

Un robot conférencier capable de planifier ses missions pour le « AAI Robot Challenge »

Éric Beaudry

<http://planiart.usherb.ca/~eric/>

Étudiant à la maîtrise en informatique

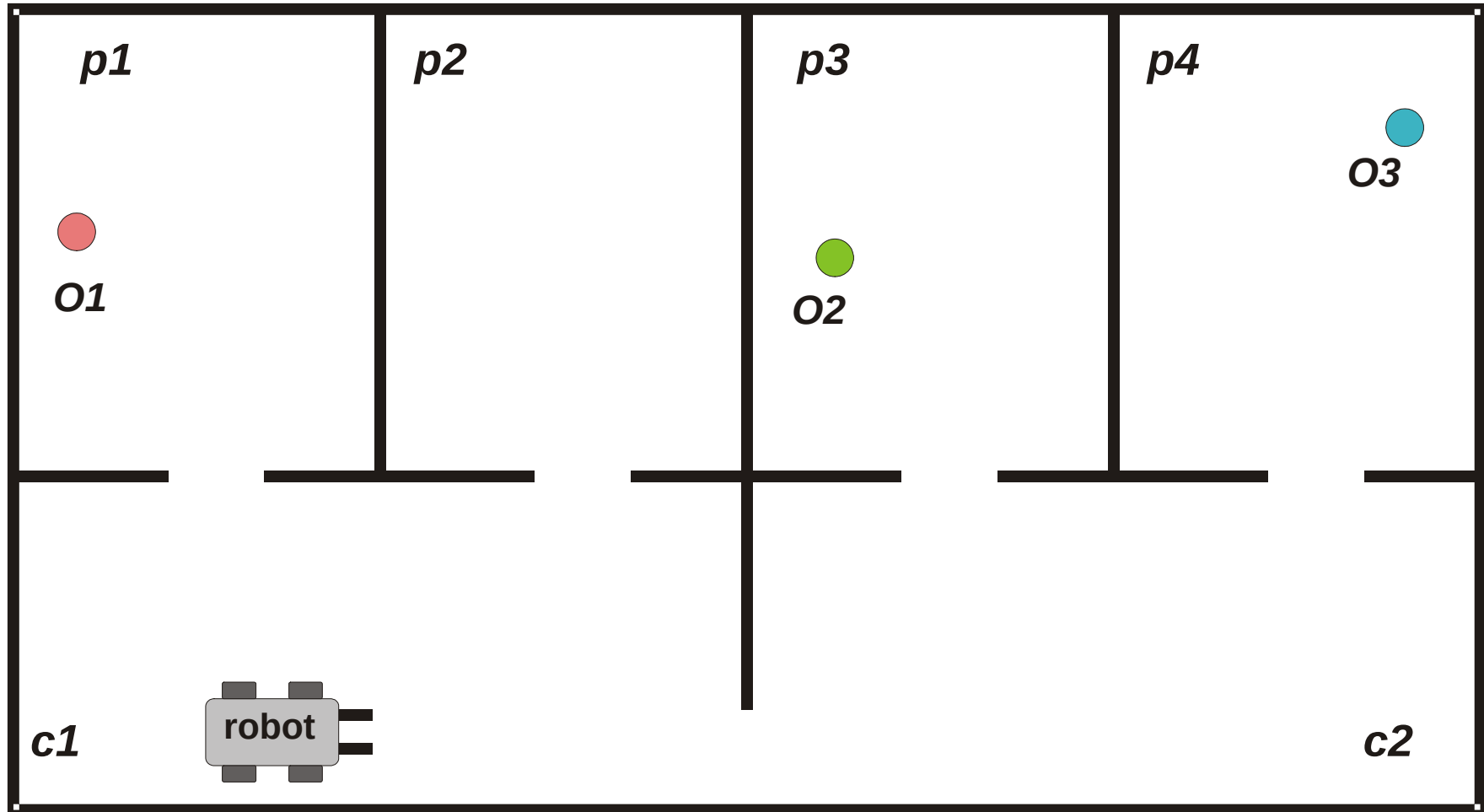
Laboratoires Planiart et Laborius

11 avril 2006

Plan de la présentation

- Applications et domaines robotiques (« AAI Challenge »)
- Introduction robotique à la robotique
- Planification en intelligence artificielle
- Planificateur ConfPlan
- Intégration de ConfPlan dans une architecture décisionnelle
- Expérimentations

Un robot livreur de colis

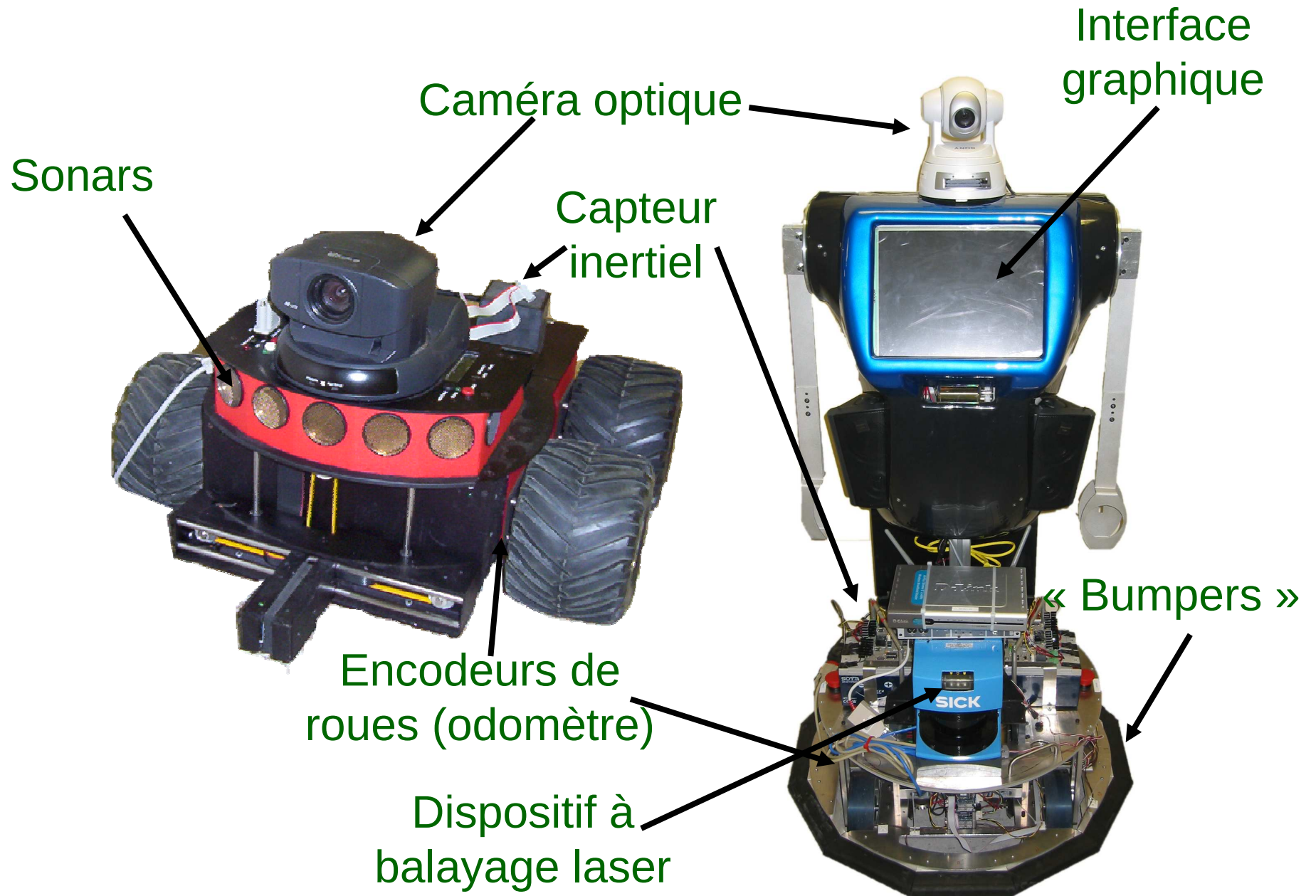


Le « AAAI Robot Challenge »

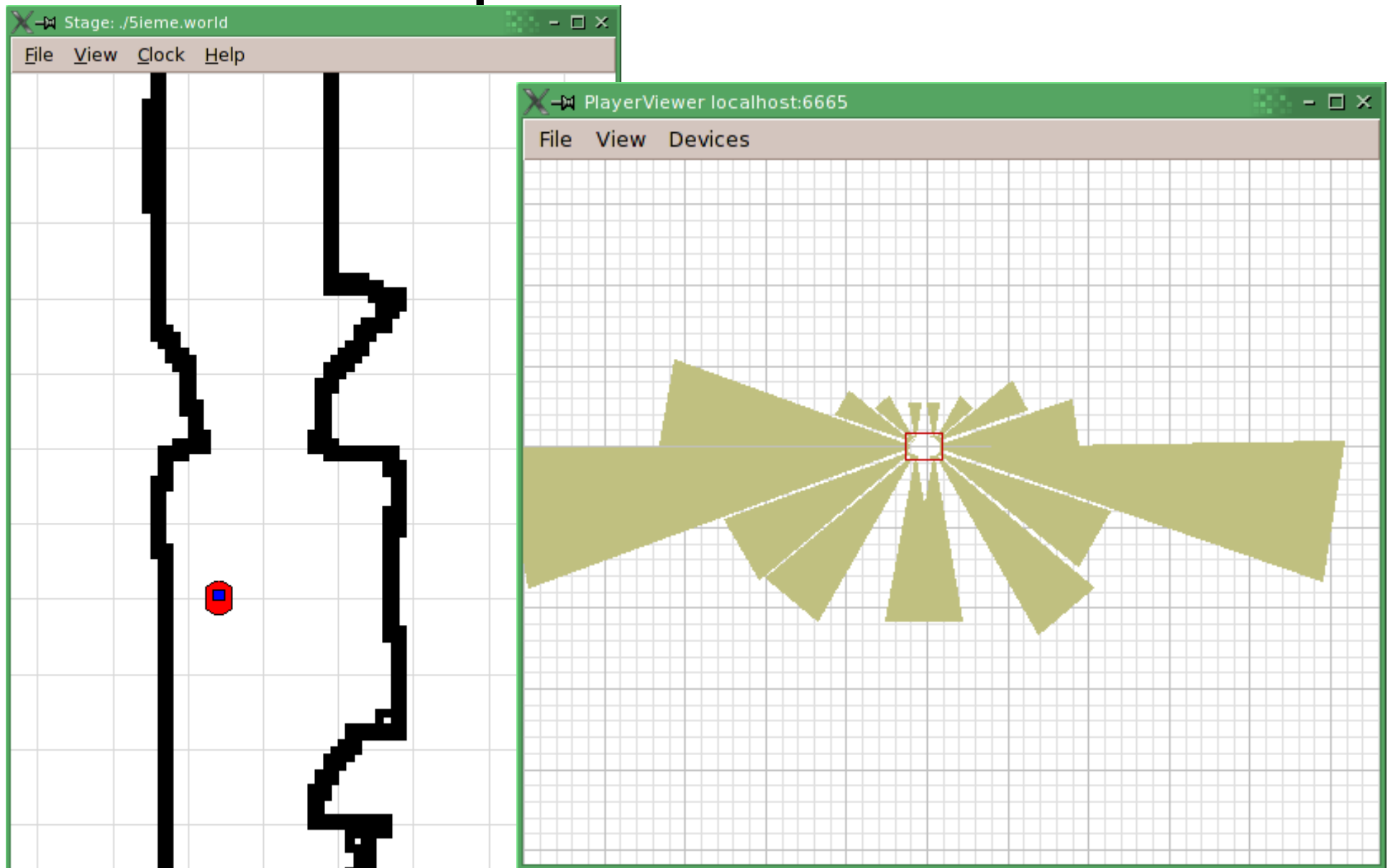
- Réaliser un robot mobile et autonome capable de participer à une conférence scientifique
 - Trouver le kiosque d'inscription et s'y inscrire
 - Donner une conférence
 - Assister à des conférences
 - Réaliser des tâches bénévoles : livrer des messages ou des objets, surveiller des salles, etc.
 - Prendre des photos
 - Interagir avec les autres participants



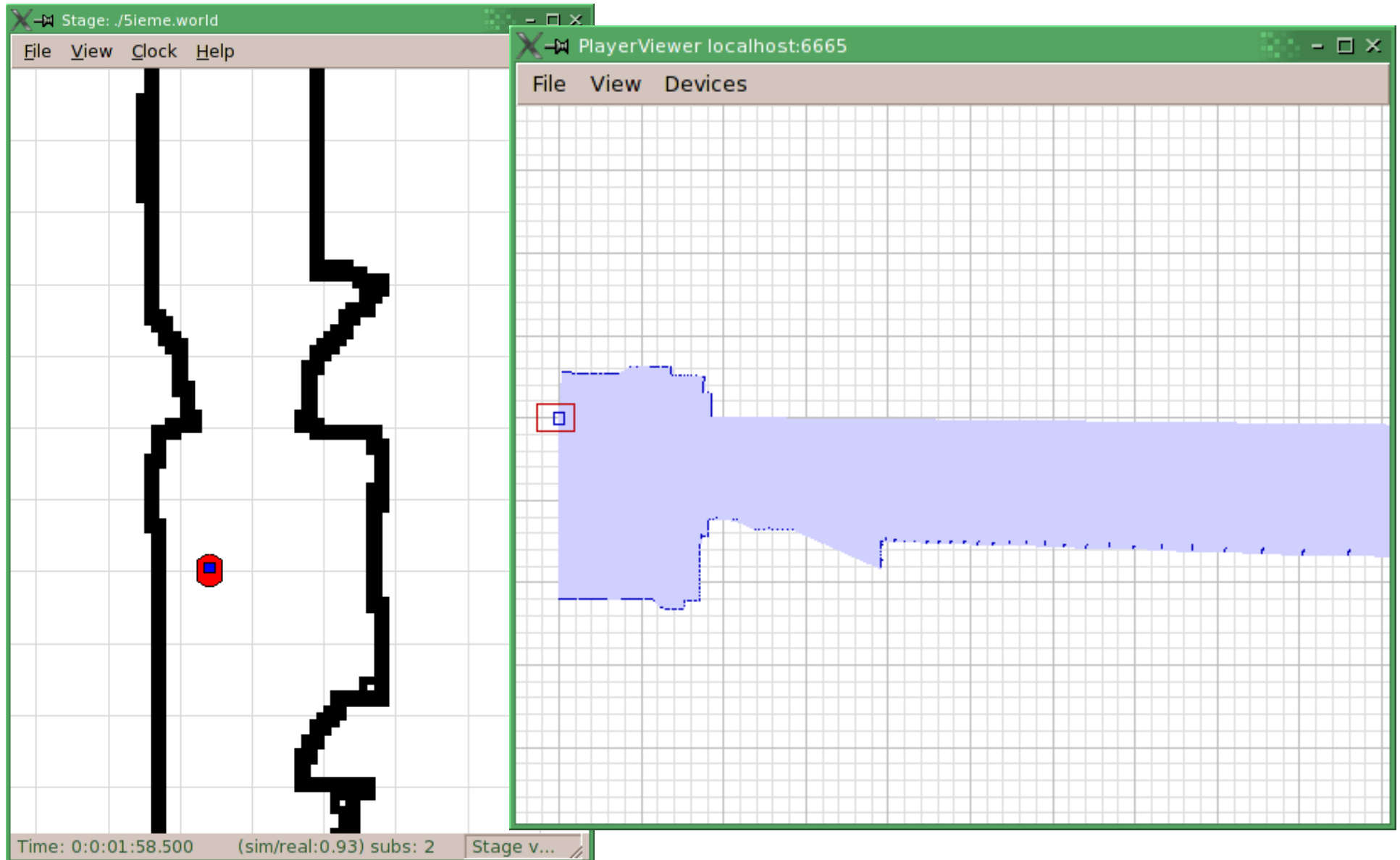
Capteurs d'un robot

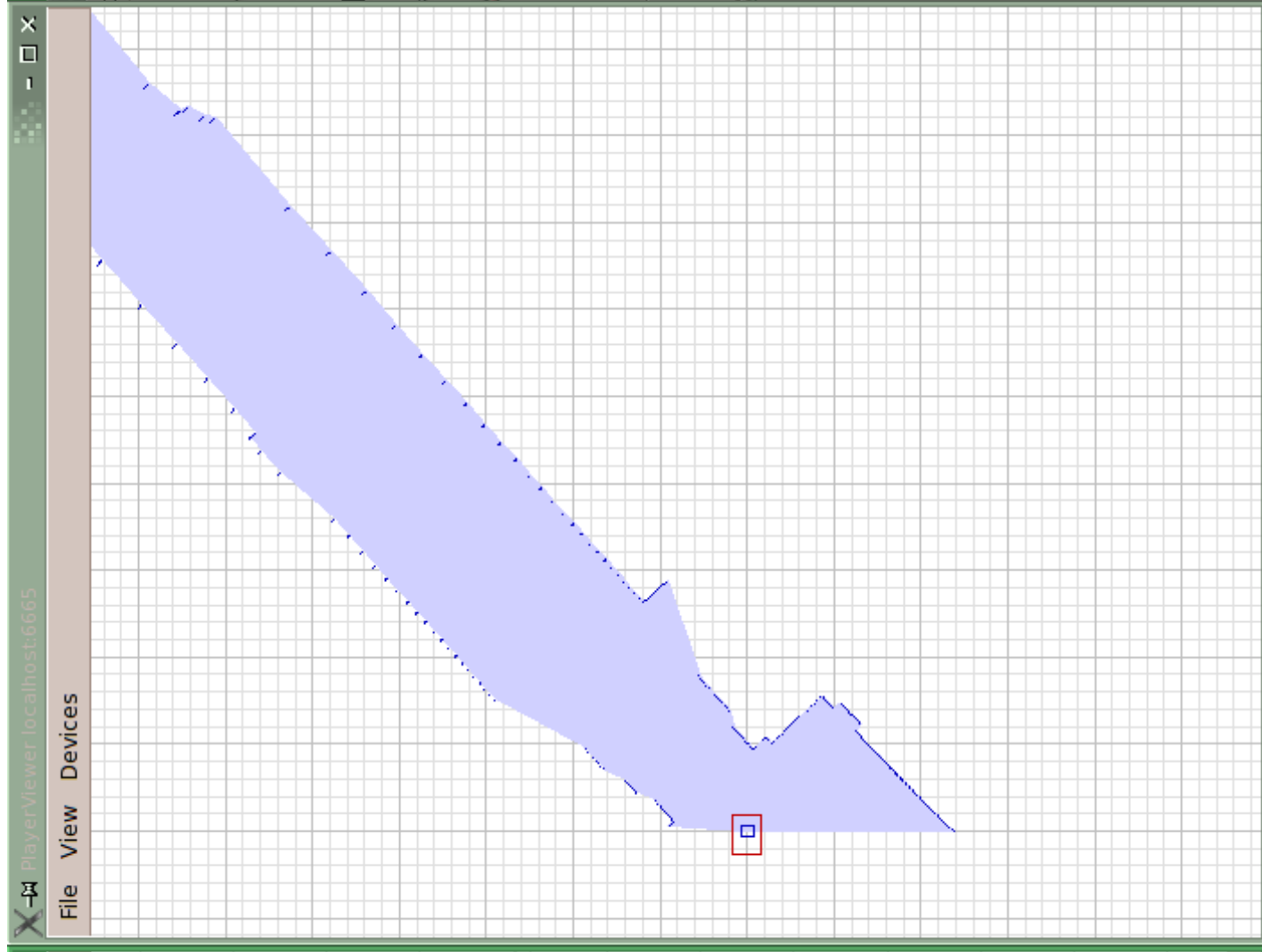
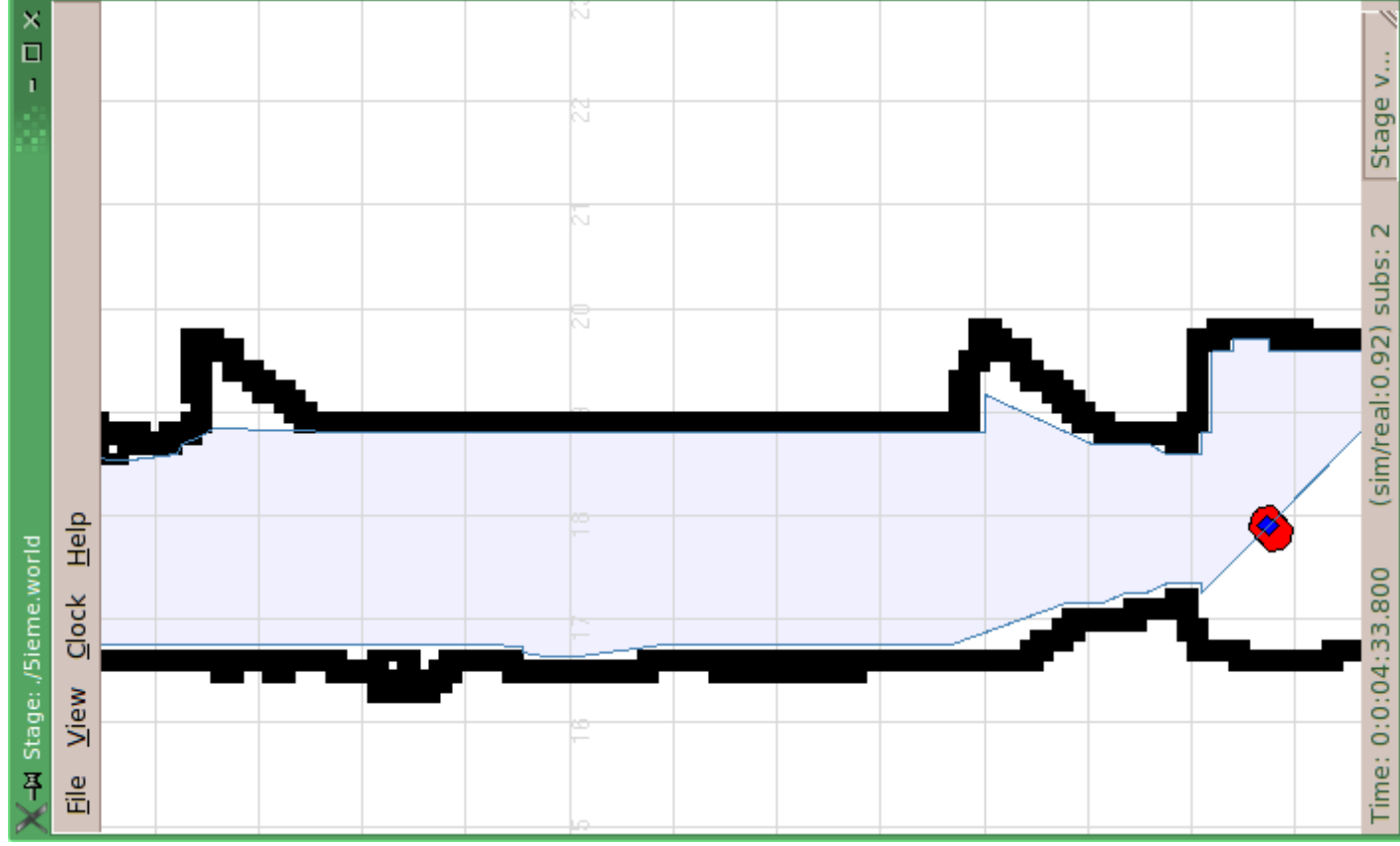


Capteurs : sonars

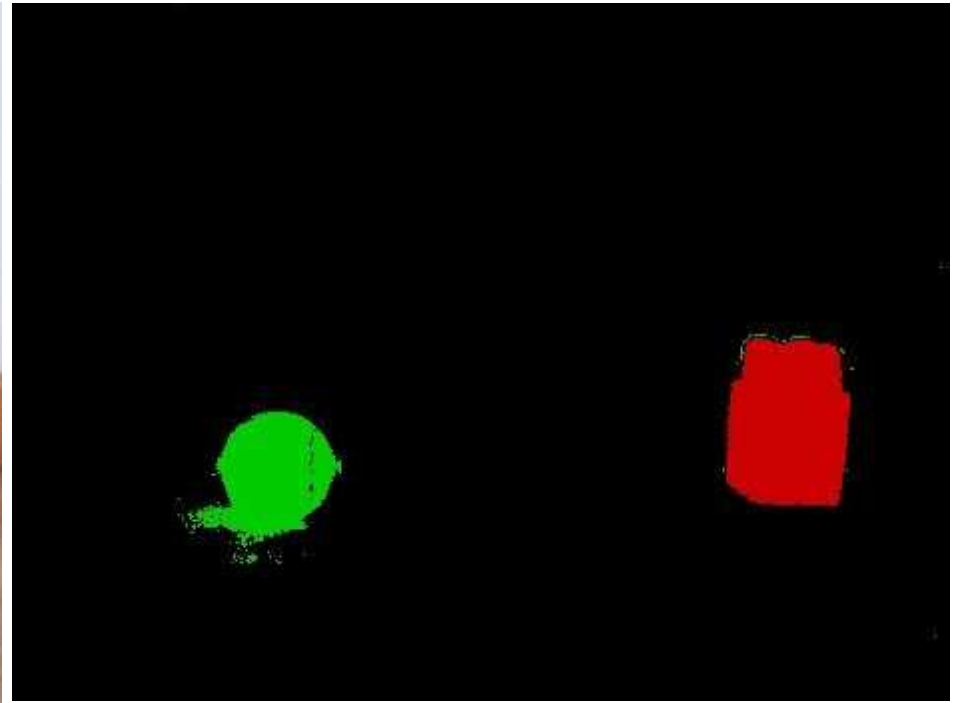
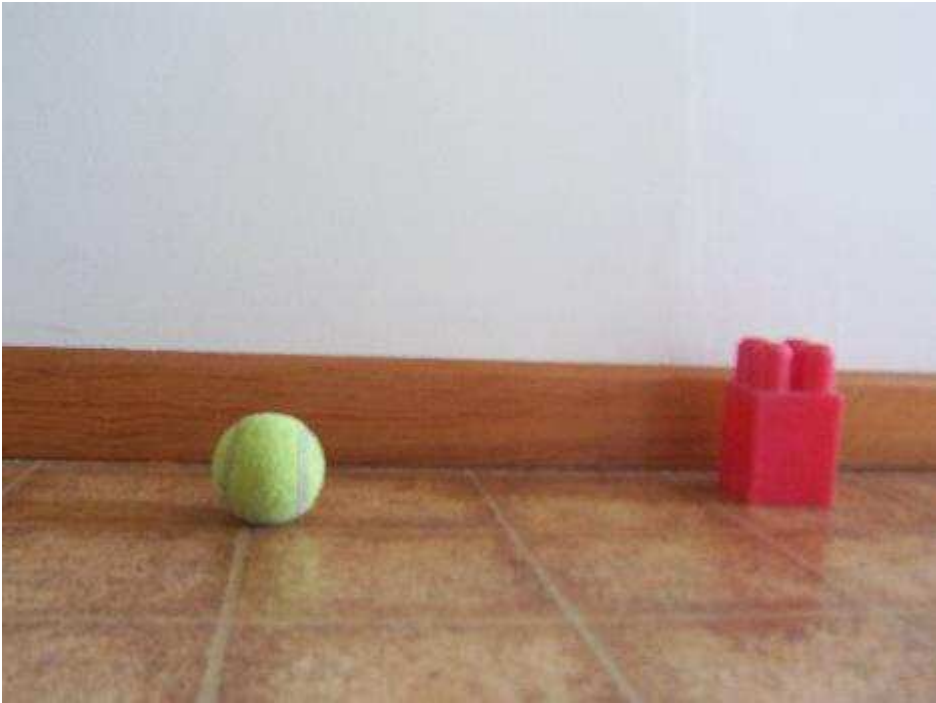


Capteurs : balayage laser

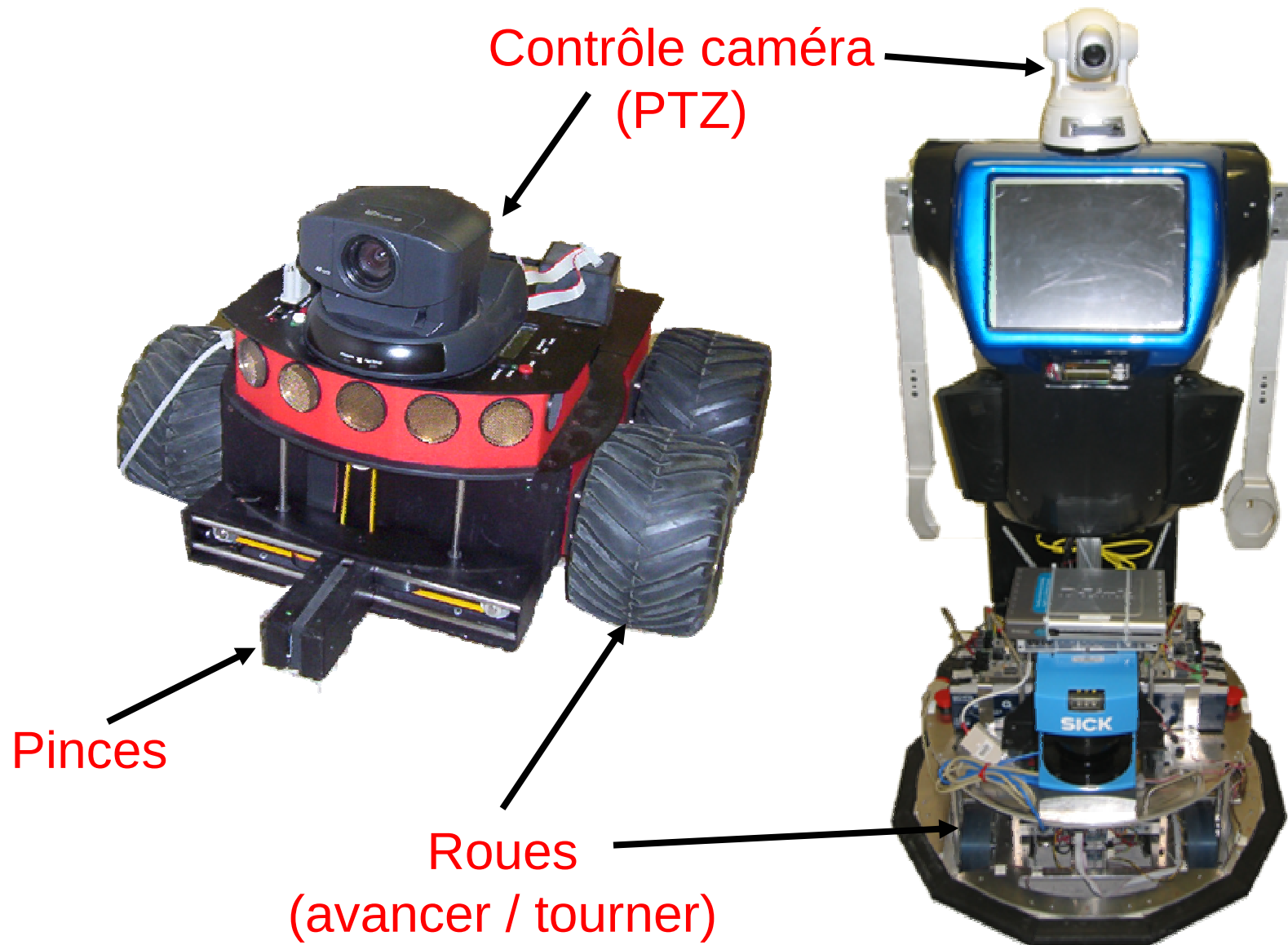




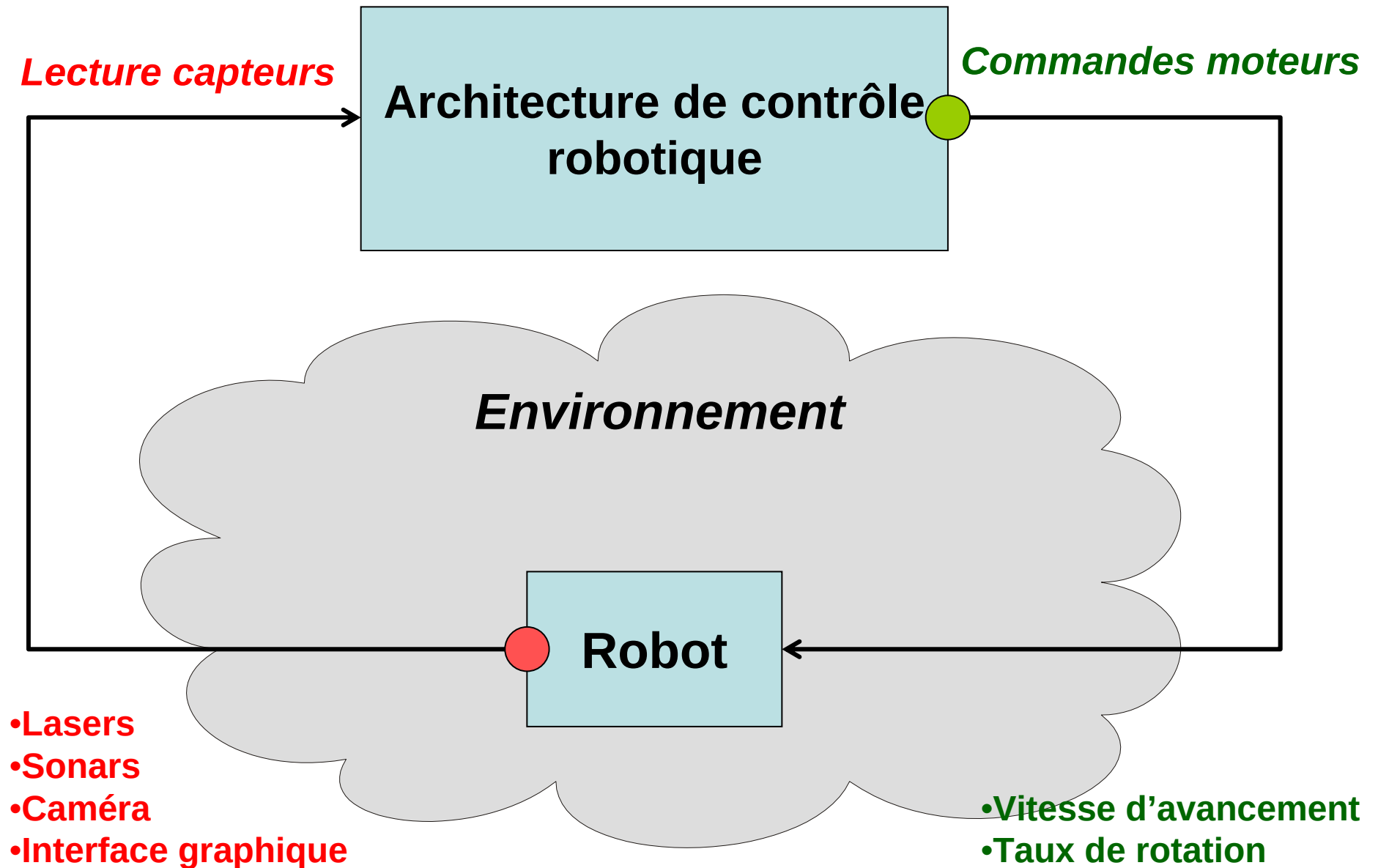
Capteurs : caméra



Actionneurs d'un robot



Boucle de contrôle



Modules comportementaux

(« *Behaviors producing modules* »)

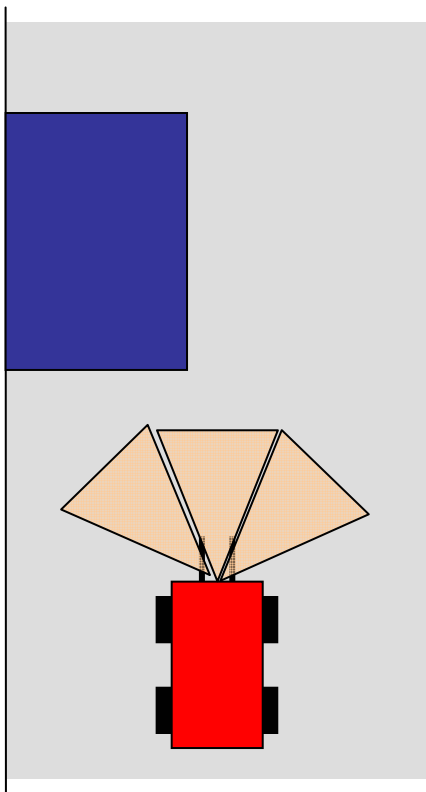
- Un **module comportemental** est une boucle de rétroaction générant des commandes pour 1+ actionneurs à partir des données de 0 ou plusieurs capteurs
- Aussi appelé **comportement**
- Responsable d'accomplir une seule fonction
 - Évitement d'obstacles
 - Suivit de trajectoires
 - Suivit de couleurs, objets, ...
 - Longer des murs
 - ...

Exemples de comportement : évitement d'obstacles (1)

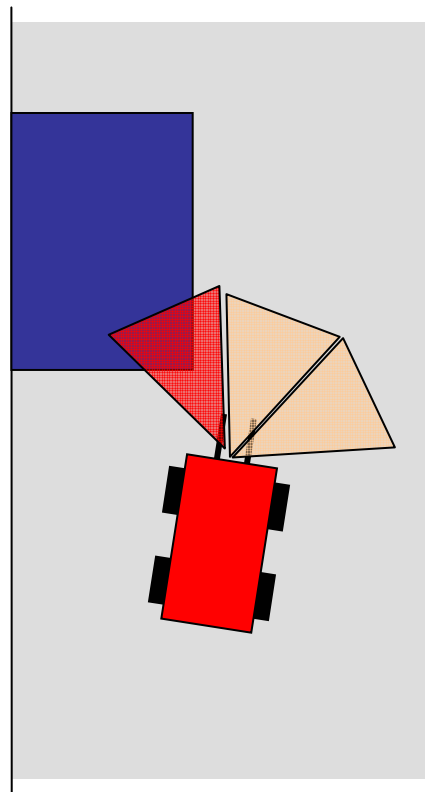
Obstacle	Action
(aucun)	(aucune)
à gauche	tourner à droite
à droite	tourner à gauche
devant	tourner à gauche ou à droite
devant + gauche + droite	faire demi-tour

Exemples de comportement :

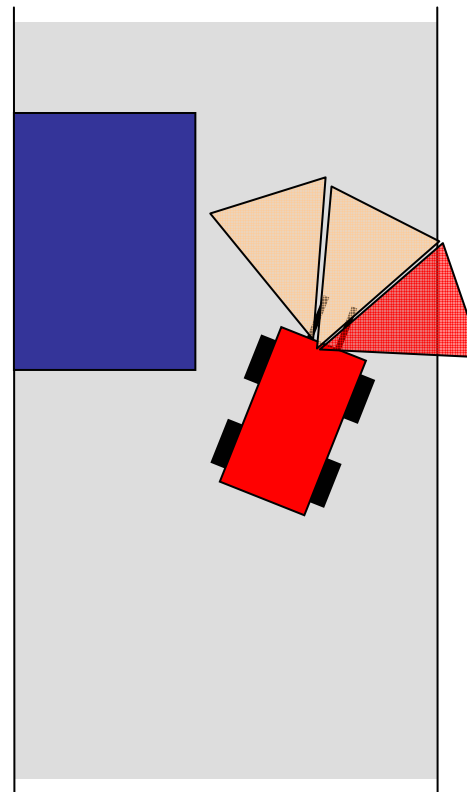
évitement d'obstacles (2)



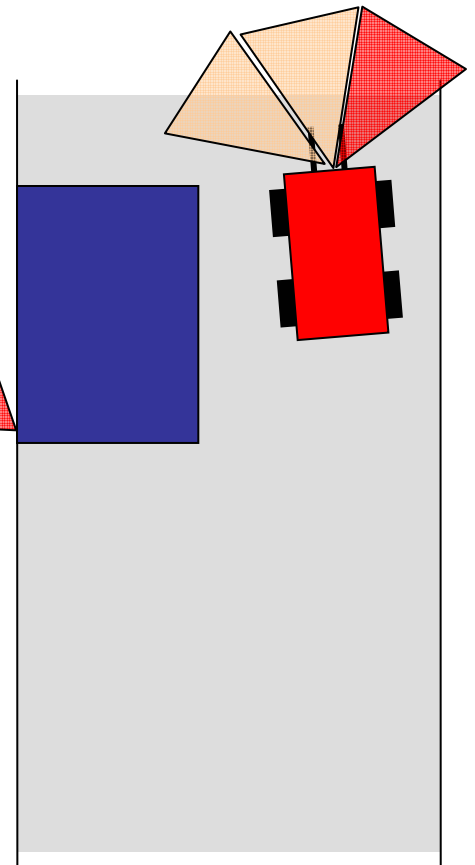
aucune action



tourner à droite



tourner à gauche

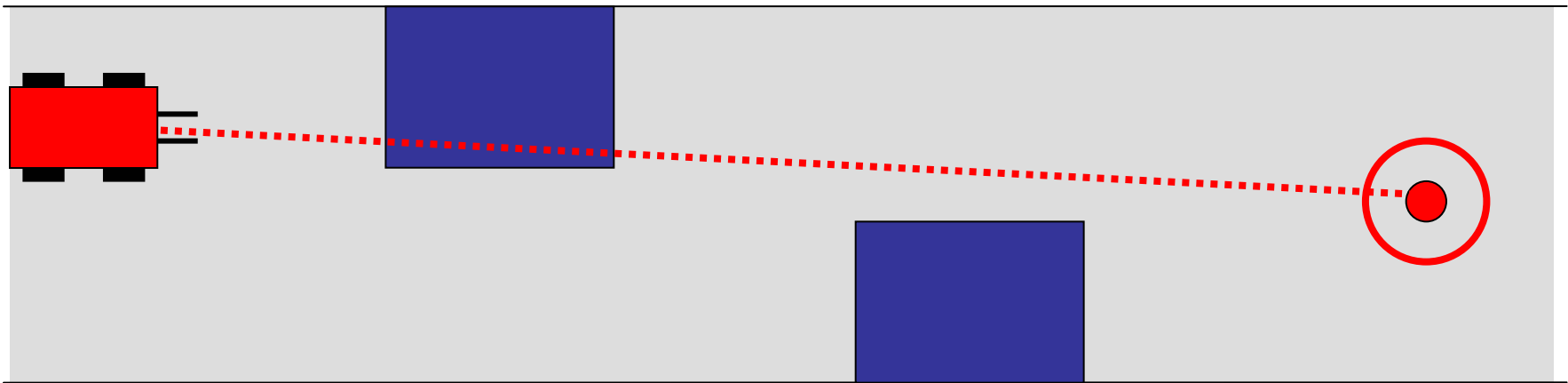


légèrement à gauche

Exemples de comportement :

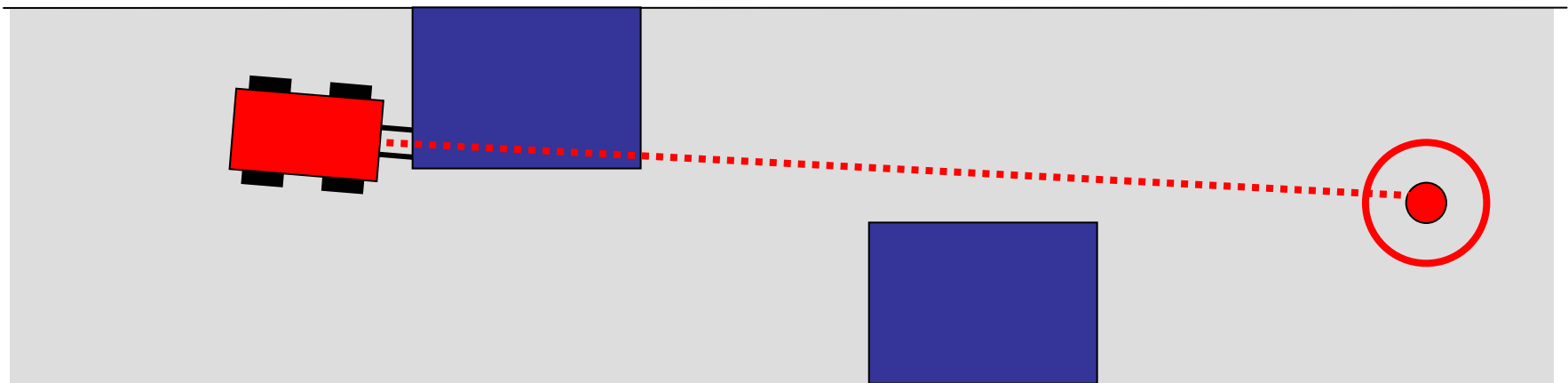
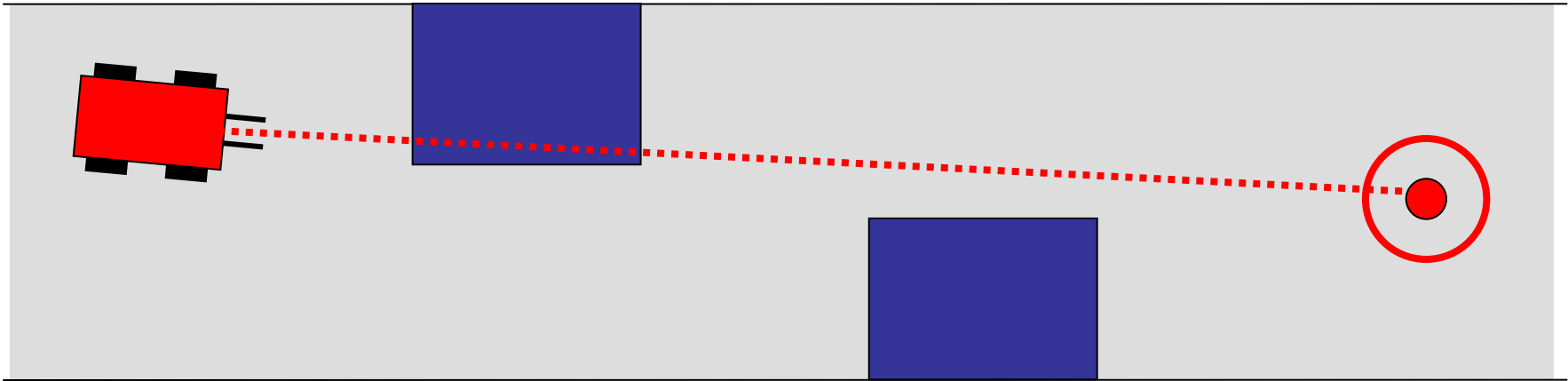
« GOTO » (1)

- Diriger le robot en ligne droite sur une cible



Exemples de comportement :

« GOTO » (2)

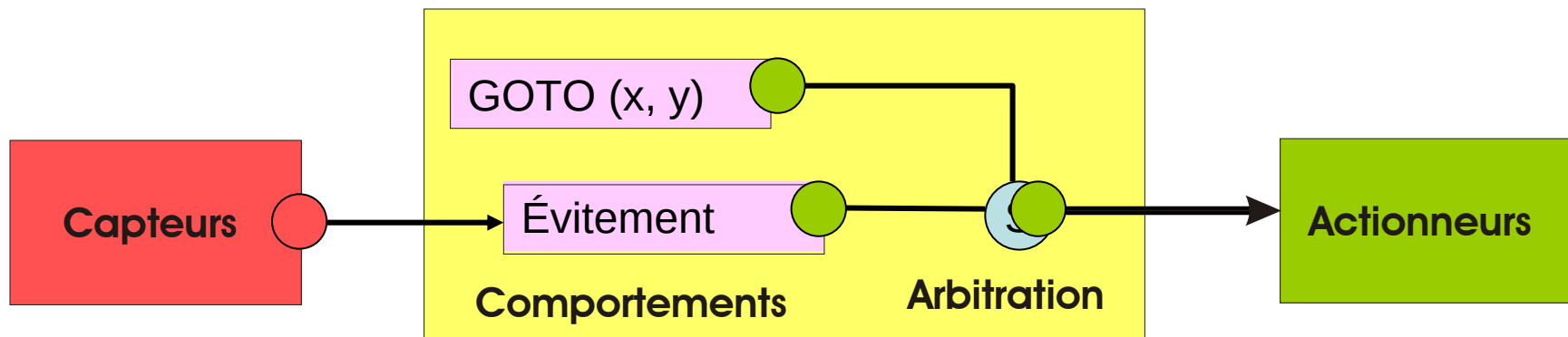


Architecture comportementale

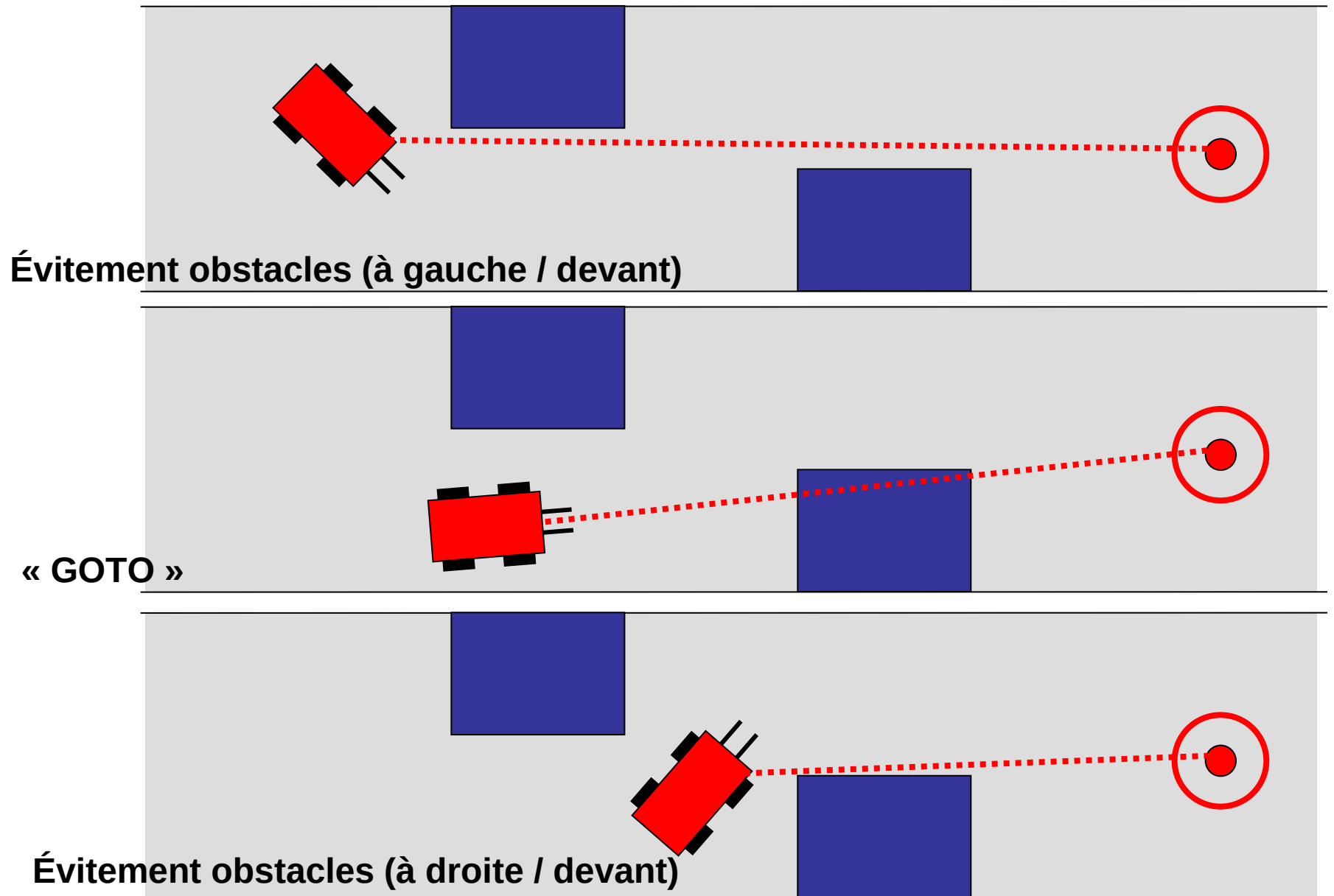
- Combinaison des comportements
- Arbitration de comportements
 - Priorité
 - Somme pondérée (« motor schema »)
 - Logique floue
 - Etc.

Architecture comportementale

- L'évitement d'obstacles est prioritaire
- Quand l'évitement génère une commande, il « écrase » celle du suivi de trajectoire



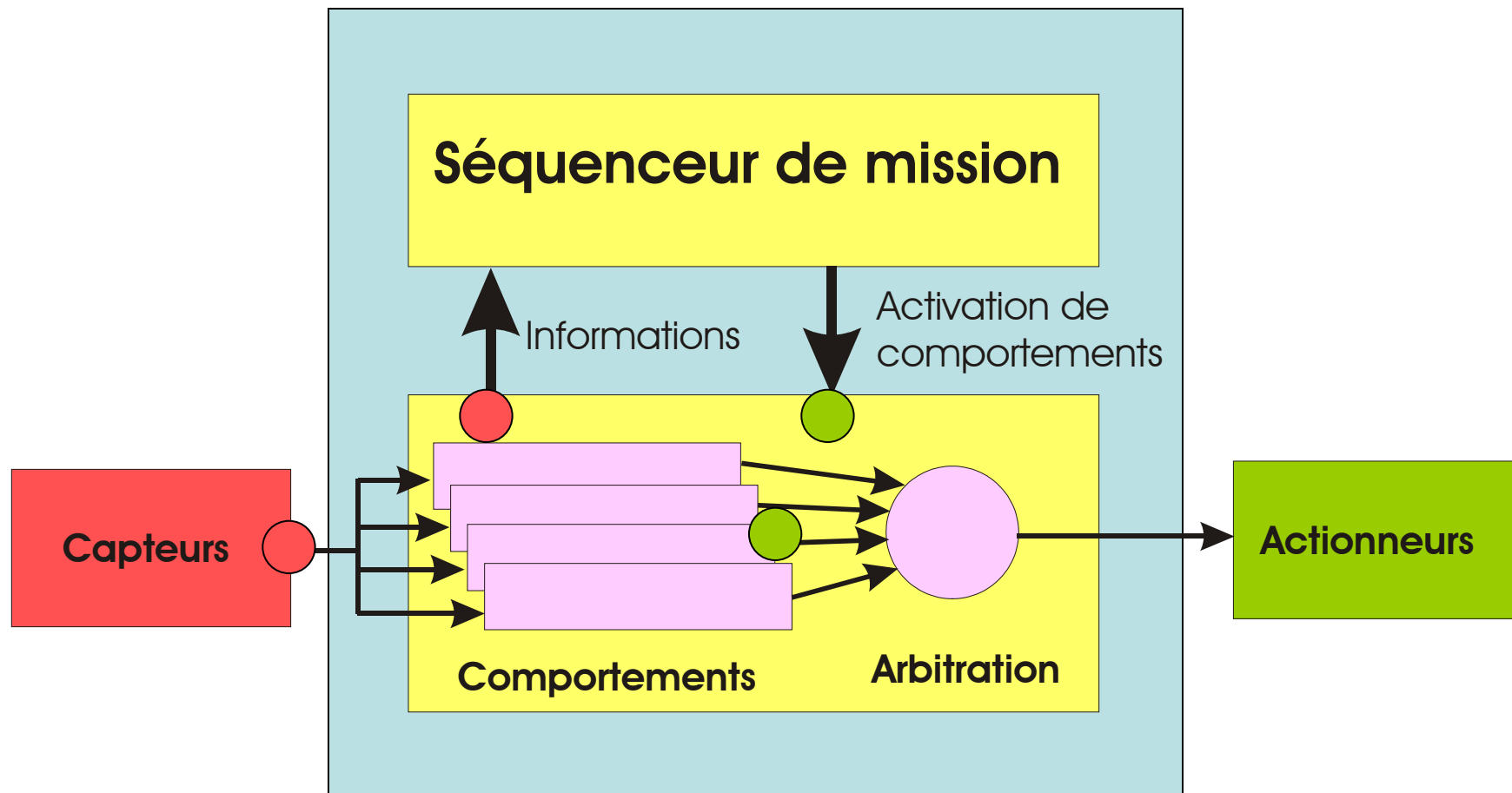
Arbitration de comportements



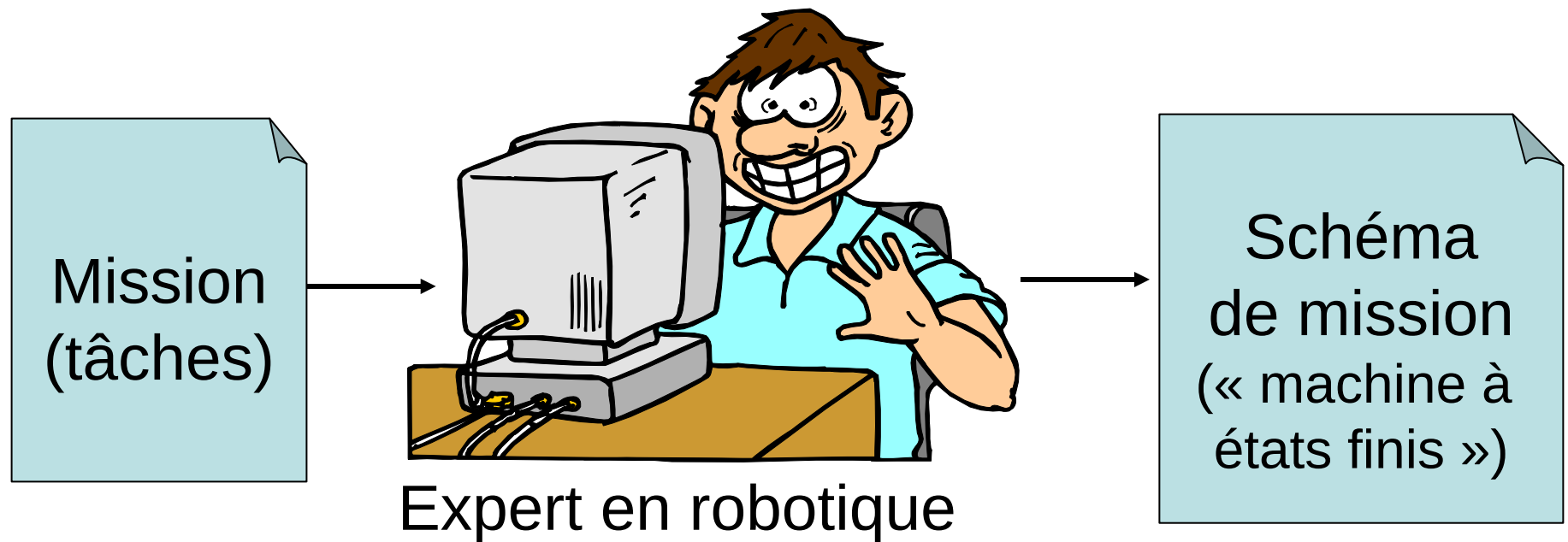
Architecture comportementale

- Problème :
 - Manque de flexibilité pour coordonner l'exécution de tâches complexes

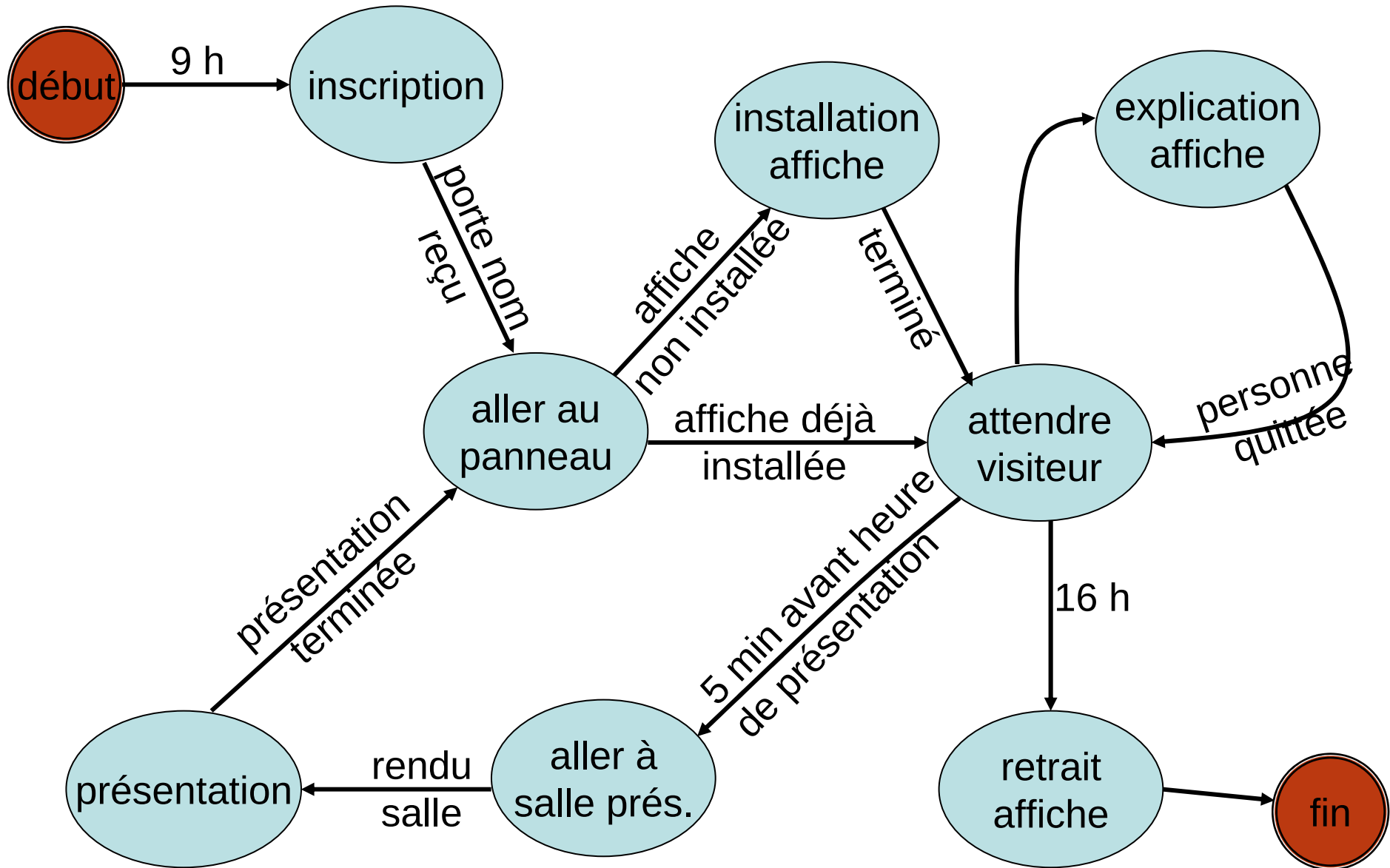
Architecture robotique de type hybride



Comment programmer des missions ?



Exemple de schéma de mission à l'aide d'une « machine à états finis »



Limitations

- Demande beaucoup de temps pour programmer les machines à états finis
- Impossible de prévoir tous les cas possibles (nous n'avons pas de boule de cristal)
- Comportement non optimal
- Manque de flexibilité
- Risques d'erreurs



Solution : « robot, débrouille-toi ! »

Fonctionnalités (actions primitives)

- Se déplacer
- Prendre et livrer des messages
- Afficher et dicter des messages
- Détecter des personnes
- Surveiller

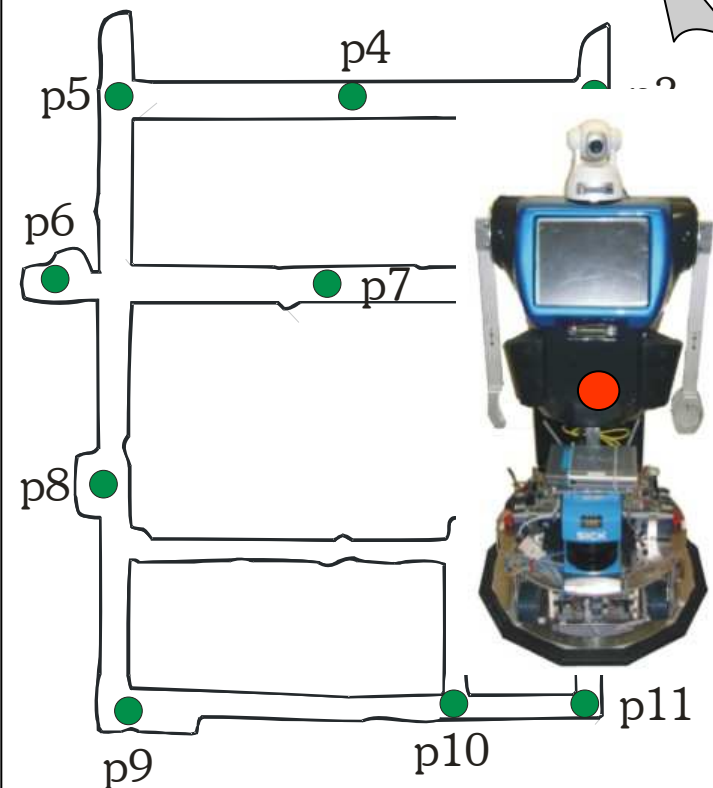
Spécifications techniques

- Vitesse moyenne: **0,12m/s**
- Autonomie batterie
 - en déplacement: **60 min**
 - en veille : **240 min**
- Temps de recharge: **240 min**

État initial

- Position du robot
- Heure courante
- Niveau de charge de la batterie
- ...

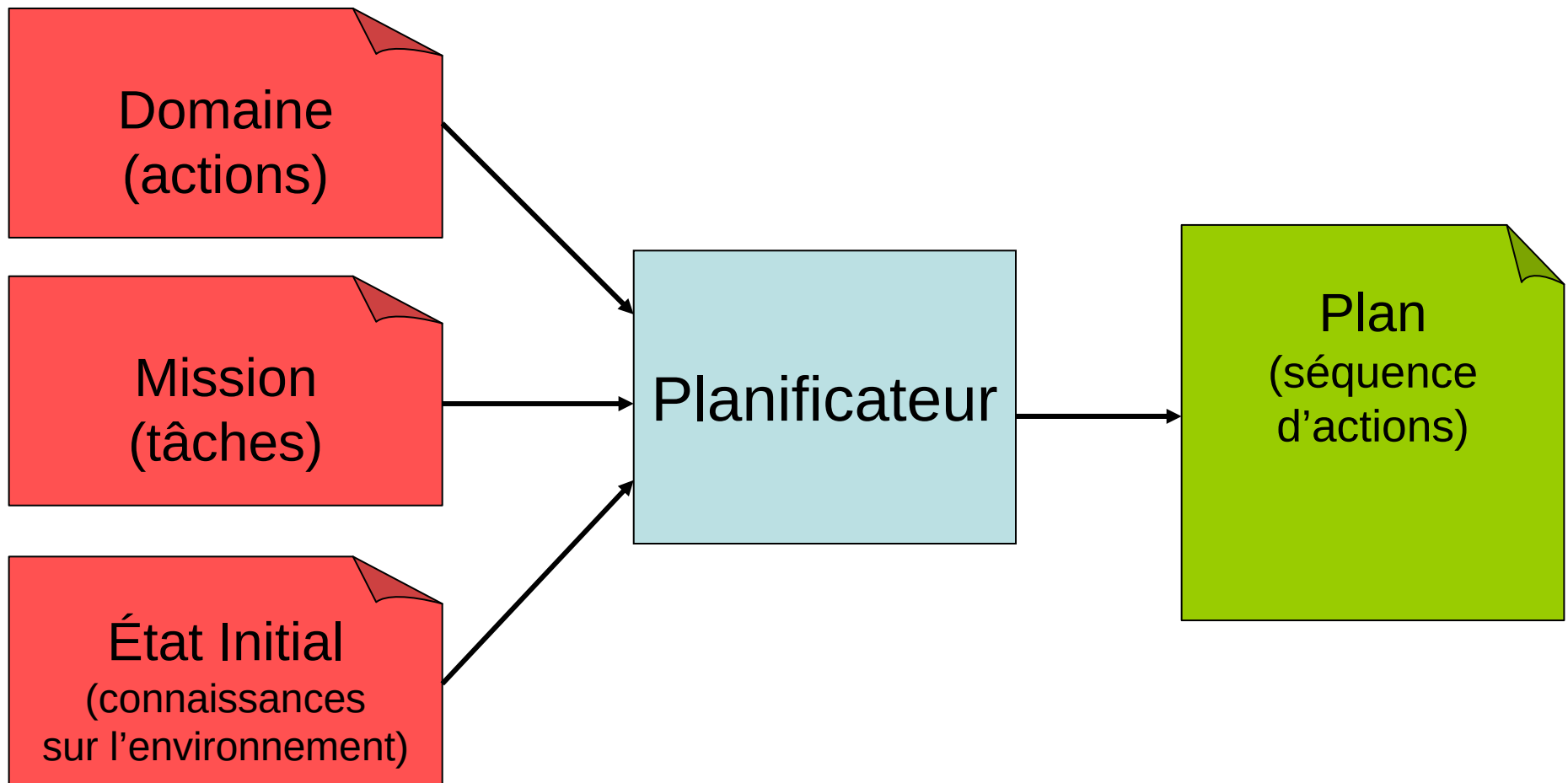
Carte



Mission

- Livrer message de *p1* à *p3*
- Présenter à 10h30
- Présenter une affiche

Modèle de planification



Hypothèses de planification

- On présume que les modules de bas niveau fonctionnent bien
 - Localisation
 - Navigation
 - La batterie se décharge de façon constante
 - ...
 - Bref, un robot dans un monde idéal!

Spécification d'un domaine de planification

- Spécification indépendante du planificateur
 - Le planificateur est vu comme une boîte noire
- Représentation symbolique
 - États : position, niveau énergie, messages, ...
 - Monde : carte, table de distances, horaire (programme)
 - Actions : se déplacer, s'inscrire, photographier, surveiller, écouter, présenter, ...

Modèle des actions

SeDéplacer(A, B)		PrendreMessage(M, P)	
Préconditions	• $\text{Position}(\text{Robot}) = A$ • $\text{Énergie} \geq k \cdot \text{distance}(A, B)$	Préconditions	• $\text{Position}(M) = P$ • $\text{Position}(\text{Robot}) = P$
Effets	• $\text{Position}(\text{Robot}) = B$ • $\text{Énergie} -= k * \text{distance}(A, B)$ • $\text{Temps} += \text{distance}(A, B) / v$	Effets	• $\text{Position}(M) = \text{<non défini>}$ • $\text{MessagesReçus} += \{ M \}$

Note: les effets des actions sont « déterministes », il n'y a pas d'incertitude lié au succès des actions.

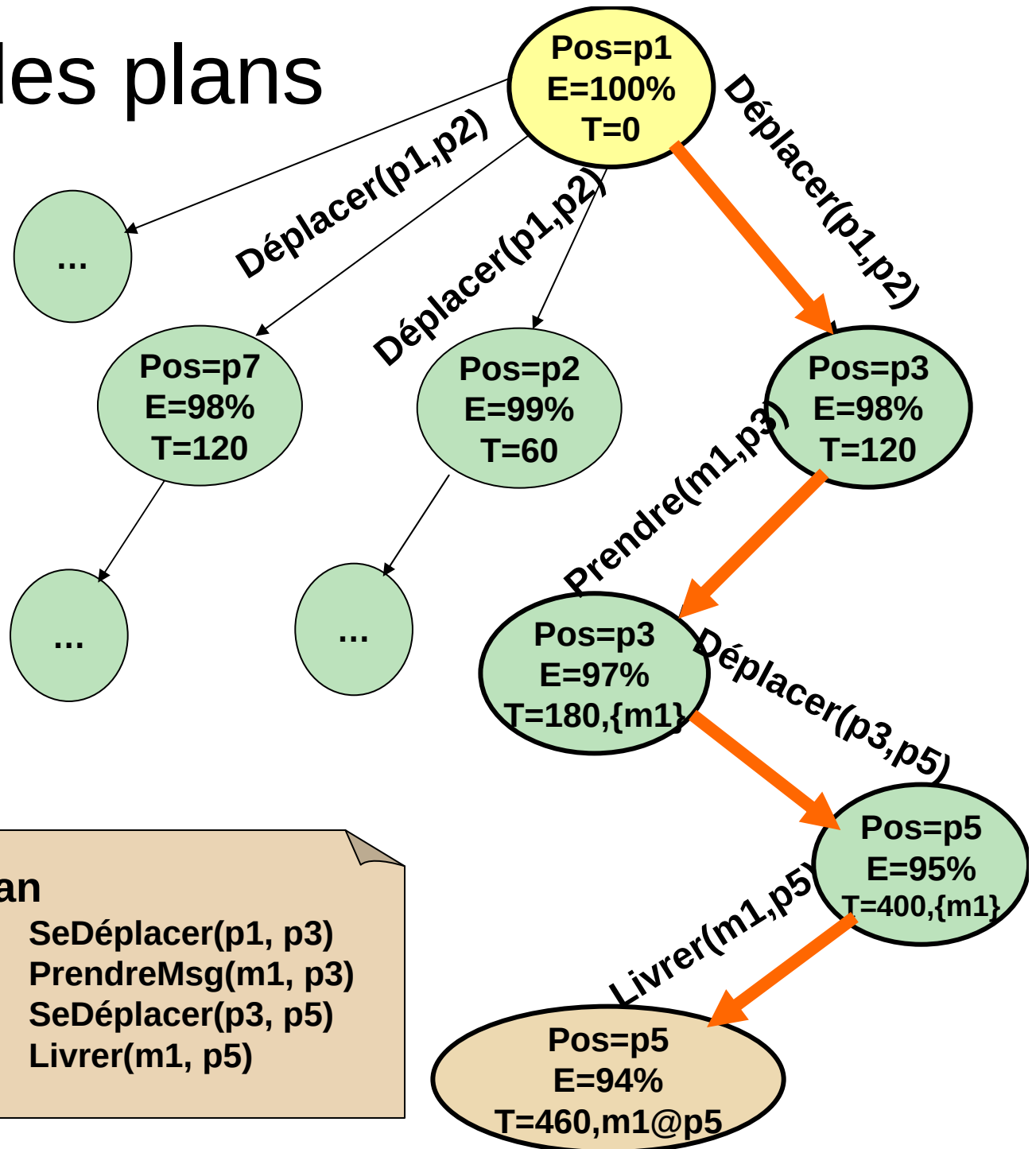
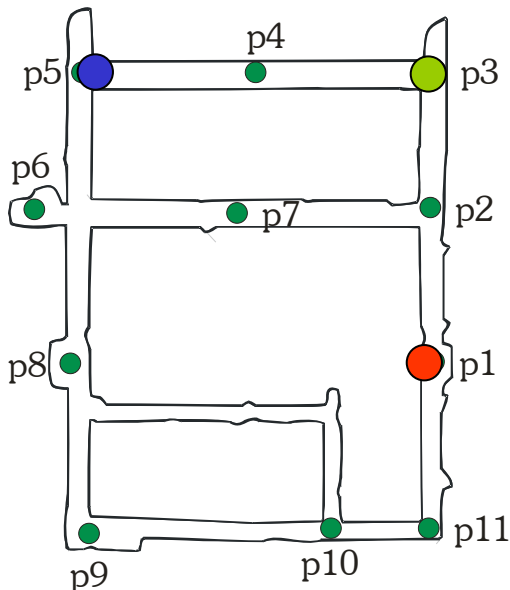
Génération des plans

Mission

Livrer message m1
de p3 à p5

État initial

Position = p1
Énergie = 100%
Temps = 0s



Plan

1. SeDéplacer(p1, p3)
2. PrendreMsg(m1, p3)
3. SeDéplacer(p3, p5)
4. Livrer(m1, p5)

Génération de plans (2)

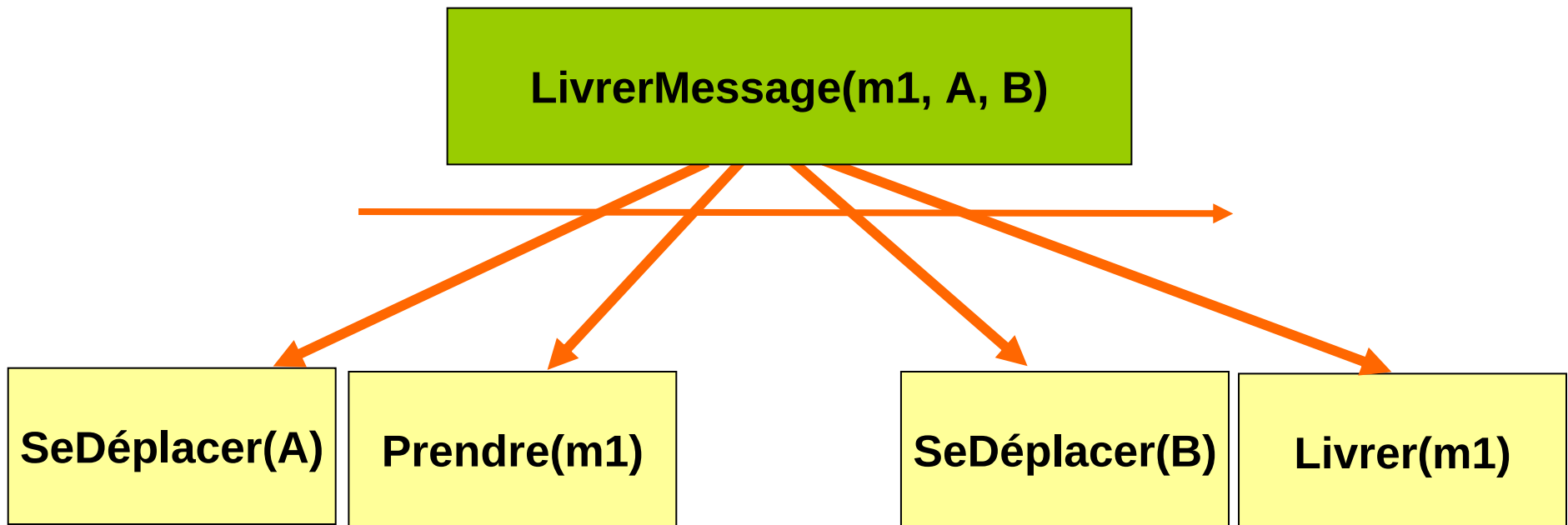
- La génération de plans est un problème dont la complexité algorithmique est **exponentielle** $O(a^n)$
- Nécessite beaucoup de **mémoire** et de **temps de calcul**
- Les algorithmes de planification classiques **sont inappropriés** pour des problèmes réels de robotique
- Besoin de développer de nouveaux algorithmes de planification

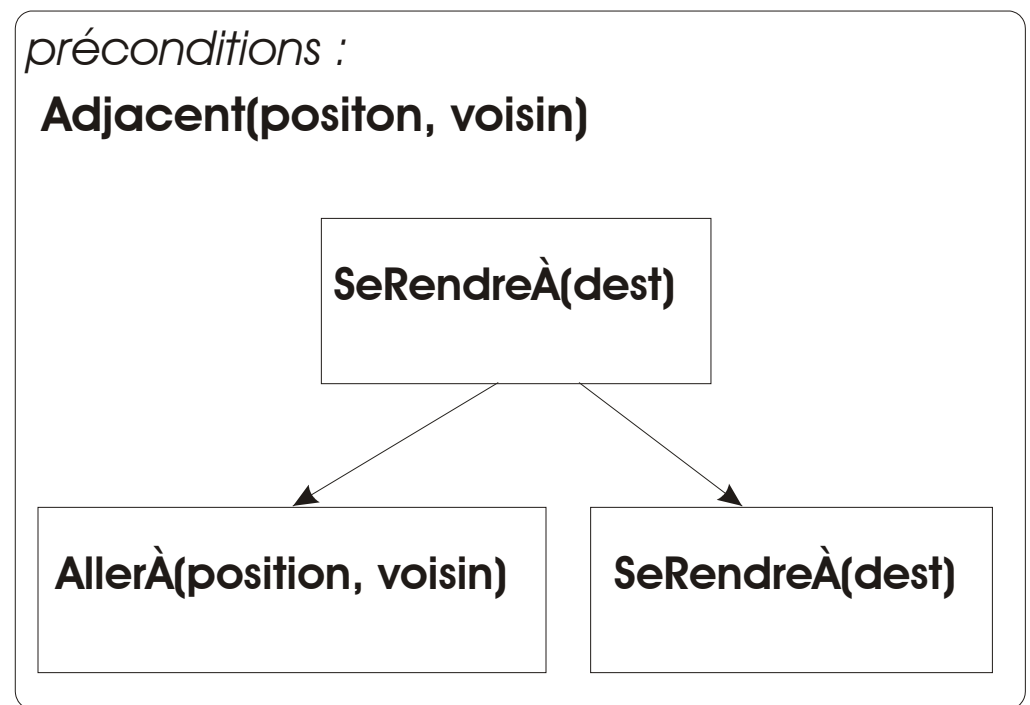
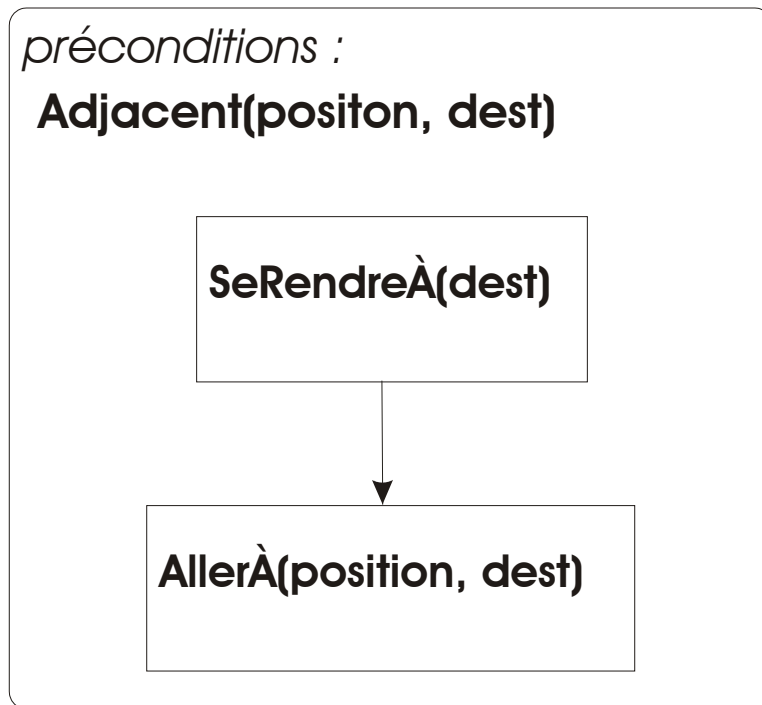
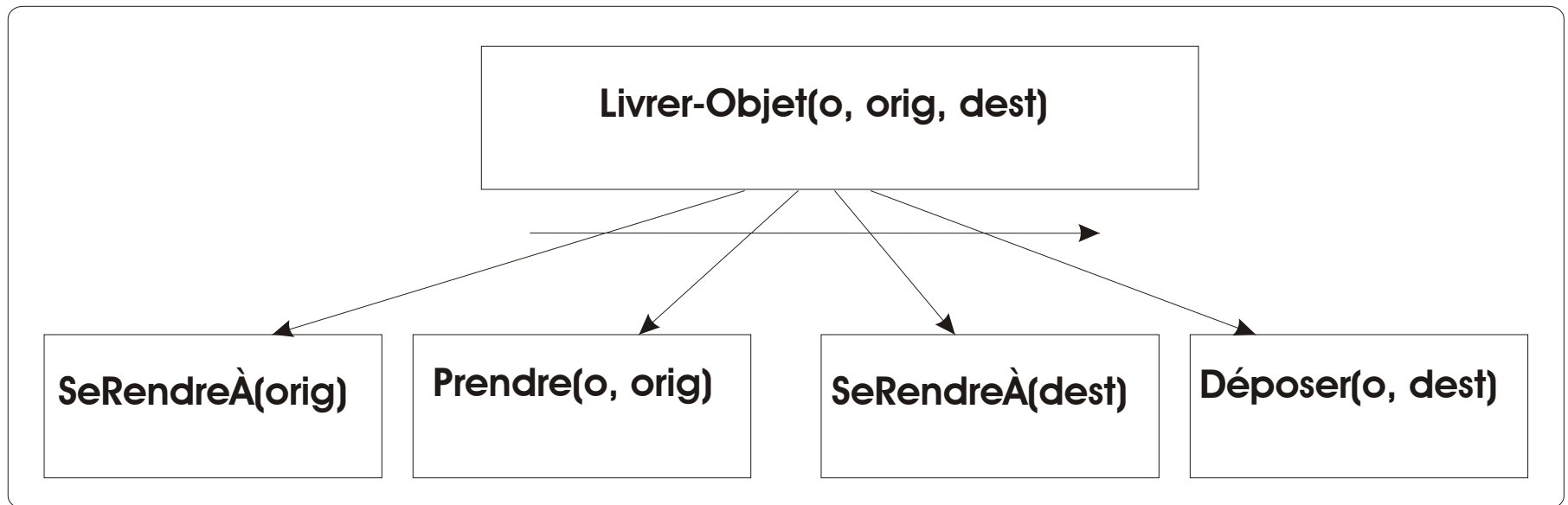
ConfPlan : nouveau planificateur

- Basé sur HTN (hierarchical task network / réseau hiérarchique de tâches)
- Gestion de contraintes de temps
- Gestion de priorités
- Algorithme « Anytime »

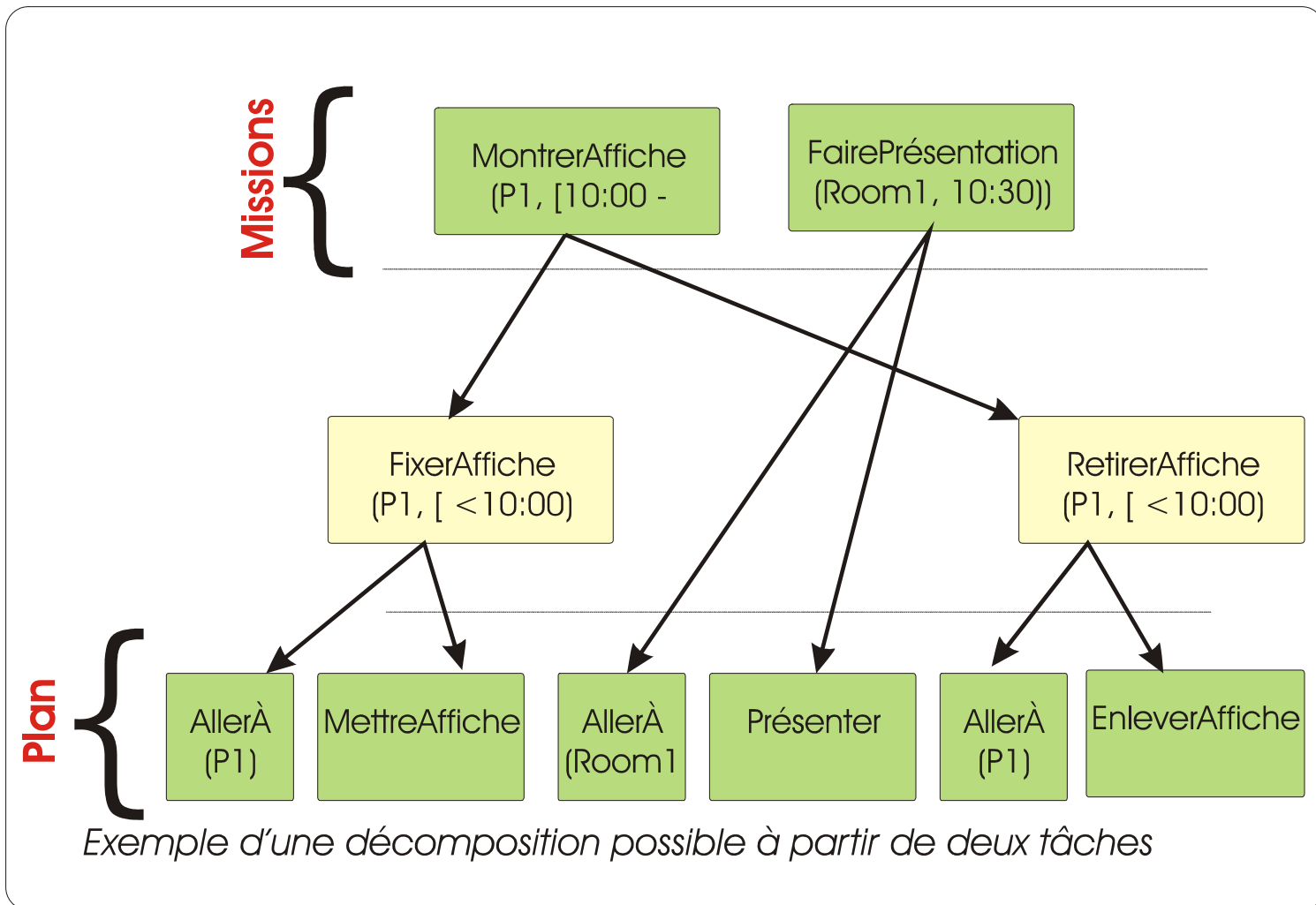
Approche HTN

- Utilisation de stratégies de recherche pour améliorer les performances
- Ajout de contraintes en utilisant des modèles de décompositions (réseaux hiérarchiques de tâches)





Génération de plans par HTN



Approche HTN (2)

- Missions : ensemble de tâches partiellement ordonnées.

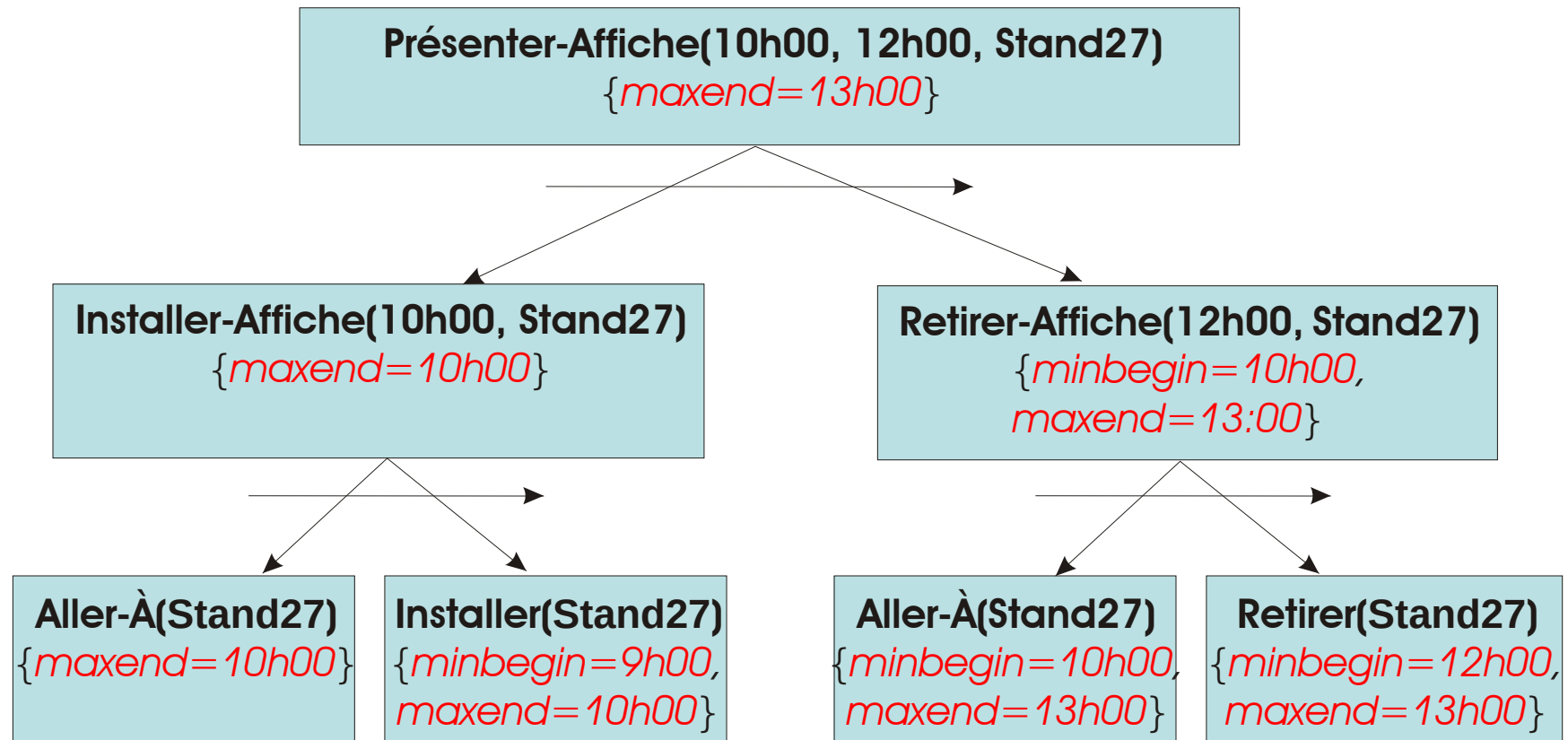
$$M = (T, O) , T = \{t_1, t_2, \dots, t_n\} \text{ et } O = \{t_{o1} < t_{o1}, t_{o2} < t_{o3}, \dots, t_{om-1} < t_{om}\}$$

- Algorithme de planification HTN
 - Recherche dans un espace de décomposition
 - Décomposition totale des tâches non primitives en tâches primitives (actions)
 - Ordonnancement des tâches

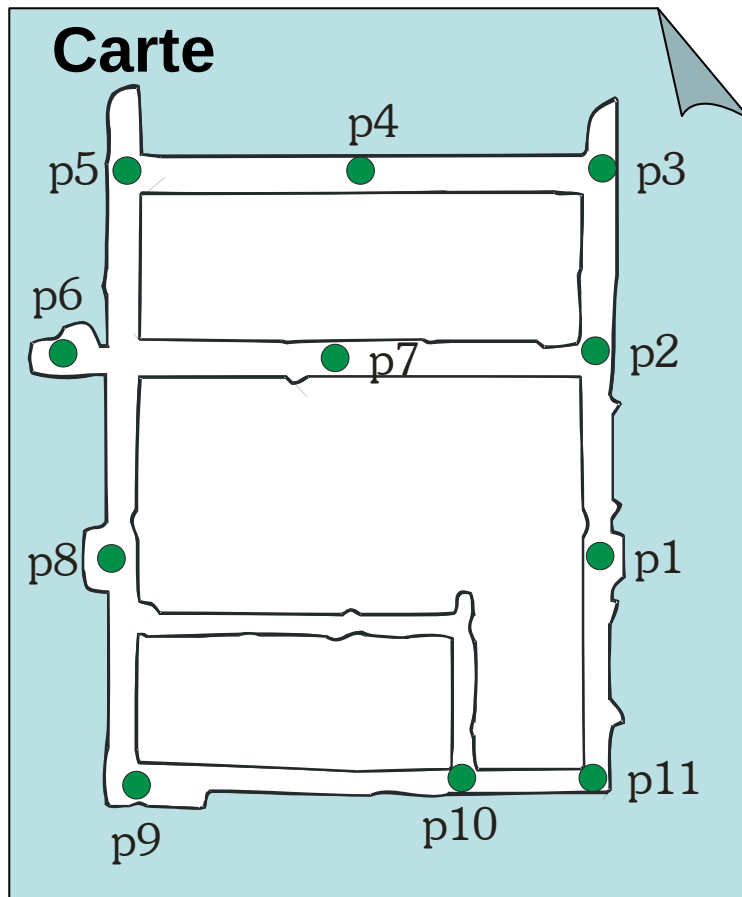
Gestion des contraintes de temps

- Attributs spéciaux sur les tâches:
 - minbegin : temps de commencement le + tôt
 - maxbegin : temps de commencement le + tard
 - minend : temps de terminaison le + tôt
 - maxend : temps de terminaison le + tard
- Création d'ordres partiels quand :
$$t_1.\text{maxend} \leq t_2.\text{minbegin} \rightarrow t_1 < t_2$$

Gestion des contraintes de temps



Exemple de mission



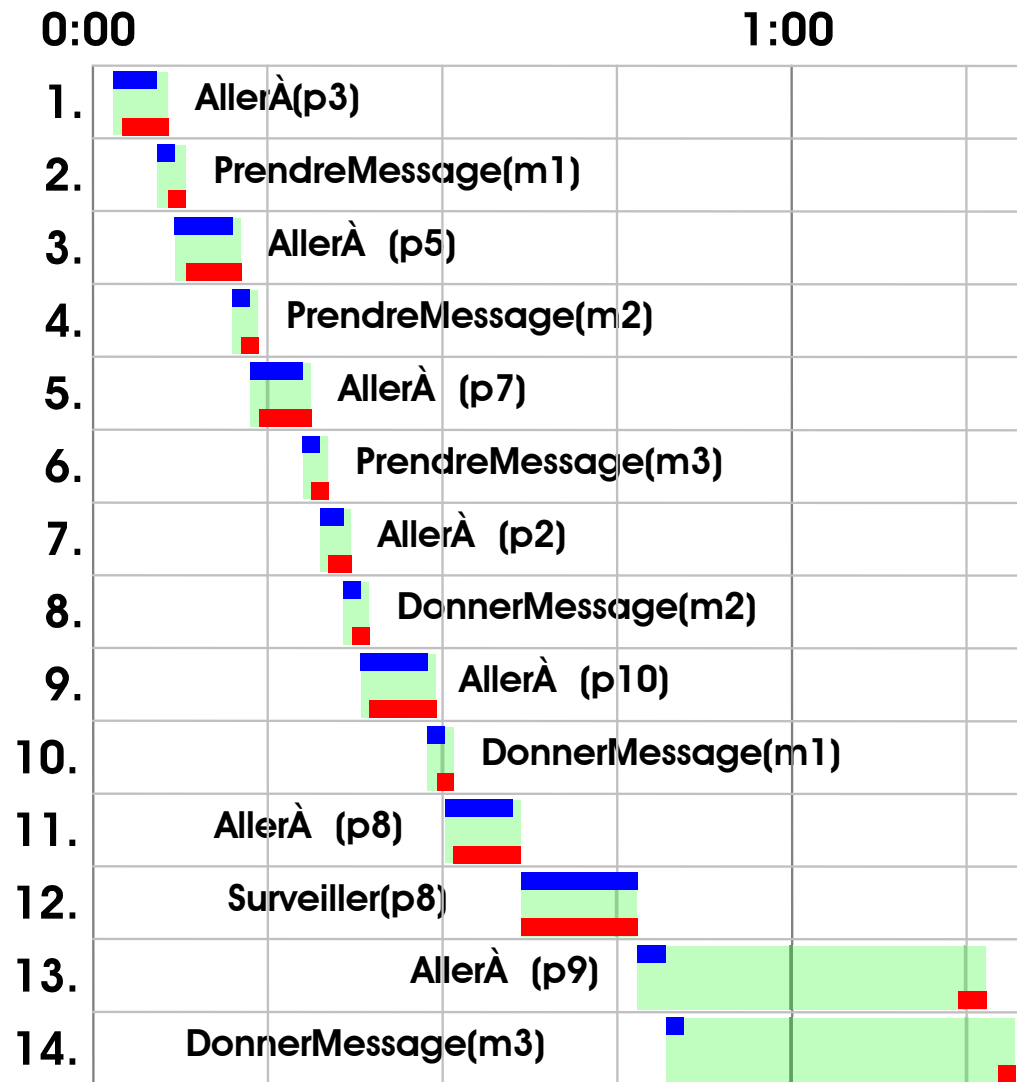
Mission

- Livrer(p5, p2, m1)
- Livrer(p3, p9, m2)
- Livrer(p7, p10, m3)
- Surveiller(p8, 0h35m, 7m)

État initial

- PositionRobot = p1
- t=0h00m
- Énergie = 100%

Exemple de plan

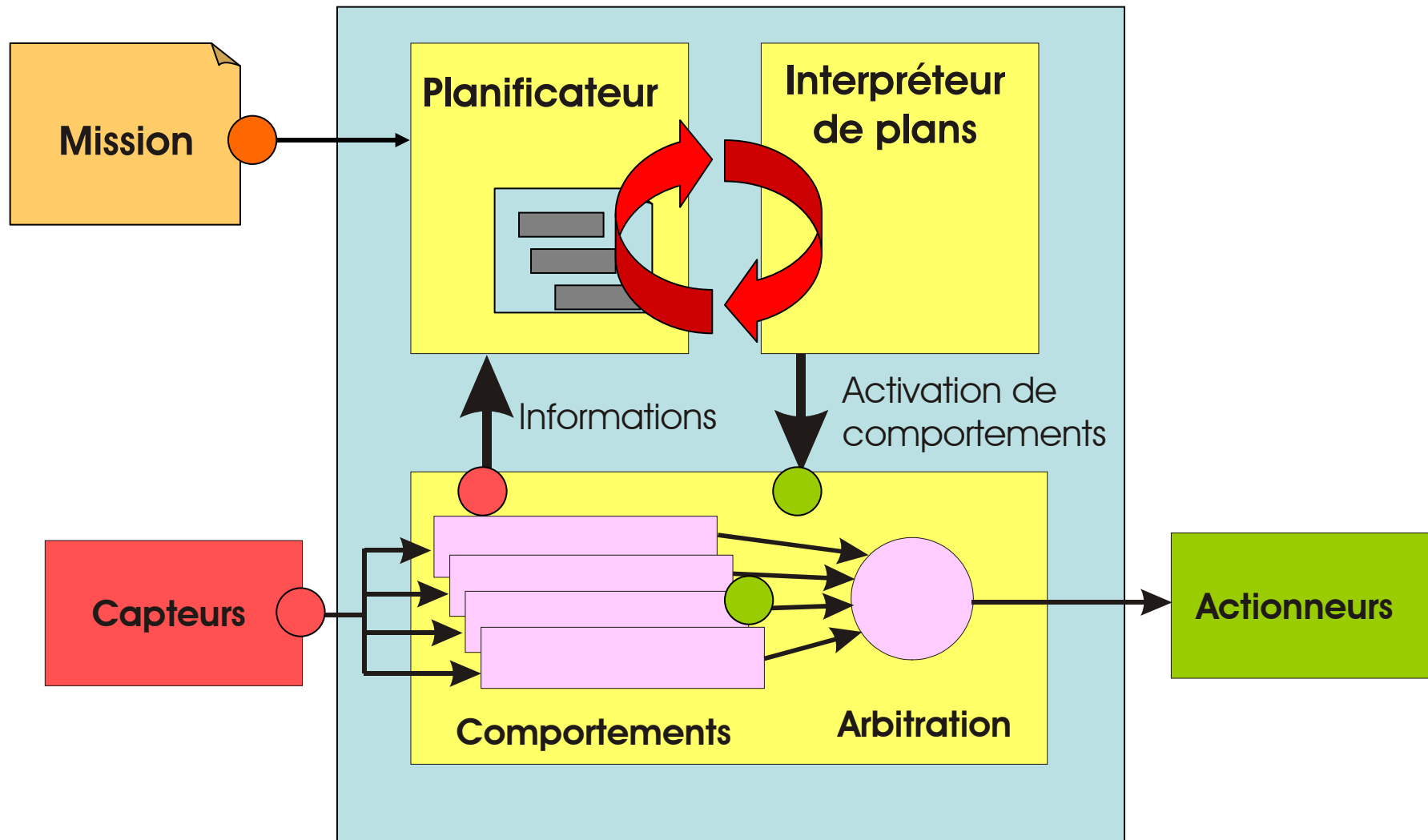


Intégration de ConfPlan dans une architecture décisionnelle

Exécution de plans

- Exécution pas à pas
- Validation continue des plans
 - L'hypothèse que tout se passe bien n'est pas toujours réaliste
 - Replanification en cas d'échecs
- Autres défis à résoudre :
 - Planifier en même temps que l'exécution d'un plan
 - Monde réel (domaine continu) vs représentation symbolique (domaine discret)
 - Conflits de tâches
 - ...

Boucle réactive



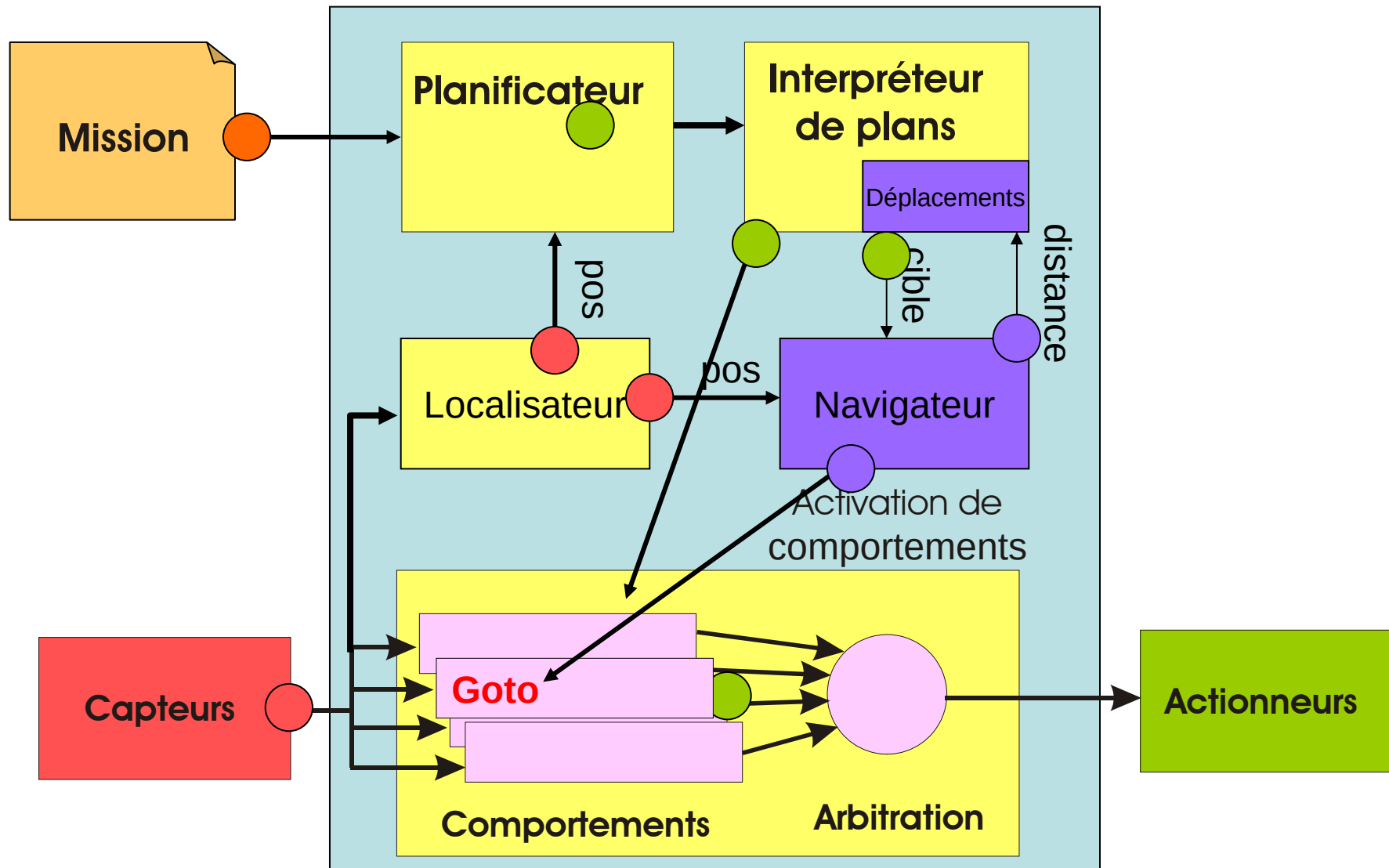
Replanification au besoin

- En cas d'échec (présent ou anticipation)
 - Validation continue du plan courant
- Changement de la mission
- L'exécution progresse mieux que prévue
 - Possibilité d'opportunisme : réaliser + de tâches ou + rapidement

Intégration avec la localisation et navigation (1)

1. Le planificateur génère un plan
2. L'interpréteur exécute le plan
3. Pour les actions de déplacement, l'interpréteur fixe une cible au navigateur
4. Le navigateur calcul un chemin (ex.: D*)
5. Le navigateur configure le chemin en activant successivement des points GOTO
6. Mise à jour continue de la distance restante
7. L'interpréteur se sert de la distance restante pour valider le plan courant

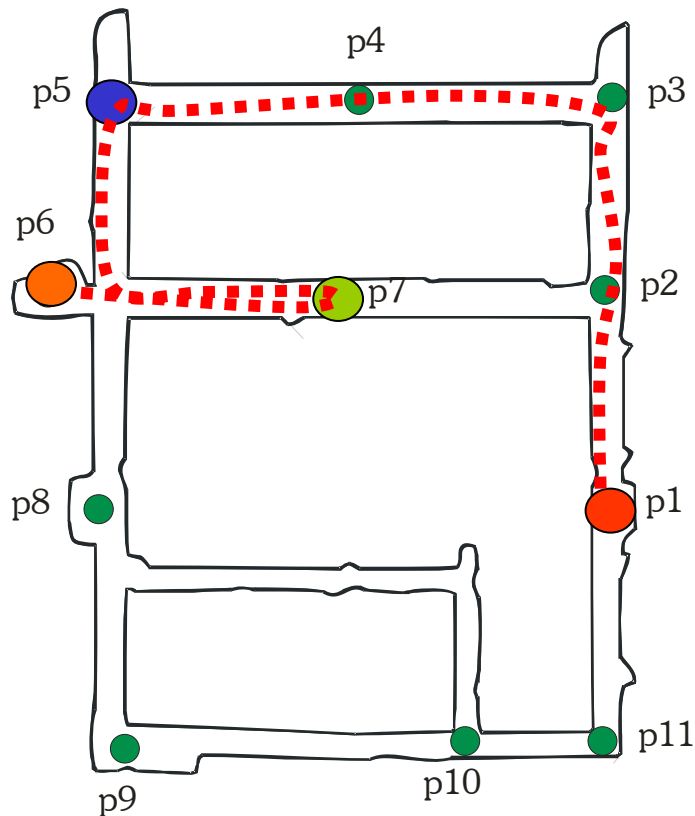
Intégration avec la localisation et navigation (2)



Exemple

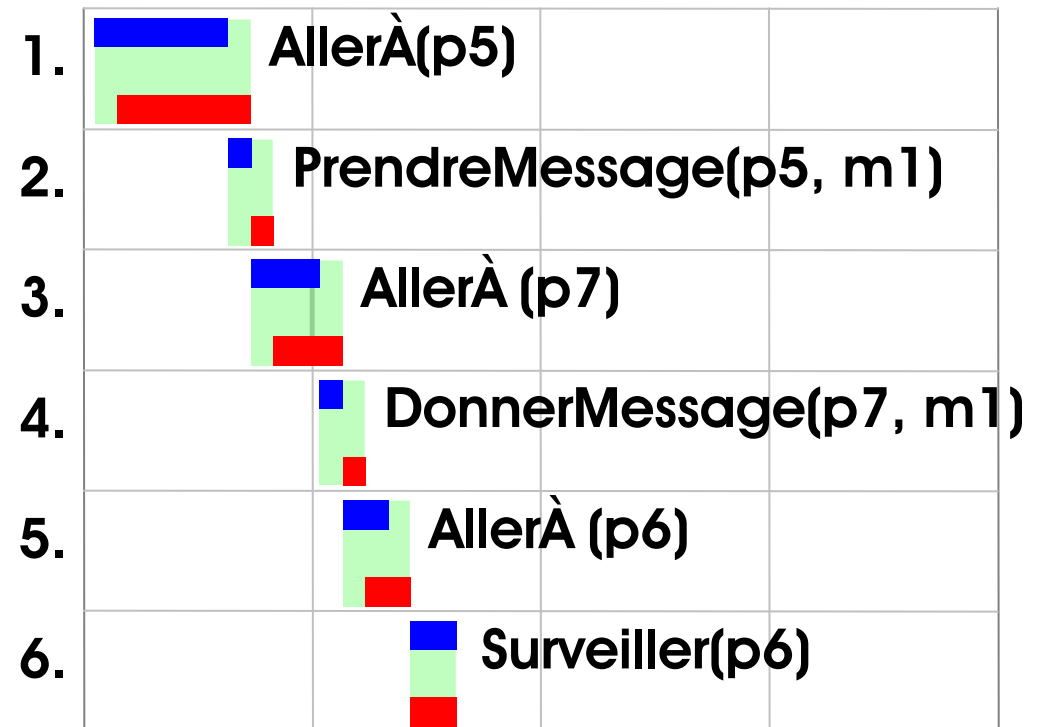
Mission :

- Livrer message p5 ● -> p7 ●
- Surveiller p6 à t=0h22m ●

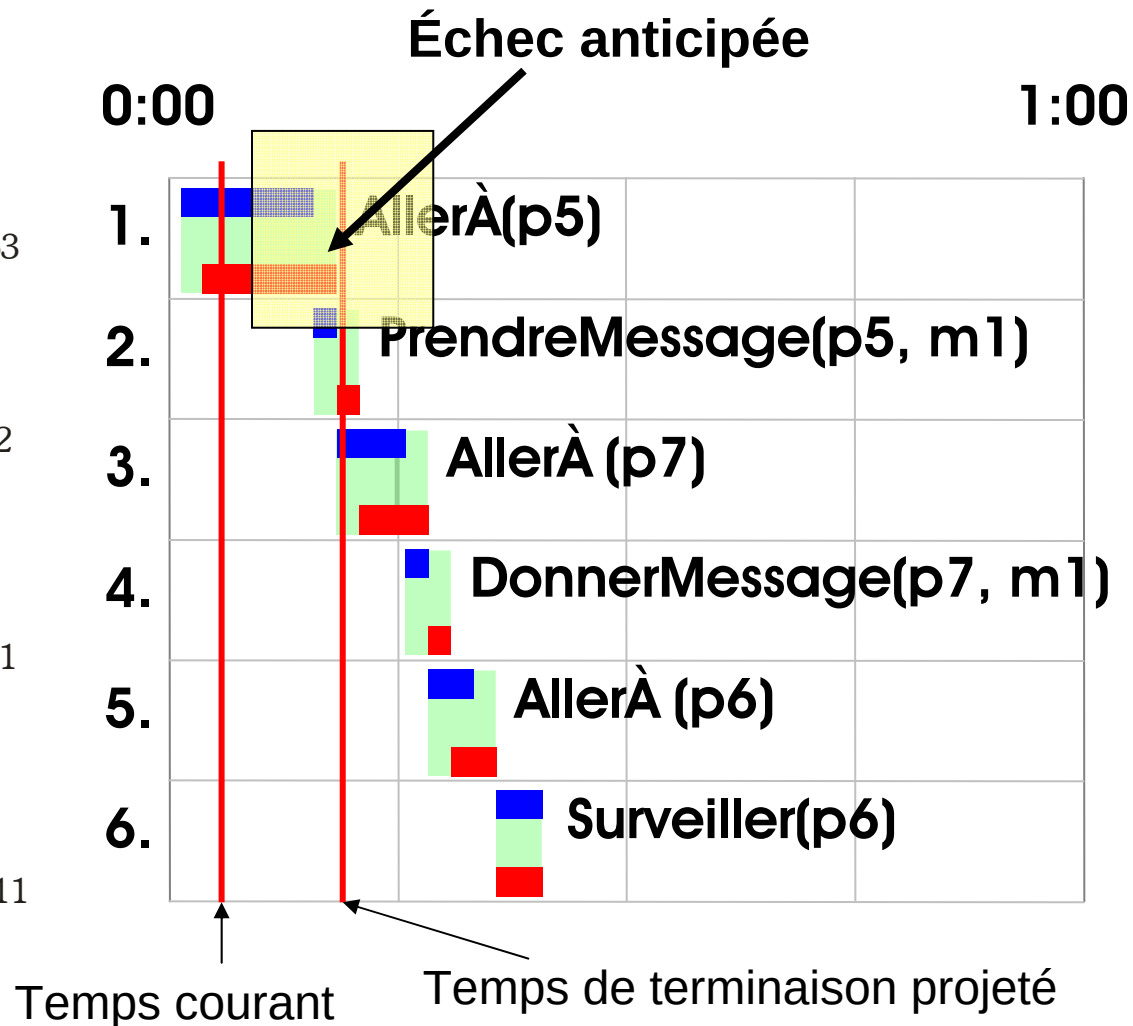
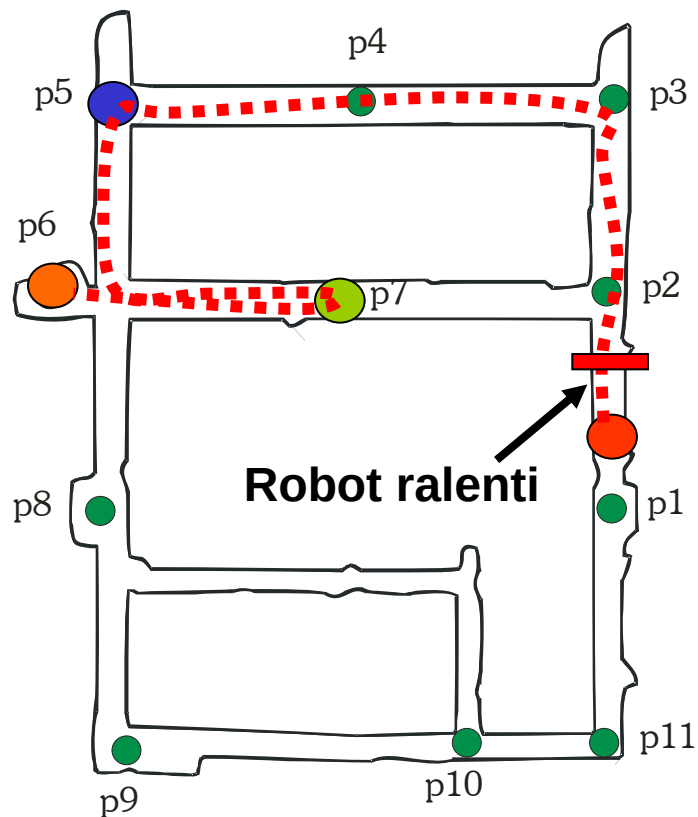


0:00

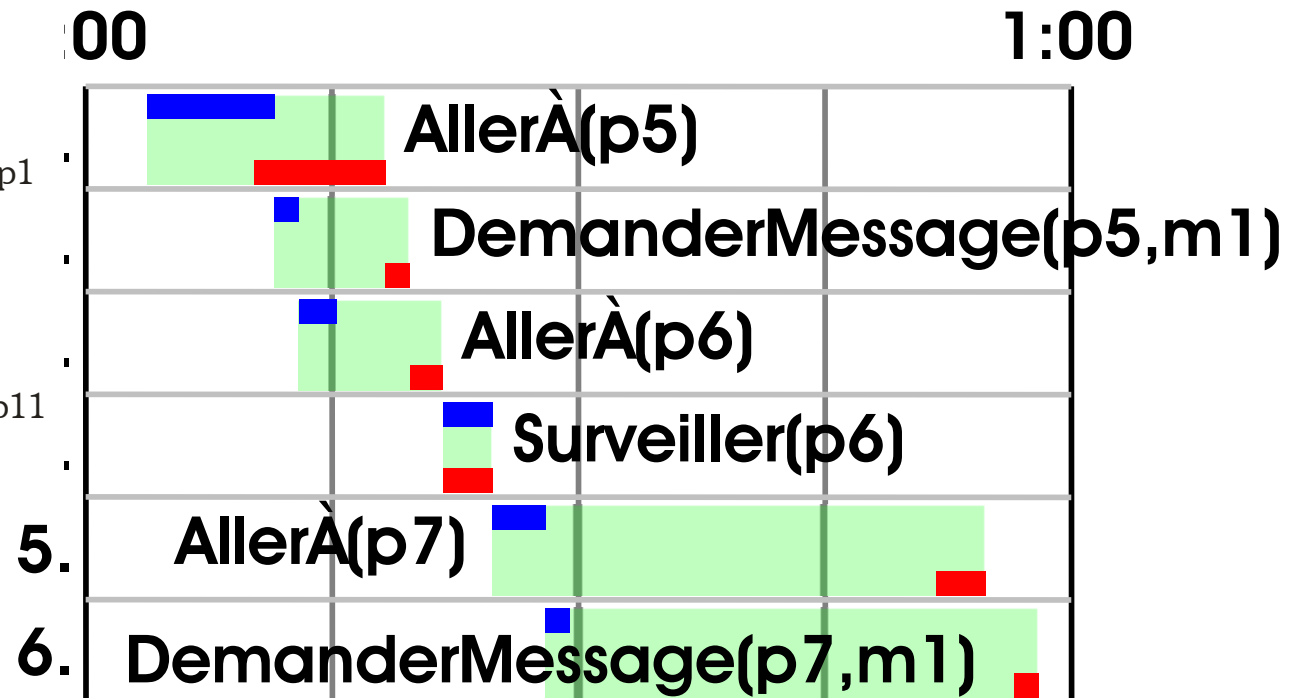
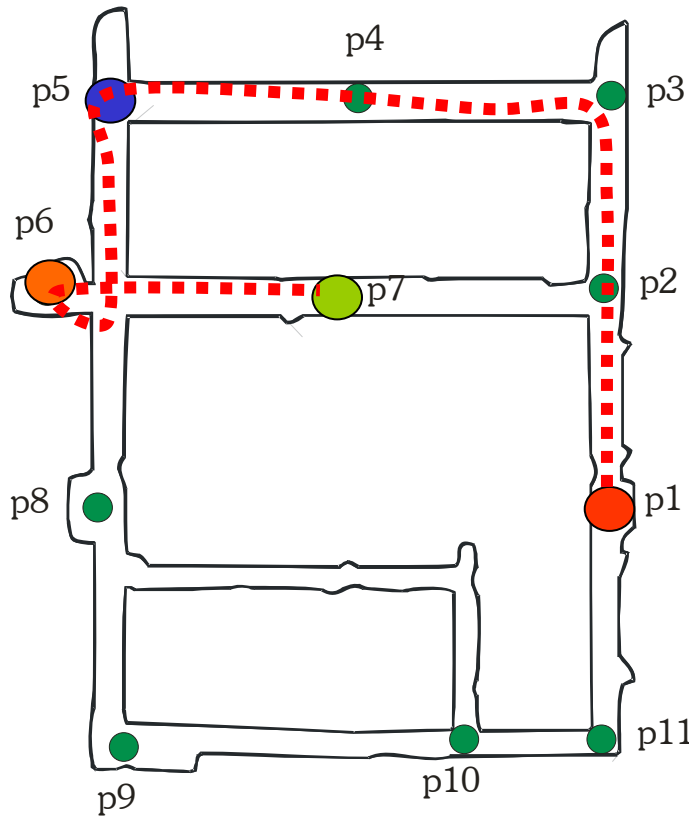
1:00



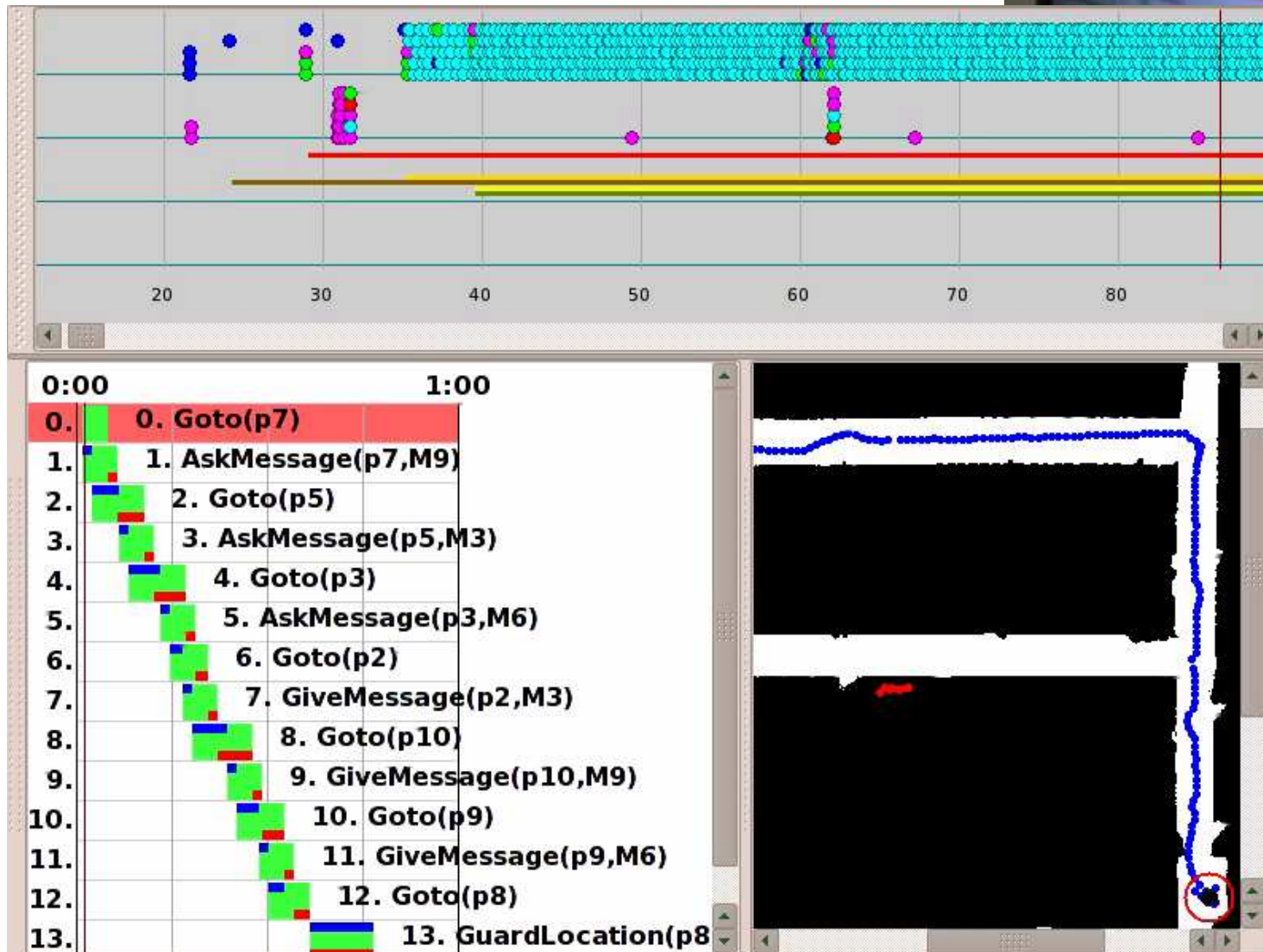
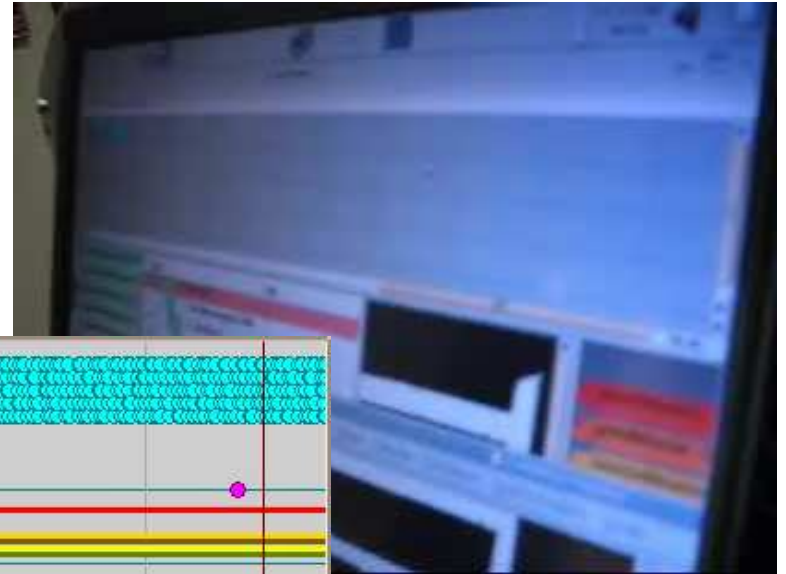
Exemple (suite)



Exemple (suite)



Expérimentations





Questions ?

<http://planiart.usherb.ca/~eric/>

Merci au **CRSNG** pour
son support financier

