Modélisation et analyse des systèmes à événements discrets à l'aide des réseaux de Petri récursifs

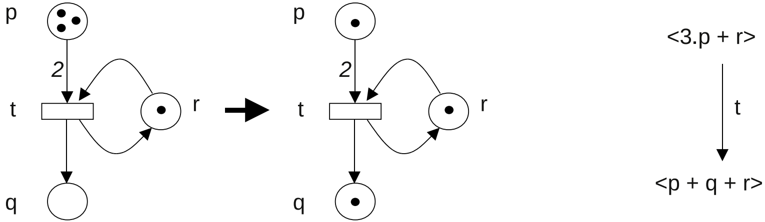
Serge Haddad

LSV CNRS & ENS Cachan & INRIA MEXICO

Séminaire Groupe francophone RdP (GDR MACS)
le 6 mai 2011 à Villetaneuse
(en collaboration avec Denis Poitrenaud, LIP6)

Plan

Les réseaux de Petri



Syntaxe

- Places
- Transitions
- Arcs valués
- Jetons (marques)

Fonctionnement

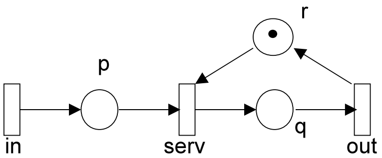
- Franchissabilité
- Franchissement

Sémantique

- Ensemble d'accessibilité
- Graphe d'accessibilité
- Système de transitions
- Structure de Kripke
- Séquences de franchissement

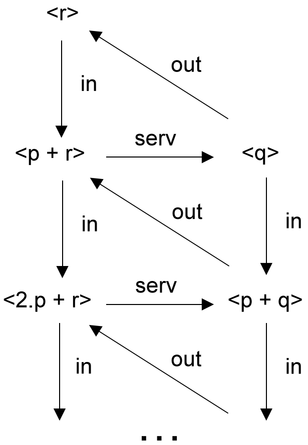
Un modèle de systèmes infinis

Une file d'attente



Son invariant : $m(q) + m(r) = 1$

Son graphe d'accessibilité



Vérification de propriétés standard

Accessibilité d'un marquage

- $\{(R, m_0), m\} : \exists ?\ \sigma\ m_0 \xrightarrow{\sigma}_R m$
- Décidable *(mais algorithme non primitif récursif)*

Caractère borné d'un réseau

- $(R, m_0) : \exists ?B\ \forall p \in P\ \forall m\ m_0 \rightarrow_R m \Rightarrow m(p) \leq B$
- Décidable et EXPSPACE-complet

Taille du graphe d'accessibilité d'un réseau borné

- Fonction non primitive récursive
- Intérêt des méthodes de réduction a priori
  (symétrie, composition, ordre partiel, etc.)

Blocage, couverture, état d'accueil, vivacité, ... : toutes décidables

Vérification de propriétés spécifiques

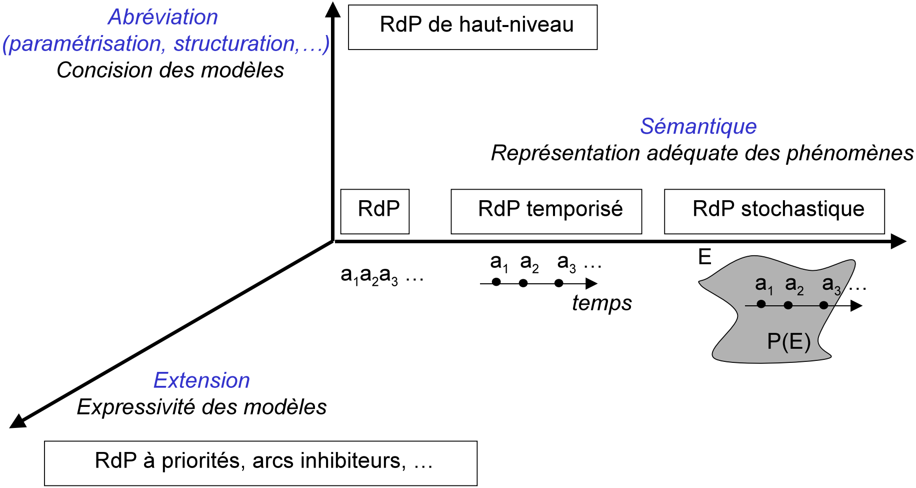
Vérification de formules de logique temporelle

Logique	propositionnelle	événementielle
CTL	indécidable	indécidable
LTL	indécidable	décidable

Equivalence de modèles

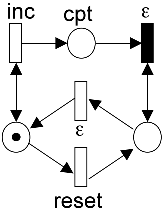
Bisimulation	RdP
RdP	indécidable
Automate	décidable

Variantes de réseaux de Petri

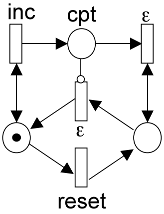


Extensions usuelles de réseaux de Petri

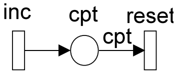
Priorité



Arcs inhibiteurs



Auto-modifiant



Expressivité équivalente à celle des machines de Turing

Indécidabilité de l'accessibilité

Indécidabilité des autres propriétés standard . . .

(excepté quelques unes moyennant des restrictions significatives)

L'état du réseau est toujours un vecteur d'entiers

Plan

Les réseaux de Petri récursifs séquentiels

(version simplifiée)

Un réseau de Petri récursif séquentiel est un RdP étendu par :

- $T = T_{el} \uplus T_{ab}$, une partition en transitions élémentaires et abstraites
- $Init : T_{ab} \rightarrow \mathbb{N}^P$, un marquage initial pour chaque transition abstraite
- $End : \mathbb{N}^P \rightarrow \{\mathbf{true}, \mathbf{false}\}$, un prédicat sur les marquages exprimé par une *formule de Presburger*

Une formule de Presburger

- a pour constantes les entiers,
- a pour variables libres les places du réseau,
- dont les termes sont construits à l'aide de l'addition
- et les formules sont soit des comparaisons de termes, soit des négations, des conjonctions ou des quantifications de formules

Exemples

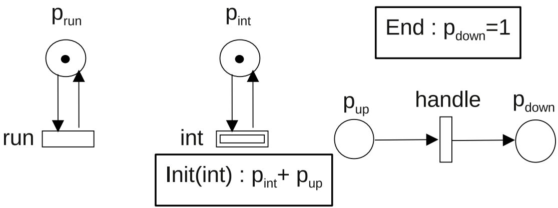
Le marquage de p est supérieur à deux fois le marquage de q : $p > 2q(> q + q)$

Le marquage de p est impair : $\exists x \; p = 2x + 1$

Exemple de RdPR séquentiel

Un mécanisme d'interruption simple

- Une tâche s'exécute en permanence.
- Elle peut être interrompue et le pilote prend la main, pour traiter l'interruption puis rendre la main à la tâche.
- Le pilote peut aussi être interrompu, etc.

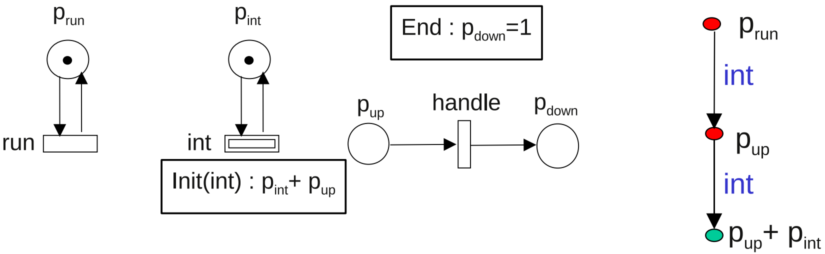


Observation : très souvent un RdPR a plusieurs composantes connexes.

Sémantique d'un RdPR séquentiel

Un état d'un RdPR séquentiel est :

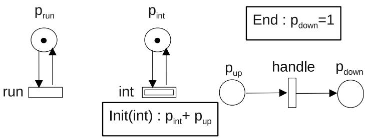
- une liste de marquages *(instances de réseau)*
- dont les arcs sont étiquetés par des transitions abstraites

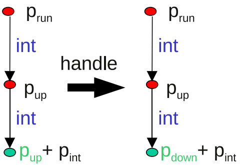


Un pas élementaire

Quelque soit la nature du pas,
celui-ci s'exécute dans le dernière instance de réseau de la liste.

Un pas élémentaire est un franchissement d'une transition élementaire et modifie le marquage de la même façon que dans un RdP.

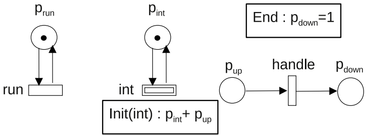


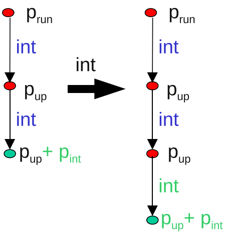


Un pas abstrait

Un pas abstrait est un franchissement d'une transition abstraite (disons t).

- Il consomme les jetons des places entrée de t
- Il étend la liste avec le marquage $Init(t)$ et étiquette l'arc correspondant par t.

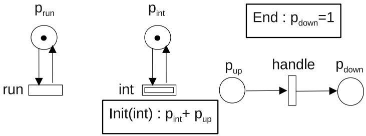


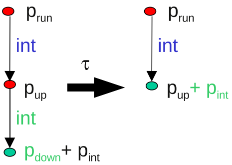


Un pas de fin

Un pas de fin est un franchissable lorsque le marquage courant satisfait End. Son franchissement

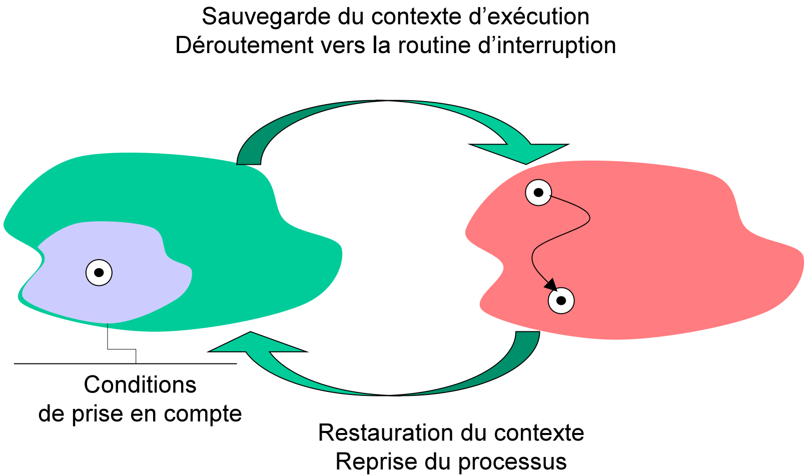
- supprime le marquage courant
- et, dans le nouveau marquage courant, produit les jetons dans les places sorties de la transition qui pointait sur l'ancien marquage courant.
- Si la liste devient vide alors le RdPR est mort.



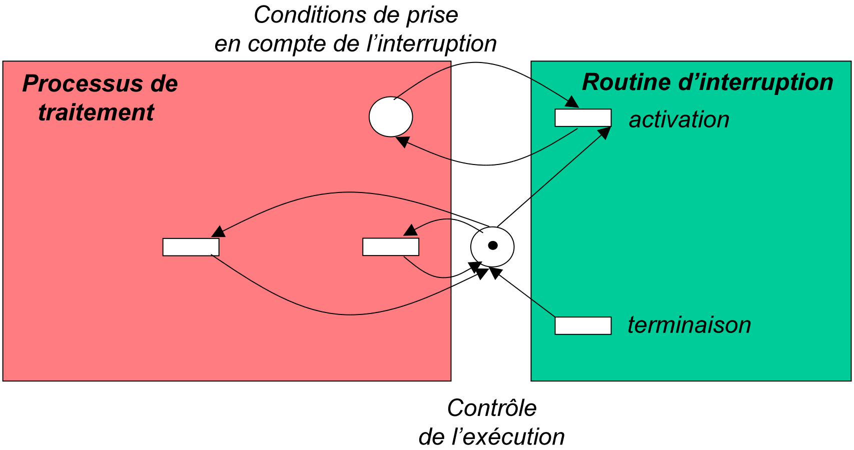


Plan

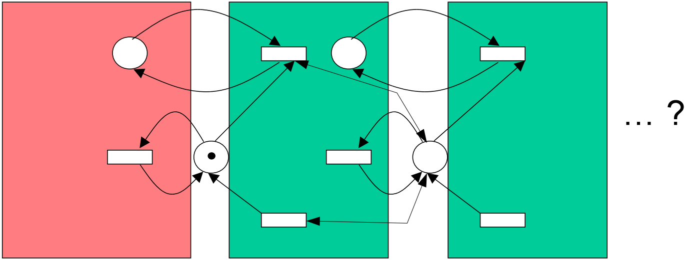
Mécanisme d’interruption



Modélisation RdP d'une interruption



Modélisation d'interruptions multiples



Impossibilité théorique

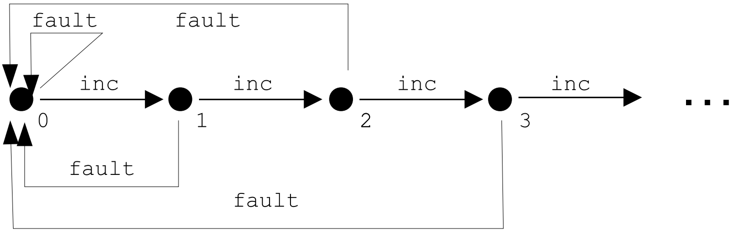
Un système inactif avec deux interruptions a et b.
L'activation et la terminaison sont étiquetées par le type d'interruptions.

Le langage associé est le langage des palindromes qui n'est pas un langage de RdP.

Abstraction d'une faute

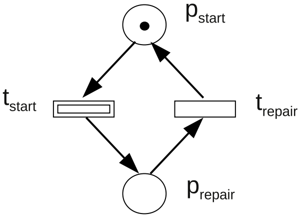
Un compteur d'événements

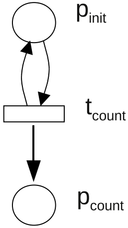
- A chaque événement, incrémentation du compteur
- Lors d'une faute, le compteur est réinitialisé



Ce graphe ne peut être le graphe d'accessibilité d'un RdP ...
même avec des transitions inobservables.

Modélisation RdPR d'une faute

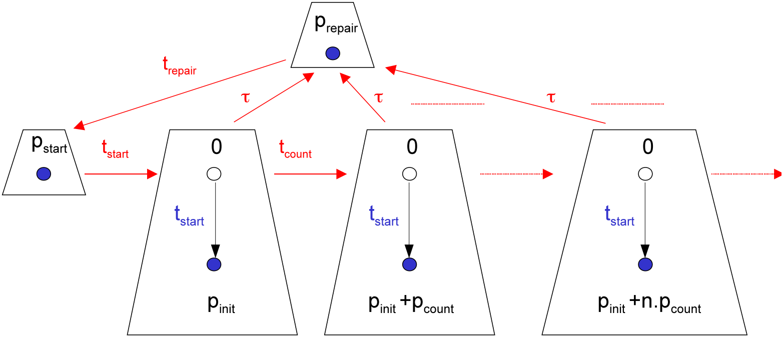




$\boxed{\text{Init}(t_{start}) : p_{init}}$

$\boxed{\text{End} : p_{init}=1}$

Graphe d'accessibilité de la modélisation

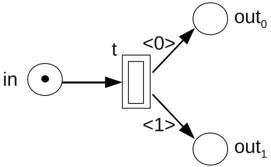


Les réseaux de Petri récursifs séquentiels

(version standard)

Nécessité de conditions de terminaison multiples

- I, un ensemble fini d'indices de terminaison
- $End = \{End_i\}_{i \in I}$, une famille de formules de Presburger
- L'incidence avant d'une transition abstraite dépend de la terminaison
  $Post : P \times T_{el} \cup (T_{ab} \times I) \to \mathbb{N}$

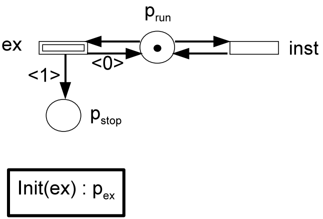


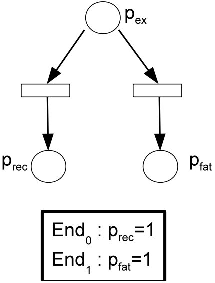
Quand un réseau créé par t se termine, selon la condition de terminaison satisfaite il produit un jeton dans out_0 ou dans out_1.

Modélisation RdPR d'une exception

Une exception est provoquée par le flot d'exécution du processus et nécessite l'intervention du système.

A l'issue du traitement de l'exception, le processus peut reprendre son exécution (e.g. allocation mémoire) ou se terminer (e.g. division par zéro).





Plan

Pouvoir d'expression des RdPR

Comportement du système en terme de langages de mots finis
(le plus commode)

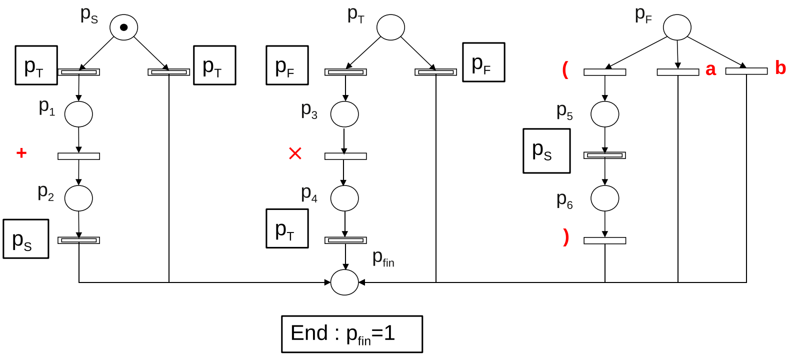
- Langages des réseaux de Petri : nécessite l'étiquetage des transitions (avec transitions inobservables) et la spécification d'un marquage final.
- Langages algébriques engendrés par une grammaire algébrique ou de manière équivalente par un automate à pile.

La famille des langages des RdPR
inclut strictement l'union de ces deux familles de langages

Simulation d'une grammaire algébrique

Une grammaire pour les expressions arithmétiques

$$S ::= T \times S \mid T \quad T ::= F + T \mid T \quad F ::= (S) \mid a \mid b$$

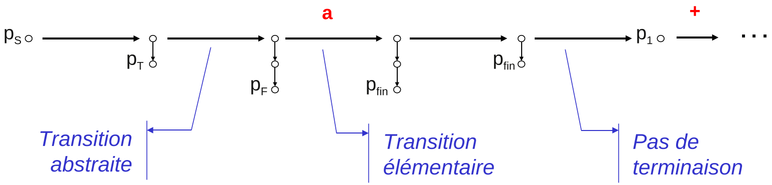


Simulation en action

Séquence correspondant à la dérivation la plus à gauche de $a + b$

$$S \rightarrow T + S \rightarrow F + S \rightarrow a + S \xrightarrow{a} +S \xrightarrow{+} S \dots$$

Séquence correspondante dans le RdPR



RdPR séquentiels et automates finis

Existence d'une simulation *faible* du produit synchronisé d'un RdPRS $\mathcal{R}$ muni d'un état final et d'un automate fini $\mathcal{A}$ par un RdPRS $\mathcal{R}'$

- $\mathcal{R}'$ inclut les comportements du produit synchronisé.
- Les comportements additionnels de $\mathcal{R}'$ ne permettent pas d'atteindre l'état final.

La famille de langages des RdPR séquentiels est close par intersection avec les langages réguliers

Décidabilité des propriétés standard

Accessibilité
Propriétés de sûreté telles que la violation d'une exclusion mutuelle

Caractère borné des places
Gestion de ressources

Caractère fini du système
Applicabilité du model checking pour les systèmes finis

Récursivité du langage
Existence de séquences fixées d'événements (pour le test)

Ces propriétés restent décidables
dans le cadre général des réseaux de Petri récursifs

Décidabilité des propriétés spécifiques

Vérification d'une formule de logique temporelle linéaire événementielle
Dans le cas de RdPR *a priori* k-bornés, le problème est PSPACE-complet.

Bisimulation d'un RdPR séquentiel et d'un automate fini
avec une restriction :

Les termes des formules de Presburger sont de la forme

$$\sum_{p \in P} a_p p$$

avec $a_p \geq 0$

Plan

Caractéristiques supplémentaires

Du point de vue sémantique

- Un pas d'un RdPR peut avoir lieu dans n'importe quelle instance de réseau.
- Par conséquent, l'état du réseau est un arbre orienté de marquages et non plus une liste.
- Un pas de terminaison dans un instance de réseau détruit le sous-arbre enraciné en cette instance.

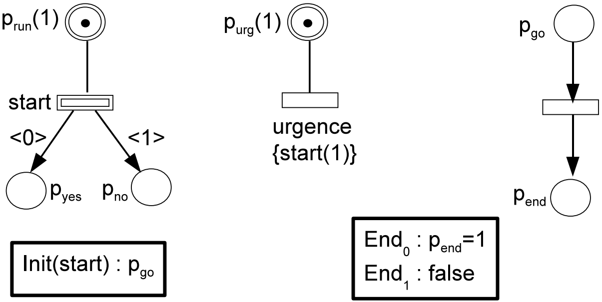
Du point de vue syntaxique

- Chaque place a une capacité (éventuellement infinie). Q est l'ensemble des places bornées.
- Les transitions peuvent avoir des arcs test connectés aux places de Q (utile pour les transitions abstraites).
- Le marquage initial d'une transition abstraite peut dépendre du marquage des places de Q dans l'instance où le franchissement a lieu.
- Le franchissement d'une transition élémentaire peut détruire des sous-arbres créés par certaines transitions abstraites (avec spécification de la condition de terminaison).

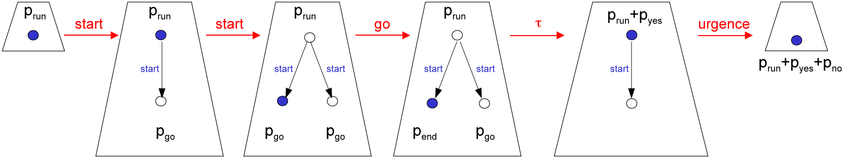
Exemple de RdPR

Un mécanisme d'urgence

- Le système lance des tâches non critiques à la demande.
- En cas d'urgence, le système détruit toutes les tâches non critiques pour se consacrer à la tâche urgente.



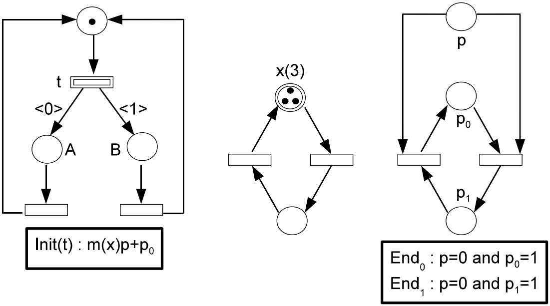
Une séquence d'accessibilité de la modélisation



Un exemple plus élaboré

Un appel de procédure à distance

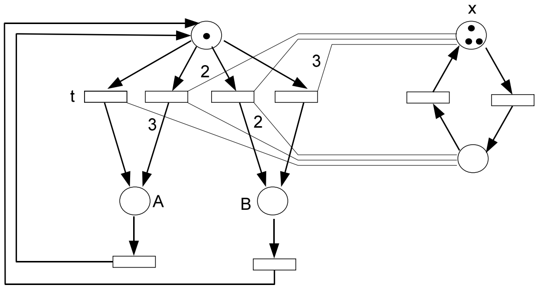
- Un processus met à jour une variable numérique bornée par des incrémentations et des décrémentations.
- Il peut appeler une procédure distante qui calcule la parité de la variable.
- Le calcul de la parité se fait parallèlement aux mises à jour de la variable *(appel par valeur)*.



Modélisation par réseau de Petri

Problèmes ouverts

- Existe-t-il une modélisation dont la structure est indépendante de la borne de x ?
- Existe-t-il une modélisation dont la taille varie en fonction du logarithme de la borne de x ?



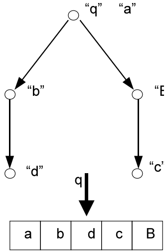
Simulation d'une machine de Turing

Produit synchronisé d'un RdPR et d'un automate

Dans le RdPR,

- L'état de la machine est codé dans la racine de l'arbre
- Les deux parties de la bande sont codées par les deux branches de l'arbre
- L'automate synchronise les franchissements dans la racine et les deux feuilles

Un état courant



Indécidabilité uniquement due à l'activité concurrente dans les sommets *(le réseau est 1-borné)*

Conséquences

La famille des langages de RdPR n'est pas close
par intersection avec les langages réguliers.

Le model checking de LTL événementiel est indécidable pour les RdPR.

Les RdPR sont strictement plus expressifs que les RdPR séquentiels.

Conclusion

Compromis entre pouvoir d'expression et décidabilité

Adéquat pour la modélisation de systèmes à événements discrets

Application à la conception de systèmes dont l'organisation est dynamique
(e.g. système multi-agents)

Nécessité d'un outil et d'une méthode de conception

Quelques références bibliographiques

S. Haddad and D. Poitrenaud
Recursive Petri Nets - Theory and Application to Discrete Event Systems
Acta Informatica 44(7-8), pages 463-508, 2007

S. Haddad and D. Poitrenaud
Checking Linear Temporal Formulas on Sequential Recursive Petri Nets
In TIME'01, pages 198-205. IEEE Computer Society Press, 2001

S. Haddad and D. Poitrenaud
Modelling and Analyzing Systems with Recursive Petri Nets
In WODES'00, pages 449-458. Kluwer Academic Publishers, 2000

S. Haddad and D. Poitrenaud
Theoretical Aspects of Recursive Petri Nets
In ICATPN'99, LNCS 1639, pages 228-247. Springer, 1999

A. El Fallah Seghrouchni and S. Haddad
A Recursive Model for Distributed Planning
In ICMAS'96, pages 307-314. AAAI Press, 1996