

Algorithmique de la Planification

Cours 2 : La planification et l'IA

Pierre REGNIER
IRIT - Université Paul Sabatier
<http://www.irit.fr/~Pierre.Regnier>



0. Plan de l'exposé

1. L'Intelligence Artificielle (IA)
 - 1.1. Les buts de l'IA
 - 1.2. Les paradigmes de l'IA
 - 1.3. La problématique de l'agent autonome
2. La planification en IA
 - 2.1. Planification et agent autonome
 - 2.2. Les planificateurs : requêtes et propriétés
 - 2.3. Planification et exécution
 - 2.4. Applications de la planification
 - 2.5. Le cadre classique
3. Approches de la planification
 - 3.1. Les approches essentielles
 - 3.2. Les problèmes posés par la logique
 - 3.3. Tableau récapitulatif
4. Conclusion

1.1. Les buts de l'IA

- **Les buts de l'IA :**

- objectif théorique : **étudier** les facultés mentales, fabriquer des modèles calculatoires (fondés sur le traitement symbolique ou numérique de l'information), pour cerner les principes et les mécanismes qui rendent possibles les phénomènes cognitifs ;
- objectif pratique : **simuler** certaines des facultés mentales, sans obéir aux restrictions qu'imposent l'adoption d'un paradigme et les observations des sciences cognitives (recherche sur les outils matériels et logiciels, représentation et traitement des connaissances...).

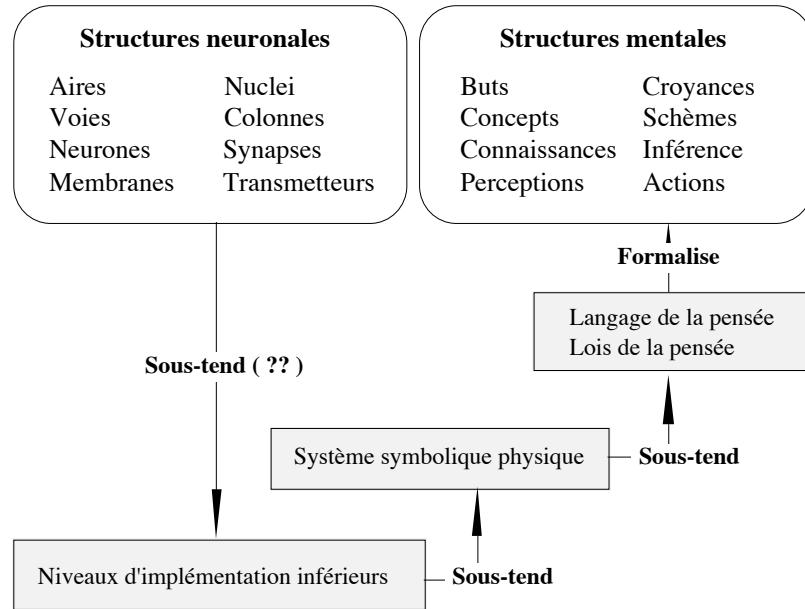
1.2. Les paradigmes de l'IA

- **Les paradigmes de l'IA :**

- le **symbolisme** (représentation, traitement symbolique, discret, de l'information, machine de Turing, von Neumann, ordinateur...) ;
- le **connexionnisme** (implémentation, apprentissage, neurones formels, réseau connexionniste, traitement numérique de l'information) ;
- **l'enaction** (vie artificielle, multi-agents).

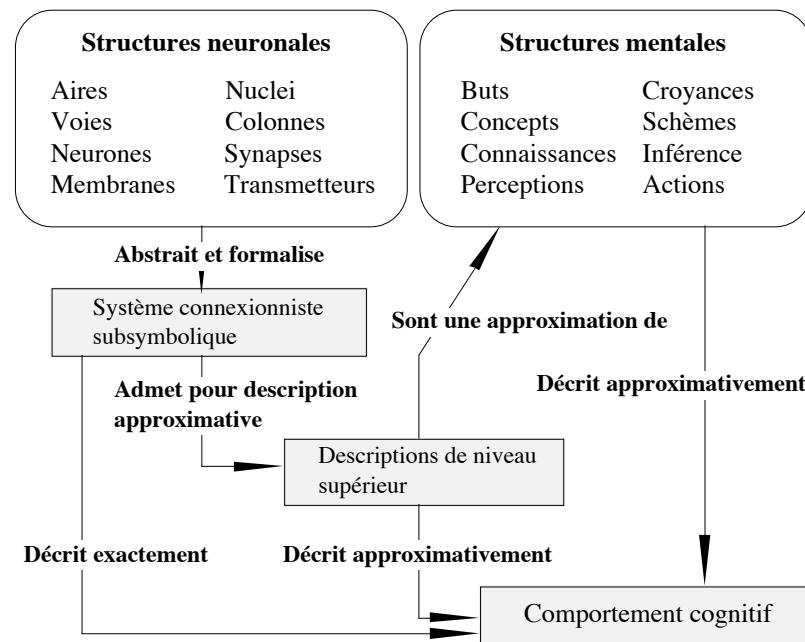
1.2. Les paradigmes de l'IA

- Le symbolisme [Smolensky, 92] :



1.2. Les paradigmes de l'IA

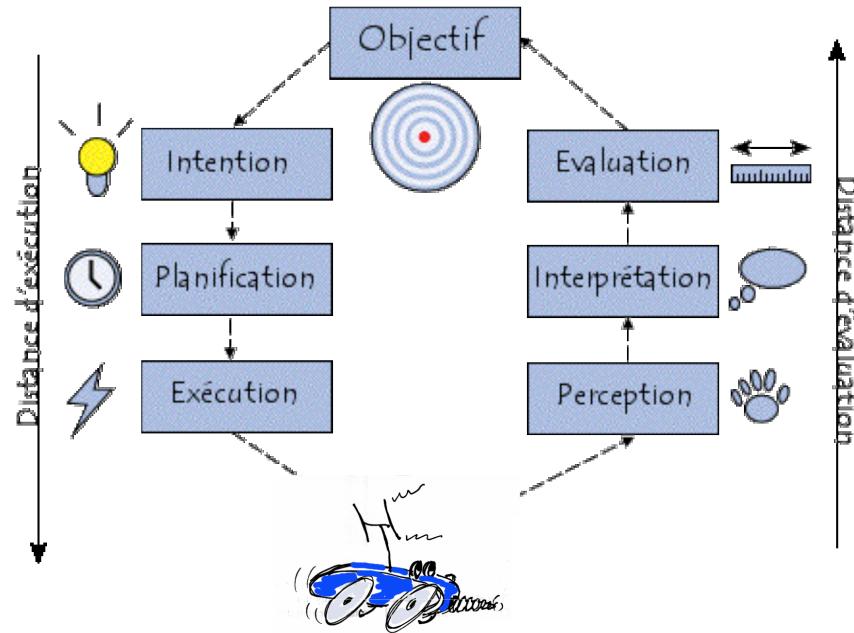
- Le connexionnisme [Smolensky, 92] :



2.1. La problématique de l'agent autonome

- En sciences cognitives, la « théorie de l'action » [Norman, 1986] modélise les actions et réactions de l'homme. Cette théorie repose sur deux concepts :
 - l'homme conçoit des modèles simples pour définir son comportement ;
 - l'homme décompose ses actions selon une boucle comprenant 7 étapes :
 - établissement d'un but** : représentation mentale d'un objectif à atteindre ;
 - formulation d'une intention** : décision d'atteindre l'objectif fixé ;
 - génération d'un plan d'action** : la suite d'actions à mener ;
 - exécution du plan d'action** : consiste à activer le système moteur ;
 - perception de l'état du système**, décrivant sous forme de variables psychologique la perception du changement par l'organisme ;
 - évaluation** de l'état atteint par rapport aux objectifs initialement fixés.
- Si l'objectif initial est complexe, il est décomposé en sous-objectifs successifs réalisables. Deux distances caractérisent l'écart entre le but de l'utilisateur et sa réalisation, et lui permettent d'estimer l'effort nécessaire pour atteindre l'objectif fixé :
 - la **distance d'exécution** : écart entre le but et l'état du système (branche descendante) ;
 - la **distance d'évaluation** : écart entre l'état du système et le but (branche ascendante).

2.1. La problématique de l'agent autonome

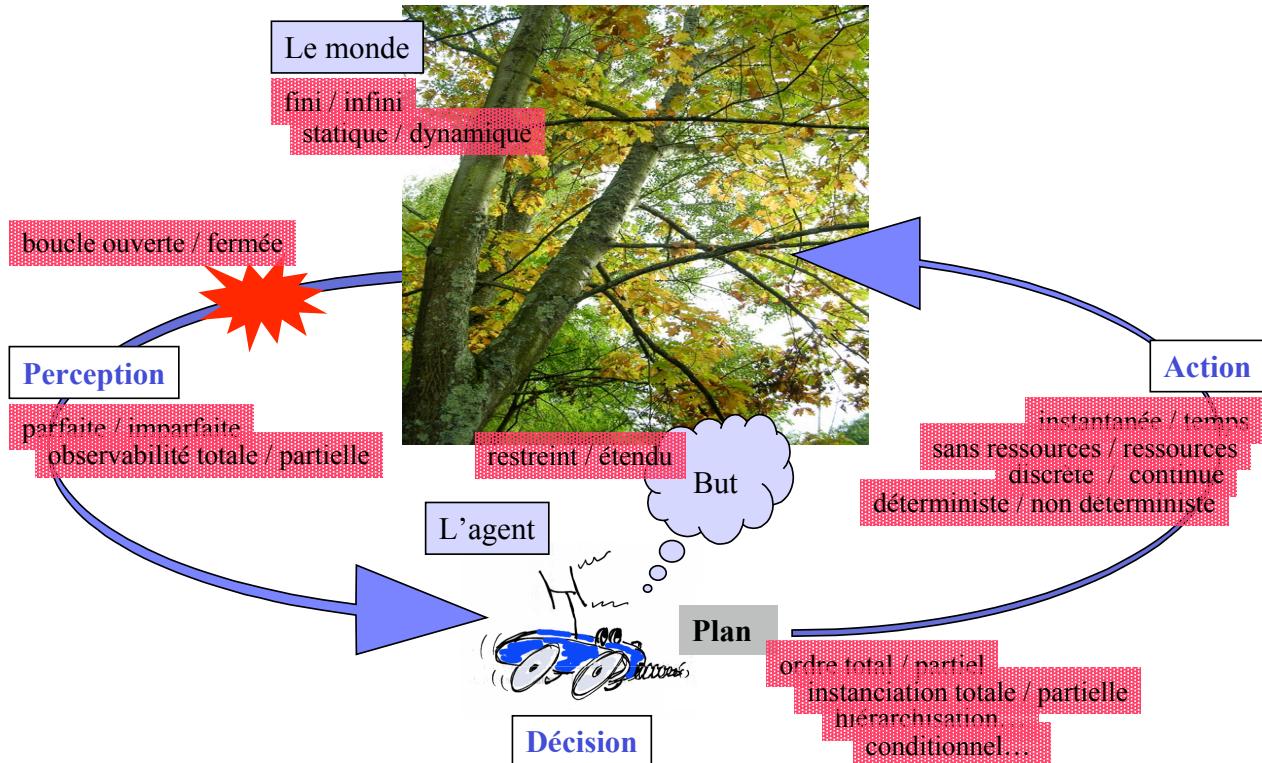


2.1. La problématique de l'agent autonome

- L'IA et la problématique de l'**agent autonome** :
 - comment permettre à un agent d'agir de manière autonome dans des environnements particuliers pour atteindre un but préalablement fixé ?



2.1. La problématique de l'agent autonome



2.1. La problématique de l'agent autonome

- **La planification** est un domaine de recherche qui cherche à concevoir des systèmes générant automatiquement, grâce à une procédure formalisée, un système intégré de décisions appelé **plan**. Le mot « planification » désigne aussi le **processus** qui élabore cet ensemble d'actions. Le **planificateur** est le système qui le produit.
- Le **plan** est une collection organisée de descriptions d'opérations ; il est destiné à guider l'action d'un ou plusieurs agents exécuteurs (systèmes robotiques ou humains) qui ont à agir dans un monde particulier pour atteindre un but préalablement défini.
- **Exécution** : réalisation d'actions effectuée dans ce monde particulier en suivant les directives du plan. Elle vise à réaliser la prédiction que constitue ce plan.
- **Plan-solution** : c'est un plan dont l'exécution permet de faire évoluer l'univers de l'état initial vers un état satisfaisant le but.
- **Flexibilité d'un plan-solution** : capacité que possède le plan à pouvoir être modifié (ordre d'exécution d'actions non ordonnées, instanciation de variables non instanciées...) sans perdre sa correction.

2.2. Planificateurs : requêtes et propriétés

- **Planificateur : les requêtes envisageables**
 - existe-t-il ou non un plan-solution ?
 - donner un plan-solution ;
 - donner un plan-solution obéissant à des critères donnés : possibilité d'exécution d'actions en parallèle, instanciation partielle des opérateurs, nombre d'actions, nombre de niveaux, temps d'exécution borné, utilisation de ressources, destiné à un agent particulier...
 - donner une famille de plans-solutions optimaux selon un critère particulier : même nombre d'opérateurs ; mêmes opérateurs ; satisfont certaines contraintes, temporelles ou d'utilisation de ressources...
 - donner (rapidement) un plan « utilisable » ;

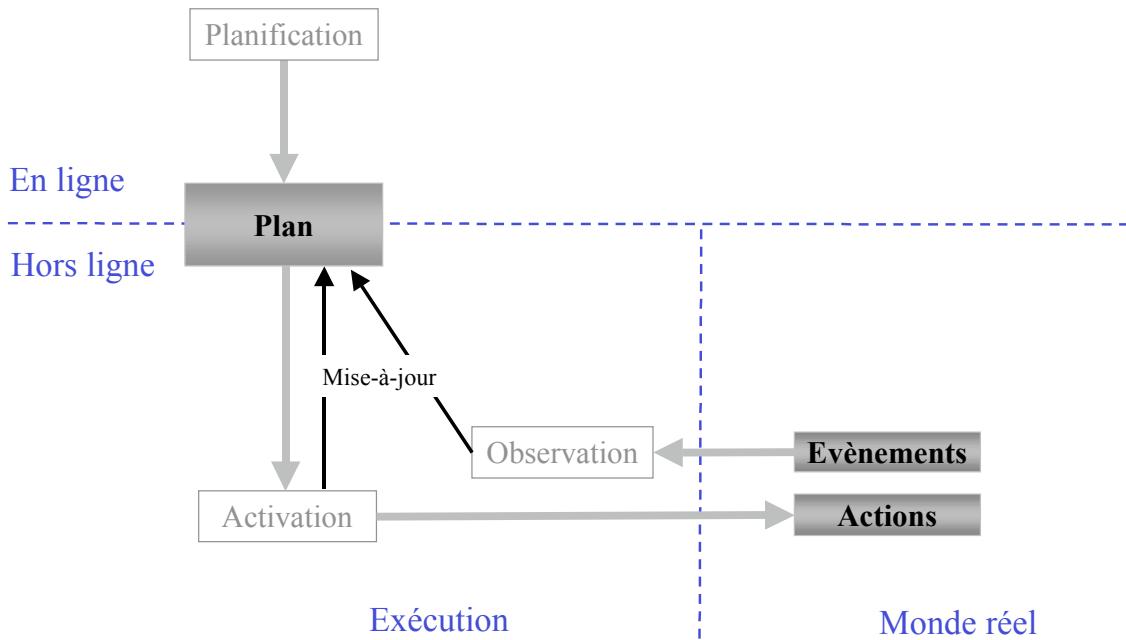
2.2. Planificateurs : requêtes et propriétés

- **Planificateur : les propriétés essentielles**

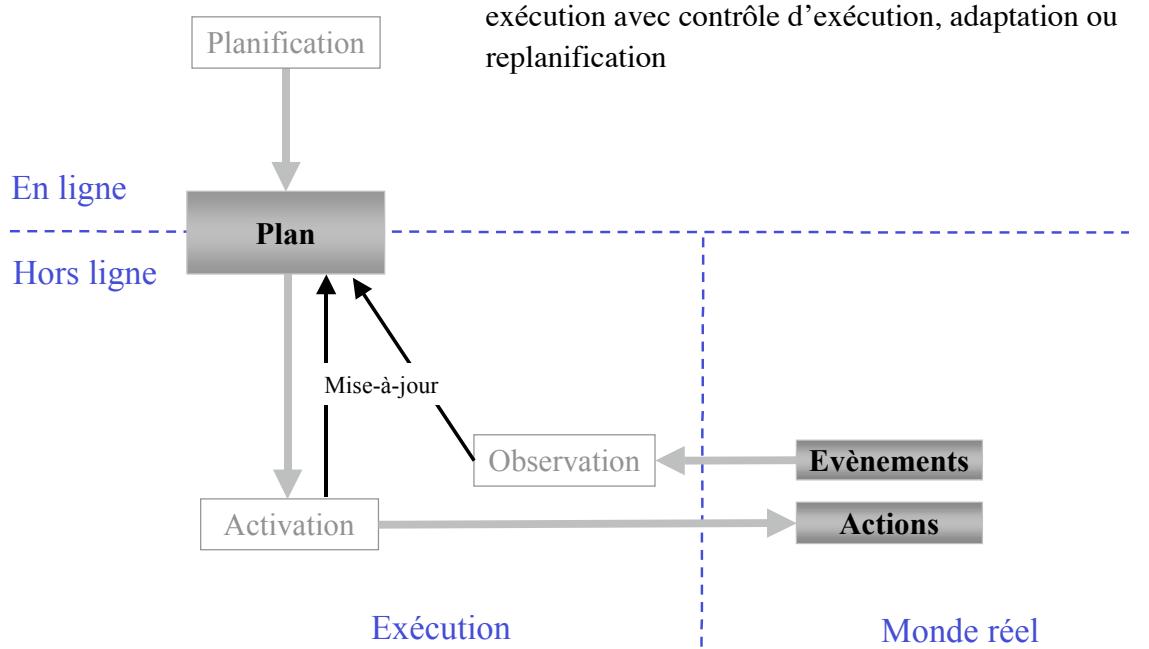
- **planificateur sain** : ssi, étant donné un problème de planification, et étant donné le formalisme de description de problèmes utilisé, tous les plans qu'il est susceptible de produire pour résoudre ce problème sont des plans-solutions ;
- **planificateur complet** : ssi, étant donné un problème de planification, et étant donné le formalisme de description de problèmes utilisé, il produit un plan-solution lorsqu'il existe ;
- **planificateur systématique** : ssi deux feuilles quelconques de l'arbre de recherche correspondent à des plans-solutions différents ;
- **planificateur optimal** (suivant un critère particulier qui mesure la qualité des plans) ssi les plans-solutions qu'il produit sont optimaux selon ce critère (nombre d'actions, nombre de niveaux, degré de parallélisme, temps d'exécution, quantité de ressources consommées...).

2.3. Planification et exécution

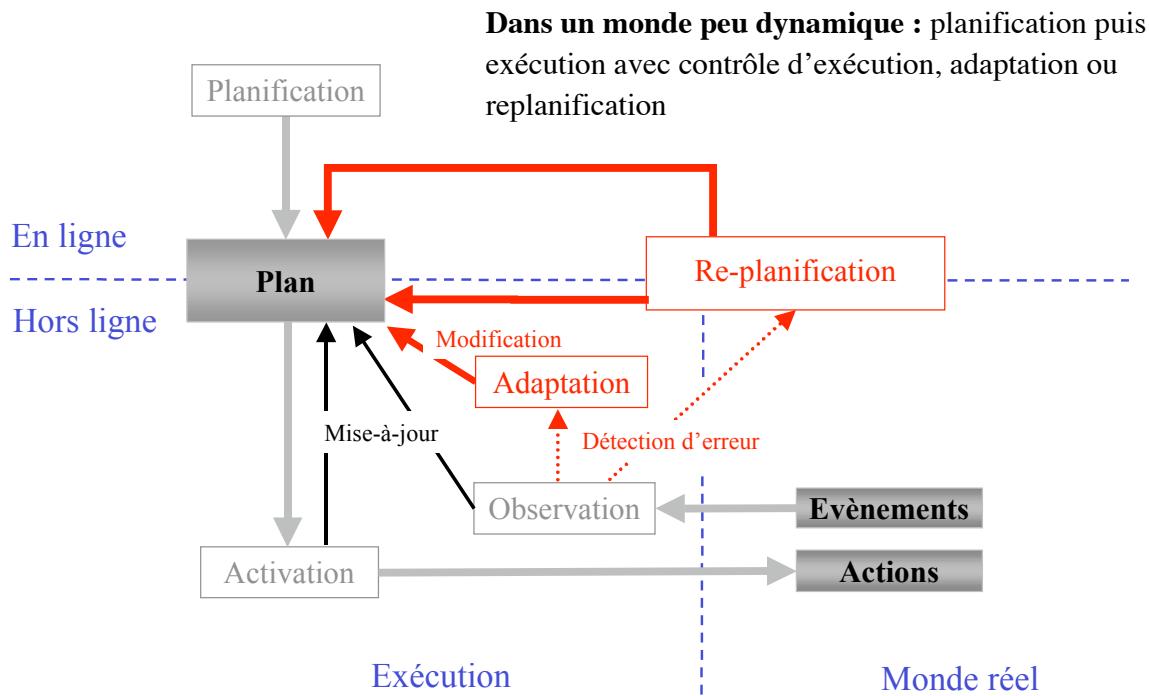
Dans un monde statique : planification puis exécution



2.3. Planification et exécution

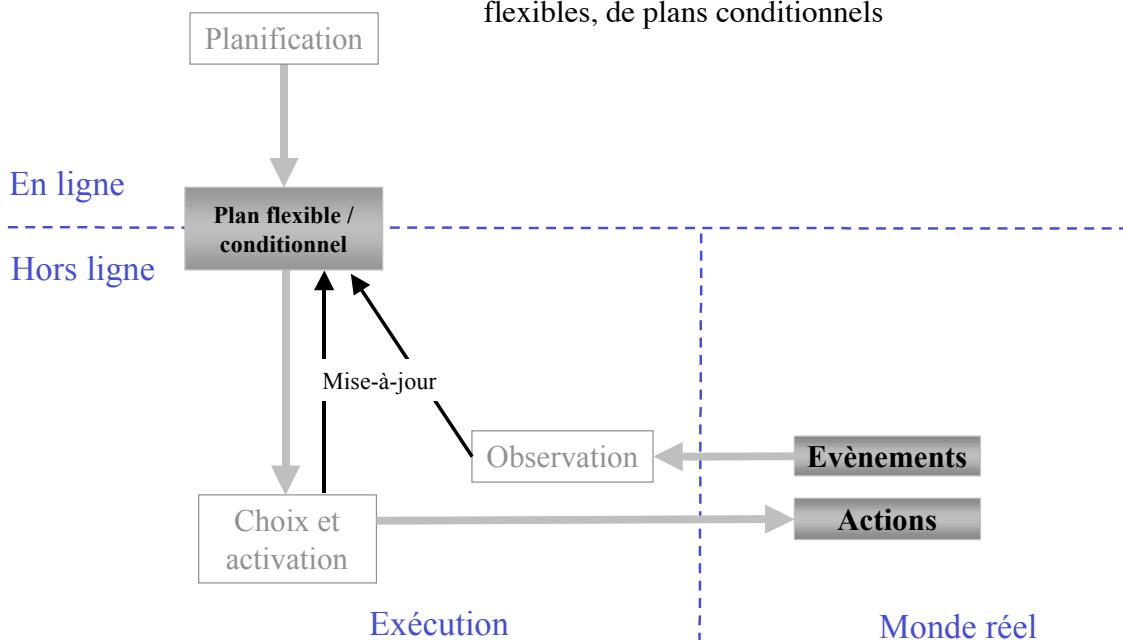


2.3. Planification et exécution



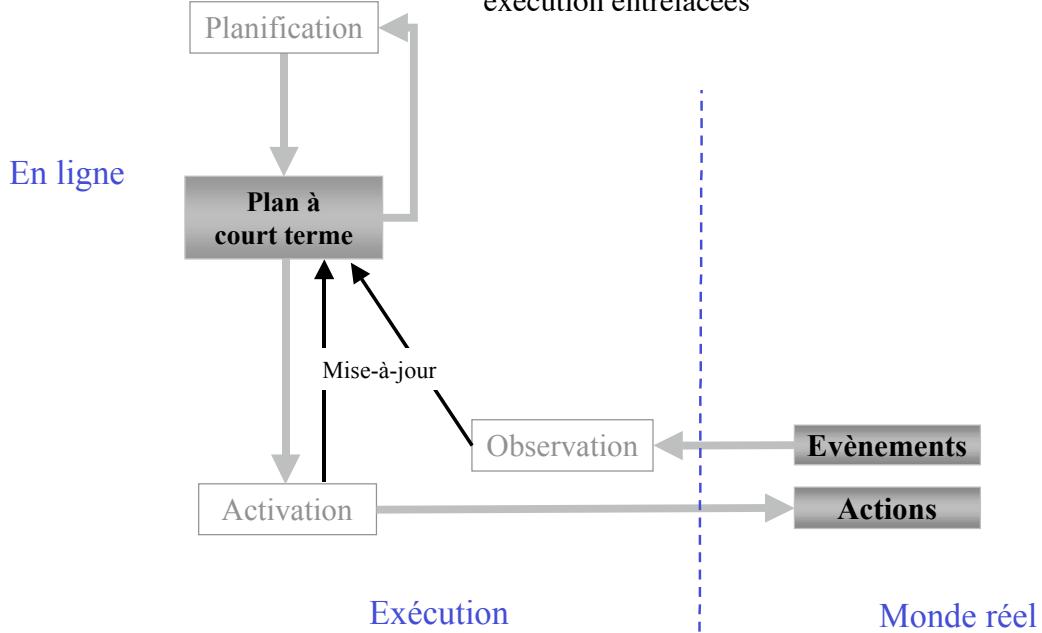
2.3. Planification et exécution

Dans un monde dynamique : synthèse de plans flexibles, de plans conditionnels



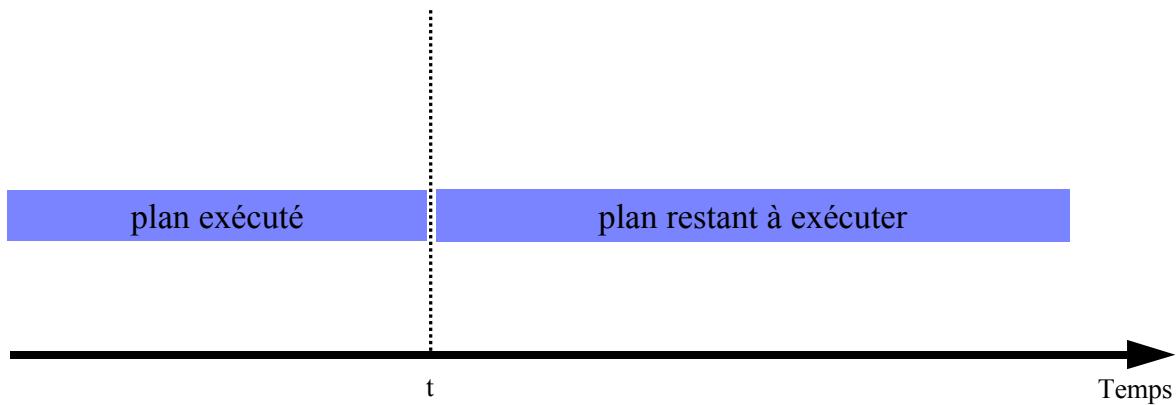
2.3. Planification et exécution

Dans un monde dynamique : planification et exécution entrelacées



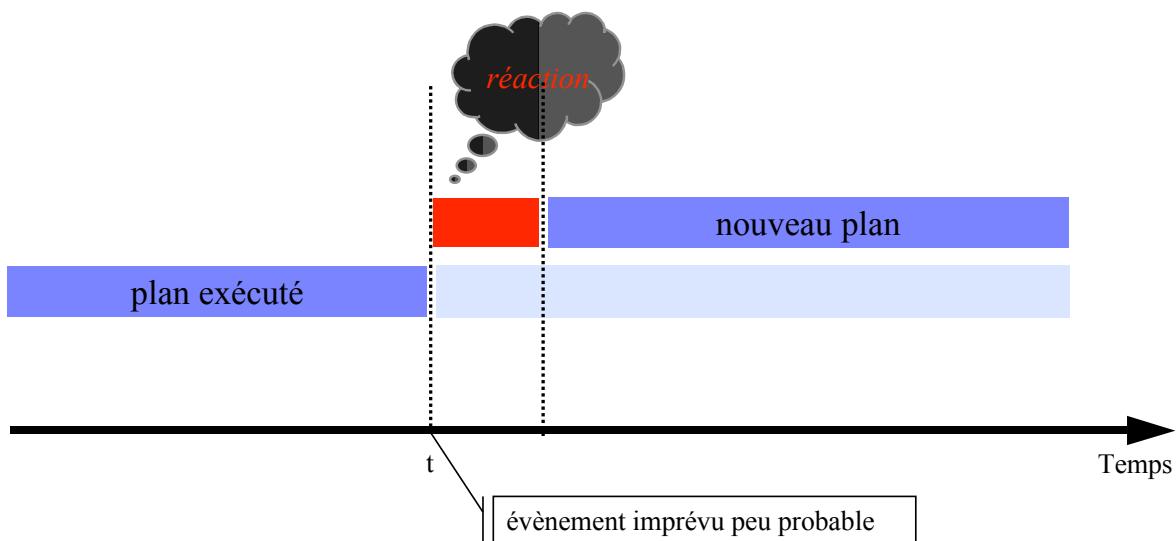
2.3. Planification et exécution

Dans un monde statique : planification puis exécution



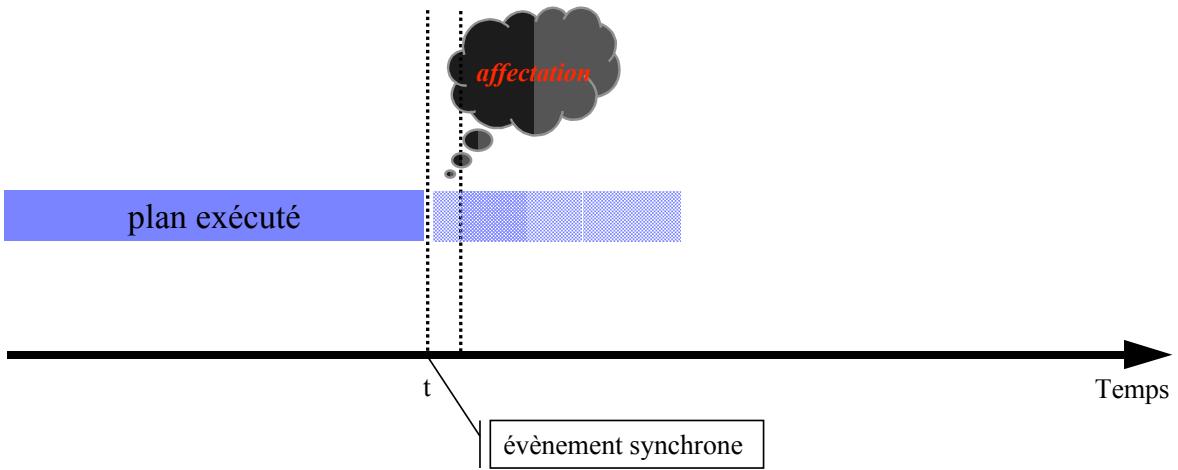
2.3. Planification et exécution

Dans un monde peu dynamique : planification puis exécution avec contrôle d'exécution, et adaptation du plan ou replanification



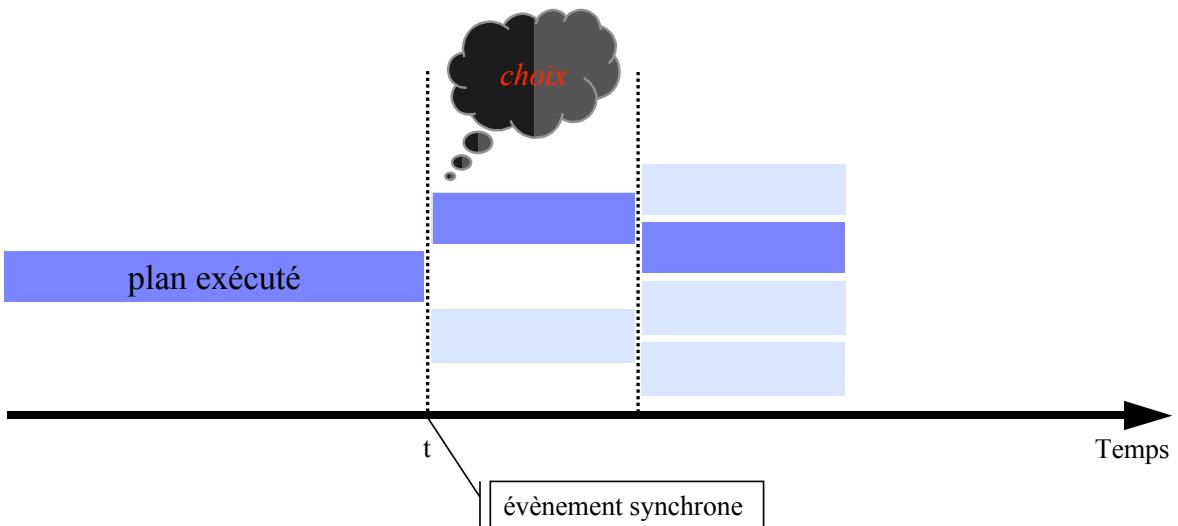
2.3. Planification et exécution

Dans un monde dynamique : synthèse de plans flexibles, de plans conditionnels



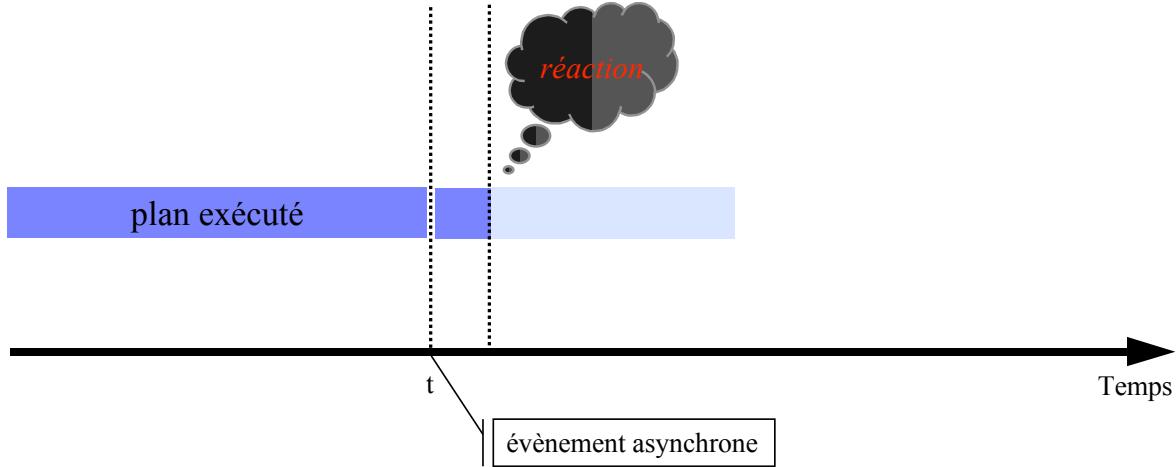
2.3. Planification et exécution

Dans un monde dynamique : synthèse de plans flexibles, de plans conditionnels



2.3. Planification et exécution

Dans un monde dynamique : planification et exécution entrelacées

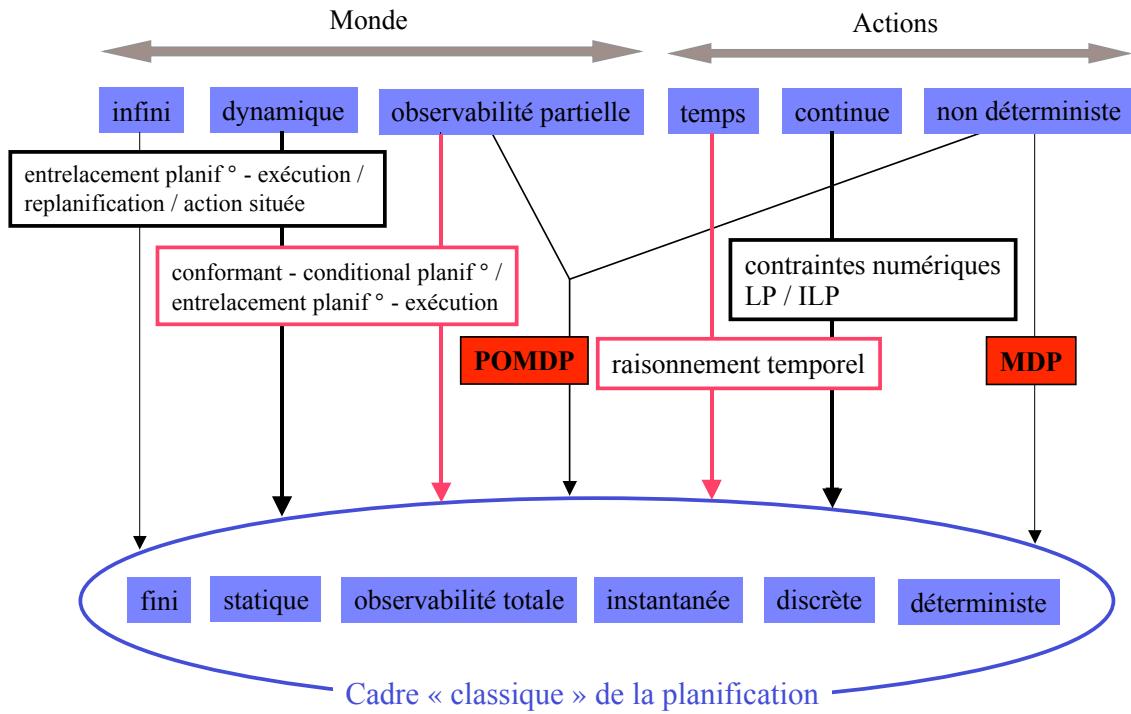


2.4. Applications de la planification

- Les applications potentielles de la planification sont nombreuses et variées :
 - systèmes robotiques autonomes : exploration en milieu hostile, spatial, militaire, chirurgie, manufacture, transport...
 - génération automatique de plans de transport, d'évacuation...
 - aide interactive à l'apprentissage...



2.5. Le cadre classique



2.5. Le cadre classique

- Pourquoi travailler dans ce cadre restrictif ?**
 - développer des algorithmes indépendants du domaine qui soient performants pour des représentations simples de l'action ;
 - tirer ensuite parti des progrès effectués pour enrichir cette représentation tout en restant efficace ;
 - objectif : finir par pouvoir traiter des problèmes réels.
- Gros progrès ces 10 dernières années :**
 - des compétitions spécialisées régulières (IPC) ont été mises en place ;
 - différents types de planificateurs (STRIPS, ADL, experts, sous-optimaux, optimaux, numériques, temporels...) sont peu à peu comparés ;
 - les performances s'améliorent constamment ;
 - les langages de représentation de domaines de planification se sont développés : STRIPS, ADL, PDDL, PDDL 2.1, PDDL 2.2...
 - les benchmarks qui sont utilisés sont de plus en plus inspirés de problèmes réels.

2.5. Le cadre classique

Conférences et compétitions

2004 :

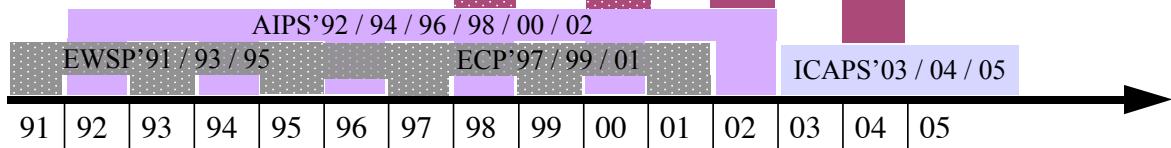
Airport, Pipes, Promela, PRS, Satellite, Settlers, UMTS
Plans -> 1000 actions

20 planificateurs
 domaines-indépendants
 sous-optimaux, optimaux
 PDDL 2.2

14 planificateurs,
 domaines-indépendants, temporels et numériques
 PDDL 2.1

16 planificateurs
 domaine-indépendants, à base de connaissances
 PDDL

5 planificateurs
 domaine-indépendants
 PDDL (Strips, ADL)

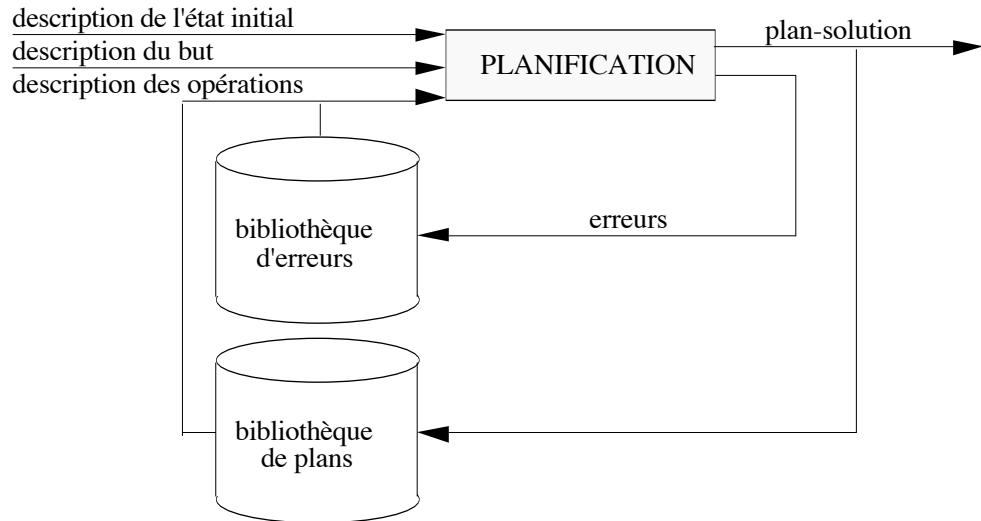


3.1. Les approches essentielles

- **Planification : les différentes approches de l'IA :**
 - 1969 : Logique et démonstration de théorèmes, calcul des situations [Mc Carthy, Hayes, 69] ;
 - 1971 : Synthèse de plans, système STRIPS [Fikes, Nilsson, 71] ;
 - 1974 : Réutilisation de plans, système HACKER [Sussman, 74] ;
 - 1975 : Décomposition en tâches hiérarchisées, système NOAH [Sacerdoti, 75].

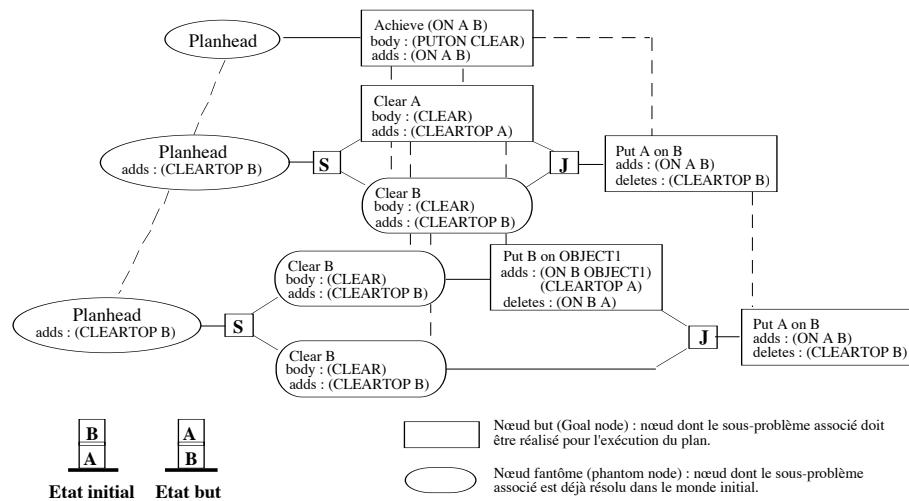
3.1. Les approches essentielles

- La planification comme **réutilisation de plans** (case-based planning). HACKER [Sussman, 74], monde des cubes.



3.1. Les approches essentielles

- La planification comme **expertise** (knowledge-based planning) : systèmes HTN (Hierarchical Task Network). NOAH [Sacerdoti, 75], pannes électromécaniques.

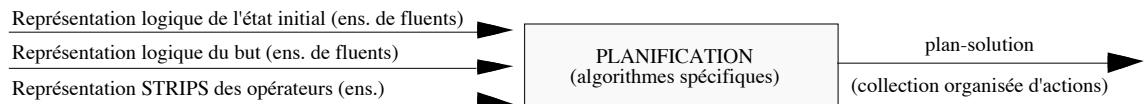


3.1. Les approches essentielles

- La planification comme **synthèse de plans**
 - **Logique** (calcul des situations [McCarthy, Hayes, 69]...)



- **Logique et opérateurs STRIPS** [Fikes, Nilsson, 71] (classical planning)



3.2. Les problèmes posés par la logique

- **Les problèmes posés par la logique :**
 - **Qualification des actions** [McCarthy, 1977] : décrire toutes les conditions préalables à l'exécution des actions. Problème : les conditions initiales sont nombreuses (voir innombrables), par exemple :

Tourner-la-clef-de-contact :

$$[t]\text{tourner-la-clef} \rightarrow [t+1]\text{démarrer}$$

$$[t]\text{tourner-la-clef} \wedge [t]\neg\text{batterie-à-plat} \wedge [t]\neg\text{météorite...} \rightarrow [t+1]\text{démarrer}$$
 - **Ramification des actions** [Finger, 1987] : décrire toutes les conséquences de l'exécution des actions (effets de bord). Problème : ces interdépendances sont nombreuses, par exemple :

Déplacer-le-bureau :

$$[t]\text{bureau-dans-pièce1} \rightarrow [t+1]\text{bureau-dans-pièce2}$$

$$[t]\text{bureau-dans-pièce1} \wedge [t]\text{objet1-dans-pièce1...} \rightarrow$$

$$[t+1]\text{bureau-dans-pièce2} \wedge [t+1]\text{objet1-dans-pièce2...}$$

3.2. Les problèmes posés par la logique

- **Les problèmes posés par la logique :**

- **Problème du décor (persistance, frame problem) [McCarthy, Hayes 1969] :** décrire tout ce que l'exécution d'une action ne modifie pas. Problème : d'innombrables propriétés ne sont pas modifiées, par exemple :

Ouvrir-la-porte :

$$[t] \text{ouvrir-la-porte} \rightarrow [t+1] \text{porte-ouverte}$$

$$[t] \text{ouvrir-la-porte} \wedge [t] \text{maison-bleue...} \rightarrow [t+1] \text{maison-bleue...} \quad (\text{FA})$$



- **Erosion des croyances [Shoham, McDermott, 1988]** : la tendance à persister diminue au cours du temps, par exemple :

$$[6:00] \text{Pierre-au-lit} : [8:05] \text{Pierre-au-lit?} \quad [13:00] \text{Pierre-au-lit?}$$

- **Persistances propres aux littéraux [Cordier, Siegel, 1995]** : selon ce qu'il représente, chaque fluent persiste plus ou moins longtemps, par exemple : *immobile, endormi, aveugle, ¬mort, mort...*

3.3. Tableau récapitulatif

Catégorie	Synthèse de plans		Réutilisation de plans	Expertise	Epistémique
Approche	Logique	Logique et op. STRIPS	Basée sur les cas	Systèmes experts	Connexionniste
Aspects modélisés	<ul style="list-style-type: none"> • Synthèse formelle d'un plan à partir d'opérateurs 	<ul style="list-style-type: none"> • Synthèse formelle d'un plan à partir d'opérateurs STRIPS 	<ul style="list-style-type: none"> • Mémorisation, indexation, réutilisation et modification des plans, • Apprentissage Symbolique 	<ul style="list-style-type: none"> • Connaissance des experts humains sur un domaine 	<ul style="list-style-type: none"> • Réaction rapide à des saillances pertinentes du monde
Représentation	Symbolique	Symbolique	Symbolique	Symbolique	Connexionniste
Apprentissage Formalisation	Non Oui	Non Oui	Oui Non	Non Non	Envisageable Non
Entrées	<ul style="list-style-type: none"> • Etat initial (fbf) • But (fbf) • Opérateurs (fbf) 	<ul style="list-style-type: none"> • Etat initial (fluents) • But (fluents) • Opérateurs 			
Sortie	Fbf représentant un plan-solution d'opérateurs	Plan-solution d'opérateurs STRIPS	Plan-solution d'opérateurs	Plan-solution avec des opérateurs de différents grains	Action située
Mécanismes utilisés pour planifier	<ul style="list-style-type: none"> • Démonstration de théorème 	<ul style="list-style-type: none"> • Recherche dans un espace d'états • Recherche dans un espace de plans • GRAPHPLAN • S A T • C S P 	<ul style="list-style-type: none"> • Indexation de plans • Modification de plans • Correction de plans 	<ul style="list-style-type: none"> • Règles de production • Décomposition hiérarchique de tâches • Autres (ad hoc) 	<ul style="list-style-type: none"> • Architecture connexionniste