Ensemble d'outils pour l'évaluation de performance dans CADP

Christophe Joubert

VASY'03

Saint Pierre de Chartreuse (Isère)

10 Juin – 13 Juin 2003

INRIA Rhône-Alpes

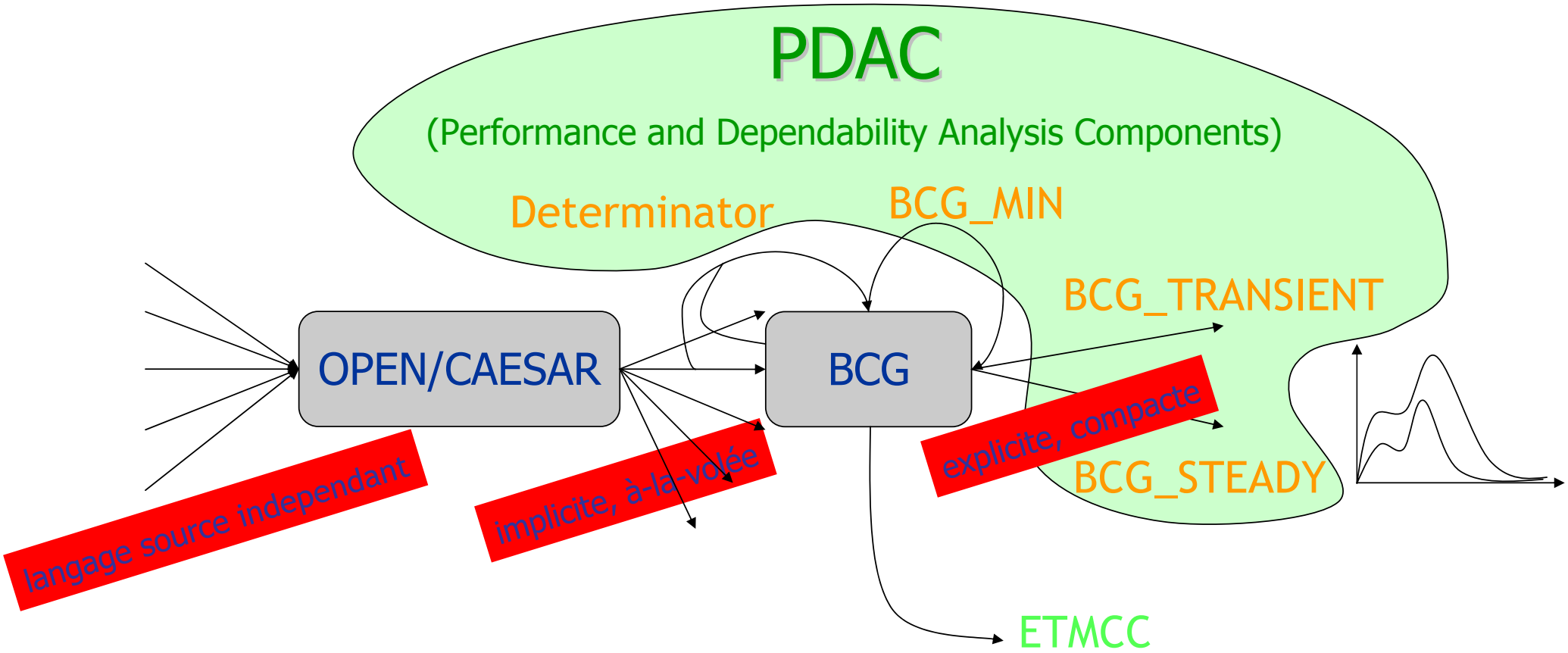
Projet VASY

(contributions de H. Garavel et H. Hermanns)

Plan

1. Quelles réalisations ?
2. Quelles applications ?
3. Quel contexte théorique ?
4. Quelles perspectives ?

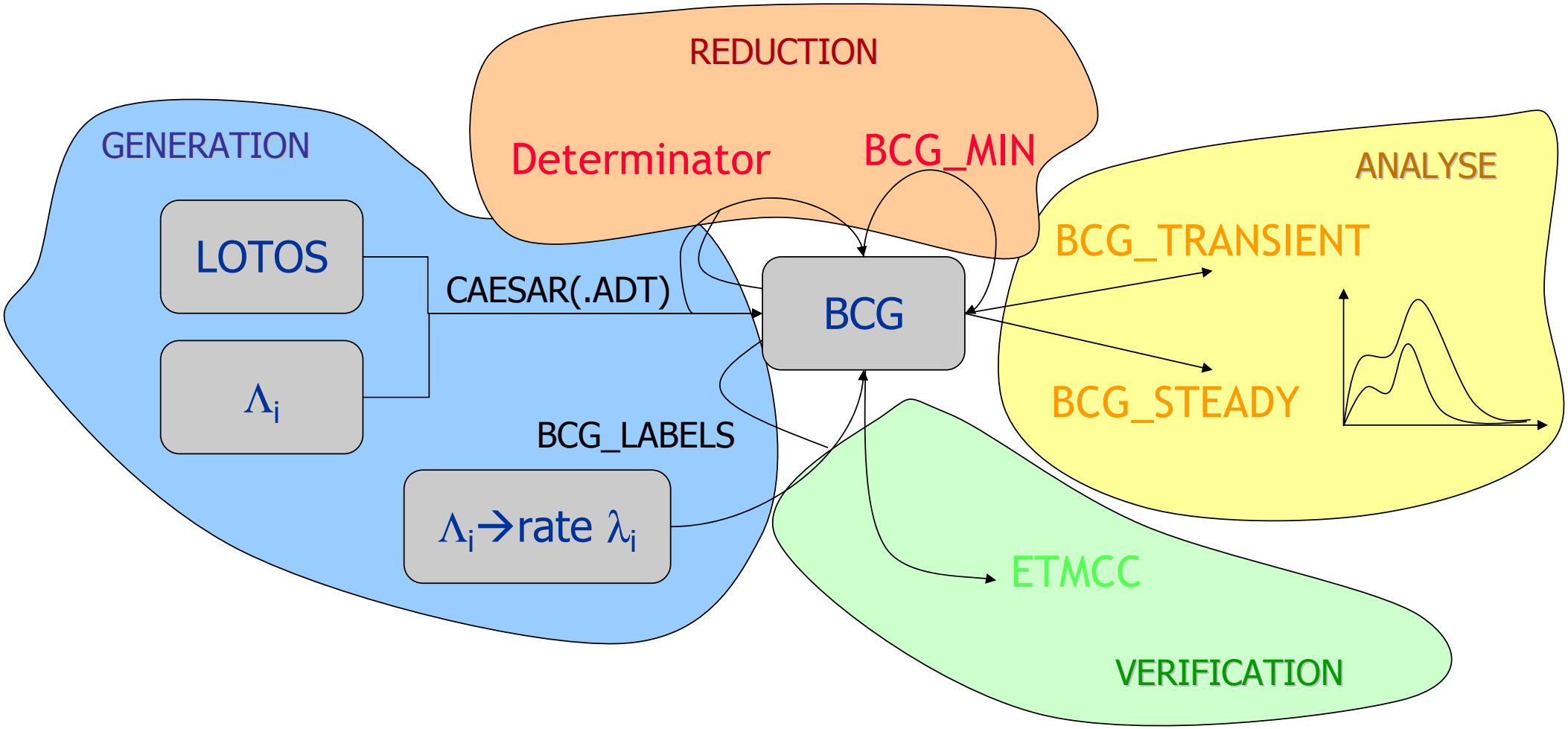
Réalisations : ensemble des outils



Réalisations : simulateurs et réducteurs

- Moyen de spécifier les charactéristiques de performance et de dépendance
- Algorithmes de construction des modèles de performance et de dépendabilité
  - Réduction à-la-volée de modèle stochastique (DETERMINATOR) [Ciardo-Zijal-96,Deavours-Sanders-99] ➔ partiel !
- Méthodes numériques basiques d'analyse (simulateurs de modèle markovien)
  - Analyse d'équilibre par *Gauss-Seidel* (steady-state, BCG_STEADY) [Stewart94]
  - Analyse d'état transitoire par *Fox-Glynn* (transient, BCG_TRANSIENT) [Stewart94]
- Méthodologie
  - Génération d'un STE stochastique
  - Réduction du STE stochastique en CTMC
  - Conversion d'un STE explicite (BCG format) en une matrice de transition
  - Connection de la matrice avec le simulateur markovien
  - Calcul d'un critère de performance (= débit, latence,...) à partir du vecteur probabiliste solution

Réalisations : processus d'analyse



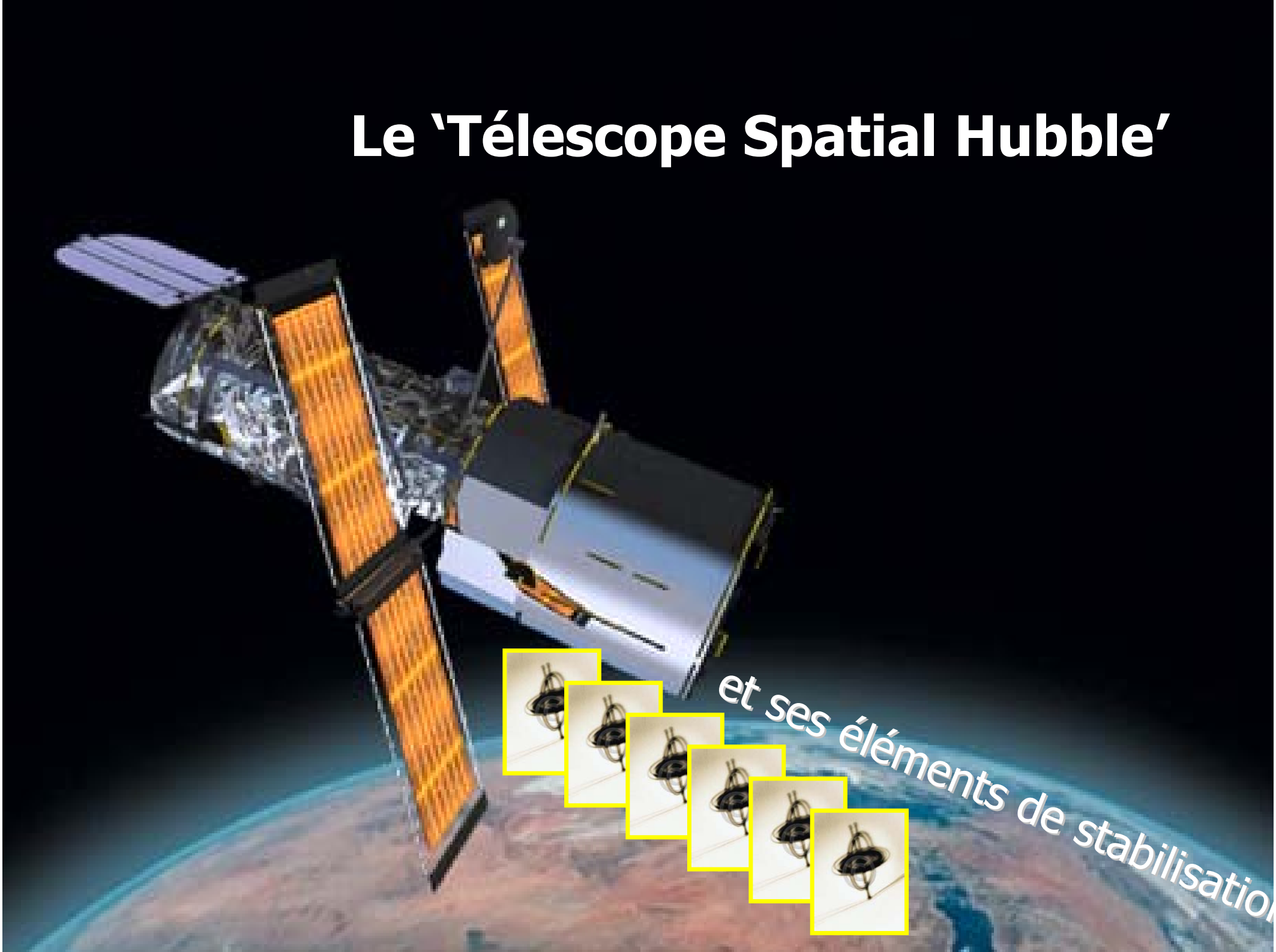
Plan

1. Quelles réalisations ?
2. Quelles applications ?
3. Quel contexte théorique ?
4. Quelles perspectives ?

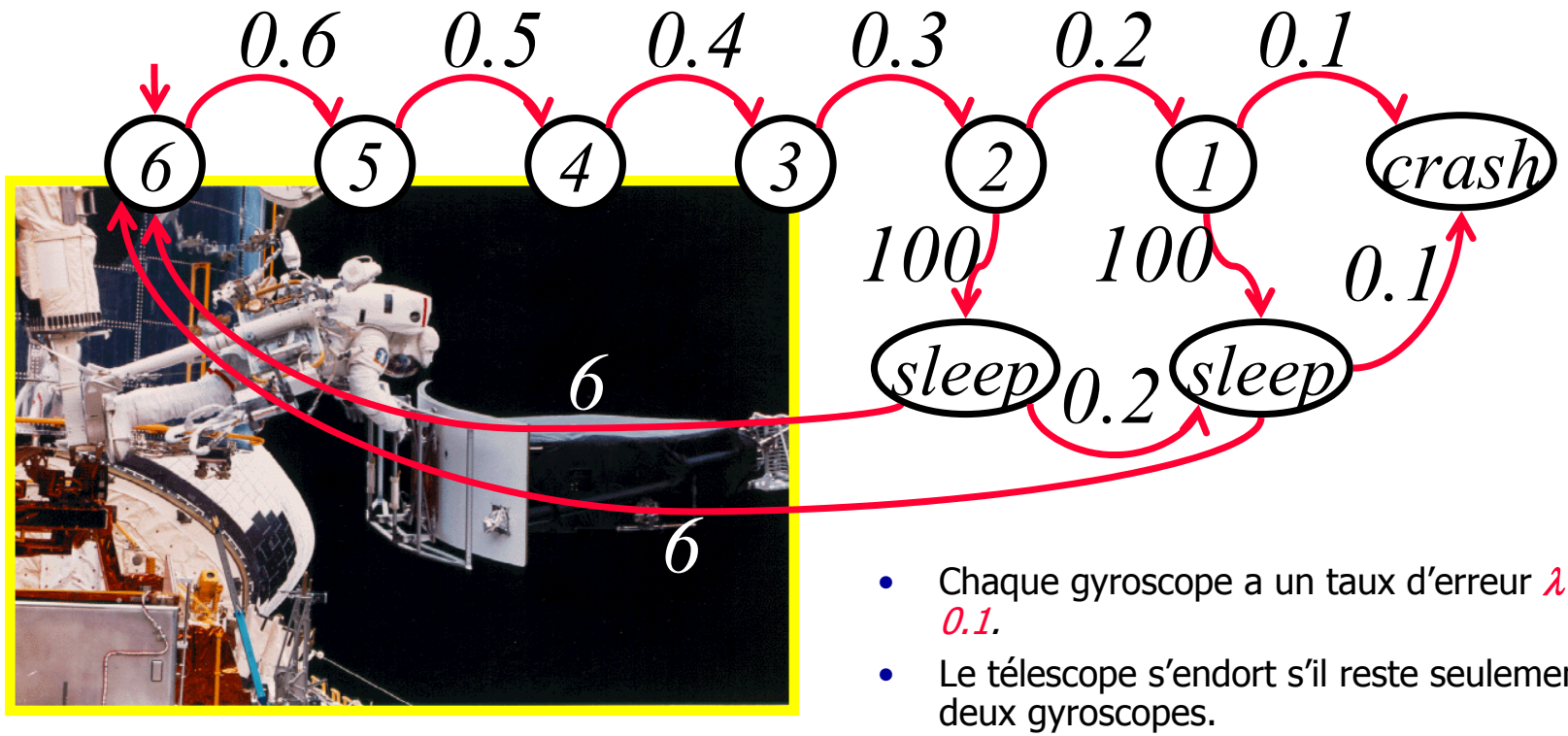
Applications : études de cas

- Les stabilisateurs de vol du satellite Hubble
  - Un exemple jouet pour les démos en direct (demo_30)
  - Utilise les outils CAESAR.ADT, CAESAR, SVL, DETERMINATOR, BCG_MIN, et BCG_TRANSIENT
- Le protocole d'arbitrage du bus SCSI-2
  - Mise en évidence du phénomène de famine lors du mauvais positionnement du contrôleur (demo_31)
  - Utilise CAESAR.ADT, CAESAR, SVL, DETERMINATOR, BCG_MIN, et BCG_STEADY

Le ‘Télescope Spatial Hubble’



Un modèle markovien simple de Hubble

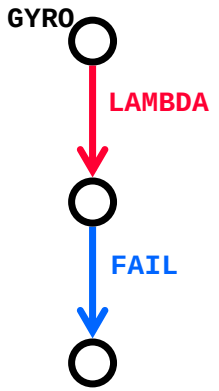


- La station terrestre prépare une mission avec navette pour réparer le télescope, ce qui prend environ 2 mois.
  ($\nu = 6$).

- Chaque gyroscope a un taux d'erreur $\lambda = 0.1$.
- Le télescope s'endort s'il reste seulement deux gyroscopes.
- Passer à l'état endormi requiert du temps ($\mu = 100$).
- Sans gyroscope opérationnel, le télescope tombe en panne.

Hubble en LOTOS

```
behaviour
   HUBBLE [LAMBDA, MU, NU]
where

process HUBBLE [LAMBDA, MU, NU] : noexit :=
      hide FAIL in
         (
            (
               GYRO [LAMBDA, FAIL] (true of Bool)
               |||
               GYRO [LAMBDA, FAIL] (true of Bool)
               |||
               GYRO [LAMBDA, FAIL] (true of Bool)
               |||
               GYRO [LAMBDA, FAIL] (true of Bool)
               |||
               GYRO [LAMBDA, FAIL] (true of Bool)
               |||
               GYRO [LAMBDA, FAIL] (true of Bool)
            )
            |[FAIL]|
            CONTROLLER [FAIL, MU, NU] (6 of Nat, false of Bool)
            >>
            (* system reset *)
            HUBBLE [LAMBDA, MU, NU]
         )
```



Hubble en LOTOS (suite)

```
process CONTROLLER [FAIL, MU, NU] (C : Nat, SLEEP : Bool) : exit :=
         (* Still gyros left *)
         [(C > 0)] ->
            (* Ah, a gyro failed. Let's count down. *)
            FAIL;
               CONTROLLER [FAIL, MU, NU] (C - 1, SLEEP)
         []
         (* Hubble starts tumbling. *)
         [(C < 3) and (SLEEP eq false)] ->
            (* Time to turn on the SLEEP mode. *)
            MU;
               CONTROLLER [FAIL, MU, NU] (C, true)
         []
         (* Sleep mode is on. *)
         [(SLEEP eq true)] ->
            (* Let's wait for the space mission to reset the system. *)
            NU;
               exit
         []
         (* No gyros left. *)
         [C == 0] ->
            (* Crash! *)
            i;
               stop
      endproc
   endproc
```

Trajectoire d'analyse pour l'example Hubble

caesar

_open

BCG

```
"NU" -> "repair; rate 6",
"MU" -> "suspend; rate 100",
"LAMBDA" -> "fail; rate 0.1"
```

bcg_labels

determinator

_open

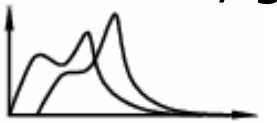
BCG

BCG

bcg_min

bcg_transient

Excel, gnuplot

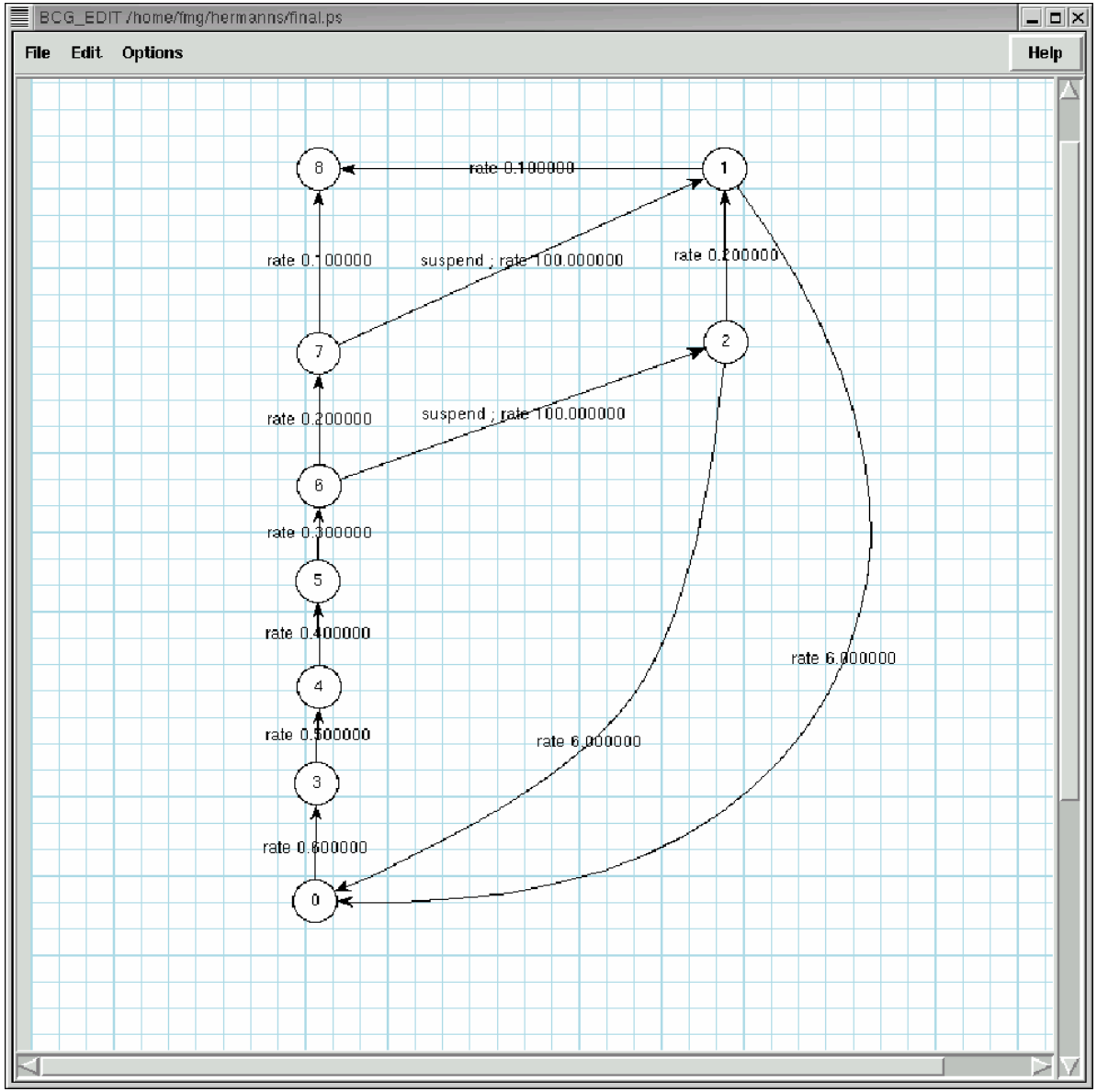


...un bon candidat pour le langage de script SVL

SVL script pour Hubble

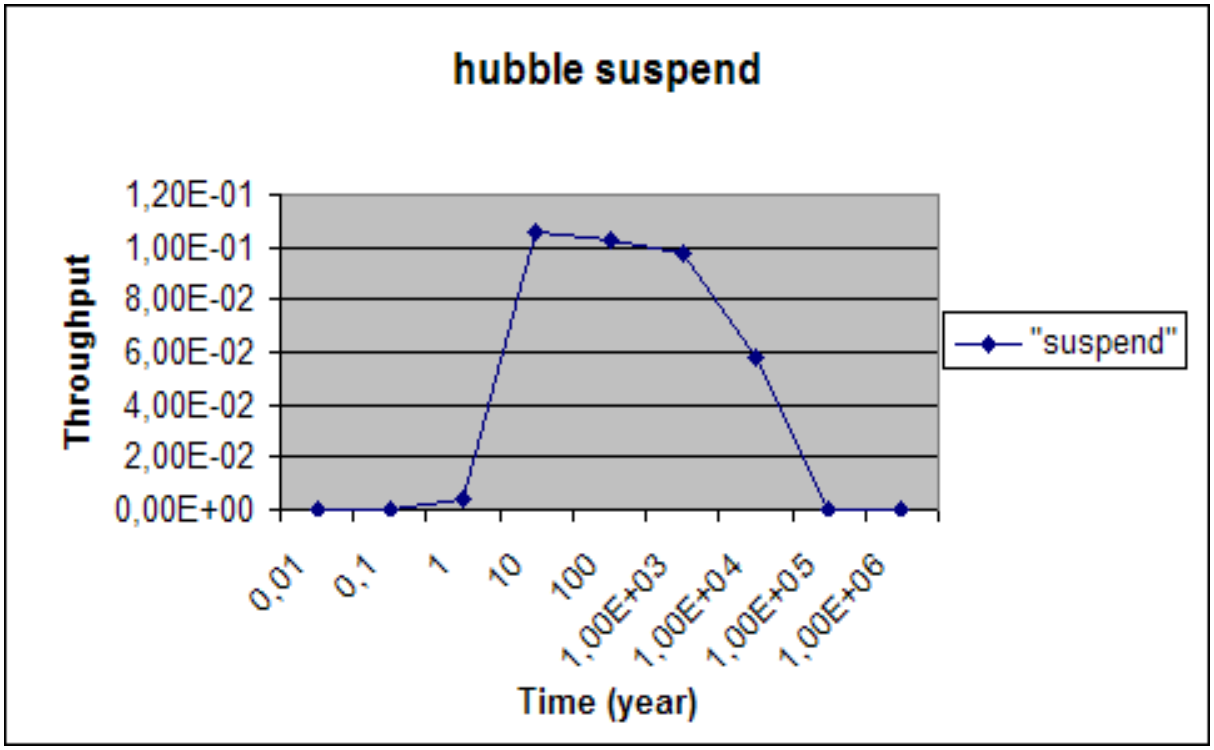
```
% CAESAR_OPTIONS="-debug"
% CAESAR_OPTIONS="-monitor"

(* generate the LTS *)
"hubble.bcg" = generation of "hubble.lotos";

"spec.bcg" =
          total rename (* turn into an IMC *)                                          (*1*)
             "NU" -> "repair; rate 6",     (* to prepare a shuttle mission, for reset takes 1/2 a year,    *)
             "MU" -> "suspend; rate 100",  (* to suspend the scientific, progtam takes 1/100 of a year,    *)
             "LAMBDA" -> "fail; rate 0.1"  (* the average lifetime of a,  gyroscope is 10 years.           *)
             in "hubble.bcg";

(* remove nondeterminism *)                                                          (*2*)
% bcg_open spec.bcg determinator.a -monitor "ctmc.bcg"

(* look for internal transitions: if absent, the graph is a Markov chain *)
% bcg_info -hidden "ctmc.bcg"

(* reduce modulo strong equivalence *)                                             (*2'*)
% BCG_MIN_OPTIONS="-rate -epsilon 1e-6"
"final.bcg" = strong reduction with bcg_min of "ctmc.bcg";

(* analyse for various time points *)
% for TIME in .01 .1 1 10 100 1e3 1e4 1e5 1e6  # measured in years
% do
%    echo
%    echo "analysing after ${TIME} years"
%    echo "------------------------"
%    ./bcg_transient -thr hubble.msr "final.bcg" time="$TIME"                       (*3*)

% done
```

CTMC minimisé de Hubble



Activité de Hubble (*suspend*) dépendante du temps

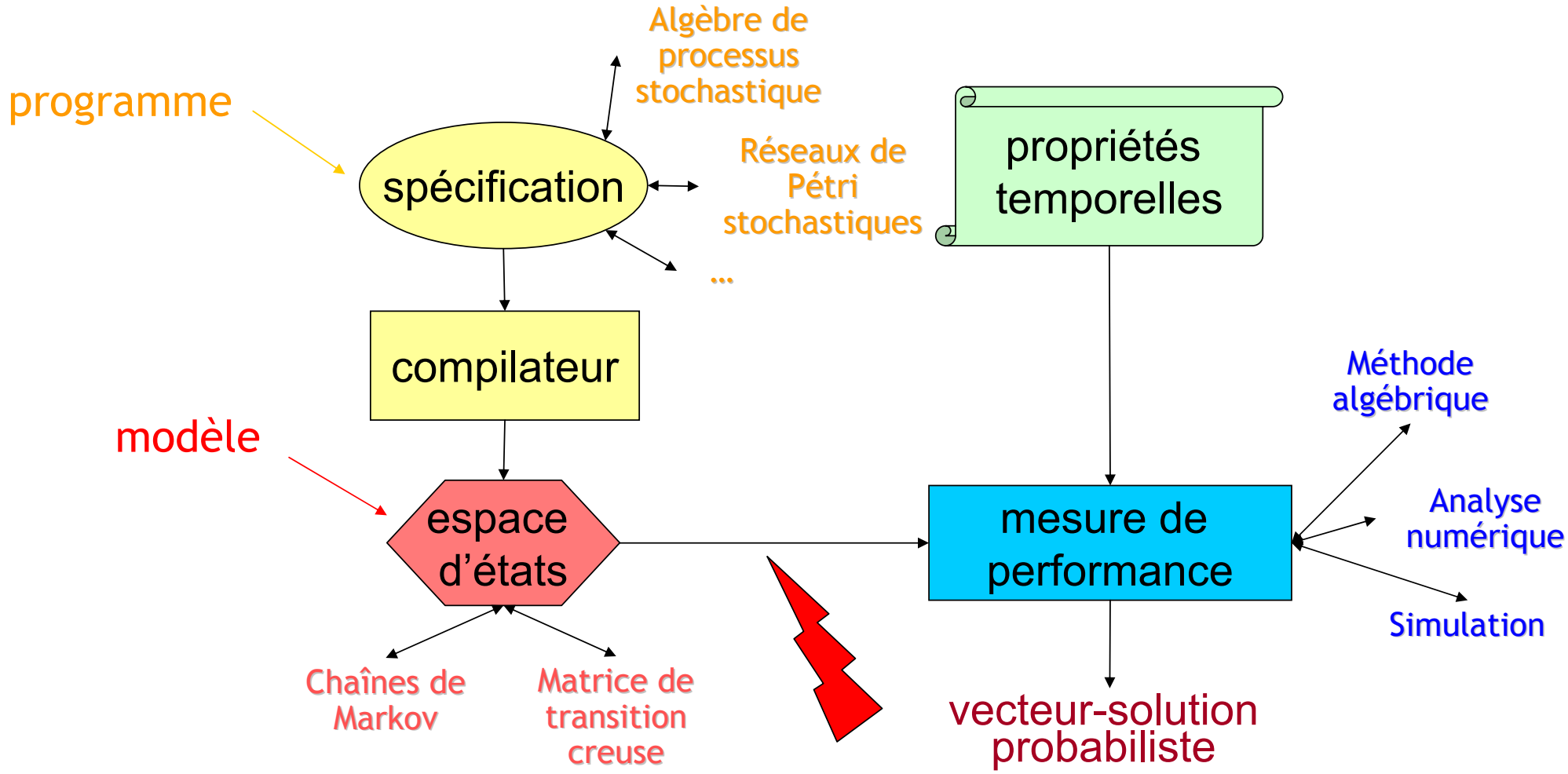
"time"	"repair"	"fail"	"suspend"
0,01	1,52E-11	0,5994	1,24E-09
0,1	5,45E-07	0,59403	4,34E-06
1	0,00248872	0,543138	0,00373419
10	0,105761	0,414947	0,105725
100	0,102729	0,414615	0,102786
1,00E+03	0,0974923	0,393478	0,097546
1,00E+04	0,0577739	0,233175	0,0578058
1,00E+05	0,00031195	0,00125902	0,00031212
1,00E+06	6,03E-27	2,43E-26	6,04E-27



Plan

1. Quelles réalisations ?
2. Quelles applications ?
3. Quel contexte théorique ?
4. Quelles perspectives ?

Contexte : évaluation de performances

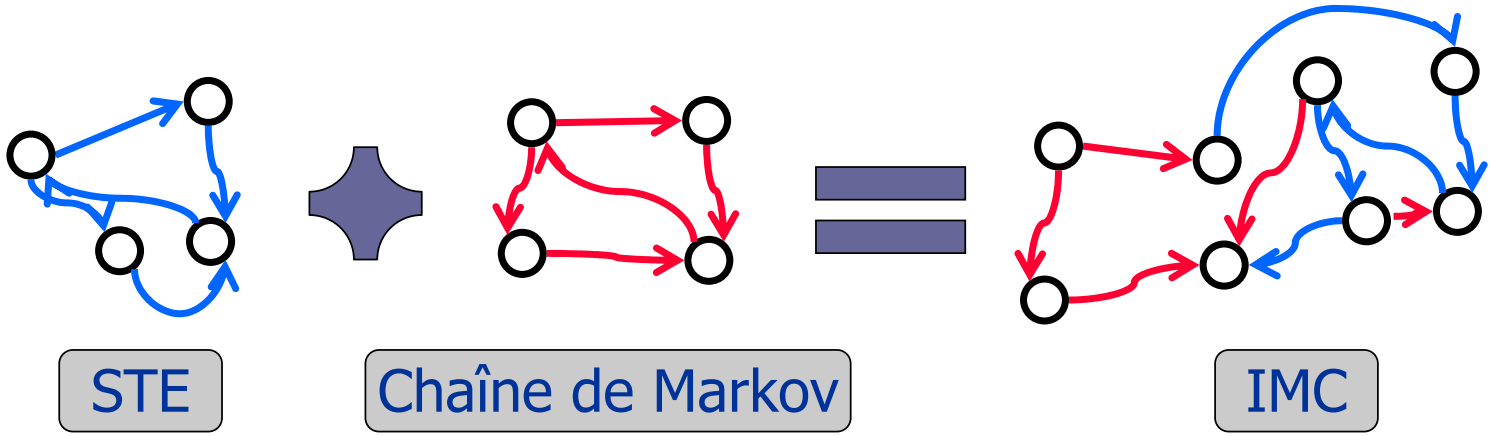


Contexte : modèle stochastique

- Processus stochastique
  - Famille de variables aléatoires $\{X_t, t \in T\}$
    - X_t est caractérisée par une fonction de distribution
- Processus Markovien
  - Processus stochastique + propriété Markovienne
    - (comportement futur entièrement défini par l'instant présent, donc totalement indépendant de son histoire)
- Chaîne de Markov
  - Processus Markovien + homogénéité + espace d'états discret
- CTMC (Continuous Time Markov Chain)
  - Chaîne de Markov + temps continu

Contexte : STE stochastique

- Modèles
  - Généralisation des STE d'algèbre de processus: $M = (S, A, T, R, s_0)$
  - ⇔ *Chaînes de Markov Interactives (IMC) [Hermanns98]* :



- **STE probabilistes** : transitions "**prob** *p*" et "*label* ; **prob** *p* "
- **STE stochastiques** : transitions "**rate** λ" et "*label* ; **rate** λ"
- **STE mixtes** : **(STE probabiliste ∪ STE stochastique) +** transitions étiquetées par des actions "*label*"

Contexte : types de STE acceptés

- BCG_STEADY:
  - STE stochastiques en entrée (taux + proba) sans deadlocks
  - Analyse itérative (Gauss-Seidel)

$$\pi_i^{(k+1)} = -\frac{1}{a_{i,i}}\left(\sum_{j<i}\pi_j^{(k+1)}a_{i,j} + \sum_{j>i}\pi_j^{(k)}a_{i,j}\right)$$

- BCG_TRANSIENT:
  - STE stochastiques en entrée (taux + proba) avec possibilité de deadlocks
  - Méthode d'uniformisation + probabilité de Poisson (Fox-Glynn)

$$\underline{\tilde{\pi}}(t) = \sum_{n=0}^{k_{ss}}\psi(\lambda t;n)\underline{\hat{\pi}}(n) + \left(\sum_{n=k_{ss}+1}^{k_{\varepsilon}}\psi(\lambda t;n)\right)\underline{\hat{\pi}}(k_{ss})$$

$$avec \quad \psi(\lambda t;0) = e^{-\lambda t}$$

$$et \quad \psi(\lambda t;n+1) = \psi(\lambda t;n)\frac{\lambda t}{n+1}, n \in \mathrm{N}$$

- DETERMINATOR:
  - STE mixtes input (rate + proba + action)
  - STE stochastiques output (si succes, sinon identique à input)
  - Extraction du non-déterminisme selon la propriété de processus stochastique « bien spécifié » (Deavours-Sanders)

Plan

1. Quelles réalisations ?
2. Quelles applications ?
3. Quel contexte théorique ?
4. Quelles perspectives ?

Conclusion

- Elargir la plate-forme CADP à la modélisation et à l'analyse de performance et de dépendabilité.

- Approche pragmatique
  - Pas d'extension de syntaxe;
  - Utilise de nombreuses parties de la boîte à outils et du script SVL;
  - Algorithmes partiels de construction de MC;
  - Algorithmes d'analyse de MC (6000 lignes de C d'outils + 12000 lignes de C de bibliothèque d'analyse de matrice creuse);
  - Linux, Unix, Windows.

- Fera partie de la prochaine version de CADP (2003)

http://www.inrialpes.fr/vasy/cadp

Perspectives

- Intégration de PDAC dans la prochaine version de CADP (2003)
- Connecter CADP avec d'autres simulateurs très performants : PRISM, …
- Générer automatiquement des contraintes stochastiques au sein de spécifications formelles (ou dans un STE classique)
- Vérifier par bisimulation et par logique temporelle probabiliste
- Analyser en parallèle des chaînes de Markov

Références

- [CZ96] G. Ciardo et R. Zijal. *Well-defined stochastic Petri nets*. MASCOTS'96
- [DS99] D.D. Deavours et W.H. Sanders. *An efficient well-specified check. PNPM'99*
- [GH02] H. Garavel et H. Hermanns. *On Combining Functional Verification and Performance Evaluation using CADP*. FME'02
- [GL01] H. Garavel et F. Lang. *SVL: A Scripting Language for Compositional Verification*. FORTE/PSTV'01
- [Her98] H. Hermanns. *Interactive Markov Chains, And the Quest of Quantified Quality*.
- [Her01] H. Hermanns. *Construction and Verification of Performance and Reliability Models*. EATCS'01
- [HJ03] H. Hermanns et C. Joubert. *A Set of Performance and Dependability Analysis Components for CADP*. TACAS'03
- [Kun86] K.S. Kundert. *Sparse Matrix Techniques*. CASD'86
- [Ste94] W.J. Stewart. *Introduction to the numerical solution of Markov chains*.