options
 ch_ordinaire.cpn
 Step: 0
 Time: 0
 Options
 History
 Declarations
 Standard declarations
 colset UNIT = unit;
 colset INT = int;
 colset BOOL = bool;
 colset STRING = string;
 Monitors
 page

Binder 0
 page

None

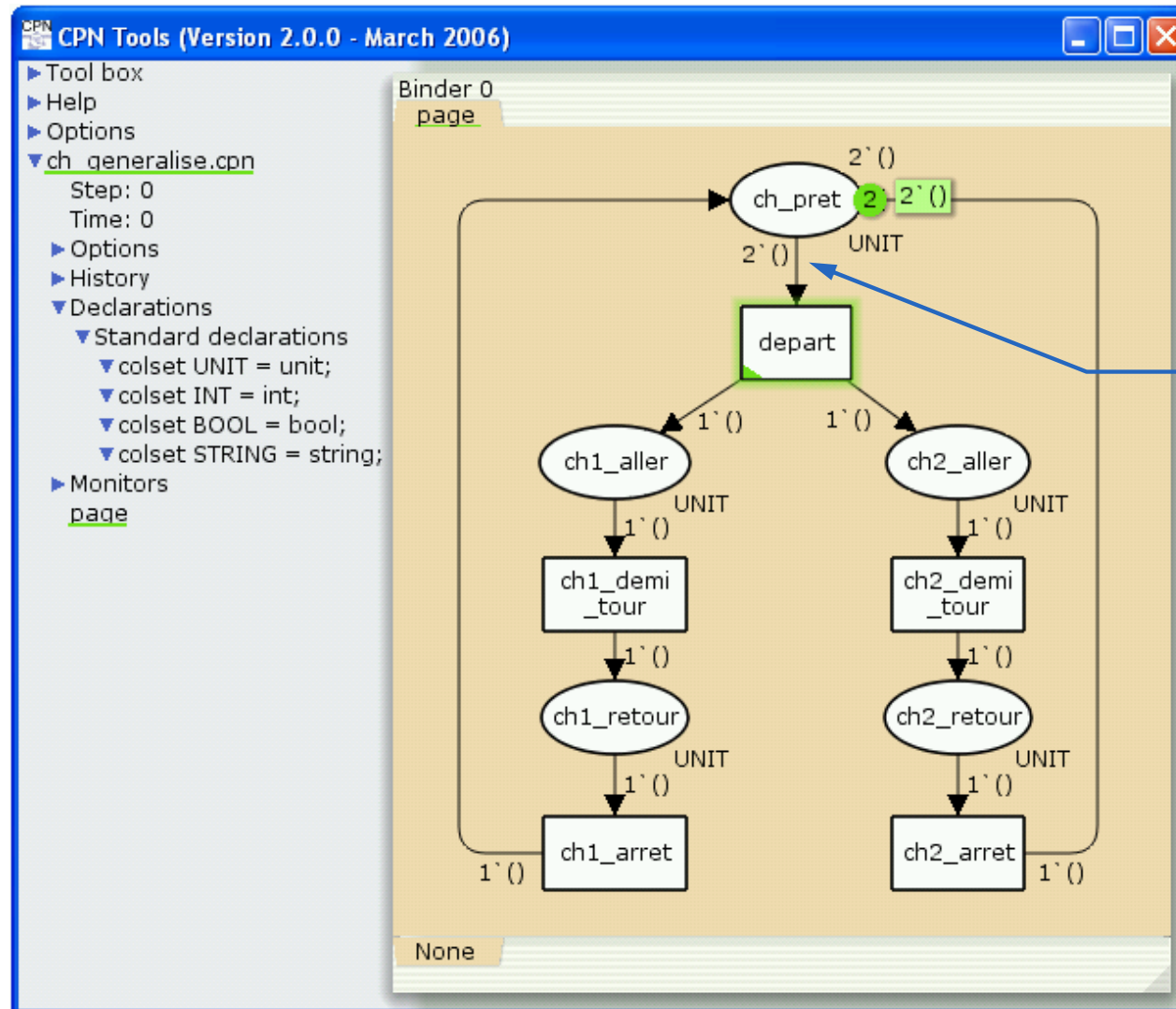
Petri net diagram showing places (ovals) and transitions (rectangles):

 - Places: ch1_pret, ch2_pret, ch1_aller, ch2_aller, ch1_retour, ch2_retour, ch1_arret, ch2_arret.
 - Transitions: depart, ch1_demi_tour, ch2_demi_tour.

Diagram details:

 - ch1_pret and ch2_pret are initial places, each containing 1 token (green circle) and labeled 1'().
 - Transitions depart, ch1_demi_tour, and ch2_demi_tour are labeled UNIT.
 - Places ch1_aller and ch2_aller are labeled UNIT.
 - Places ch1_retour and ch2_retour are labeled UNIT.
 - Places ch1_arret and ch2_arret are labeled UNIT.
 - Transitions ch1_demi_tour and ch2_demi_tour are labeled UNIT.
 - Places ch1_retour and ch2_retour are labeled UNIT.
 - Places ch1_arret and ch2_arret are labeled UNIT.
 - Transitions depart, ch1_demi_tour, and ch2_demi_tour are labeled UNIT.

Modèle CPN tools (RdP généralisé)



Pondération
2 jetons ()

Modèle CPN tools : couleurs, jetons et variables

■ CH

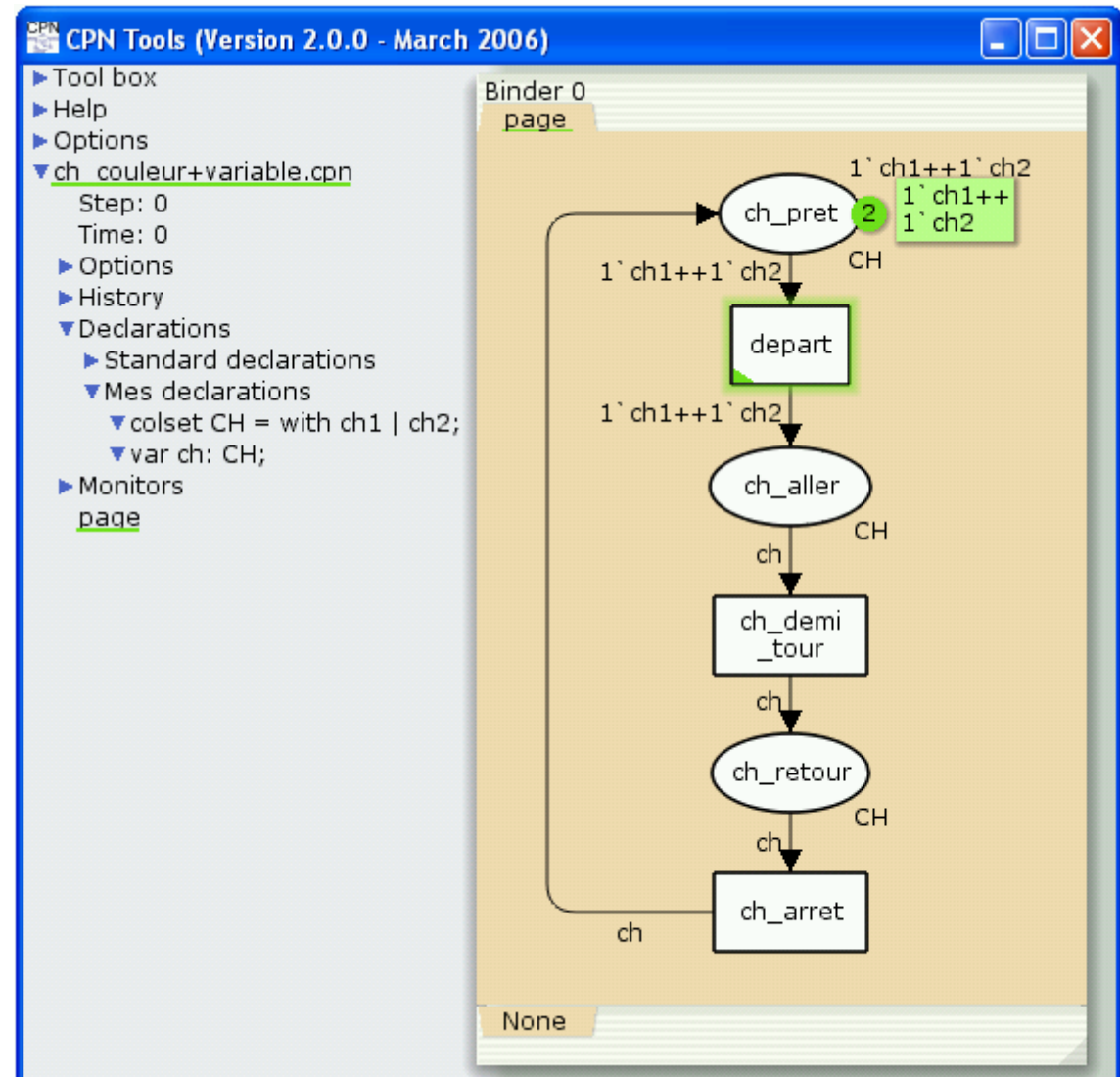
- nouvelle couleur définie comme un type énuméré
color CH : with ch1 | ch2 ;
(modélise les chariots)

■ ch

- variable de type CH
var ch : CH;

■ 1`ch1 ++ 1`ch2

- « multi-set » de 2 éléments,
1 jeton ch1 et un jeton ch2



Modèle CPN tools : couleurs complexes

■ ETAT

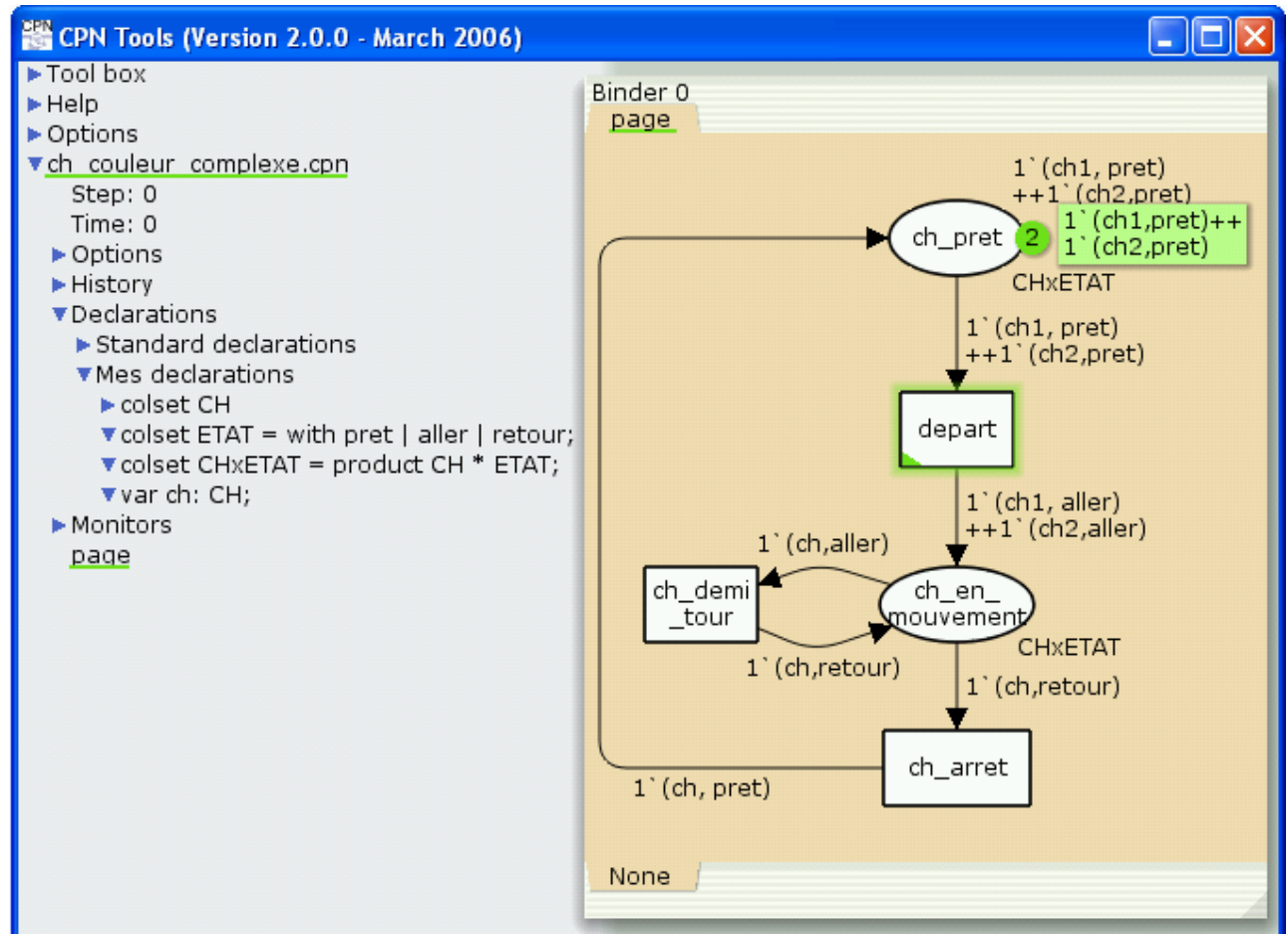
- nouvelle couleur colset ETAT :
with pret | aller | retour ;
(modélise les états possibles d'un chariot)

■ CHxETAT

- nouvelle couleur définie comme un produit cartésien
colset CHxETAT :
product CH * ETAT ;

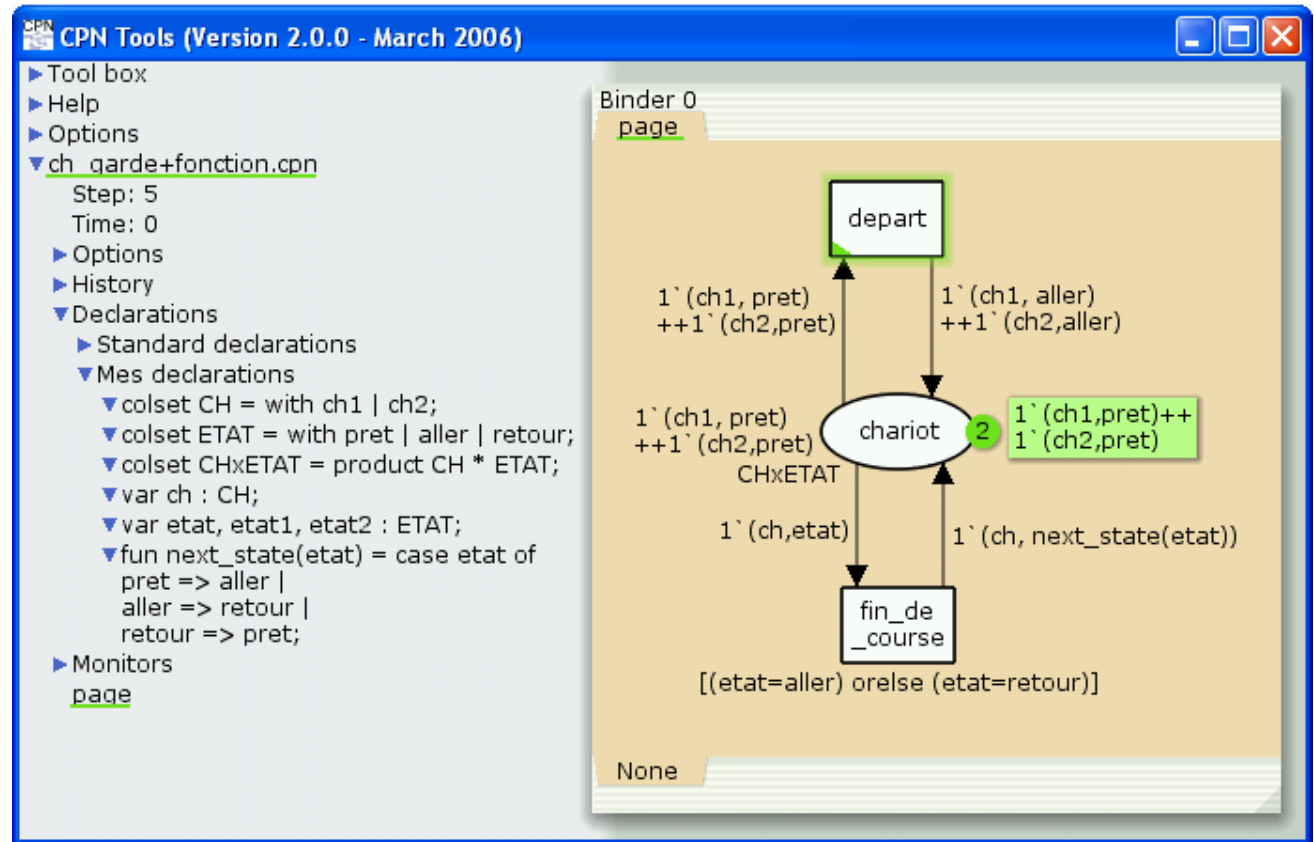
■ 1'(ch1, aller)

- jeton de couleur CHxETAT qui modélise le chariot ch1 à l'aller



Modèle CPN tools : garde et fonction

- [(etat=aller) orelse (etat=retour)]
 - garde : condition supplémentaire pour le franchissement de transitions
- next_state(etat)
 - fun next_state(etat) =
case etat of
pret => aller |
aller => retour |
retour => pret ;
 - fonction dont le paramètre d'entrée est un jeton de couleur ETAT et qui retour un jeton de couleur ETAT

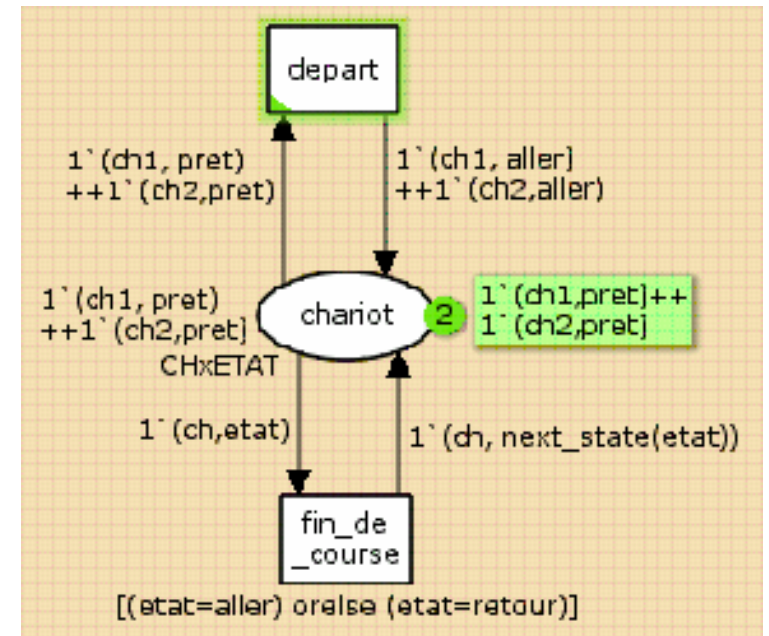
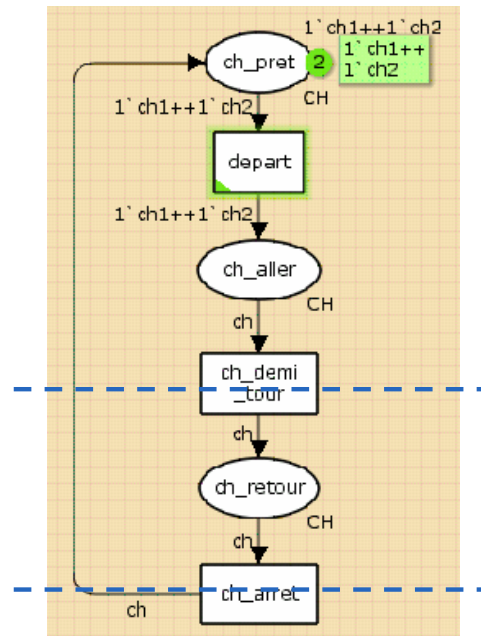
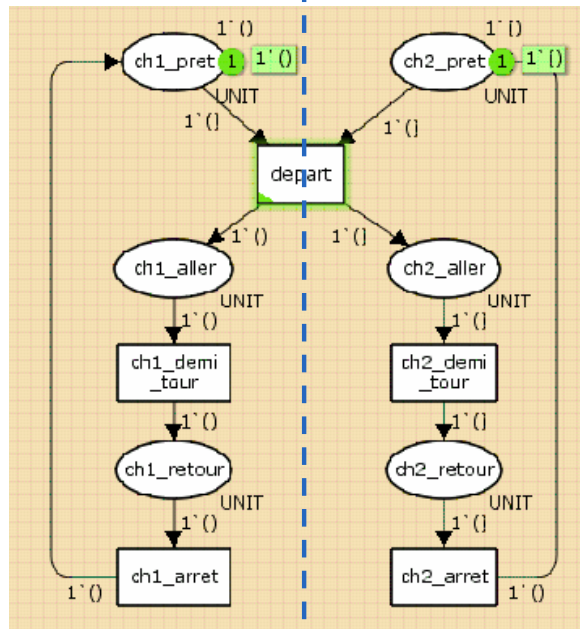


Modèle CPN tools : modèle complètement replié

du comportement des deux chariots

Repliage
sur les chariots

Repliage
sur l'état des chariots



Modèle CPN tools : RdP temporisé (déplié)

■ UNIT

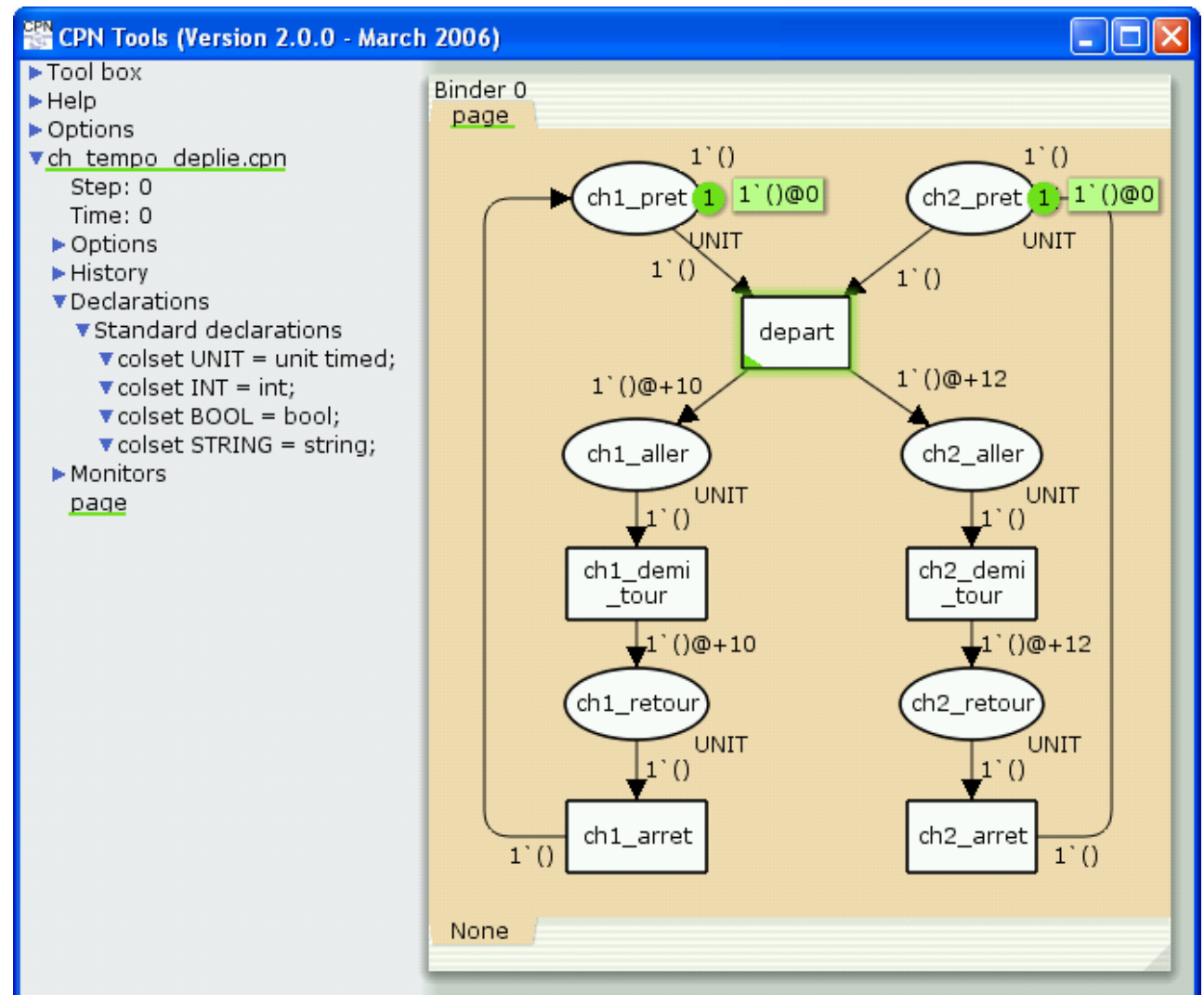
- couleur enrichie de l'attribut « timed »

■ $1'()@0$

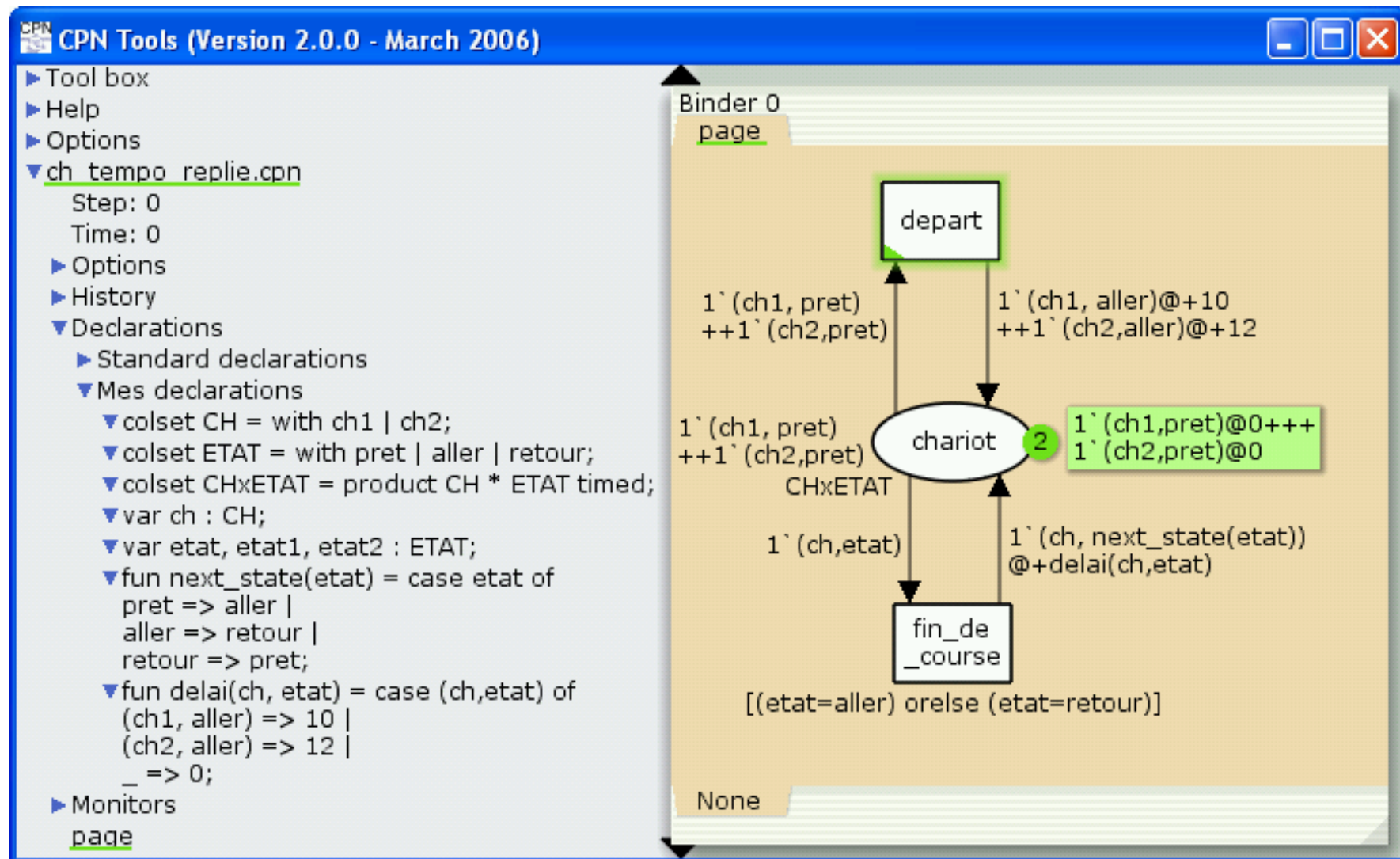
- 1 jeton « () » qui sera disponible pour une transition à la date 0 (pas de réservation)

■ $1'()@+12$

- l'attribut date du jeton produit sera $\text{current_date} + 12$



Modèle CPN tools : RdP temporisé (replié)



Pour modéliser une action deux options : une place ou une transition

