

# Raisonnement symbolique et géométrie pour la robotique mobile

*J. Guitton, J.L. Farges*

ONERA

The logo for ONERA, consisting of the word "ONERA" in a serif font above a blue curved line that spans the width of the text.

## Plan

- Introduction
- Vers la planification hybride
  - Types de planification
  - Comparaison d'approches
  - Connexion planificateurs
- Implémentation de l'architecture
  - Les composants
  - Le formalisme
  - L'algorithme
  - L'interaction
  - Exemple illustratif
- Conclusion

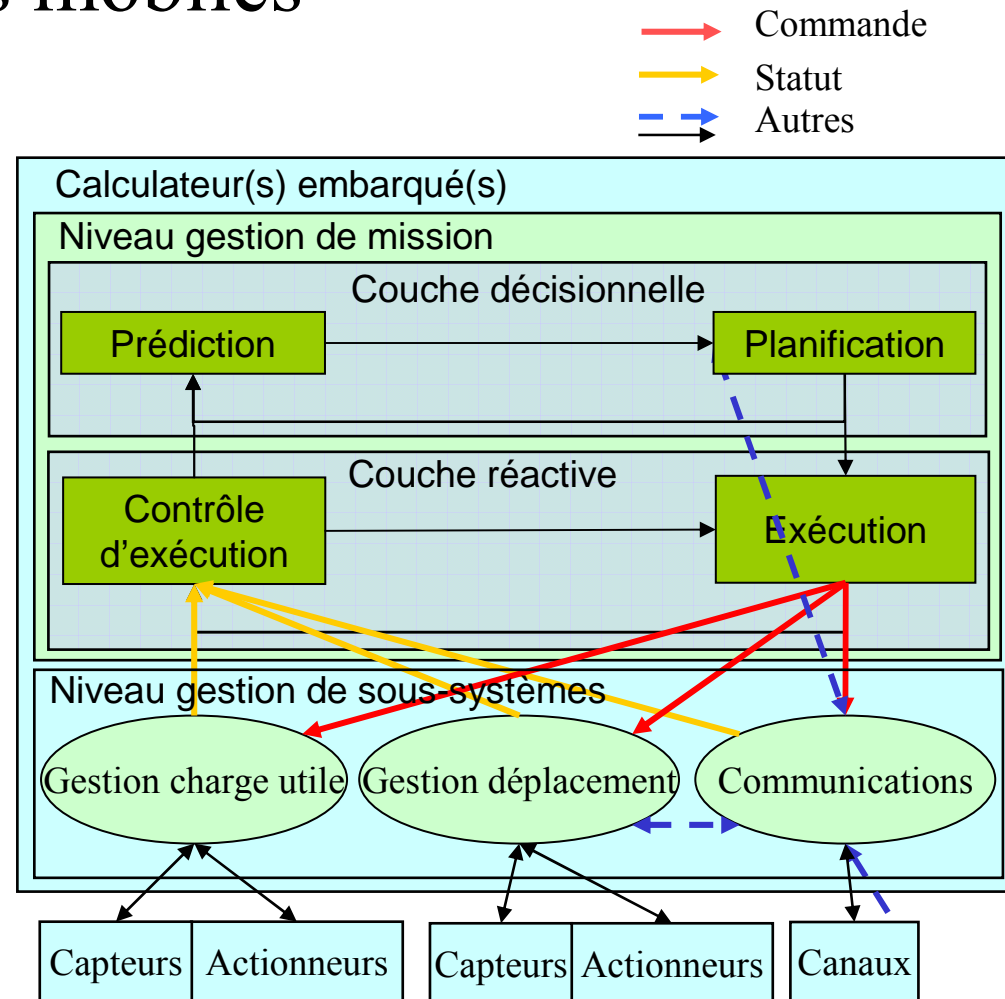
## Introduction (1/2)

- **Autonomie des robots mobiles**

- Gestion de mission
- Couche décisionnelle
- Planification

- **Raisonnement**

- Symbolique
  - Actions charge utile
- Géométrique
  - Déplacements



## Introduction (2/2)

- Approches pour la prise en compte de déplacements
  - Modélisation sous forme de course d'orientation (Smith, 2004)
  - Affectation de tâches + planification de chemins (Rabideau et al., 1999)
  - Interpréteur de grammaire + algorithmes pour les graphes (voyageur de commerce, plus court chemin...) (Brummit et Stentz, 1998)
  - Pas de prise en compte des contraintes géométriques pour les actions.
- Connexion planification de tâches - planification de chemins
  - Hiérarchie: Tâches > chemins
  - Planificateur de chemins utilisé par le planificateur de tâches (Lamare et Ghallab, 1998)
  - Approches pour des problèmes spécifiques (Zacharias et al., 2006)
  - Planification de tâches et chemins intégrée (Cambon et al., 2004)
  - Pas d'évaluation de l'intérêt, pas de définition générale de la connexion

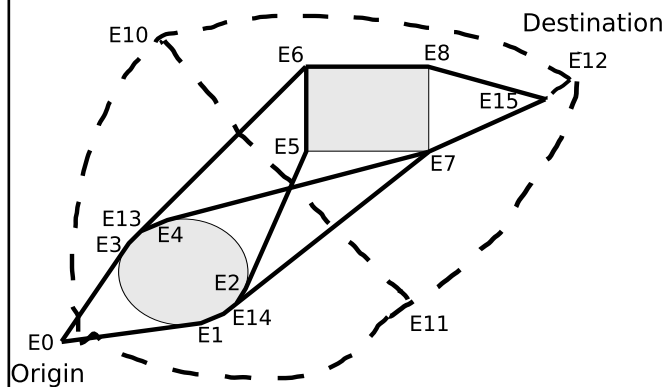
## Vers la planification hybride - Types de planification (1/2)

- Planification de tâches
  - Planificateurs dépendants du domaine
  - Planificateurs indépendants du domaine
    - Opérateur  $o = \langle \text{entête}(o), \text{préconditions}(o), \text{effets}(o) \rangle$
    - Problème de satisfaction de buts  $P_B = \langle O, \text{état initial}, \text{buts} \rangle$
  - Planificateurs configurables en fonction du domaine
    - Utilisation de connaissances spécifiques au domaine
    - Règles de contrôle de la recherche du plan
    - Réseaux de tâches hiérarchiques
  - Réseaux de tâches hiérarchiques
    - Méthode  $m = \langle \text{entête}(m), \{ \text{décomposition}_1(m), \text{décomposition}_2(m), \dots \} \rangle$
    - $\text{décomposition}_i(m) = d_i = \langle \text{préconditions}(d_i), \text{tâches}(d_i) \rangle$
    - Tâche : soit opérateur, soit méthode
    - Problème de planification de tâches  $P_T = \langle O, M, \text{état initial}, \text{tâches} \rangle$

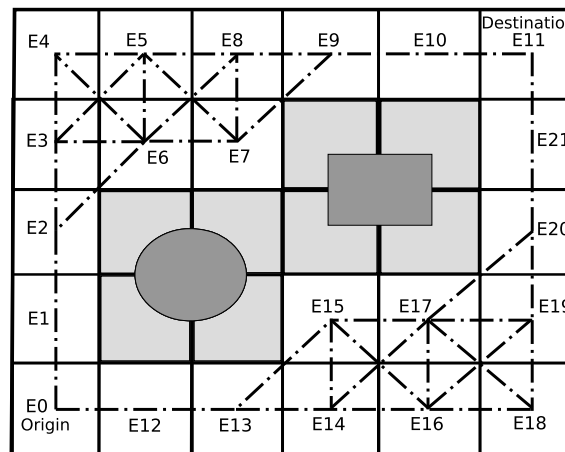
## Vers la planification hybride - Types de planification (2/2)

- Planification de chemins

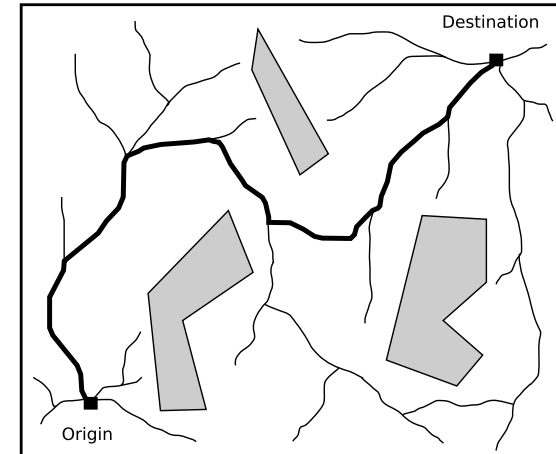
- Trouver un chemin de l'origine à la destination satisfaisant
  - Les contraintes de l'environnement
  - Les contraintes de la cinématique du robot
- Approches de construction de graphe dans l'espace de configuration à base de:



squelettes



cellules



graphes aléatoires

- Optimisation du chemin par A\*, Dijkstra...

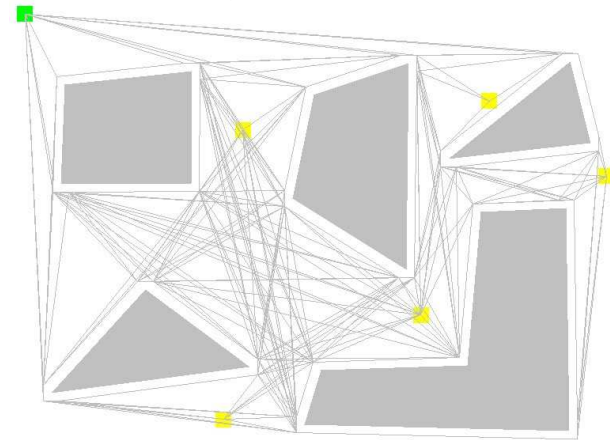
## Vers la planification hybride - Comparaison d'approches (1/2)

- Problème

- Domaine des *rovers* (3ème compétition de planification)
- Graphe de visibilité
- Problèmes mono robot
- 5 problèmes: nombre d'objectifs croissant

- Solution par planification de tâches uniquement

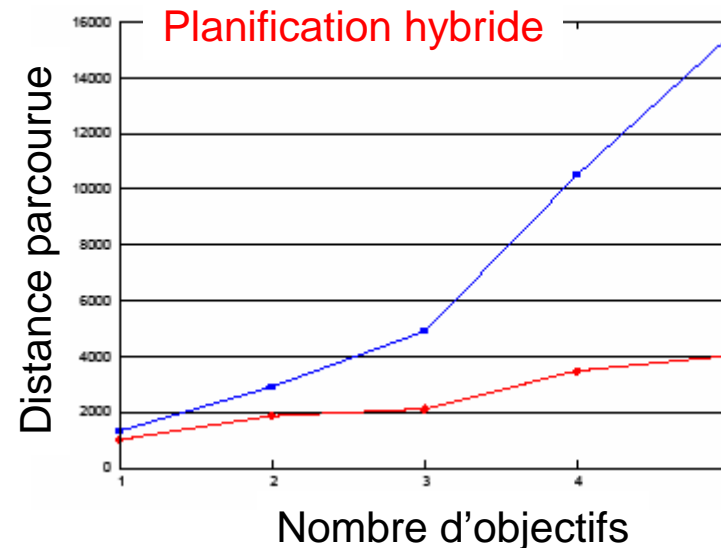
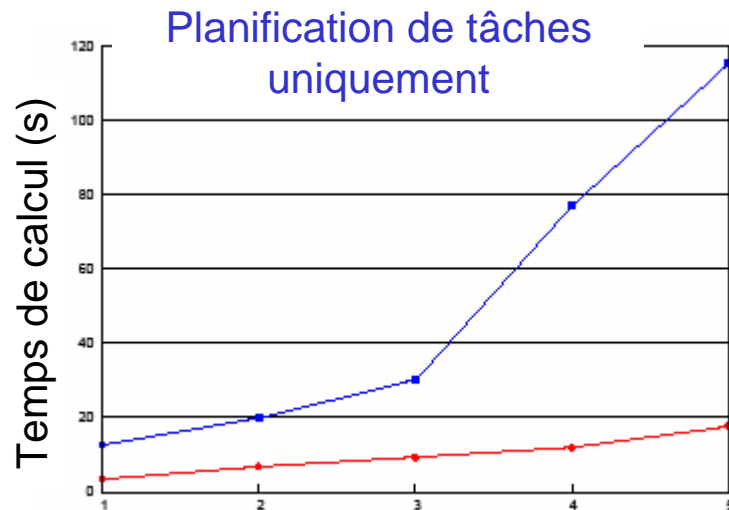
- Méthode naviguer
  - De origine à destination
  - Évite les boucles
- Opérateurs
  - naviguer : de point de passage a voisin
  - marquage / démarquage
- Prédicats indiquant les voisins



- Solution par planification hybride

- Opérateur naviguer
  - De origine à destination
  - Sous-traite à un planificateur de chemin basé sur Dijkstra: minimise la distance

## Vers la planification hybride - Comparaison d'approches (2/2)



- Résultats Nombre d'objectifs
  - Net avantage à la planification hybride pour les deux critères
- Commentaires
  - Dans la 3ème compétition SHOP2 utilise des appels LISP
  - La méthode naviguer peut être modifiée pour tenir compte de la distance
    - Diminution de la distance parcourue
    - Augmentation du temps de calcul



## Vers la planification hybride - Connexion planificateurs (1/2)

- Types de connexions

- Hiérarchique

- Planification de tâche complète puis requête vers le planificateur de chemin
    - Raffinement en planification de chemin

- Hiérarchique avec retour d'information

- Le premier opérateur impliquant un chemin impossible est indiquée
    - La recherche du planificateur de tâches reprend au niveau de cet opérateur

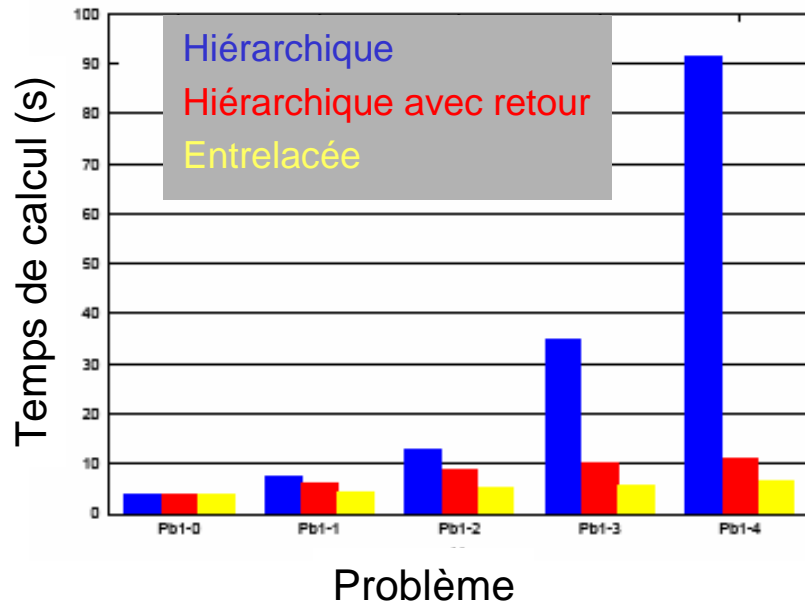
- Entrelacée

- Appel au planificateur de chemin comme **précondition** à l'opérateur

- Problème

- Pour chaque objectif deux points destination **logiquement** possibles pour l'opérateur
  - Pb1-0 : les deux points **géométriquement** possibles pour les 4 objectifs
  - ...
  - Pb 1-4 : seulement 1 point **géométriquement** possible pour les 4 objectifs

## Vers la planification hybride - Connexion planificateurs (2/2)

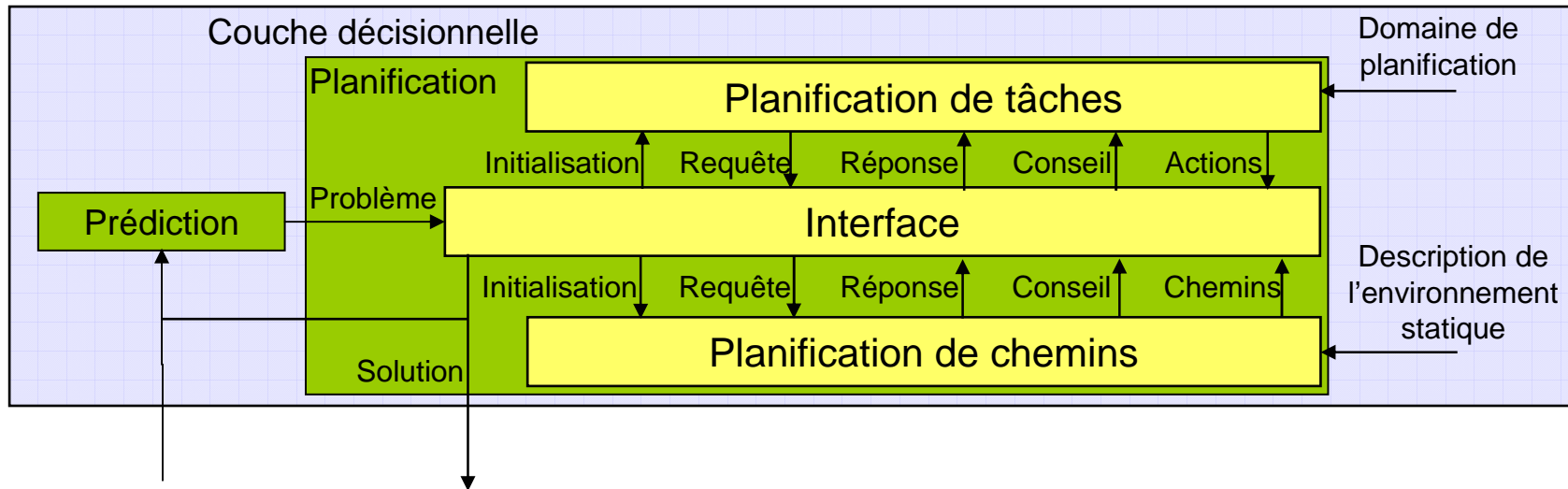


Problème	Hierarchique	Hierarchique avec retour	Entrelacée
Pb1-0	4	4	4
Pb1-1	8	8	5
Pb1-2	17	11	6
Pb1-3	43	13	7
Pb1-4	112	14	8

Nombre de requêtes du planificateur de tâches au planificateur de chemins

- Résultats
  - Corrélation entre le temps de calcul et le nombre de requêtes
  - L'exécution entrelacée conduit à un temps de calcul plus faible
- Commentaires
  - La planification entrelacée est moins aveugle que la planification hiérarchique
  - Avec la planification entrelacée les requêtes impossibles ne sont pas répétées

## Implémentation de l'architecture - Les composants (1/2)



- Planificateur de tâches
  - Tâches HTN partiellement ordonnées
  - Domaine
    - Opérateurs
    - Méthodes
  - Initialisation
    - État initial
    - Tâches à décomposer

## Implémentation de l'architecture - Les composants (2/2)

- Planificateur de chemins
  - Répond aux requêtes :
    - Chemin entre deux zones spécifiées par des contraintes géométriques
    - Possibilité de respecter des contraintes de mouvement pendant l'action
  - Prend en compte les contraintes cinématiques si nécessaire
  - Maintien un historique de manière à :
    - Suggérer des améliorations au planificateur d'action
    - Optimiser son propre temps de calcul
  - Pour le moment : graphe de visibilité + Dijkstra (Cf. tests précédents)
- Interface
  - Évite la connexion directe pour :
    - Changer plus facilement de planificateur de chemin
    - Permettre la prise en compte de problèmes multi robots
  - Buts :
    - Initialiser les planificateurs de manière cohérente
    - Gérer les échanges entre eux

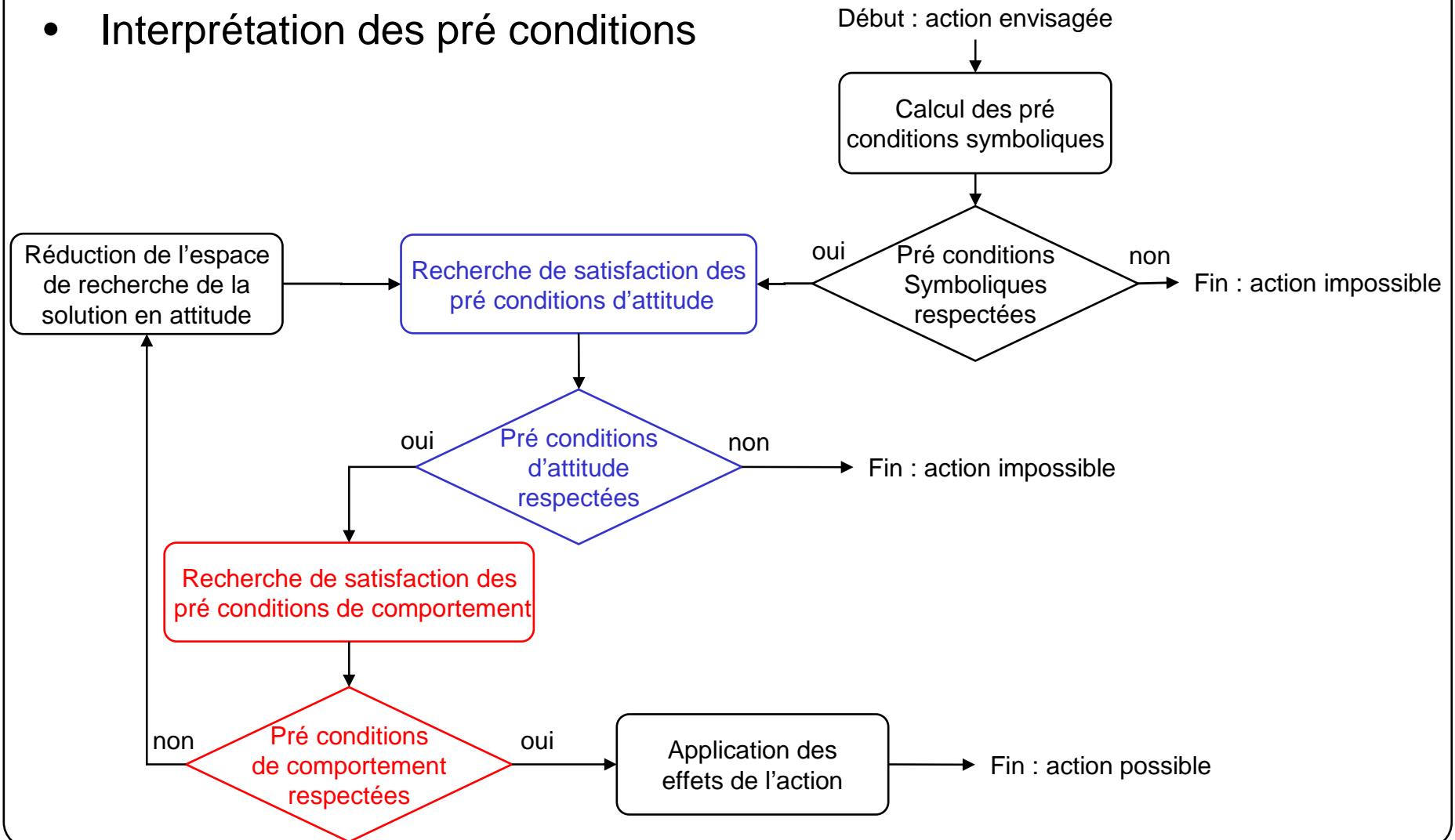
## Implémentation de l'architecture - Le formalisme (1/2)

- Absence d'opérateur naviguer
- Les mouvements sont des pré conditions indiquant des contraintes géométriques pour l'exécution de certaines actions
  - Préconditions d'attitude pour commencer l'action
  - Préconditions de comportement pour mener l'action
  - Correspondent à des points ou des domaines

```
(Operator (!Balayer ?r ?o)
;; préconditions symboliques
((robot ?r) (zone ?o) (capteur_lateral ?c) (a ?r ?c) (est_calibre ?c))
;; préconditions d'attitude
((distance(r.pos,o.pos) >= 10)(distance(r.pos,o.pos) <= 20)
(angle(r.pos,o.pos,r.cap) = 90)(r.vitesse = r.vitesse_max))
;; précondition de comportement
((duree 10)(constant(r.cap))(constant(r.vitesse)))
;; effets
((a_image_synthetique ?r ?o)))
```

## Implémentation de l'architecture - Le formalisme (2/2)

- Interprétation des pré conditions



## Implémentation de l'architecture - L'algorithme (1/2)

- Nœud = (état, liste de tâches)
- Développement de nœud récursif à partir de (état initial, tâches à décomposer)
- Développer (état, liste de tâches)
  - Si liste de tâches vide un plan a été trouvé
  - Sinon
    - Si la première tâche est un opérateur traiter opérateur (état, liste de tâches)
    - Sinon traiter méthode (état, liste de tâches)

## Implémentation de l'architecture - L'algorithme (2/2)

- Traiter opérateur (état, liste de tâches)
  - Le retrouver par son nom et unifier les préconditions symboliques
  - Pour chaque unification
    - Émettre une requête de satisfaction des préconditions d'attitude
    - Si ok
      - Émettre une requête de satisfaction des préconditions de comportement
      - Si ok
        - » Appliquer les effets et développer (état résultant, liste de tâches réduite)
- Traiter méthode (état, liste de tâches)
  - La retrouver par son nom
  - Pour chaque décomposition
    - unifier les préconditions symboliques
    - Pour chaque unification
      - Développer (état, tâches(décomposition) + liste de tâches réduite)



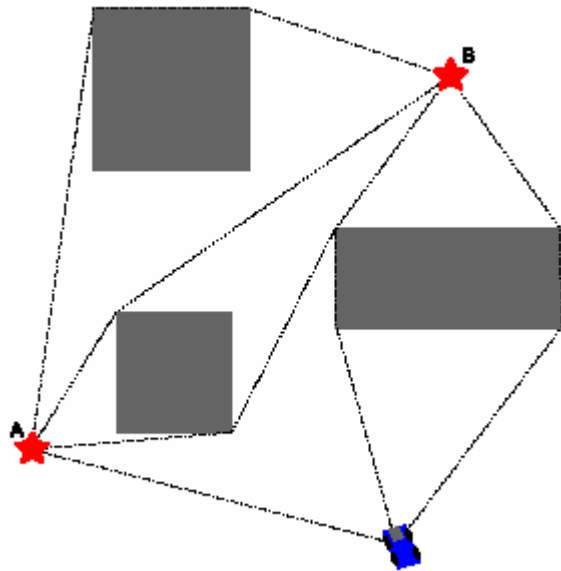
## Implémentation de l'architecture - L'interaction (1/2)

- Vocabulaire commun pour l'interaction
  - Concepts (robot, objectif...)
    - Permet d'identifier les agents et les objets physiques
  - Propriétés globales (durée...)
    - Associées à un mouvement
  - Propriétés de concepts (r.cap, o.position...)
    - Spécifiques à chaque concept
    - Ontologie possible par type de robot mobile
  - Règles (=, constant...)
    - S'appliquent aux propriétés
- Éléments encapsulés dans des messages

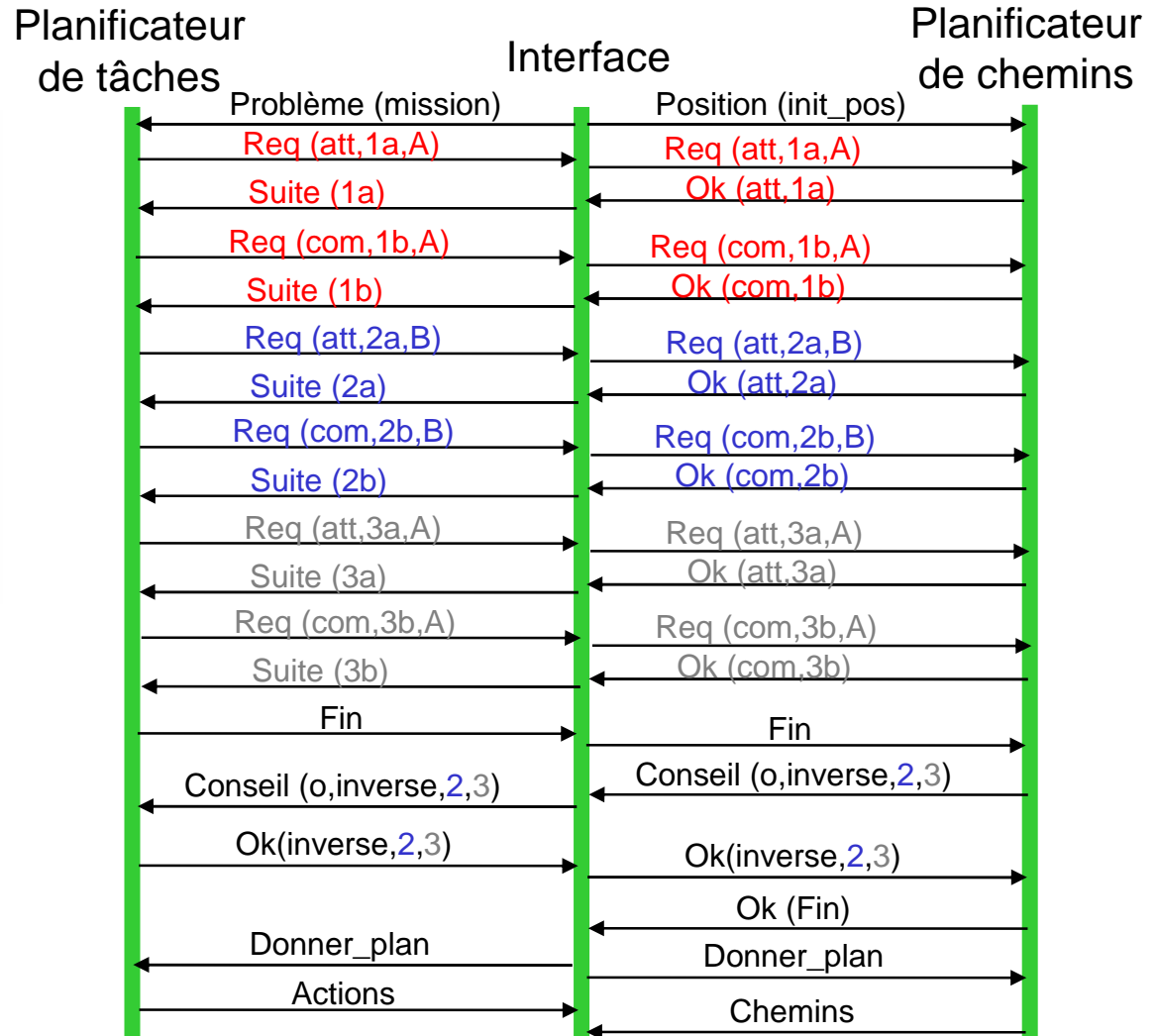
## Implémentation de l'architecture - L'interaction (2/2)

- Requêtes
  - Requêtes de planification
    - Type (attitude, comportement), agent, identifiant d'action, contraintes
  - Requêtes système
    - Initialisation, étape suivante, finalisation...
  - Requêtes d'information
    - Demande de conseil heuristique (quel est l'objectif le plus proche?...)
- Conseils
  - Type (h, o, r), contenu, agent, actions concernées
  - Heuristiques
  - D'optimisation
    - Spontané et basé sur les métriques du planificateur de chemins
  - De réparation
    - Pour orienter la reprise du planificateur de tâches

# Implémentation de l'architecture - Exemple illustratif



- Tâches
  - Photo en A
  - Photo en B
  - Prélèvement en A



## Conclusion

- **Comparaisons expérimentales**

- La planification hybride présente un avantage important en termes de temps de calcul et de qualité de solution par rapport à la planification de tâches
- L'exécution entrelacée des planificateurs de tâches et de chemins permet de réduire le nombre de requêtes et le temps de calcul par rapport à l'exécution hiérarchique
- D'où l'intérêt d'une architecture de planification avec:
  - Un planificateur de tâches intégrant des pré conditions géométriques
  - Un planificateur de chemins
  - Des échanges de requêtes et de conseils à l'aide d'un langage commun

- **Perspectives**

- Extension au multi agent
- Extension à la sélection de buts
- Contexte en ligne et réparation de plan
- Test sur des missions de recherche et sauvetage