



PEPS 2006

Leonardo Brenner, Brigitte Plateau, Ihab Sbeity

Laboratoire ID-IMAG, Grenoble, France.
Projet Sure-Paths





1 Introduction

2 Grammaire/Parser

3 Découpage





Le Logiciel PEPS?

Qu'est-ce que c'est le PEPS?

PEPS est un logiciel pour l'évaluation de performance basé sur le formalisme de Réseaux d'Automates Stochastiques (SAN) et algèbre tensorielle.

- algorithmes pour algèbre tensorielle;
- techniques d'aggregation;
- méthodes des calculs de mesures stationnaires et transitoires, etc.





PEPS Version 2003

Il a deux problèmes:

- La grammaire n'est pas très puissante;
- La structure du logiciel. Tous les modules (classes) sont mélangé ensemble.





PEPS 2006 : nouveautés

On propose:

- Nouvelle grammaire/parser plus sophistiquée, qui permet la description d'autres formes répliquées;
- Découpage en modules indépendants.





Grammaire/Parser

Quoi de neuf?

On a proposé une grammaire plus puissante qui permet la modélisation de systèmes avec de composant répliqués en plusieurs niveaux (automates, états, transitions, événements).

Pour ça, on a proposé:

- Fonctions plus sophistiquées qui permettent de définir une notion de voisinage entre les automates, les états, ... ;
- Domaines de répliqués plus complexes;
- Réplication d'événements et de transitions au-delà de la réplication d'automates et d'états déjà existante.





Partage de ressources

Le modèle

On a un nombre R de ressources qui sont partagés parmi P processus.

Modèle SAN

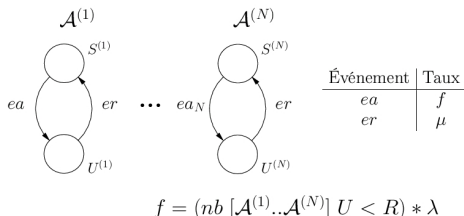
Il faut:

- créer un automates pour chaque ressource;
- chaque ressource a son propre ensemble d'événements



Partage de ressources

PEPS 2003



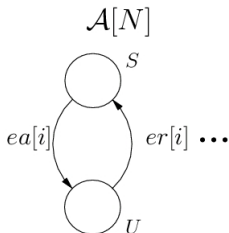
Mais, c'était incorrect

On utilisait le même événement local pour tous les automates.
C'est incompatible avec la définition de SAN.



Partage de ressources

PEPS 2006

 $N = [1..10]$ $i = \text{current aut}$ $f = (nb [\mathcal{A}[N]] U < R) * \lambda$

Événement	Taux
$ea[N]$	f
$er[N]$	μ

Maintenant, chaque automates a ses propres événements locaux.





File d'attente

Le modèle

Files d'attente avec possibilité de perte.

Modèle SAN

Il faut:

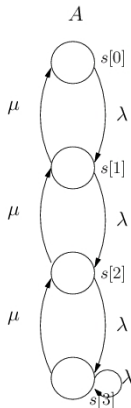
- créer un état pour chaque état de la file;
- le dernier état ne se comporte pas de la même façon que les autres états.





File d'attente

Comment on fait?

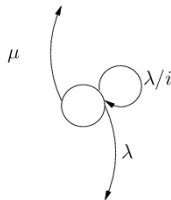


$$n = 3$$

$$N = [0..n]$$

$$i = \text{current stt} == n$$

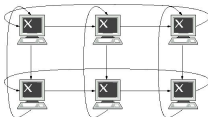
Événement	Taux
λ	10
μ	5



Réseaux

Le modèle

Chaque noeud du réseaux se communique avec ses voisins.



Modèle SAN

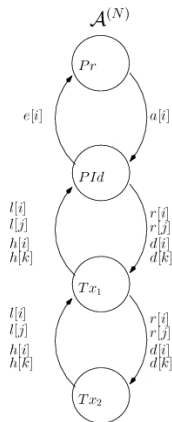
Il faut:

- créer un automate pour chaque noeud;
- chaque noeud a son ensemble d'événements qui doivent se synchroniser avec ses voisins.



Torus

Comment on fait?



$$n = 2$$

$$nn = (n * 2) - 1$$

$$N = [0..nn]$$

$i = \text{current aut}$

$$j = ((\text{current aut div } n) * n) + ((\text{current aut} + 2) \bmod n)$$

$$k = ((n - ((\text{current aut div } n) * n) + (\text{current aut mod } n))$$

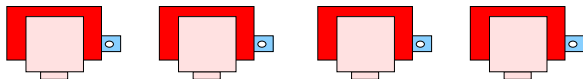
Événement	Taux
$e[N]$	μ
$a[N]$	λ
$r[N]$	5
$d[N]$	10
$l[N]$	2
$h[N]$	3



Chaîne de production

Le modèle

On a une chaîne de production avec N robots. Tous les robots ont un comportement identique sauf le premier et le dernier.



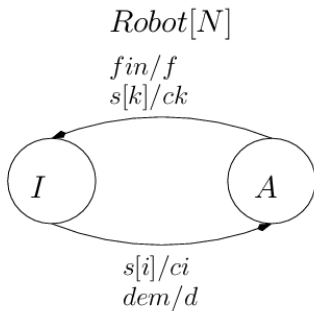
Modèle SAN

Il faut:

- créer un automate pour chaque robot;
- chaque robots a son ensemble d'événements qui doivent synchroniser avec ses voisins;
- le premier et le dernier ont un comportement différent.

Chaîne de production

Comment on fait?



$N = [1..10]$ $NE = [1..9]$
i = *current aut* - 1
k = *current aut*
ci = *current aut!* = 1
ck = *current aut!* = 10
f = *current aut* == 10
d = *current aut* == 1

Événement	Taux
<i>dem</i>	λ
<i>fin</i>	μ
<i>s</i> [NE]	5



Découpage

PEPS 2003

- Un seul programme;
- Composé de plus de 35 classes;
- Difficile à maintenir

PEPS 2006

- Plusieurs programmes (modules);
- Chaque modules sert à un travail spécifique;
- Mis à jour et insertion de nouveaux modules plus facile.

