

Model-Checking Aléatoire :
une approche entre test et vérification

Marc BOYER

ENSEEIHHT – TéSA / IRIT (IRT)

2, rue Camichel

31071 TOULOUSE Cedex 7

Marc.Boyer@enseeiht.fr

Jean-Christophe PINCE

M3-Systems

1, rue des oiseaux

31410 Lavernose

pince@m3systems.net

10 Mars 2004

List of Slides

1 – La vérification formelle en milieu industriel	3
2 – Une opportunité : un DRT avec une PME	4
3 – Model-checking aléatoire : entre test et exhaustivité	5
3.1 – Principe de l’approche	5
3.2 – Comparaison théorique des approches	6
3.3 – Comparaison pratique des approches	9
4 – Le cas d’étude : SOIF	11
4.1 – Le système	11
4.2 – Les résultats	12
5 – Face aux approches modernes	13
6 – Conclusion	15

1 – La vérification formelle en milieu industriel

Vous savez ce que c'est...

- surtout du test
- un peu de preuve
- bien peu de model-checking
 - besoin de développer un prototype dans un langage spécifique
 - peur de la complexité
 - problème de taille des systèmes

2 – Une opportunité : un DRT avec une PME

- Volonté du responsable de voir ce qu'était devenu les RdP
- Un stagiaire sans *a priori*
 - connaissance de SDL
- Un cas d'étude temps-réel : couche unifiée pour satellites

Démarche :

- choix du langage IF (outillé, proche de SDL)
- prise en main, premier prototype
- une idée pour limiter l'explosion de la taille : mettre un `random` dans les transitions

et pourquoi pas ? ...

3 – Model-checking aléatoire : entre test et exhaustivité

3.1 Principe de l'approche

Le moteur de model-checking suppose qu'une action sans paramètre est déterministe (en fonction de l'état de départ).

En mettant un code C++ caché dans la transition, on trompe le moteur.

Il ne tire ces transitions que tant qu'il découvre de nouveaux états.

En n'utilisant ce *truc* que dans les transitions de génération d'entrée, on a :

- génération partielle des séquences d'entrée
- génération exhaustive des réponses à ces entrées

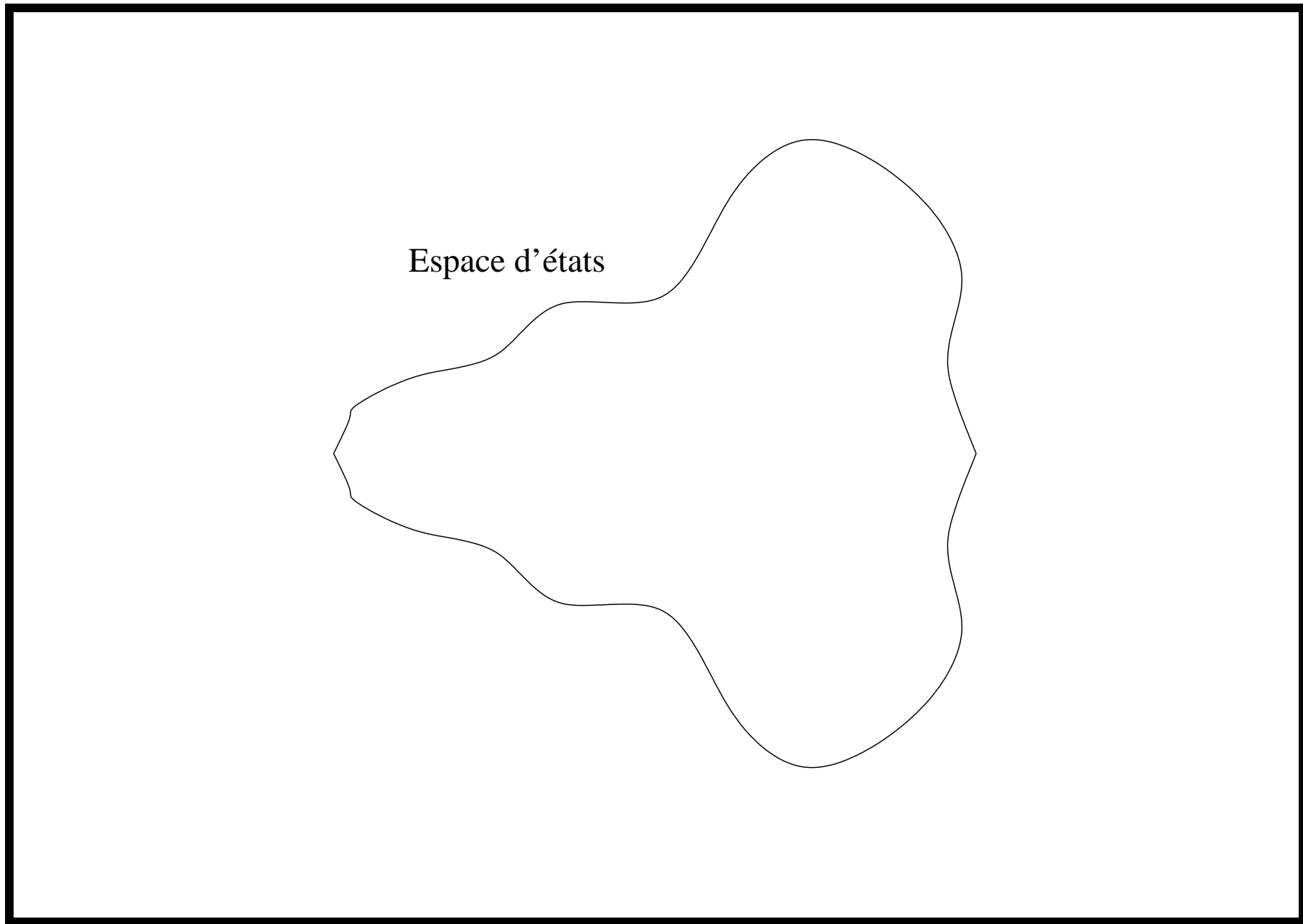
⇒ on gagne sur le test pour un système où la réponse est non déterministe (systèmes parallèles)

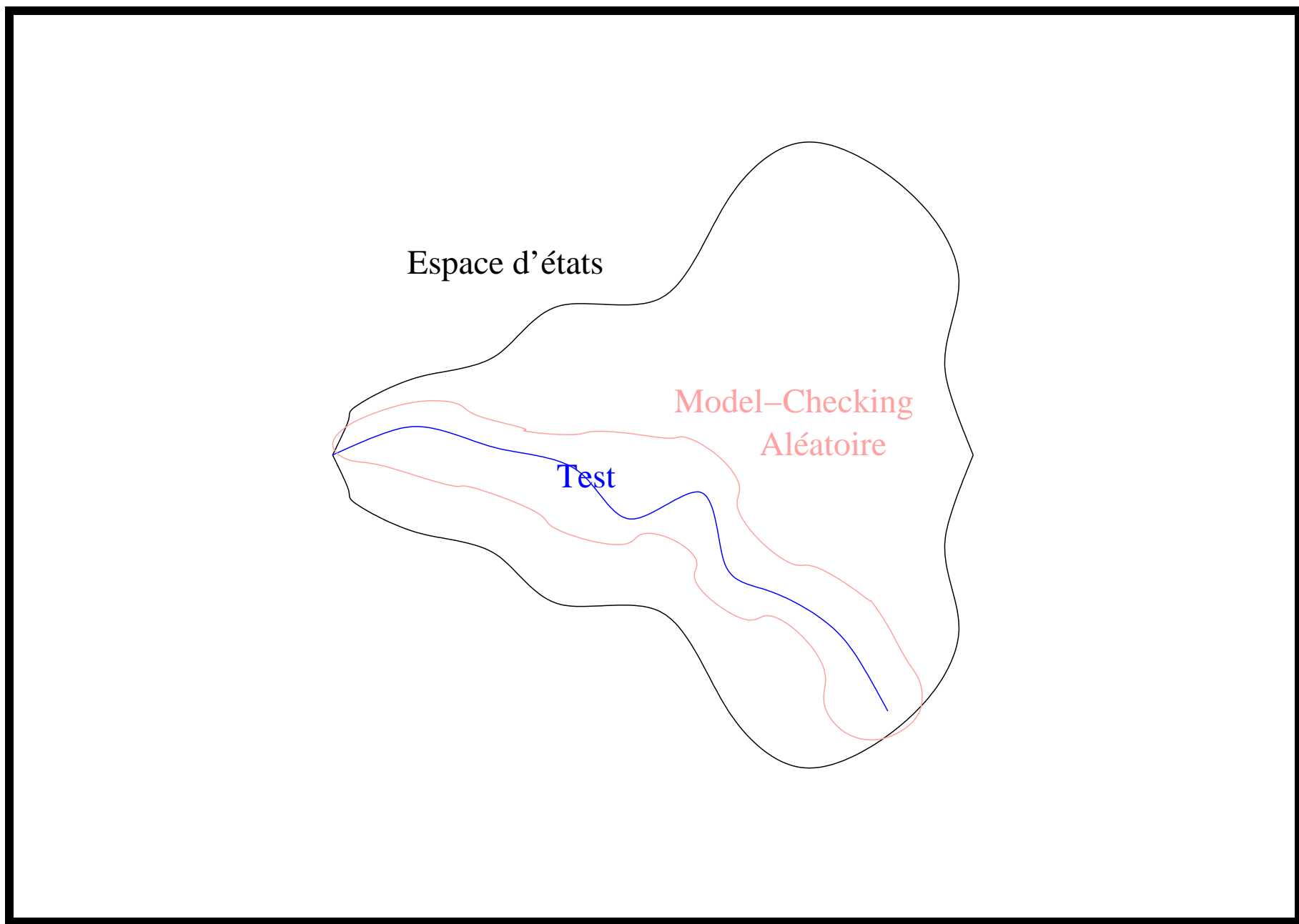
3.2 Comparaison théorique des approches

Model-checking : générer l'ensemble des comportements du système et vérifier les propriétés souhaitées, sans distinction entrée/sortie

Test : générer des entrées (signifiante ?) et vérifier les propriétés des sorties observées

Model-checking aléatoire : générer des entrées aléatoires et vérifier les propriétés de toutes les sorties possibles





3.3 Comparaison pratique des approches

Model-checking :

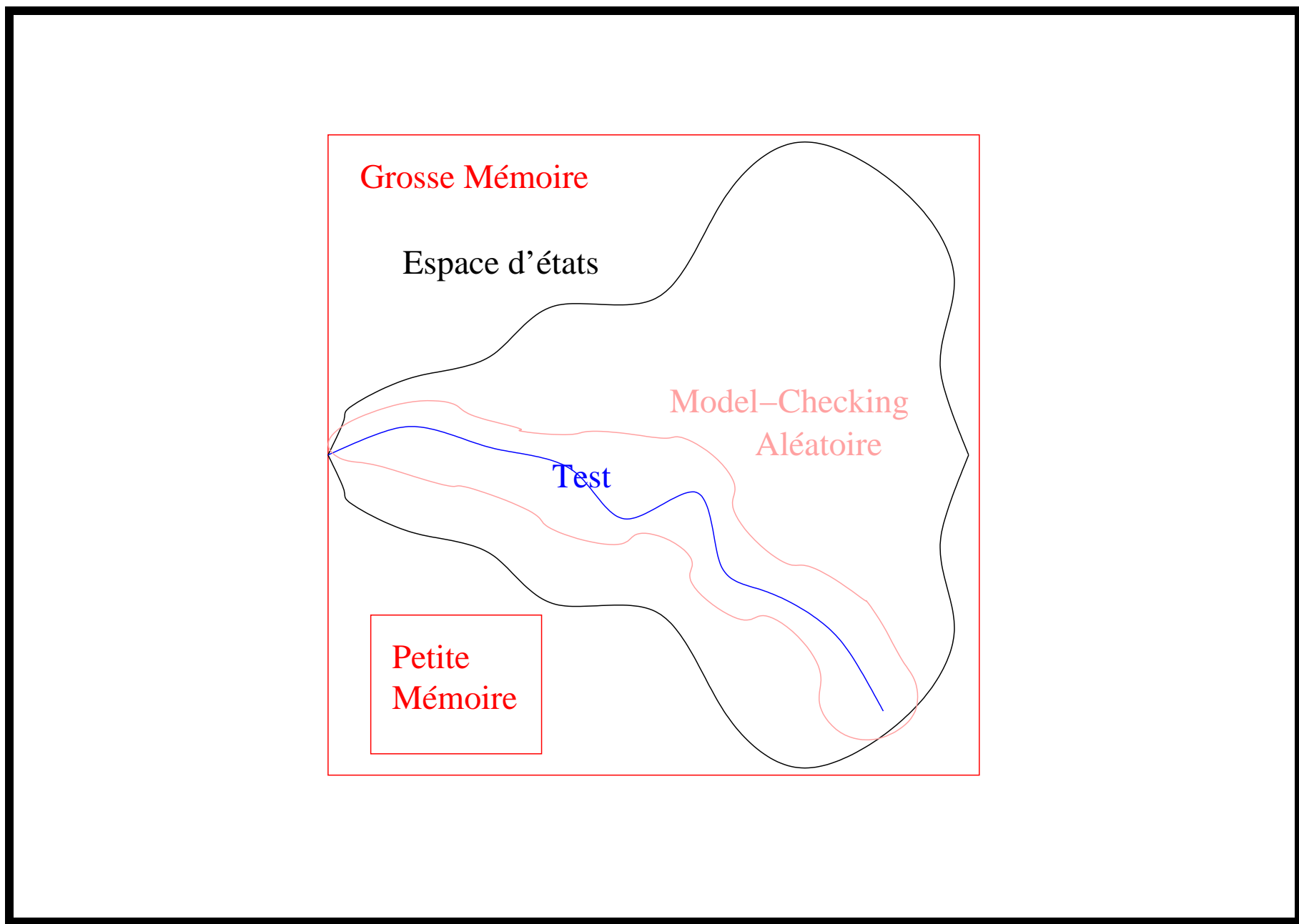
- nécessité de développer un *modèle* du système
- problème de taille, besoin d'abstraire
- ⇒ expertise nécessaire pour que les propriétés de l'abstraction soient transférables au modèle

Test : – test du *système*

- pas d'exhaustivité

Model-checking aléatoire :

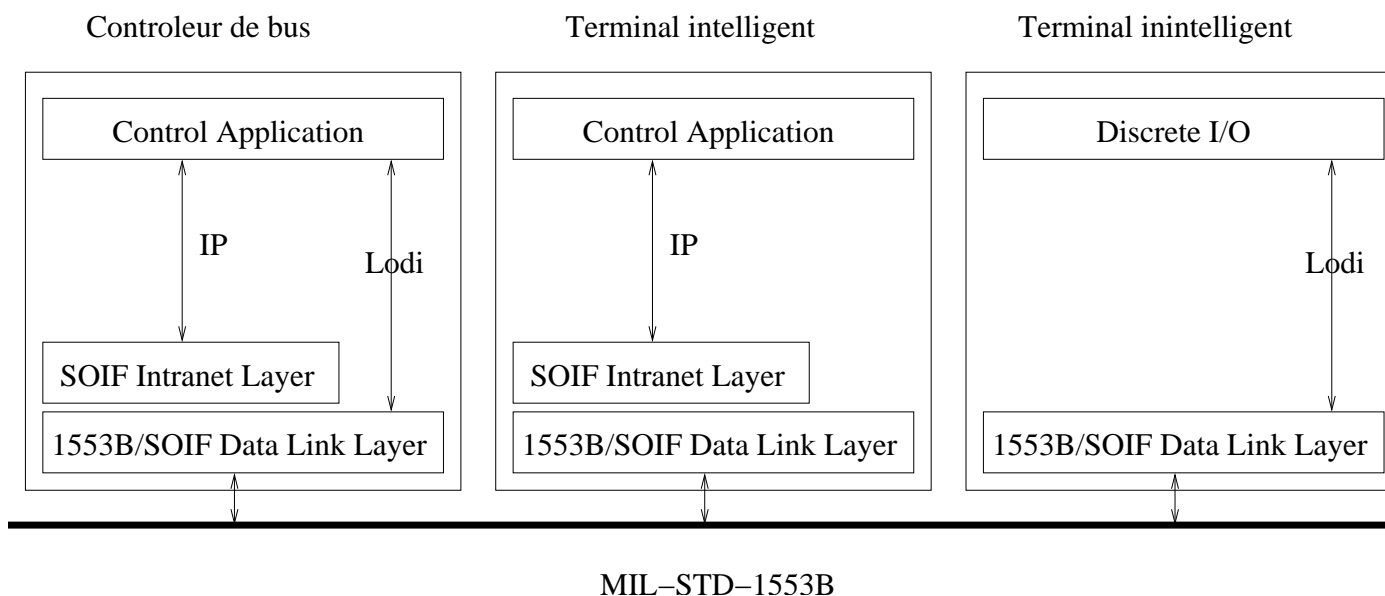
- nécessité de développer un *modèle* du système
- pas d'exhaustivité



4 – Le cas d'étude : SOIF

4.1 Le système

- norme pour satellite
- flux périodique temps réel (Lodi) et IP (commandes)



4.2 Les résultats

	Model-checking					Model-checking aléatoire				
	états	trans .	P1	P2	P3	états	trans.	P1	P2	P3
s0	35.080	35.379	Non	Non	Non	2.762	2.791	Non	Non	Non
s1	49.525	49.827	Non	Non	Non	3.966	3.989	Non	Non	Non
s2	53.152	53.451	Non	Non	Oui	4.212	4.235	Non	Non	Non
s3	out of memory					538.046	551.509	Non	Non	Non
s4	out of memory					3.586	3.608	Non	Non	Non
s5	out of memory					5.316	5.339	Non	Non	Non
s6	out of memory					8.316	8.408	Non	Non	Non
s7	out of memory					2.507.214	2.574.529	Non	Non	Oui

P1 : Blocage P2 : Boucle infi nie P3 : Débordement

5 – Face aux approches modernes

La comparaison initiale était partielle (partiale ?)

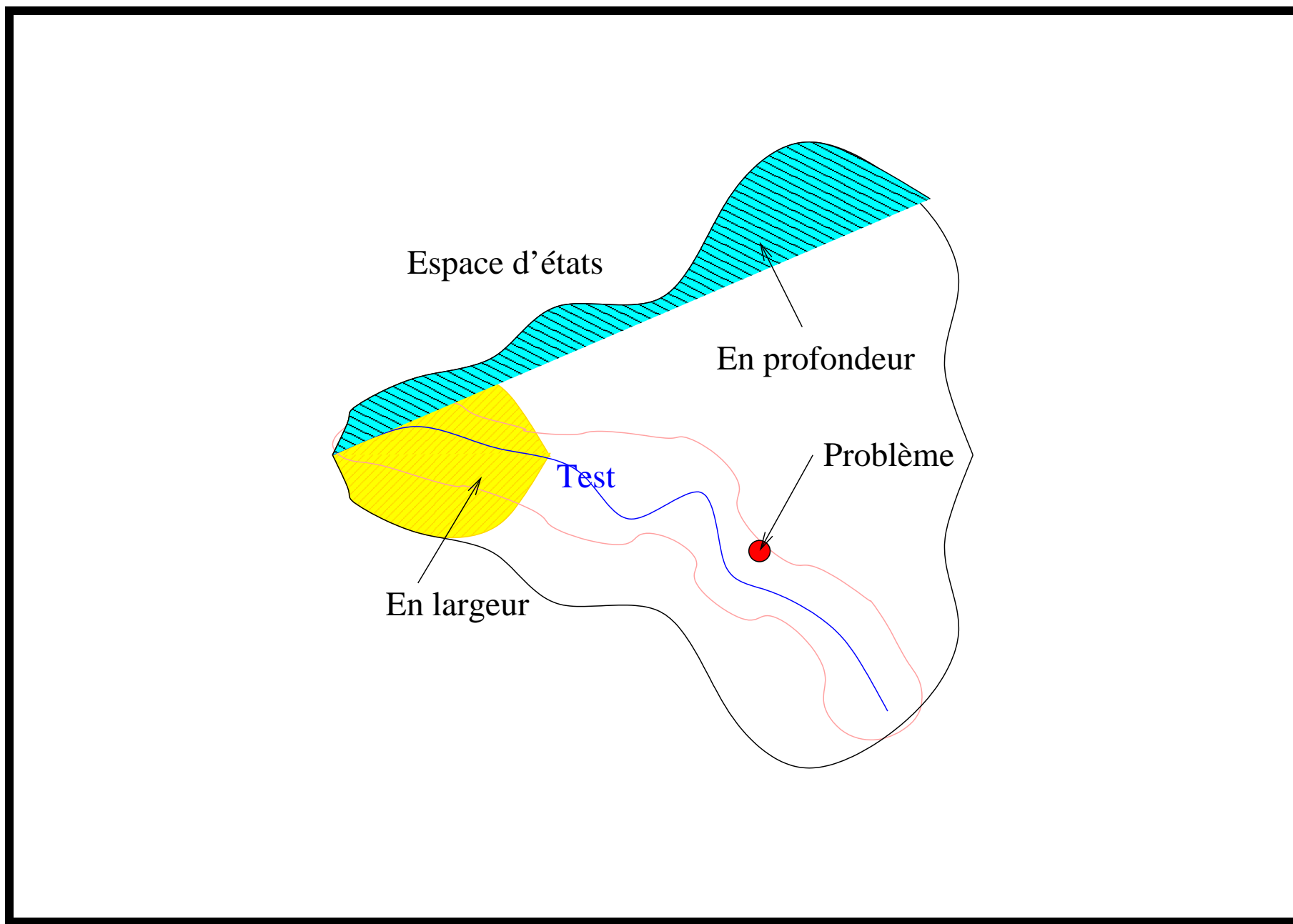
Model-checking à la volée :

- en entrée : un système *et* une propriété
- explore uniquement la partie nécessaire pour évaluer la propriété
- guidage en profondeur ou en largeur d'abord

Test : séquence d'entrée significative et non aléatoire

Notre approche :

- explore les entrées "sur une profondeur", les réponses de façon exhaustive
- possibilité de guider les entrées



6 – Conclusion

- dans la pratique :
 - IF proche de SDL
 - possibilité d’absorber le système et pas une abstraction cousue main
 - pratique proche du test
 - ⇒ usage autonome de l’industriel possible
- dans la théorie : étude nombre moyens d’états couverts
- outillage :
 - reprendre l’expertise du test pour des entrées significatives
 - si accès à un moteur : simplement une politique de parcours différente