Gestion des croyances de l'homme et du robot et architecture pour la planification et le contrôle de la tâche collaborative homme-robot

Matthieu Warnier

Université de Toulouse

THÈSE

En vue de l'obtention du

DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE

Délivré par :

Institut National des Sciences Appliquées de Toulouse (INSA de Toulouse)

Présentée et soutenue par :

Mathieu WARNIER

Le 10 décembre 2012

Titre :

Gestion des croyances de l'homme et du robot et architecture pour la planification et le contrôle de la tâche collaborative homme-robot

École doctorale et discipline ou spécialité :

ED MITT : Domaine STIC : Intelligence Artificielle

Unité de recherche :

LAAS-CNRS

Directeur de Thèse :

M. Rachid ALAMI

Rapporteurs :

M. Dominique DUHAUT
M. François CHARPILLET

Autres membres du jury :

Mme. Adriana TAPUS
M. Peter FORD DOMINEY
M. Felix INGRAND

Table des matières

Merci à Rachid qui m'a fait confiance quand je suis allé toquer à sa porte fin 2007.

Merci à mes parents qui m'ont compris et soutenu dans mon choix de quitter un poste dans l'industrie pour venir faire ma thèse.

Merci à toutes les personnes que j'ai côtoyées au cours de ces cinq années au LAAS.

Merci à tous les doctorants avec qui j'ai passé de très bons moments. Beaucoup sont devenus de vrais amis.

A tous un grand merci !

Résumé.

Ce travail de thèse a pour objectif de définir et mettre en oeuvre l'architecture décisionnelle d'un robot réalisant une tâche en collaboration avec un homme pour atteindre un but commun. Il s'inscrit dans le contexte du développement des robots assistants qui prennent en compte l'homme explicitement dans l'élaboration de leurs actions. De tels robots pourraient aider des personnes handicapées à rester autonomes chez elles. Ils pourraient également travailler en association étroite avec des opérateurs humains dans un cadre industriel, pour les soulager dans leur travail et augmenter leur productivité de manière beaucoup plus souple que les robots industriels actuels. Dans le cadre de cette thèse nos scénarios sont focalisés sur de la manipulation conjointe d'objets.

Ce travail a été réalisé au sein de l'équipe HRI (human robot interaction) au sein du groupe RIS (Robot interactionS) au LAAS CNRS à Toulouse.

Un certain nombre de fonctionnalités existaient déjà ou ont été développées conjointement avec ce travail au sein de l'équipe. De plus une architecture logicielle permettant de faciliter grandement l'intégration de ces différents composants était disponible.

Ce travail a d'abord consisté en l'étude puis en la formalisation des différentes capacités nécessaires. Il s'est traduit concrètement par l'approfondissement de certains des modules fonctionnels existants, par l'auteur ou par d'autres membres de l'équipe en lien étroit avec l'auteur. Une couche de contrôle de haut niveau a été mise en oeuvre par l'auteur pour permettre l'intégration et la mise en oeuvre de ces différentes capacités.

Ces différentes capacités sont décrites plus précisément dans les lignes qui suivent.

Tout d'abord a été abordée la représentation symbolique de l'état physique du monde. L'état physique du monde perçu par les capteurs est synthétisé sous la forme de faits symboliques qui représentent les possibilités de perception et d'action des agents sur les objets, ainsi que les positions des objets et des agents. L'auteur a finalisé un module existant et rendu celui-ci plus rapide pour permettre son utilisation en ligne. Ces faits sont

stockés dans une ontologie proposée par un autre membre de l'équipe et avec laquelle l'auteur a développé une interface. L'auteur a ensuite proposé une nouvelle formalisation pour prendre en compte explicitement les différences de croyances sur l'état physique du monde entre les agents.

Une formalisation des actions et des buts considérés a ensuite été proposée. Dans le cadre de nos scénarios de manipulation conjointe d'objets, les actions utilisées sont les suivantes : prendre un objet posé sur un support, poser un objet sur une table ou lâcher un objet dans un conteneur. Pour chacune de ces actions ont été définis les préconditions et les effets associés en terme de faits symboliques géométriques.

Puis a été abordé le suivi de l'évolution de l'état du monde. Le robot doit pouvoir mettre à jour rapidement l'état du monde suite à des changements. Pour cela un système simple mais robuste de monitoring d'actions élémentaires a été défini. De plus un mécanisme de gestion de l'attention décide de l'orientation des caméras de manière à focaliser l'acquisition d'informations compte tenu du contexte.

Ensuite, la question de la réalisation effective des actions par le robot a été prise en compte. Une typologie des différentes trajectoires, en fonction du triplé pince, objet et support, a été définie en collaboration avec les collègues de la planification de mouvement. Des automates ont été définis par l'auteur, pour déterminer la séquence de plans géométriques associée à une action symbolique donnée, pour un état du monde donné.

La planification des buts et l'exécution des plans produits ont été enfin abordées et donnent une cohérence à l'ensemble. Le robot doit être en mesure de planifier et d'exécuter les plans produits. On utilise le planificateur développé au sein de l'équipe. Il permet de produire un plan partagé qui intègre à la fois les actions du robot et celles de l'homme afin d'atteindre le but commun. L'auteur a développé un moteur simple d'exécution de ces plans partagés. Les actions sont exécutées de manière séquentielle en utilisant les capacités de réalisation des actions du robot et de monitoring des actions de l'homme. Un échec du plan donne lieu à une tentative de re-planification.

Puis dans un deuxième temps, le robot a été doté de la capacité à déterminer que d'autres agents ont des croyances distinctes sur la position des objets dans l'environement du fait de capacités d'observation et de compréhension distinctes mais aussi tout simplement parce que des actions sont réalisées en leur absence. Ces divergences ont été prises en compte explicitement dans le planificateur de tâche par l'ajout d'actions de communication qui permettent de rétablir des croyances communes si et quand il y a besoin.

Cette architecture décisionnelle a été mise en place sur un robot. Des expérimentations ont permis d'illustrer la capacité du robot à accomplir un but avec l'homme. Nous sommes parvenus à envisager le problème dans sa

globalité. Ce cadre global nous ouvre de nombreuses perspectives.

Première partie

Introduction et formalisation

Chapitre 1

Introduction

1.1 Les robots d'assistance à la personne

Depuis déjà plusieurs décennies, les robots sont utilisés avec succès dans l'industrie où ils permettent d'améliorer la productivité.

De plus en plus ils vont assister les hommes en dehors des usines, chez les gens [42], dans les hôpitaux ou les maisons de retraite [43, 61], et même dans l'espace pour aider les spationautes [7]. Les robots compagnons pourraient être une des solutions au problème de la population vieillissante dans les pays développés. Un tel robot pourrait permettre aux personnes âgées de garder leur indépendance en réalisant des tâches du quotidien et en interagissant de manière sécurisée et amicale. Cela réduirait les dépenses de santé et donnerait une meilleure qualité de vie aux personnes âgées.

1.2 Contexte de mes travaux

Notre travail a été effectué dans le cadre du projet européen CHRIS : "Cooperative human robot interaction systems". Ce projet s'est déroulé de mars 2008 à février 2012. Il avait pour ambition fondamentale d'aborder les principales problématiques nécessaires à une interaction homme-robot sécurisée. Plus spécifiquement, le projet est typiquement concerné par des scénarios où un homme et un robot réalisent une tâche de manière coopérative en un même endroit pour atteindre un but commun. Les compétences dont le robot devait être doté sont la communication d'un but partagé (verbalement ou au moyen de gestes), la perception et la compréhension des intentions (à partir de mouvements), la cognition nécessaire à l'interaction, et la compliance active et passive. Ce sont les prérequis pour de nombreuses applications en robotique de service. De manière logique dans la mesure où des membres de

l'équipe ont tenu un rôle important dans la définition des objectifs de ce projet, la plupart de ces objectifs s'insèrent plutôt harmonieusement dans notre vision du robot assistant qui est expliquée dans la section suivante.

1.3 Notre vision du robot d'assistance

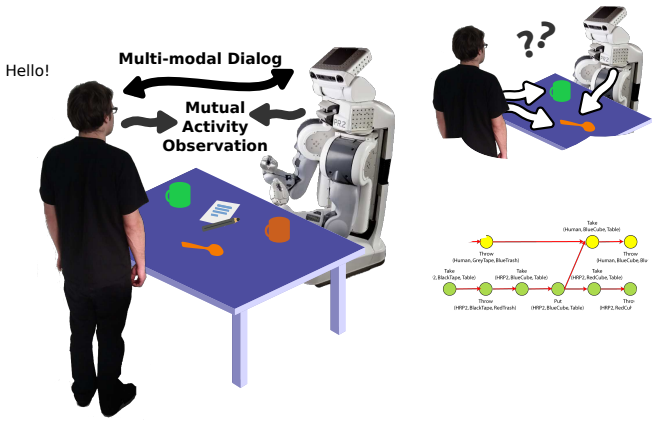


FIG. 1.1 – Robot raisonnant sur l'homme et lui-même à plusieurs niveaux : perception, action, interaction. Les sources d'information sont le dialogue multimodal et l'observation de l'environnement et des activités de l'homme.

Nous considérons des scénarios où un homme et un robot partagent un espace commun et un but commun (*shared goal*).

Le robot doit réaliser la tâche qui permettrait d'atteindre ce but en présence ou en interaction avec l'homme, et ce de la manière la plus satisfaisante selon des critères d'efficacité, de sécurité, d'acceptabilité, d'intentionnalité. Pour cela, le robot doit pouvoir planifier, délibérer de manière réactive, anticiper, raisonner.

Concrètement le robot doit d'abord être en mesure de localiser les objets, l'homme et lui-même. Il doit également raisonner sur ses propres capacités de perception et d'action comme sur celles de l'homme. Il doit pouvoir ensuite comprendre et même si possible anticiper les actions de l'homme. Il lui faut aussi planifier ses propres mouvements en prenant en compte l'homme. Le robot doit planifier la tâche permettant d'atteindre le but de haut niveau en partageant le travail avec l'homme. Il lui faut aussi disposer de protocoles permettant de communiquer et de se synchroniser avec l'homme dans la réalisation de cette tâche commune. Le robot doit bien entendu être en mesure de procéder avec l'homme à l'exécution de cette tâche en réagissant de ma-

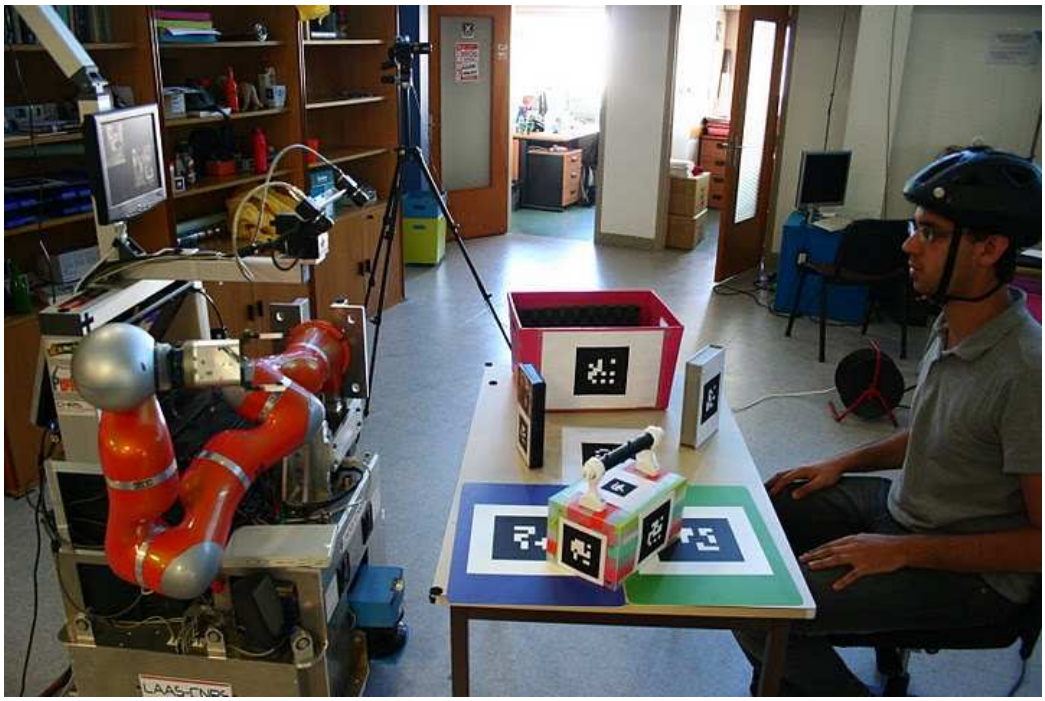


FIG. 1.2 – Jido face à un homme avec la table recouverte d'objets au milieu.

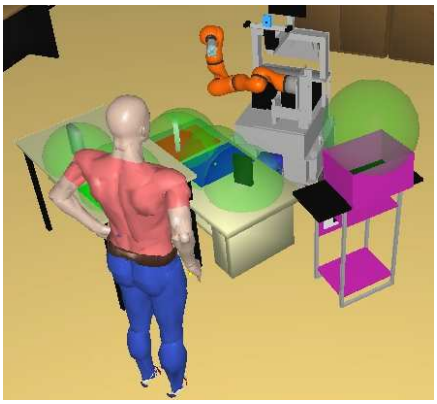
nière appropriée selon le déroulement de l'exécution. La figure 1.1) donne un exemple schématique de notre vision.

1.4 Robot et scénarios

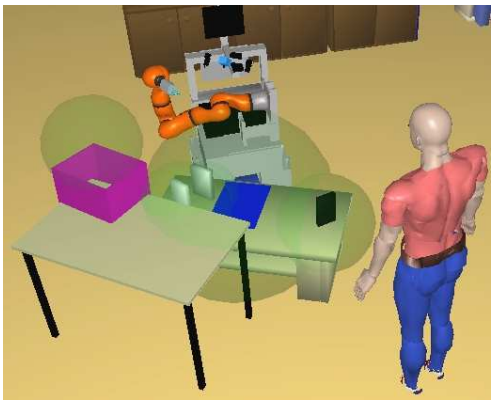
Nous avons utilisé le robot Jido situé à gauche dans la figure 1.2. Le robot a sa base fixe mais il peut manipuler des objets au moyen d'un bras Kuka. Les objets sont détectés par la caméra au moyen de tags grâce à ARToolkit [5]. L'homme est détecté par un capteur Kinect. La direction de sa tête peut aussi être détectée au moyen d'un système de capture de mouvement.

Nous avons choisi deux tâches différentes dans le cadre de nos projets. La tâche *nettoyer la table* consiste à mettre toutes les cassettes qui se trouvent sur une table dans une corbeille. La tâche *obtenir le jouet* consiste à attraper un petit objet initialement recouvert par une grosse boîte à fond vide. Il y a de très nombreux scénarios différents pour la tâche *nettoyer la table* selon le nombre de cassettes et l'atteignabilité initiale par les deux agents de la corbeille et des cassettes. La figure 6.6 donne l'exemple de 4 affichages du modèle 3d pour 4 scénarios différents. La difficulté est liée au fait que pour atteindre le but il faut réaliser un certain nombre d'actions qui impliquent le

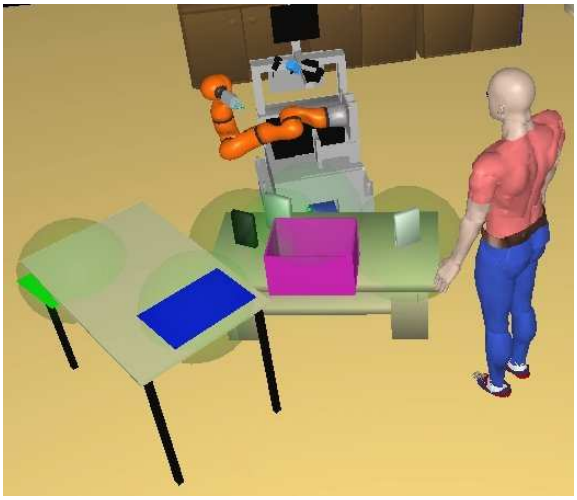
robot ou/et l'homme selon la disposition initiale des objets et des agents.



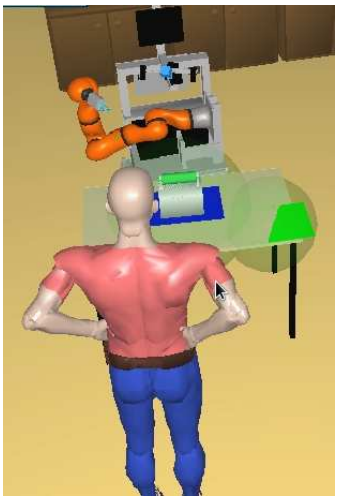
(a) corbeille atteignable par l'homme uniquement



(b) corbeille atteignable par le robot uniquement



(c) corbeille atteignable par les deux agents



(d) obtenir le jouet

FIG. 1.3 – Quatre scénarios. Un scénario est défini par une tâche et les positions initiales des agents et des objets.

1.5 Architecture et plan du manuscrit

1.5.1 Architecture

Voici la couche décisionnelle du robot. Elle correspond à l'instance actuelle de notre vision introduite par la section 1.3. Elle est constituée de plusieurs

entités qui ont chacune un rôle bien spécifique comme décrit dans la figure 1.4. Cette architecture permet de définir de manière précise ma contribution et de présenter le plan du manuscrit.

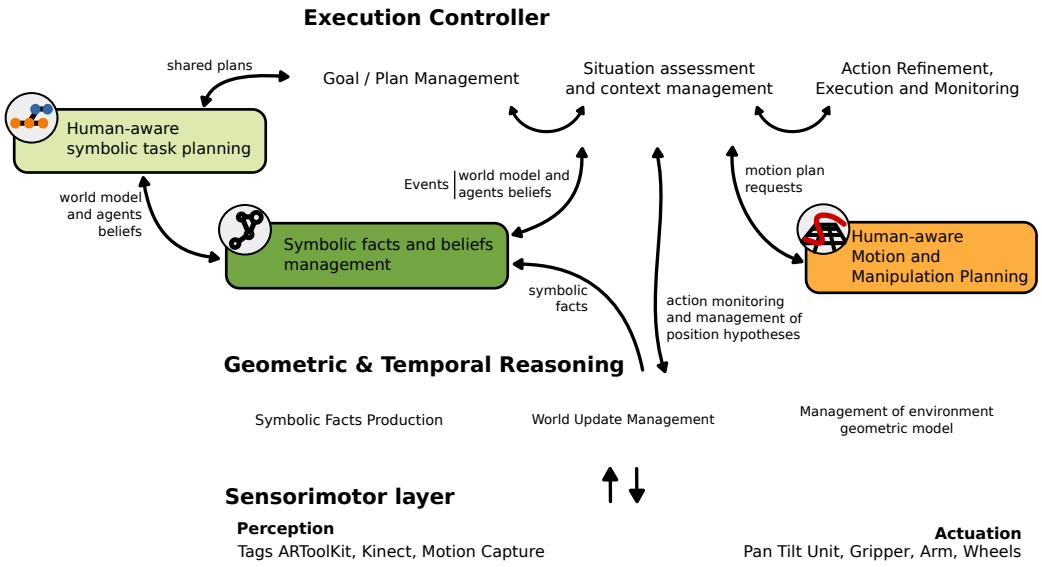


FIG. 1.4 – Architecture de la couche décisionnelle

Ma contribution centrale a consisté en l'élaboration du contrôleur de haut niveau (*execution controller* en haut de la figure 1.4). Il peut être subdivisé en 3 sous-activités qui sont présentées ci-dessous :

1. Construction et mise à jour de l'état du monde.
2. Gestion des buts et des plans.
3. Exécution et monitoring des actions.

Le contrôleur de haut niveau articule les composants identifiés dans la figure 1.4. Nous donnons ci-dessous le nom de l'instanciation de ces composants dans l'architecture actuelle.

- SPARK : module de raisonnement et de connaissance de l'espace. (Spatial Reasoning and Knowledge) [54]
- ORO : module de gestion de la connaissance symbolique [33]
- HATP : un planificateur de tâches qui prend en compte l'homme. (Human-Aware Task Planner) [3]
- Des planificateurs de mouvement, de placement et de prise prenant en compte l'homme.[55, 36, 41]

De ce fait, ma contribution a également consisté en la spécification et la réalisation d'interfaces avec ces différents composants. Je suis aussi directement intervenu sur le composant SPARK en proposant de nouvelles fonctionnalités de monitoring d'actions et d'hypothèses de positionnement qui étaient nécessaires à la sous-activité de mise à jour des états du monde.

1.5.2 Plan du manuscrit

Le manuscrit est découpé en quatre parties. La partie I fait office d'introduction générale. Elle est constituée du chapitre d'introduction 1 et du chapitre 2 qui propose une formalisation des états et processus d'un robot cognitif agissant de manière collaborative avec l'homme. Cette formalisation permet de mieux contextualiser les contributions dans le domaine. Je conclus le chapitre en l'appliquant à d'autres contributions du domaine et à la mienne.

La partie II décrit la première version du système complet. Les trois sous-activités du contrôleur d'exécution qui sont présentées ci-dessus sont déjà présentes. Le chapitre 3 décrit la sous-activité de construction et de mise à jour de l'état du monde. On explique d'abord la production de faits symboliques géométriques à partir de la représentation géométrique cartésienne du monde. Puis on présente les hypothèses de positionnement qui permettent de spatialiser un objet en l'absence de perception à partir de la connaissance d'un état symbolique de cet objet. On décrit ensuite les "moniteurs" d'actions qui permettent de reconnaître les actions de l'homme. Enfin on finit par introduire le mécanisme d'attention qui permet de focaliser la perception du robot. L'articulation de tous ces processus est également explicitée. Le chapitre 4 décrit la sous-activité de gestion des buts et des plans. Le planificateur de tâches est décrit rapidement. On aborde ensuite les automates de gestion des buts et des plans qui permettent d'articuler l'exécution des actions du robot et le monitoring des actions de l'homme. Le chapitre 5 explicite la sous-activité d'exécution et de monitoring des actions. La réalisation des actions du robot repose sur l'exécution d'automates qui traduisent une action symbolique en une séquence de mouvements planifiés et d'ouvertures ou de fermetures de la pince. Le "monitoring" d'une action est directement lié à l'attente du déclenchement du "moniteur" spécifique de cette action. Le chapitre 6 démontre le fonctionnement de cette première version du système au moyen d'expérimentations. On décrit rapidement l'architecture LAAS qui facilite l'implémentation. Le fonctionnement complet du système est ensuite illustré au moyen d'un scénario simple pour lequel on explicite tous les processus. Puis on présente les résultats d'une campagne intensive d'expérimentations menée en février 2012.

La partie III décrit l'enrichissement du système précédent par l'ajout de la gestion des croyances distinctes pour les agents. Le chapitre 7 explique comment ces croyances distinctes sont générées. Un algorithme a été conçu et mis en oeuvre pour permettre au robot de raisonner explicitement sur l'état des croyances de l'homme sur la position des objets dans le monde. Le robot peut ainsi identifier les différences avec ses propres croyances. Cette différence se reflète ensuite dans les faits symboliques synthétisant le monde géométrique cartésien. La fin du chapitre propose une extension de ces raisonnements sur les croyances distinctes à d'autres attributs que la position des objets. Le chapitre 8 décrit la gestion des croyances distinctes au niveau du planificateur de tâches. Il consiste en l'ajout d'actions de communication pour rétablir la consistance des croyances de l'homme et du robot seulement si c'est nécessaire et au moment le plus opportun. Le chapitre 9 présente des application de ces nouvelles capacités de raisonnement du robot. Deux exemples d'application au dialogue sont introduits. Enfin, trois expérimentations illustrent la génération des croyances distinctes et leur utilisation par le planificateur symbolique.

La partie IV propose quelques perspectives futures. Elle introduit d'abord deux nouvelles campagnes d'expérimentations pour mieux évaluer le système actuel puis décrit quelques pistes d'amélioration du système. Le chapitre 10 propose quelques perspectives d'amélioration du système. On s'intéresse d'abord à l'amélioration de la mise à jour de l'état du monde en introduisant une formalisation des états atteignables, puis en réfléchissant à l'utilisation d'une représentation probabiliste des états et son intérêt pour la perception, la reconnaissance d'actions et l'attention. Finalement on propose une perspective d'optimisation de l'utilisation du planificateur de mouvement en anticipant la planification des mouvements les plus probables.

Le chapitre 11 conclut le manuscrit en tentant de tirer les leçons du système proposé et d'en rappeler également les limites.

Chapitre 2

Formalisation des états et processus

2.1 Description de la formalisation

Quelles sont les connaissances et les briques décisionnelles que devraient posséder un robot cognitif qui réalise des buts impliquant le déplacement dans l'environnement et la manipulation d'objets réels de manière collaborative avec l'homme ? Je propose dans cette partie une formalisation en termes d'états et de processus. Mon ambition est d'être le plus général possible même si cette formalisation est fortement inspirée et illustrée par mon travail et les articles que j'ai lus au cours de ma thèse. Cette formalisation dessine un cadre global qui me permet d'introduire le vocabulaire et de situer mon travail ainsi que d'autres contributions. Pour plus de clarté, je repousse la mention des références bibliographiques à la fin de ce chapitre dans la section où je contextualise certaines références au moyen du formalisme introduit en première partie de chapitre.

2.1.1 Description générale

La figure 2.1 propose une description générale des états et des processus. Les états sont représentés par des ovales. Ils peuvent être eux-mêmes subdivisés en rectangles arrondis pour représenter les différentes croyances des agents sur ces états et la croyance des autres agents sur ces états. Les rectangles arrondis grisés représentent les valeurs qu'a l'homme de ces croyances. Elles sont inconnues par le robot. Les rectangles arrondis colorés représentent les croyances du robot sur ces croyances. Les processus sont représentés par des rectangles. Les rectangles grisées représentent les processus de l'homme inconnu du robot. Les rectangles beiges représentent les processus du robot.

A chaque processus de l'homme peuvent être associés deux processus du robot qui correspondent à l'émulation du processus de l'homme par le robot et à son inversion. L'émulation est un modèle qu'a le robot de ce que devrait produire le processus de l'homme pour des entrées données. L'inversion est un modèle qu'a le robot de ce qui aurait dû être les entrées du processus de l'homme compte tenu des sorties qui sont connues. De même, le robot peut raisonner sur le fait que l'homme émule et inverse les processus du robot. Le schéma peut être divisé en deux parties verticalement. La partie haute concerne les états et processus liés à l'homme. La partie basse les états et processus liés au robot. Les états au centre représentent les états *physiques* du monde. Par états *physiques*, on entend qu'un observateur extérieur qui pourrait se déplacer à sa guise dans l'environnement alors que le temps se serait arrêté pourrait déterminer tous ces états. Ils ne dépendent que de la position des objets et des agents dans le monde. Plus on s'éloigne du centre verticalement, plus l'on progresse vers les états *mentaux* d'un agent. À l'inverse des états *physiques*, les états *mentaux* ne sont pas directement accessibles par l'observation du monde à un instant donné. Pour pouvoir y accéder de manière exacte, il faudrait rentrer à l'intérieur des programmes du robot ou du cerveau de l'homme. De fait, ils ne peuvent qu'être inférés à partir de l'activité des agents.

2.1.2 Les états physiques

Dans ce schéma on distingue deux niveaux de représentation des entités dans le monde :

- Niveau 1 : niveau géométrique cartésien.
- Niveau 2 : niveau géométrique symbolique.

Pour la représentation de niveau 1, on adopte un repère cartésien. Les entités sont représentées par des volumes élémentaires possiblement articulés. Les capteurs de type vision sont décrits géométriquement par leur direction et focales qui permettent de calculer l'image perçue.

La représentation de niveau 2 est à un niveau plus synthétique et discrétisé que la représentation de niveau 1 introduite ci-dessus. Ces faits appartiennent pour l'essentiel aux trois catégories suivantes :

- Positions. "Box isOn Table"
- Perspective taking : la perception des agents. "Human looksAt Box"

– Affordances : la capacité d'action des agents. "Box isReachableBy Human"

Qu'est qui entraîne la mise à jour de ces états ? :

– Acquisition de nouvelle information. (perception, dialogue, résultats des actions.)
– Raisonnement au sein d'un des niveaux.
– Mise à jour entre les niveaux.

La perception permet de mettre à jour le niveau 1 de manière directe (le robot perçoit un objet à la position x,y,z) ou indirecte en utilisant du raisonnement spatial (Je ne vois pas la cassette sur la table. Si elle est sur la table alors elle est à une position inclue dans cette union de zones invisibles pour moi). La projection des effets d'une action (l'homme a saisi la cassette) ou l'intégration d'informations fournies par le dialogue (l'homme dit au robot que la cassette est sur la table) vont mettre à jour le niveau 2. Des raisonnements de type sens commun permettent de raisonner directement au niveau 2 : le beurre est habituellement dans le réfrigérateur. "Butter isIn Fridge". Toute mise à jour dans l'un de ces deux niveaux doit être transmise à l'autre niveau. Niveau 1 à 2 : calculs géométriques des faits à partir des positions. Niveau 2 à 1 : calculs inverses consistant en la spatialisation de faits symboliques par l'échantillonnage dans l'ensemble des positions géométriques qui vérifient la description symbolique.

Nous décrivons ci-dessous un exemple de "résonance" entre les deux niveaux :

– Nouvelle information au niveau 2 : L'homme dit au robot que la boîte est sur la table.
– Niveau2 –> Niveau1 : spatialisation de isOn table.
– Raisonnement Niveau1 : Le robot explore sans succès la surface de la table. Il existe une seule zone de la table qu'il ne peut observer, c'est celle qui se trouve derrière la grosse boîte dans la perspective du robot.
– Niveau1 –> Niveau2 : La boîte est derrière la grosse boîte.

La notion d'incertitude est essentielle. La perception peut produire des positions ayant un certain niveau d'incertitude. En cas d'absence de perception, cette incertitude est encore plus flagrante dans la mesure ou les hypothèses sur la position peuvent être alors disjointes compte tenu des états symboliques atteignables.

La prise en compte fine des incertitudes devrait permettre d'estimer la qualité de l'état de la connaissance actuelle du monde. C'est un élément essentiel à prendre en compte pour arbitrer entre exploration et exploitation. Par exemple, pouvons-nous raisonnablement valider le résultat d'une action compte tenu du niveau d'incertitude sur l'état du monde ou doit-on préalablement raisonner et explorer pour améliorer cette connaissance ? On identifie trois sous modèles pour les états du monde physiques :

- Rb (Robot believes some state of the world) : ce que croit le robot sur un état du monde.
- RbHb (Robot believes that human believes some state of the world) : ce que le robot croit que l'homme croit sur un état du monde.
- RbHbRb (Robot believes that human believes that robot believes some state of the world) : ce que le robot croit que l'homme croit que le robot croit sur un état du monde.

Les figures 2.2 et 2.3 permettent d'illustrer dans quelle mesure ils peuvent être effectivement distincts. Un homme et un robot peuvent être chacun assis en face de la table ou absent. Il y a deux grand conteneurs ouverts et deux petites boîtes. L'homme ou le robot peuvent déplacer les petites boîtes. Le point de départ de l'objet est représenté par un cercle rouge et le point d'arrivée par un cercle vert. Le temps s'écoule de gauche à droite. Les deux figures se suivent temporellement. On représente ici 5 modèles distincts. Les modèles B, C et D représentent respectivement Rb, RbHb et RbHbRb. Le modèle A représente la connaissance exacte de l'état du monde perçu par un observateur omniscient. Le modèle E représente la connaissance exacte de l'état du monde de l'homme qui peut être bien sûr différente de la croyance qu'a le robot de cette connaissance. L'empilement vertical de différentes images permet de représenter les différents modèles à un instant donnée. On entoure une petite boîte d'une sphère noire transparente pour représenter le fait que l'agent imagine la position de cet objet qu'il ne peut pas percevoir de manière exacte.

Dans la figure 2.2, la quatrième vignette en partant de la gauche montre le robot qui déplace la boîte noire alors que l'homme est absent. Quand l'homme revient, il ne perçoit pas directement cette transition si bien que les croyances des différents agents sont désormais en partie distinctes comme indiqué dans la dernière colonne de vignettes.

Le plus souvent le RbHb (croyance du robot sur la croyance de l'homme) et le RbHbRb (croyance du robot sur la croyance de l'homme sur la croyance du robot) sont identiques. Cependant, si le robot comprend que l'homme se trompe sur la perception par le robot d'une transition, alors ces modèles sont

alors distincts. La perception et le raisonnement associé sont bien sûr hors de portée de la plupart des robots actuels hors simplifications importantes.

States and processes for human robot collaborative task achievement

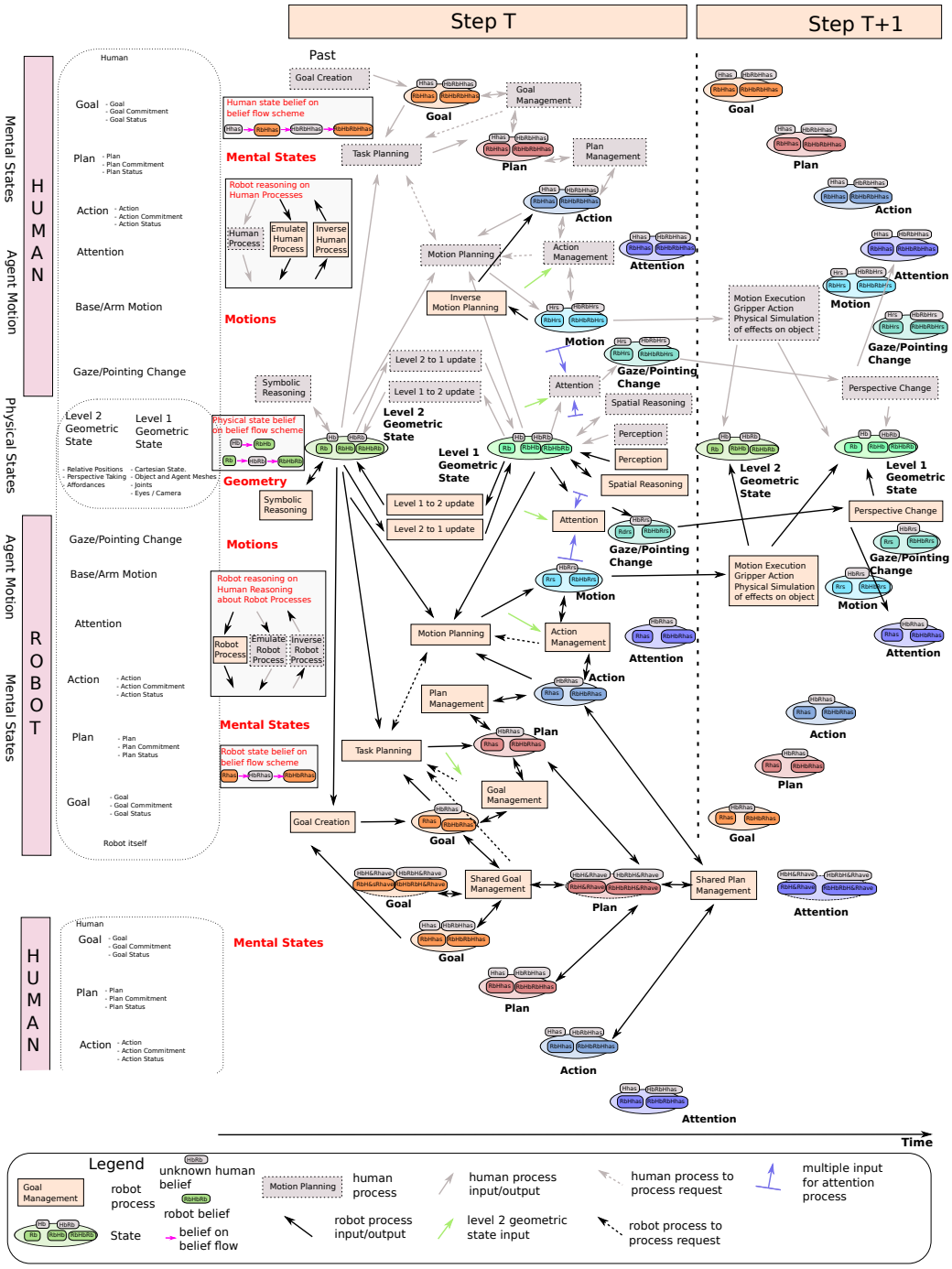


FIG. 2.1 – *
États et processus

FIG. 2.2 – *
Historique d'interaction 1

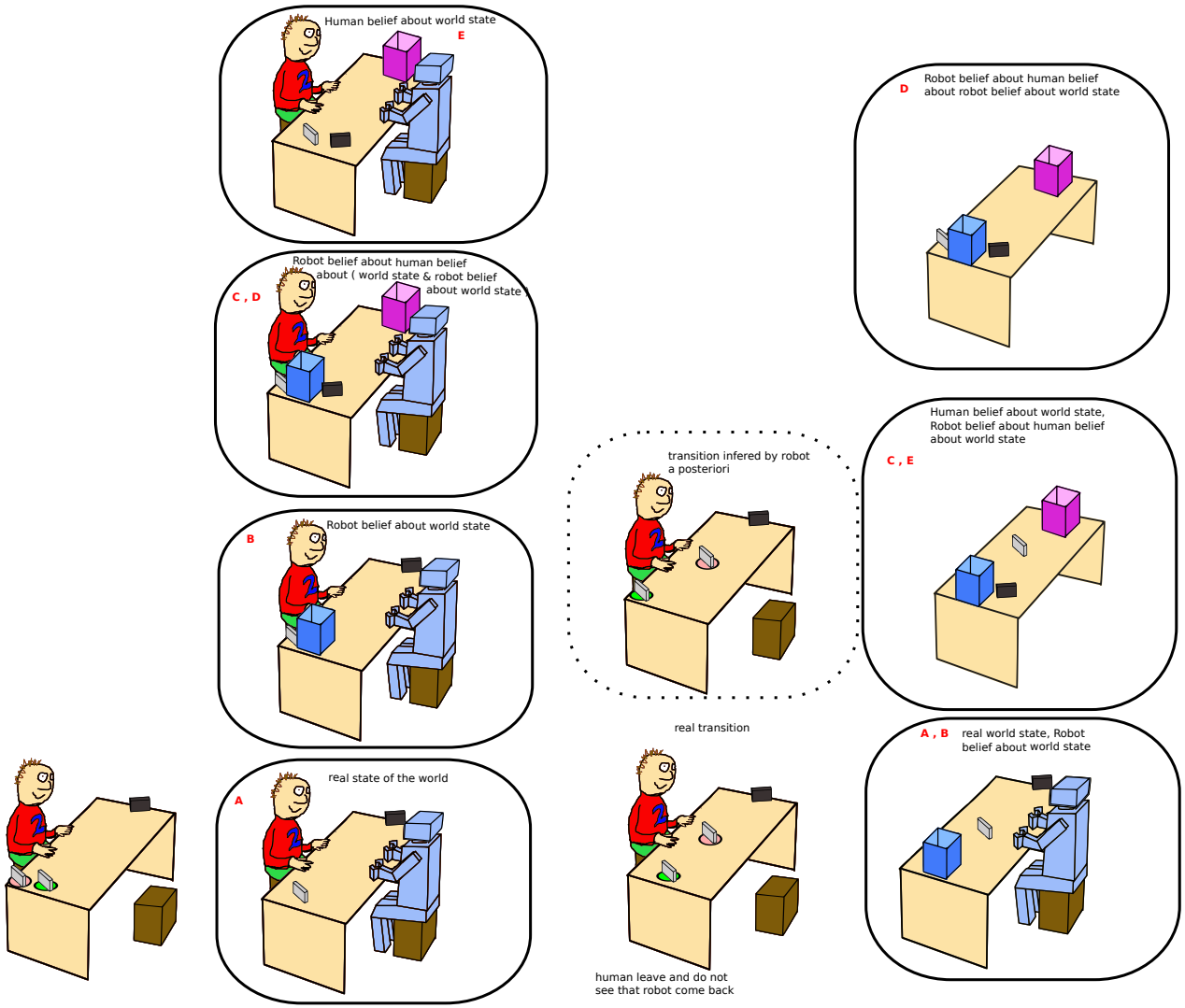


FIG. 2.3 – *
Historique d'interaction 2

2.1.3 Les états mentaux

Comme expliqué plus haut, les états *mentaux* ne sont pas directement accessibles par l'observation du monde à un instant donné et ne peuvent qu'être inférés à partir de l'activité des agents. On identifie les quatre états mentaux suivants :

- Buts
- Actions
- Plans
- Attention

Un but correspond à un sous-ensemble des états du monde que l'agent souhaite atteindre. Par exemple, si le but est d'avoir une boîte de céréales sur la table et qu'il y a deux boîtes de céréales accessibles dans le placard. Le but est atteint si l'on arrive à un état du monde dans lequel il y a au moins une des deux boîtes de céréales sur la table. Dans la suite du document, nous adoptons la définition suivante pour le concept d'action. Une action est un opérateur de l'agent permettant de réaliser un but directement accessible. C'est-à-dire que l'état actuel est proche du but et que cette action a donc une traduction immédiate en un nombre réduit de mouvements et de déplacements entraînant un résultat clairement identifié sur un objet ou un agent. Une action peut être initialisée si l'état du monde vérifie un certain nombre de préconditions propres à cette action. La réalisation nominale de l'action produit des effets connus spécifiques de cette action. Dans notre contexte, on peut considérer les actions suivantes : prendre un objet accessible, déposer un objet sur un support accessible, se déplacer d'un point à un autre qui est proche et facilement accessible depuis le premier. Le choix du niveau de granularité dans la définition des actions peut paraître arbitraire. Il dépend bien sûr du contexte applicatif et de la modélisation du domaine. Dans certains contextes on pourrait avoir une granularité beaucoup plus grossière. Considérons l'emploi du temps d'un commercial d'une entreprise. Une action pourrait être d'aller rencontrer un client. On tait toutes les étapes correspondant à la réalisation de cette action. À l'inverse on pourrait avoir une granularité beaucoup plus fine. Considérons par exemple l'utilisation d'une main robotique. Dans ce contexte, la modification de l'angle d'une articulation peut être considérée comme une action. Nous avons choisi un niveau de granularité adapté à notre contexte, correspondant grosso modo en un découpage intuitif en termes soit de mouvement des bras en lien avec un objet alors que le corps de l'homme ou la base du robot est fixe ou de déplacement de faible distance du corps de l'homme ou de la base du robot. Un plan est un ensemble de séquences d'actions dont la réalisation doit per-

mettre d'atteindre le but. Ce plan peut être plus ou moins détaillé en termes d'ordonnancement des actions et d'allocation des actions à un acteur donné. Par état d'attention on désigne les objets ou agents sur lesquels l'agent focalise sa perception. C'est l'état mental le plus facilement déductible de l'état physique du monde comme il est directement lié à l'orientation des capteurs. Il peut demeurer néanmoins une ambiguïté importante s'il existe plusieurs candidats potentiels pour une direction des capteurs donnée.

On distingue les états mentaux du robot qu'il choisit ou calcule, de ceux de l'homme que le robot ne peut qu'inférer ou être informé par le dialogue.

États mentaux du robot :

- Rhas (Robot has a mental state) : le robot a un état mental donné.
- RbHbRhas (Robot believes human believes robot has a mental state) : le robot croit que l'homme croit que le robot a un état mental donné.

Le robot choisit ses états mentaux compte tenu du contexte. Concernant la croyance de l'homme sur son état mental, le robot doit raisonner sur la capacité de l'homme à interpréter ce qu'il a perçu ou que le robot lui a communiqué.

États mentaux de l'homme :

- RbHhas (Robot believes human has a mental state) : le robot croit que l'homme a un état mental donné.
- RbHbRbHhas (Robot believes human believes robot believes human has a mental state) : le robot croit que l'homme croit que le robot croit que l'homme a un état mental donné.

Le robot devrait inférer les états mentaux de l'homme ou les déduire de l'information que l'homme lui communique. Le robot devrait inférer que l'homme croit que le robot croit que l'homme a un état mental donné, au moyen du modèle qu'a le robot de la capacité de l'homme à raisonner sur la compréhension du robot des états mentaux de l'homme.

L'agent doit se choisir un but. On peut définir des règles de déclenchement de but en fonction d'une situation donnée. Par exemple, on pourrait avoir la règle suivante : il faut débarrasser la table si celle-ci est en chantier depuis au moins cinq minutes et qu'il n'y a plus personne. Une fois qu'il a un but, l'agent doit mettre en oeuvre sa réalisation en trouvant un plan qu'il doit réaliser et en vérifiant si le but est atteint. La planification de but consiste à obtenir un plan en vue d'atteindre le but. Une fois qu'il a un plan, l'agent doit

réaliser les actions qui le composent tant que le plan reste valide. Pour chaque action, l'agent doit vérifier qu'elle peut être mise en oeuvre, puis la mettre en oeuvre et vérifier qu'elle a été effectivement réalisée. Nous donnons ici un aperçu très global de ces différents processus. Dans la pratique ils peuvent être raffinés énormément.

Dans la partie basse du schéma, nous avons reproduit les états mentaux de l'homme pour les mettre en face de ceux du robot. L'objectif est d'introduire les états mentaux partagés qui sont clefs dans le contexte de la collaboration homme-robot :

- But Partagé.
- Plan partagé
- Attention partagée.

- RbH&Rhave (Le robot croit que l'homme et le robot ont un état mental partagé) : le robot croit que l'homme et le robot ont un état mental partagé.
- RbHbRbH&Rhave (Le robot croit que l'homme croit que le robot croit que l'homme et le robot ont un état mental partagé) : le robot croit que l'homme croit que le robot croit que l'homme et le robot ont un état mental partagé.

Les états mentaux partagés et leurs processus associés requièrent un niveau de communication et de synchronisation supplémentaire par rapport aux états mentaux individuels. Les deux agents doivent s'accorder sur le choix de l'état mental partagé. Ils doivent s'assurer de l'engagement de l'autre agent. Ils doivent aussi partager le statut final de l'état mental partagé.

Il existe de nombreuses variantes de ces protocoles de gestion des états partagés. Ils peuvent tenir compte de l'habitude des agents à travailler ensemble et de leurs capacités respectives. (Experts. Parents et enfants.) Par exemple, dans quelle mesure les actions sont exécutées en parallèle ou l'une après l'autre, quel niveau de monitoring des actions de l'autre agent choisit-on ?

2.1.4 Les mouvements

On parle ici de mouvements au sens large. Il s'agit aussi bien des déplacements de l'agent complet, que des mouvements du bras, de l'ouverture ou de la fermeture de la main ou de la pince mais aussi des changements d'orientation du regard. Les mouvements sont un pont entre les états mentaux et la représentation du monde physique. Ils trahissent l'intentionnalité, l'attention.

Sous-états pour un mouvement du robot :

- Rrs (Robot realizes a motion) : le robot exécute un mouvement.
- RbHbRrs (Robot believes human believes robot realises a motion) : le robot croit que l'homme croit que le robot réalise un mouvement.

La planification de mouvement est une fonctionnalité essentielle dont doit être doté notre robot. Le robot doit posséder un répertoire de mouvements qu'il est en mesure de planifier et d'exécuter. Dans le cadre de l'interaction avec l'homme, cette planification de mouvement doit prendre en compte l'homme explicitement à plusieurs niveaux ; sécurité, visibilité, confort, expressivité...

Le robot doit aussi raisonner sur la perception et la compréhension de ses mouvements par l'homme.

La séparation entre la planification symbolique et la planification géométrique peut parfois conduire à une impasse. Le plan symbolique produit peut ne pas être exécutable si certaines informations géométriques ont été négligées à cette étape. Un certains nombre d'efforts visent à renforcer le lien entre planification symbolique et géométrique pour prendre en compte des contraintes géométriques fortes a priori ou suite à un premier échec de la planification symbolique.

Sous-états pour les mouvements de l'homme :

- RbHrs (Robot believes human realizes a motion) : le robot croit que l'homme réalise un mouvement.
- RbHbRbHrs (Robot believes human believes robot believes human realizes a motion) : le robot croit que l'homme croit que le robot croit que l'homme réalise un mouvement.

Le robot devrait être en mesure d'interpréter les mouvements de l'homme en termes d'action et d'attention. Il s'agit en quelque sorte d'avoir un modèle de la planification de mouvement de l'homme pour pouvoir interpréter de manière inverse l'action ou l'intention qui le cause. Le robot doit aussi modéliser la connaissance que l'homme a sur la capacité du robot à interpréter ses actions.

S'ils ne sont pas directement observés, le robot déduit les effets de ses actions et de celles de l'homme sur le monde. Il peut aussi inférer le niveau de compréhension par l'homme de ces effets.

2.1.5 Gestion de l'attention.

Par attention on entend la politique de décision d'orientation des capteurs. Il s'agit des yeux pour les humains et généralement les caméras qui sont leurs équivalents pour nos robots qui sont anthropomorphes. La direction du capteur joue d'abord un rôle dans le choix d'acquisition d'informations mais elle remplit également un rôle de communication de l'attention. (Un agent regarde un autre agent non pas pour voir ses yeux mais aussi pour lui communiquer quelque chose).

L'attention doit arbitrer entre de multiples demandes compétitives sur la ressource orientation capteur. Elle est fortement influencée par la qualité des états du monde physique. La nécessité de réduire les incertitudes sur ces états déclenche des mécanismes d'exploration. Ce comportement de base est fortement modulé par les états mentaux qui vont focaliser sur un besoin d'observation ou de communication précis lié à l'action et au but en cours.

Théoriquement, on devrait pouvoir modéliser l'attention en termes de théorie de la décision. Compte tenu de l'état de la connaissance du monde, de l'utilité d'acquérir telle information ou communiquer tel message modulée par la probabilité d'acquérir effectivement cette information ou de communiquer ce message, quelle direction maximise l'utilité ?

2.1.6 Historique d'interaction / inférences

Le présent n'a de sens qu'à la lumière du temps qui s'est écoulé jusque là. Les états de l'homme et du robot à un instant donné sont construits ou doivent être interprétés dans ce sens.

Le contenu de ces états résonne temporellement entre eux selon les flèches d'influences représentées sur le schéma. L'homme comme le robot peuvent en "jonglant" d'un modèle à l'autre réaliser des inférences temporelles. Inférence ascendante : de l'état du monde, des mouvements vers les états mentaux. Une succession d'actions évoque par exemple un plan qui évoque un but. Inférence descendante : des états mentaux, vers les mouvements, vers les états du monde. Si l'on connaît le but, on peut estimer le plan et l'action et en déduire la croyance éventuellement erronée de l'homme sur l'état du monde. L'homme va chercher un objet à une table. L'homme croit que l'objet est sur cette table.

2.1.7 Dialogue

Le dialogue n'est pas mentionné explicitement dans le schéma. Tout peut passer par le dialogue : informer, interroger sur des états physiques ou men-

taux, synchroniser les différents modèles. Le dialogue sera plus ou moins explicite selon l'état de la connaissance des différents agents et de leur capacité d'inférence.

2.1.8 Théorie de l'esprit

La théorie de l'esprit désigne la capacité d'un agent à attribuer à un autre agent des états et processus mentaux et à raisonner dessus. Elle est décrite plus en détail dans la sous séction suivante 2.2.1. La théorie de l'esprit est au coeur de notre schéma. Les états mentaux et processus que le robot attribue à l'homme sont explicités.

2.2 Utilisation de la formalisation

Ce schéma, qui n'est certainement pas pour autant complètement exhaustif, propose un nombre important d'états et de processus nécessaires à un robot cognitif interagissant avec l'homme. Comme explicité précédemment, il est utile pour situer mon travail mais également des travaux significatifs du domaine.

2.2.1 Littérature

En psychologie, la prise de perspective (*perspective taking*), introduite par Flavell et Tversky [16, 60], désigne la capacité d'un individu à raisonner sur la compréhension qu'ont les autres du monde en termes de perception visuelle, de description spatiale et d'affordance. Elle est essentielle à l'interaction. Elle est devenue populaire depuis quelques années dans le domaine de l'interaction homme-robot. Breazeal [9] présente un algorithme d'apprentissage qui prend en compte l'information sur la perspective visuelle de l'enseignant. Johnson [28] utilise la prise de perspective visuelle dans le contexte de la reconnaissance d'actions entre deux hommes. Trafton [59] utilise à la fois la prise de perspective visuelle et spatiale pour identifier l'objet indiqué par un partenaire humain. Ros [47] exploite la prise de perspective pour résoudre des ambiguïtés du dialogue.

La prise de perspective visuelle est calculée à partir de la représentation cartésienne de la géométrie et est décrite sous la forme de faits symboliques. À ce titre, elle fait partie des processus de type *Level* 1 *to* 2 *update* indiqués dans la figure 2.1

Mavridis [37] décrit comment une information transmise par le dialogue peut être spatialisée dans la représentation cartésienne du monde au même

titre que les données issues de la perception. Dans notre schéma il s'agit clairement d'un processus de type $Level2to1update$

La présence de l'homme doit être prise en compte dans la planification de mouvement du robot selon Kulic, Berg et Madhav [32, 6, 35], pour la navigation d'après Althaus et Sisbot [4, 53] comme pour la manipulation selon Kemp [30]. C'est le processus *motion planning* du robot qui tient compte, entre autres, de la posture de l'homme dans ses entrées.

La direction du regard impacte l'influx perceptuel mais sert également la communication avec les autres agents. Le robot doit décider où regarder et quand. C'est le processus d'*Attention* décrit dans la figure. Karaoguz [29] explore cette question. Demiris [15] décrit comment la reconnaissance d'action peut également impacter sur l'attention.

En psychologie, la théorie de l'esprit, en anglais theory of mind, (ToM) désigne la capacité à attribuer des états mentaux *e.g.* croyances, buts, intentions (plans) à soi-même et aux autres. La prise de perspective visuelle introduite plus haut est l'un des précurseurs les plus importants de la ToM. Leslie [34] démontre qu'elle apparaît très tôt dans le développement de l'enfant . La ToM regroupe une large gamme de compétences de la prise de perspective visuelle instantanée jusqu'à l'interprétation complexe des buts, plans et sentiments des autres sur une longue période. Plus sa ToM est développée, plus un agent sera performant quand il interagira avec d'autres dans un contexte collaboratif ou compétitif.

La ToM est omniprésente dans le schéma de la figure 2.1. En effet, les états de l'homme sont explicitement représentés ainsi que ses processus. La croyance de l'homme sur les états du robot est aussi présente et même sa croyance sur la croyance du robot sur ses propres états à lui.

Scassellati [48] a été l'un des premiers à introduire la ToM dans un contexte robotique. Il présente les modèles de Leslie's et Baron-Cohen's ToM et leurs utilisations potentielles sur un robot.

La capacité à inférer que d'autres ont une croyance sur l'état du monde qui diffère de la sienne est considérée comme une étape cruciale du développement de la ToM. En psychologie, la tâche avec croyance fausse ($false\ belief\ task$) a été formulée pour la première fois par Wimmer [63] en 1983.

Dans la figure 2.1, les deux états géométriques cartésiens et symboliques (en vert) sont subdivisés en 3 sous-états qui permettent d'exprimer ces divergences de croyances.

Breazal dans [10] décrit l'une des premières implémentations pour l'interaction homme - robot. Elle considère même que la tâche avec croyance fausse peut devenir un *benchmark* pour évaluer les compétences de robots

sociaux cognitifs. Breazal défend une conception de la ToM de type "Simulation Theory". Le robot utilise ses propres mécanismes comme modèle des processus de l'homme. Ainsi, le robot utilise d'abord ses propres capacités de modélisation du monde pour détecter la formation de croyances divergentes, en prenant la perspective visuelle de l'homme. Qu'est-ce que le robot verrait s'il était positioné là où est l'homme et avait le même champ de vision que l'homme ? Puis le robot va utiliser ses propres mécanismes de réalisation des actions pour interpréter le mouvement observé de l'homme, comme étant la réalisation d'une action précise. Quelle serait l'action du robot qui lui ferait réaliser un mouvement proche de celui que réalise l'homme ? Enfin le robot utilise ses propres mécanismes de traduction de buts en actions pour déterminer le but de l'homme à partir de ses actions et de ses croyances. Quel serait le but du robot qui le conduirait à réaliser les actions que réalise l'homme, si le robot avait les croyances qu'a l'homme ?

Kennedy et al dans [31] proposent une approche de la ToM intitulée *simulation "comme moi"*. L'agent utilise ses propres connaissances et capacités comme modèle d'un autre agent, pour prédire les actions de cet agent. Trois exemples d'une *simulation "comme moi"* dans un contexte social sont implémentés dans la version "embodied" de l'architecture cognitive (ACT-R) Adaptative Control of Thought-Rational. ACT-R/E (pour ACT-R Embodied). Ces exemples montrent la pertinence d'une approche simulation pour modéliser :

- la prise de perspective.
- le travail d'équipe.
- le comportement dominant/dominé.

Hiat et al dans [22] relèvent certaines limitations dans le travail de [10] et [31]. Selon eux ces auteurs utilisent un système de gestion des croyances qui suit les croyances d'un partenaire humain, et ils supposent qu'il est possible de connaître les croyances sur le monde sur la seule base des observations. Bien que les deux approches fournissent des informations utiles sur la théorie de l'esprit, elles sont limitées du fait d'hypothèses fortement restrictives de par leur caractère très optimiste : un agent parvient à déterminer de manière exacte ce qu'un autre connaît sur la seule base des observations. Les inconsistances entre le comportement prévu et celui observé sont uniquement attribuables au fait d'avoir des croyances distinctes sur l'état du monde. Hiat et al tentent donc de remettre en cause cette hypothèse selon laquelle la variabilité dans le comportement de l'homme résulte uniquement de différences sur ses croyances à propos des états du monde. La variabilité dans le comportement de l'homme peut aussi être attribuable à l'existence de chemins

multiples permettant la réalisation d'un même but pour des mêmes croyances.

Breazeal dans [19] décrit un robot qui est en mesure de manipuler les croyances de l'homme à son égard. Le robot modélise comment les états mentaux de l'homme sont modifiés par la perception visuelle du monde qui l'entoure. Cette modélisation est associée à une simulation du futur immédiat, suffisamment précise géométriquement pour permettre de prendre en compte la perspective des différents agents. Cela lui permet d'estimer quelles sont les actions qu'il doit mettre en oeuvre pour faire croire à l'homme que le robot a un but précis. Cette capacité est testée expérimentalement. Les observateurs humains, une fois qu'ils ont compris cette capacité, réévaluent positivement la capacité d'interaction du robot.

Butterfield [11] cherche à développer des modèles mathématiques qui reflètent les résultats des études expérimentales sur la ToM et plus généralement des processus d'interaction sociale. Les actions qu'un agent veut réaliser sont considérées comme des variables cachées conditionnées à la fois par les observations de cet agent sur l'état du monde et l'inférence de l'agent sur les intentions des autres agents. Chaque processus de ToM est décrit de manière probabiliste comme un Champ aléatoire de Markov, où l'interaction entre les différentes variables décide le fonctionnement global du système. Les variables cachées représentent les états de chaque agent et les variables observées représentent les observations de chaque agent. Les champs aléatoires de Markov peuvent être paramétrés pour décrire différents processus de ToM, en s'adaptant à différentes situations homme-robot par un choix pertinent des fonctions d'évidence et de compatibilité. L'auteur ne cherche pas à savoir comment observer ou inférer les états mentaux des autres agents, mais plutôt à déterminer comment ces états, une fois qu'ils sont observés, influencent de manière collective la prise de décision. Des champs de Markov sont proposés pour les différents processus suivants :

- L'impact de la certitude affichée du tuteur sur la modification de croyance de l'élève.
- L'attention partagée.
- La prise de décision collective sur le partage des actions.

S'ils ont un but commun, l'homme et le robot doivent choisir et exécuter un plan partagé. Le robot doit suivre l'implication de l'homme et décider des actions et des actes de communication les plus appropriés pour exécuter le plan de manière efficace et fluide, d'après Hoffman et Shah [23, 50]. Plusieurs théories sur la collaboration proposées par Cohen, Grosz et Clark [14, 20, 13] soulignent que les but collaboratifs ont certaines exigences spé-

cifiques par rapport aux buts individuels. Puisque le robot et la personne partagent un même but, ils doivent s'accorder sur la manière de le réaliser, ils doivent démontrer leur implication tout au long de l'exécution, etc. Plusieurs systèmes robotiques ont déjà été bâtis sur ces théories par Rich, Sidner, Tambe et Breazeal [45, 51, 57, 8] Ils ont justifié la pertinence de cette approche. Ils ont également montré combien il est difficile de gérer le $turn - taking$ entre les partenaires et d'alterner la réalisation du but et la communication de manière générique. Plus récemment, Rich [44] a proposé une méthode pour identifier le niveau d'engagement du partenaire dans la tâche. Sur cette base, Aaron [24] génère des actes de communication pour rétablir ou entretenir l'engagement du partenaire dans la réalisation de la tâche. Cela correspond aux processus *Shared Goal Management* et *Shared Plan Management* en bas de la figure 2.1.

2.2.2 Ma contribution

L'originalité de ce travail consiste essentiellement dans l'ambition de couvrir un large spectre des états et processus décrits dans le schéma. Assez naturellement cela se traduit par une ambition plus limitée au niveau des composants individuels.

Une part de ma contribution se situe au niveau des états physiques du monde, que cela soit en termes de construction ou de mise à jour. Tout cela sera explicité dans le chapitre 3. Un résumé des différents points est présenté ci-dessous.
Le calcul des états géométriques symboliques à partir de l'état géométrique a été affiné. (Level 1 to 2 update). Les actions qui définissent la dynamique du monde ont été modélisées en termes de préconditions et d'effets. Un mécanisme d'inférences "ascendantes" simple mais robuste permet d'estimer l'action en cours, à partir du déplacement de la main de l'homme repéré par des "monitors". La projection des effets des actions dans la représentation du monde symbolique niveau 2 et sa traduction dans le niveau 1 (Level 2 to 1 update) a été mise en place. On réalise une détection d'anomalies dans le niveau 2 à partir de critères géométriques au niveau 1 (Level 1 to 2 update). Un système attentionnel équilibre l'exploration, la focalisation sur l'action, la focalisation sur des événements de type action de l'homme ou sur une inconsistance de l'état du monde. On détecte des divergences dans l'état du monde de l'homme et du robot grâce à des raisonnements sur la perception des transitions.

Une mécanique de création et mise en oeuvre de buts partagés a été réalisée. Elle repose sur un certain nombre d'hypothèses simplificatrices fortes.

Le but est donné par l'homme : R&Hhas = RbHbRbR&Hhas = Rhas = RbHbRhas = RbHbRbHhas = RbHhas, et est considéré comme compris et accepté par les deux agents. Le plan est calculé par le robot et est considéré comme compréhensible par l'homme au fur et à mesure de son exécution, par la simple énumération des actions qui le constituent. Les actions sont annoncées par le robot verbalement et sont considérées comme étant acceptées et comprises par les deux agents. (RbHhas = RbHbRbHhas.) (Rhas = RbHbRhas.) La motivation de l'homme est considérée comme acquise, à l'exception de son refus d'agir trois fois d'affilée.

La troisième contribution a consisté à articuler la planification de mouvement ets de tâches. Une typologie des trajectoires de la pince du robot a été définie en collaboration avec les spécialistes du motion planning. Des automates ont été définis pour traduire une action symbolique en trajectoires élémentaires compte tenu de l'état du monde.

Au delà de ces contributions individuelles, c'est la mise en oeuvre complète du système qui est ma contribution la plus forte. Elle se fait dans un contexte certes limité mais qui permet toutefois de mesurer, de juger et d'apprécier la pertinence de l'approche globale pour la réalisation coopérative de tâches.

Deuxième partie

Une première version du système complet

Chapitre 3

Construction et mise à jour des états du monde

Dans le schéma de la figure 2.1 présentée en 2.1.1, la partie centrale du schéma est consacrée à la représentation des états du monde physique et des processus associés. Le robot doit construire et mettre à jour un état du monde précis à plusieurs niveaux. C'est évidemment une capacité cruciale pour un bon fonctionnement dans la durée de notre robot. Dans cette partie, nous allons décrire les raisonnements géométriques et temporels sur l'état du monde à disposition de notre robot, nous présenterons aussi de nouvelles idées qui n'ont pas encore été mises en oeuvre.

3.1 Production de faits géométriques symboliques

Il est indispensable de pouvoir produire une description symbolique de l'état géométrique courant sous la forme de faits symboliques. Cette description est utilisée comme résumé de l'état courant par certains processus de haut niveau, comme la planification de tâches et le contrôle d'exécution de plans. Il s'agit du processus "level 1 to level 2 update" dans le schéma de la figure 2.1 Ce travail avait déjà été largement entrepris au sein de l'équipe. La première contribution a consisté à poursuivre ce travail. La deuxième a permis de considérer des états géométriques distincts pour les différents agents. Cela permet de générer une représentation symbolique distincte en utilisant la mécanique de traduction de la géométrie existante.

L'état géométrique du monde est abstrait, sous la forme de faits symboliques qui peuvent être classifiés selon trois catégories :

– Positions relatives des objets et des agents, e.g. ⟨`GREY_TAPE isOn TABLE`⟩.

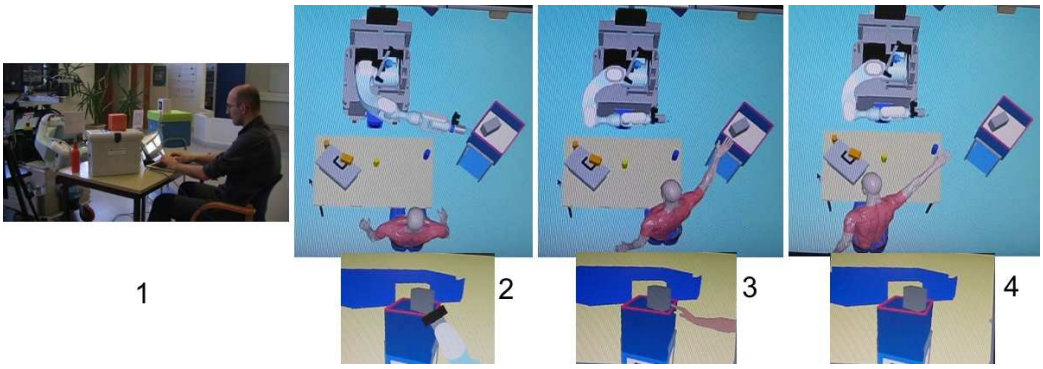


FIG. 3.1 – Un exemple illustrant la relation *reachable*. La relation est calculée de la perspective du robot comme de l'homme. La posture calculée à chaque étape est illustrée par une vue globale de la scène (haut), et une vue plus rapprochée (bas).

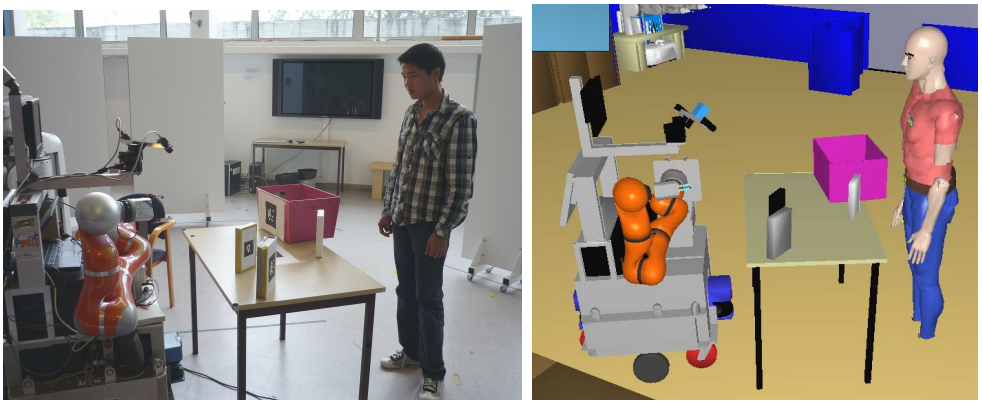
- Capacité de perception et d'action et état des agents, e.g. ⟨`ROBOT looksAt GREY_TAPE`⟩, ⟨`GREY_TAPE isVisibleBy HUMAN1`⟩.
- Mouvements des objets ou des mains ou têtes d'agents, e.g. ⟨`GREY_TAPE isMoving true`⟩,
  ⟨`ROBOT_HEAD isTurning true`⟩.

Le raisonnement avec la perspective de l'homme permet de calculer certains faits tels que : ⟨`GREY_TAPE isBehind HUMAN1`⟩, ⟨`GREY_TAPE isVisibleBy HUMAN1`⟩.

La figure 3.1 illustre différentes situations pour la relation *reachable*. Le robot et son partenaire humain sont placés l'un en face de l'autre, face à une table (Figure 3.1.1). Le robot estime d'abord si la petite boîte grise est atteignable pour lui. Il cherche pour cela une posture libre de collision permettant d'attraper l'objet. (Figure 3.1.2). Le robot utilise ensuite la perspective de l'homme pour déterminer si le même objet est également atteignable par l'homme. Dans la dernière scène, l'humain se déplace sur sa gauche, s'éloignant de l'objet (Figure 3.1.4). La situation est alors réévaluée. Le raisonneur géométrique ne trouve plus de posture de l'homme lui permettant d'atteindre l'objet, comme celui-ci est trop loin de l'objet.

La liste de faits calculés dans l'état décrit par la figure 3.1 est la suivante :

ROBOT	HUMAN1
PINK_TRASHBIN isReachable false	PINK_TRASHBIN isReachable true
WALLE_TAPE isReachable false	WALLE_TAPE isVisible true
LOTR_TAPE isReachable true	LOTR_TAPE isReachable false
GREY_TAPE isReachable true	GREY_TAPE isReachable false
WALLE_TAPE isVisible true	WALLE_TAPE isReachable true
LOTR_TAPE isVisible true	LOTR_TAPE isVisible true
GREY_TAPE isVisible true	GREY_TAPE isVisible true



(a) État initial (b) Vue du modèle 3d de l'état initial.

FIG. 3.2 – Dans cette situation, il y a 3 cassettes sur la table. Deux cassettes sont atteignables seulement par le robot : the LOTR_TAPE (noire dans le modèle 3d) et GREY_TAPE. La troisième cassette WALLE_TAPE (blanche dans le modèle 3d) et la poubelle PINK_TRASHBIN sont atteignables seulement par l'humain HUMAN1. Toutes les cassettes sont sur la table TABLE.

```
WALLE_TAPE isOn TABLE
LOTR_TAPE isOn TABLE
GREY_TAPE isOn TABLE
```

```
WALLE_TAPE isOn TABLE
LOTR_TAPE isOn TABLE
GREY_TAPE isOn TABLE
```

3.2 Instanciation géométrique de faits symboliques

Nous avons vu dans la partie précédente un exemple de mise à jour du niveau géométrique 1 vers le niveau géométrique 2. Ici nous examinons un exemple de mise à jour du niveau géométrique 2 vers le niveau 1.

3.2.1 Hypothèse de positionnement

Un état symbolique renseignant sur la géométrie est traduit/spatialisé dans l'espace géométrique cartésien. En effet, parfois, le robot connaît un état géométrique symbolique d'un objet qu'il a déduit de l'effet attendu de ses actions ou de celles de l'homme, ou obtenu par le dialogue alors qu'il ne peut le percevoir. C'est le cas par exemple si l'objet est dans un conteneur, dans une main de l'homme ou dans la pince du robot. Cette spatialisation est le plus souvent approchée. L'objet est placé au centre du conteneur ou de

la main de l'homme. Elle est a priori beaucoup plus précise dans le cas où l'on connaît la transformation qui lie l'objet à l'environnement. C'est typiquement le cas quand l'objet est dans la pince du robot à la suite d'un pick. Cette instanciation géométrique est utile pour représenter géométriquement la connaissance du robot dans l'espace cartésien. Elle est indispensable dans le cas où l'objet est dans la pince du robot pour la planification de mouvements futurs. Elle permet également d'estimer la plausibilité de l'état symbolique considéré, dans le cas où la perception fournit une position concurrente. Ceci permet un retour du niveau 1 vers le niveau 2. On peut définir pour chaque état symbolique des seuils de distance au-delà desquels l'état symbolique est considéré comme impossible du fait de son instanciation géométrique. Nous résumons cela au moyen de deux algorithmes 1 et 2. L'algorithme 1 gère la mise à jour de la position de l'objet. L'algorithme 2 décide de la suppression éventuelle d'une hypothèse de positionnement.

Les fonctions utilisées par les deux algorithmes 1 et 2 sont décrites dans le tableau 3.1.

object.hasPositionHypothesis() : fonction booléenne permettant de savoir si un *objet* a une hypothèse de positionnement.
object.updatePositionFromHypothesis() : fonction qui calcule la position d'un *objet* à partir de l'hypothèse de positionnement et la met à jour dans le modèle 3d.
object.isPerceivedNew() : fonction booléenne permettant de savoir si un *objet* a été perçu de nouveau depuis le dernier appel à elle-même.
object.updatePositionFromPerception() : fonction mettant à jour la position de l'*objet* dans le modèle 3d à partir de la dernière position perçue.
object.updatePerceptionPositionHypothesisConflictValue() : fonction calculant la distance entre la position perçue et celle calculée à partir de l'hypothèse de positionnement d'un *objet* et mettant à jour la variable stockant cette valeur.
object.getPerceptionPositionHypothesisConflictValue() : fonction récupèrant la dernière distance entre la position perçue et celle calculée à partir de l'hypothèse de positionnement d'un *objet*.
object.deletePositionHypothesis() : supprime l'hypothèse de positionnement d'un *objet*.
object.informAttentionAboutConflict() : fonction informant le processus attentionnel de la détection d'un conflit sur la position de l'*objet*.

TAB. 3.1 – Fonctions utilisées par les algorithmes 1 et 2.

Algorithm 1 chooseObjectPosition (*objects*)

```
for all objects do
  if object.hasPositionHypothesis() then
    if object.isPerceivedNew() then
      object.updatePositionFromPerception()
      object.updatePerceptionPositionHypothesisConflictValue()
    else
      object.updatePositionFromHypothesis()
    end if
  else
    if object.isPerceivedNew() then
      object.updatePositionFromPerception()
    end if
  end if
end for
```

3.2.2 Spatialisation basée agent

D'autres travaux de notre équipe de recherche réalisés par Pandey [41] relèvent aussi de l'instanciation géométrique de faits symboliques. Ils se concentrent sur la spatialisation dans le repère géométrique cartésien d'attributs symboliques relatifs à la perception et à la capacité d'action d'un agent. Il s'agit typiquement de la visibilité et de l'atteignabilité. À la différence des hypothèses de positionnement présentées ci-dessus qui produisent une position spatiale unique pour un objet, ces processus produisent des grilles 3D donnant la valeur d'un attribut donné pour chaque cellule de la grille. On peut anticiper la valeur prise par les attributs d'un objet qui serait positionné dans l'espace compte tenu des valeurs locales de ces attributs dans la grille. Un objet sera visible s'il est déplacé à un endroit de la grille de visibilité où les valeurs de visibilité sont positives. Un objet qui peut-être saisi par l'homme sera non atteignable s'il est positionné à un endroit de la grille d'atteignabilité où les valeurs d'atteignabilité sont très faibles. Cela a été conçu pour la planification de mouvements dans le but de récupérer des positions cibles vérifiant des attributs symboliques. Par exemple, où poser un objet tel qu'il soit visible et atteignable par l'homme ? C'est absolument indispensable dans le cas de l'interaction homme-robot, où l'on souhaite que l'homme puisse spécifier une tâche de manipulation du robot à partir d'attributs symboliques, et non en devant donner une position cartésienne exacte.

Algorithm 2 assessHypothesis (*object threholdSt thresholdDyn iterMax*)

Require: $object.hasPositionHypothesis()$
  $iter = 0$
  while $object.hasPositionHypothesis()$ **do**
    if $object.isPerceivedNew()$ **then**
      if $object.getPerceptionPositionHypothesisConflictValue() > threholdStatic$ **then**
        $object.deletePositionHypothesis()$
      else $\{object.getPerceptionPositionHypothesisConflictValue() > threholdDynamic\}$
        $object.informAttentionAboutConflict()$
        if $iter > iterMax$ **then**
          $object.deletePositionHypothesis()$
        else
          $iter = iter + 1$
        end if
      end if
    else
      $iter = 0$
    end if
  end while

3.3 Reconnaissance basique d'actions

Un suivi fin des actions de l'homme est crucial pour maintenir un état du monde cohérent. La compréhension complète des actions de l'homme est une tâche ardue, qui nécessite de réaliser à la fois des inférences descendantes depuis les buts et plans vers les actions, comme ascendantes à partir des mouvements effectués par l'homme et ses croyances sur l'état du monde comme explicité dans la figure 2.1 et 2.1.6. Tout cela étant possible uniquement dans la mesure où le robot possède de bons modèles de ces différents états et processus de l'homme (buts, plans, actions, mouvements ....). Dans la mesure où, dans nos scénarios, l'homme possède un répertoire d'actions assez limité, des raisonnements temporels et géométriques simples sur les trajectoires des mains de l'homme, compte tenu de la position des objets, permettent de reconnaître certaines actions de manière satisfaisante. Nous l'appelons reconnaissance basique d'action. Il s'agit d'une inférence simple de type ascendante depuis les états du monde et les mouvements vers les actions.

Par exemple, les actions *prendre*, *jeter* ou *poser* peuvent être reconnues respectivement, par la proximité entre une main vide et un objet sur la table, la position d'une main tenant un objet au-dessus d'un conteneur, ou la proximité entre une main tenant un objet et la table.

Les algorithmes 3 et 4 décrivent la création et la mise à jour des moniteurs actifs. L'algorithme 3 décrit la mise à jour de tous les moniteurs actuellement utilisés dans nos expérimentations. L'algorithme 4 est une généralisation de l'algorithme précédent à tout type de moniteur. L'algorithme 5 explique comment un moniteur qui déclenche est exploité.

Les fonctions utilisées par les trois algorithmes 3, 4 et 5 sont décrites dans le tableau 3.2.

Fonctions pour les algorithmes 3 et 4 :
monitors.findMonitor(entity,actionType) fonction booléenne permettant de savoir si le moniteur de l'action de type *actionType* sur l'entité *entity* fait partie des moniteurs courants dans la liste *monitors*.
monitors.deleteMonitor(entity,actionType) fonction qui supprime le moniteur de l'action de type *actionType* sur l'entité *entity* de la liste des moniteurs courants *monitors*.
monitors.createMonitor(entity,actionType) fonction qui ajoute le moniteur de l'action de type *actionType* sur l'entité *entity* à la liste des moniteurs courants *monitors*.
object.isOnTable(table) fonction booléenne permettant de savoir si l'objet *objet* est sur la table *table*.
object.respectCondition(condition) fonction booléenne générique permettant de savoir si l'objet *objet* respecte la condition *condition*.
Fonctions de l'algorithme 5 :
monitor.triggeredNew() : fonction booléenne permettant de savoir si un *monitor* a déclenché depuis le dernier appel.
monitor.action() : fonction renvoyant l'action instanciée (type et arguments) par un *monitor* qui a déclenché.
action.forceEffects() fonction qui vérifie si les précondition de l'action *action* sont vérifiées.
action.arePreconditionsVerified() fonction qui force les effets de l'*action* par la création ou la suppression des hypothèses de positionnement décrites dans la partie 10.2
action.informAttention() : fonction qui a pour effet d'informer le processus attentionel de la détection de l'*action*.

Tab. 3.2 – Fonctions utilisées par les algorithmes 3, 4 et 5.

Les arguments de l'algorithme 3 représentent respectivement, la liste des objets manipulables par les humains (*manipulableObjects*), la liste des conteneurs dans lesquels les humains peuvent déposer des objets (*containers*), la table (*table*), la liste des zones d'intérêt de la table (*tableAreas*) et enfin la liste des moniteurs actifs qui est mise à jour par l'algorithme (*monitors*).

Algorithm 3 updateBasicActionRecognitionMonitors (*manipulableObjects containers table tableAreas monitors*)

```
for all manipulableObjects do
  if monitors.findMonitor(manipulableObject,pick) then
    if Not(manipulableObject.isOnTable(table)) then
      monitors.deleteMonitor(manipulableObject,pick)
    end if
  else
    if manipulableObject.isOnTable(table) then
      monitors.createteMonitor(manipulableObject,pick)
    end if
  end if
end for
for all containers do
  if Not(monitors.findMonitor(container,throw)) then
    monitors.createteMonitor(container,throw)
  end if
end for
for all tableAreas do
  if Not(monitors.findMonitor(tableArea,place)) then
    monitors.createteMonitor(tableArea,place)
  end if
end for
```

Les arguments de l'algorithme 4 représentent respectivement, la liste des types d'action pour lesquels on souhaite monitorer (*actionsTypes*), la liste des listes d'entités pouvant être cibles d'un type d'action donné (*actionsTypesEntitiesSets*), la liste des conditions de création du moniteur pour chaque type d'action (*actionsTypesEntitiesConditions*) et enfin la liste des moniteurs actifs qui est mise à jour par l'algorithme (*monitors*).

Algorithm 4 updateBasicActionRecognitionMonitors (*actionsTypes actionsTypesEntitiesSets actionsTypesConditions monitors*)

```
for all actionsTypes do
  for all actionsTypesEntitiesSet[actionsType.count] do
    if monitors.findMonitor(actionsTypesEntity,actionsType) then
      conditionCount = actionsTypesConditions[actionsType.count]
      if Not(actionsTypesEntity.respectCondition(conditionCount))
      then
        monitors.deleteMonitor(actionsTypesEntity,actionsType)
      end if
    else
      conditionCount = actionsTypesConditions[actionsType.count]
      if actionsTypesEntity.respectCondition(conditionCount) then
        monitors.createteMonitor(manipulableObject,table,pick)
      end if
    end if
  end for
end for
```

Algorithm 5 interpretBasicActionRecognitionMonitors (*monitors*)

```
for all monitors do
  if monitor.triggeredNew() then
    action = monitor.action()
    if action.arePreconditionsVerified() then
      action.forceEffects()
      action.informAttention()
    end if
  end if
end for
```

3.4 Pilotage de l'attention visuelle

Par l'attention visuelle, on désigne la direction des capteurs caméras. Cette direction contrôle bien évidemment ce que perçoit le robot à un instant donné. Elle a aussi un effet sur la croyance de l'homme sur l'attention du robot, compte tenu du modèle qu'il a de la perception du robot. À ce titre, l'attention visuelle peut jouer un rôle de communication.

Le pilotage de l'attention visuelle est donc nécessaire pour arbitrer la compétition entre plusieurs demandes concurrentes. On peut définir des politiques complexes qui vont arbitrer entre optimisation du gain d'information de la perception, ou acte de communication à partir d'optimisation de l'utilité d'une direction donnée.

Dans cette partie nous allons décrire la politique simple et robuste que nous utilisons actuellement. Elle est fondée sur une pile de priorité décroissante entre les différents processus susceptibles de mobiliser l'attention visuelle qui est présentée ci-dessous :

- Appui de la verbalisation.
- Focus sur l'action en cours.
- Focus sur les "monitors" qui déclenchent.
- Focus sur les conflits entre la perception et les hypothèses de positions.
- Contrôle de l'existant basé objet.
- Exploration exhaustive basée zones.

Appui de la verbalisation. Généralement, le locuteur regarde son interlocuteur quand il lui parle. Quand le robot parle, il dirige son attention visuelle avec une priorité maximale en direction du visage de son interlocuteur.

Focus sur l'action en cours. Nous faisons l'hypothèse que les agents sont coopératifs et impliqués dans la réalisation du but. Le robot et l'homme cherchent à réaliser l'action planifiée en cours si elle leur est allouée. Pour chaque action on peut calculer une direction du regard appropriée. Cette direction a pour double objectif de communiquer le focus du robot sur l'action en cours et de focaliser la perception là où le changement doit avoir lieu.

Focus sur les "moniteurs". Le déclenchement d'un moniteur est généralement le signe d'une action de l'homme. Comme décrit dans l'algorithme 5 d'interprétation des moniteurs, le déclenchement d'un moniteur est communiqué à l'attention visuelle. S'il n'y a pas de dialogue ou d' action en cours, l'attention visuelle est alors recrutée en priorité de manière spécifique et appropriée selon le type de moniteur. La perception résultante permet alors de

constater le résultat escompté (exemple d'un *place* réussi) ou au contraire d'infirmer le résultat attendu et anticipé d'une action (exemple d'une action toucher reconnue à tort comme un *pick*)

Focus sur les conflits entre perception et les hypothèses de position. La reconnaissance d'un conflit entre la perception et une hypothèse de position, qui traduit une incohérence entre les croyances du robot et l'état réel du monde, mobilise l'attention visuelle si l'écart entre la position perçue et celle calculée par l'hypothèse est suffisamment important pour que l'hypothèse soit considérée comme suspecte, mais pas assez grand pour qu'elle soit directement écartée.

Contrôle de l'existant basé objet. Si aucun processus de priorité supérieure ne tente de mobiliser l'attention visuelle, la stabilité de l'état existant est contrôlée par la perception de toutes les positions existantes.

Exploration exhaustive basée zones. Si aucun processus de priorité supérieure ne tente de mobiliser l'attention visuelle et que le contrôle des positions existantes vient d'être réalisé, le robot procède à une exploration exhaustive de l'espace de travail à la recherche de nouveaux objets ou de la nouvelle position d'objets ayant disparu.

3.5 Premier bilan

La figure 3.3 permet de mettre en cohérence tous les processus explicités jusqu'ici.

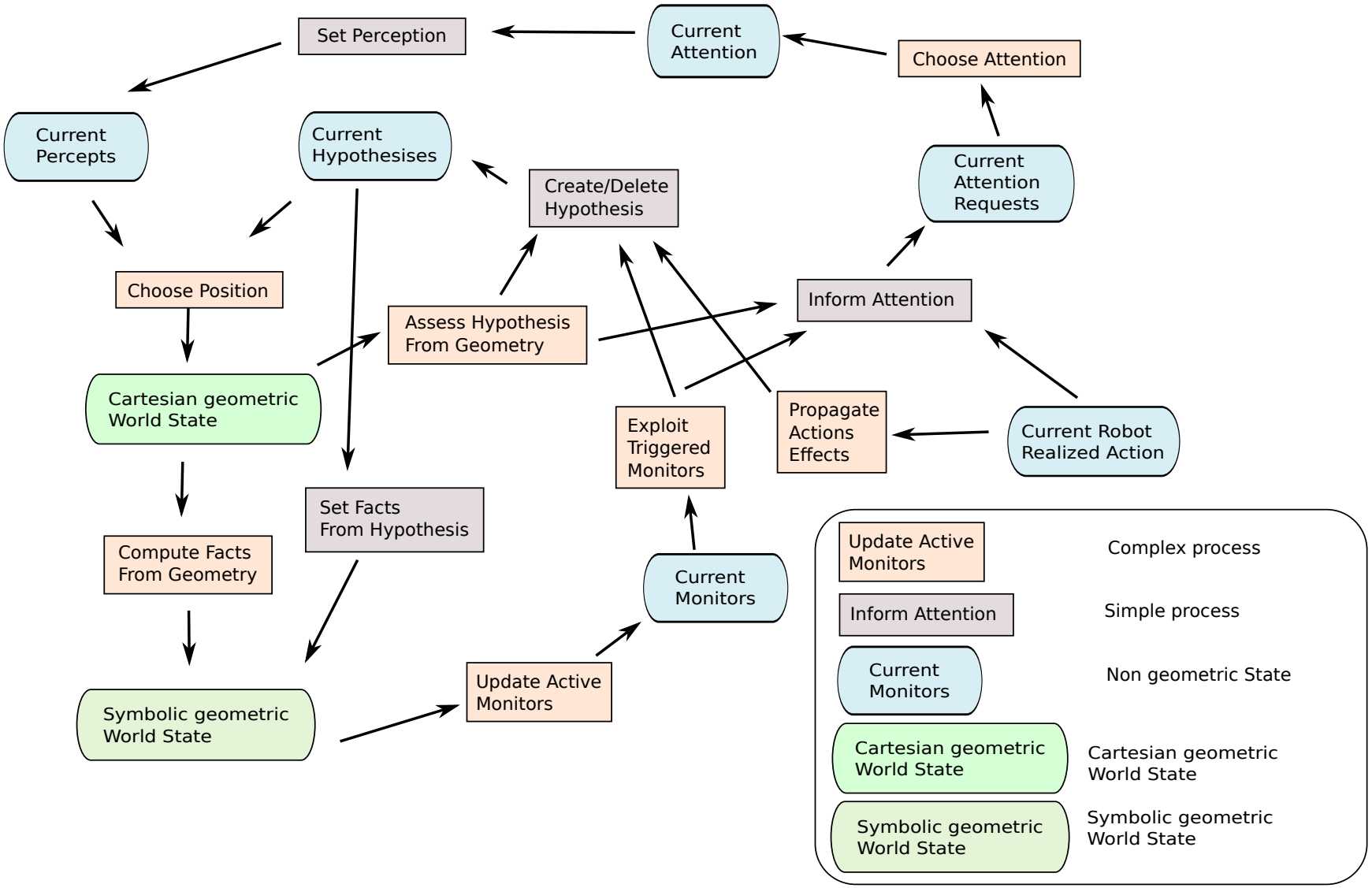


FIG. 3.3 – *
Hypothèses, moniteurs, action, attention

Comme cela sera démontré dans le chapitre 6, cette politique de choix de l'attention visuelle, associée aux mécanismes de monitoring des actions, et aux hypothèses de positionnement résultant de la réalisation des actions du robot ou de l'interprétation des moniteurs sur les actions de l'homme, permet au robot une construction et une mise à jour relativement fiable de l'état du monde. Le système présente néanmoins des limites. Nous en présentons un certain nombre ci-dessous. Le chapitre 10 propose des perspectives d'amélioration.

Caractère réactif. Le système réagit de manière réactive et automatique à chaque événement incident. Les événements ne sont pas actuellement sauvegardés pour être réexaminés a posteriori dans le but de produire un meilleur état. Par exemple, si un moniteur de prise d'objet est déclenché juste au moment où le robot verbalise une information à l'homme, l'attention est mobilisée par la verbalisation et n'est donc pas mobilisable par le moniteur. L'objet sera donc considéré comme pris même si ce n'est pas le cas. Si l'attention avait été mobilisée par le moniteur, le robot aurait directement perçu que l'objet n'avait en fait pas bougé. L'état de l'objet sera donc erroné tant que le robot n'aura pas perçu l'objet de nouveau. En pratique la zone de travail étant limitée et le robot disposant de politique d'exploration continuelle, il ne faut pas très longtemps pour que l'objet soit perçu à nouveau.

Absence d'estimation de la qualité de la connaissance de l'état. La qualité de la connaissance de l'état n'est pas accessible. Le robot ne peut donc moduler l'exploration spatialement ou temporellement en conséquence, pour aller observer là ou sa connaissance est moins bonne et pendant plus de temps.

Absence de représentation explicite des états atteignables Il n'y a pas de représentation explicite et exhaustive des états atteignables. Cela permettrait de chercher des états candidats dans le cas où l'on ne sait pas dans quel état est l'objet. Par exemple, les raisonnements actuels permettent au robot de déterminer l'état d'un objet compte tenu de la position perçue ou du résultat d'une action effectuée par le robot ou par l'homme. Par contre le robot ne dispose pas d'un mécanisme lui permettant de faire l'hypothèse que, si l'objet n'est pas visible sur la table, il est soit dans les mains de l'homme, soit dans la corbeille.

La considération des différents états atteignables et l'introduction d'une représentation probabiliste des états, qui sont présentées dans le chapitre 10, permettent en partie d'aller au delà de ces limites.

Chapitre 4

Gestion des buts et des plans

4.1 Contexte

Dans ce chapitre nous nous intéressons aux buts et aux plans. Dans la figure 2.1, on peut identifier les processus associés : "Goal Creation", "Goal Management", "Task Planning", "Plan Management", "Shared Goal Management" et "Shared Plan Management". Pour rappel, nous désignons par but un état du monde qu'un ou plusieurs agents veulent atteindre. Un plan est un ensemble de séquences d'actions dont la réalisation doit permettre d'atteindre l'état du monde relatif au but.

Le but pourra donc être décrit par un ensemble de prédicats sur l'état du monde qui doivent être vérifiés pour que le but puisse être considéré comme atteint. Dans le cadre de notre travail on considère un plan partagé entre un homme et un robot. Le plan est constitué de deux séquences. Une séquence pour l'homme. Une séquence pour le robot. On appelle lien "causal" un lien directionnel entre une action d'une séquence et une autre action de l'autre séquence, tel que l'action cible ne peut démarrer tant que l'action source n'a pas été réalisée avec succès. Le plan comme le but sont également caractérisés par un statut qui décrit l'état intermédiaire (ongoing / suspended) s'ils sont toujours actifs ou l'état final s'ils ne le sont plus (achieved / impossible / stopped). Qu'il s'agisse du but ou du plan, s'ils sont partagés, il y a une complexité supplémentaire en termes d'estimation, de communication et de négociation.

Nous pouvons décrire rapidement les processus mis en oeuvres :

Goal Creation. Qu'est ce qui entraîne la création d'un nouveau but ? On identifie les trois contextes suivants par ordre de complexité croissante en termes de capacité de reconnaissance et de décision du robot.

1. Requête directe. L'homme demande au robot de lui apporter un objet.
2. Respect d'un planning existant. Le robot devra accomplir ce but avant telle heure.
3. Identification d'un besoin, respect d'une consigne. S'il pleut il faut fermer la fenêtre. Si l'homme a soif il faut lui apporter de l'eau.

Task Planning. La planification de tâches permet d'élaborer un plan permettant d'atteindre un but à partir d'un état du monde donné. De grands efforts ont été consacrés par la communauté pour se doter de planificateurs de plus en plus performants.

(Shared) Goal Management. Par la gestion du but (partagé), on désigne tout les processus mis en oeuvre liés à un but existant. Entre autres, cela consiste en l'estimation et la communication de l'état du but, l'estimation et la communication de l'engagement des agents à atteindre ce but, l'établissement et la négociation d'un plan permettant d'atteindre ce but.

(Shared) Plan Management. Par la gestion du plan (partagé), on désigne tous les processus mis en oeuvre liés à un plan existant. Entre autres cela consiste en l'estimation et la communication de l'état du plan, l'estimation de l'engagement des agents dans la réalisation du plan, la gestion des politiques de réalisation du plan en terme de réalisation et monitoring d'actions.

Comme décrit dans la partie 2.2.2, nous avons assez largement simplifié certains de ces processus dans le système actuel. Cela est dû notamment à la difficulté du dialogue et de l'estimation d'états mentaux complexes de l'homme, et du fait que nous avons souhaité réaliser un système complet en mettant l'accent sur la réalisation effective des actions et la construction et le maintien de l'état du monde. Cela s'est fait au détriment d'une gestion plus complexe de l'interaction et des états de haut niveau.

Seules les requêtes de création de buts de l'homme sont à l'origine de nouveaux buts. La faisabilité du but et du plan est uniquement considérée sous l'angle de l'état physique du monde et de la capacité physique des agents. L'éventualité qu'un des agents refuse ou renonce à participer n'est pas prise en compte. Seul le robot est impliqué dans la planification qui est communiquée de manière minimum et progressive à l'homme. Nous disposons d'un planificateur qui prend en compte explicitement l'homme et ses croyances propres qui peuvent être distinctes de celle du robot.

Le planificateur est rapidement décrit dans la section 4.2. La gestion du plan est décrite dans la section 4.3.

4.2 Le planificateur HATP

HATP (Human-Aware Task Planner) est un planificateur hiérarchique. Le but de la planification hiérarchique (HTN) est de décomposer une tâche de haut niveau représentant le but en sous-tâches, jusqu'à atteindre un ensemble de tâches atomiques qui sont réalisables par les agents [40]. HATP est capable de produire des plans pour le robot aussi bien que pour les autres participants (humains ou robots). Le comportement du planificateur peut être modifié au travers du réglage des différents coûts des actions et en prenant en compte un ensemble de règles sociales. Ces coûts et règles permettent d'adapter le comportement du robot en fonction du niveau de coopération désiré.

4.2.1 Agents et séquences d'actions

Le robot ne planifie pas uniquement pour lui-même mais également pour les autres agents. Le plan produit, appelé *plan partagé*, est un ensemble d'actions formant une séquence pour chaque agent impliqué dans la réalisation du but commun. En fonction du contexte, certains plans partagés contiennent des liens causaux entre les actions des différents agents. Par exemple, le second agent peut devoir attendre la fin de la réalisation d'une action du premier agent avant de pouvoir commencer sa propre action. Lorsque le plan est exécuté, les liens causaux induisent des synchronisations entre agents.

La figure 4.1 illustre un plan avec deux séquences d'actions. Le but consiste à avoir une table nettoyée. Il y a quatre objets de type cassette : grey_tape, walle_tape, lotr_tape et black_tape. Il y a deux boîtes de rangement Box1 et Box2. La table est nettoyée quand les cassettes se retrouvent dans la boîte de rangement qui leur correspond. grey_tape et walle_tape dans la boîte Box1. lotr_tape et black_tape dans la boîte Box2. La tâche consiste à nettoyer la table. Initialement, grey_tape est atteignable seulement par l'homme, walle_tape, lotr_tape et black_tape sont atteignables seulement par le robot. La boîte Box1 est atteignable seulement par l'homme. La boîte Box2 est atteignable seulement par le robot. Il y a trois types d'action différents. L'action TAKE consiste à prendre un objet. L'action THROW consiste à mettre un objet dans une boîte. L'action PUTRV, consiste à poser un objet sur la table de telle sorte qu'il soit visible et atteignable par l'autre agent. La séquence du haut correspond aux actions de l'homme. La séquence du bas correspond aux actions du robot. Il y a un lien causal entre les deux séquences. Il est entre l'action PUTRV du robot sur walle_tape et l'action TAKE de l'homme sur walle_tape. L'homme ne peut attraper la cassette walle_tape tant que le robot ne l'a pas déplacée, comme walle_tape n'est pas atteignable par l'homme initialement.

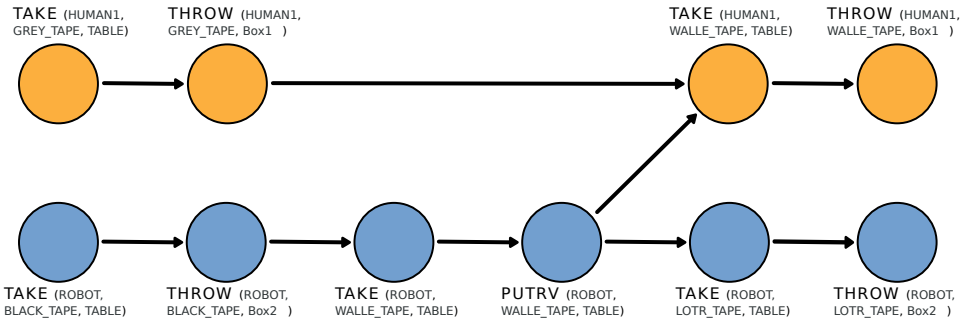


FIG. 4.1 – Un plan à 2 séquences d'action produit par HATP

4.2.2 Coûts des actions et règles sociales

À chaque action est associée une fonction de coût et une fonction de durée. La fonction de durée produit un intervalle temporel pour l'exécution de l'action et est utilisée, d'une part, comme une ligne de temps pour ordonner les séquences d'actions des différents agents et, d'autre part, comme fonction de coût additionnelle. En plus de ces coûts, HATP prend en entrée un ensemble de règles sociales. Ces règles sont des contraintes qui ont pour but de guider la construction du plan vers le meilleur plan selon les préférences de l'homme. Les principales règles sociales qui ont été définies sont :

- État indésirable. Pour éviter un état dans lequel l'humain peut se sentir en position inconfortable.
- Séquence indésirable. Pour éliminer les séquences d'actions qui peuvent être mal interprétées ou rejetées par l'homme.
- Équilibrage des efforts. Pour répartir et ajuster l'effort de travail entre les agents.
- Temps morts. Pour éviter les temps morts entre les actions de l'homme ou d'un autre agent.
- Liens imbriqués. Pour limiter les dépendances entre les actions des différents agents.

La figure 4.2 illustre un plan alternatif au plan précédent (figure 4.1) lorsque la règle sociale concernant les temps morts est utilisée. Le plan produit est le meilleur plan en accord avec une évaluation globale des différents critères de coût. On remarque que, pour appliquer cette règle, le planificateur a dû modifier l'ordre des actions du robot.

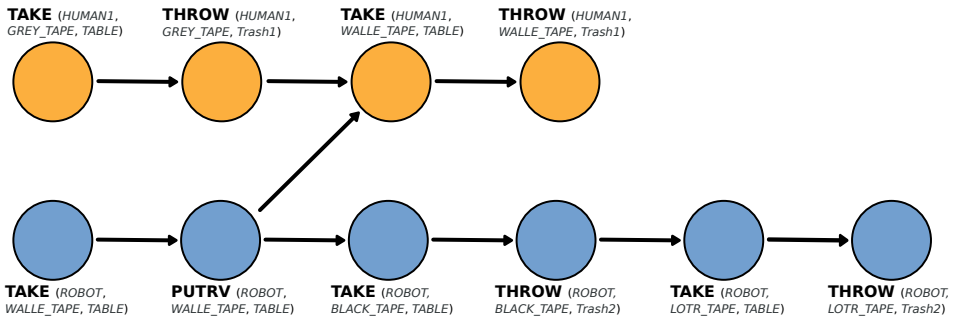


FIG. 4.2 – Le même plan avec la règle sociale des temps morts

4.2.3 Différents niveaux de coopération

En réglant les fonctions de coût et en appliquant les règles sociales, HATP peut être utilisé pour calculer de nombreux plans alternatifs. Ces plans peuvent être catégorisés en différents niveaux de coopération :

- Accomplir seul le but spécifié par l'humain.
- Partager des ressources concrètes en donnant des objets à l'humain.
- Coopérer avec l'humain en coordonnant ses actions avec les actions du robot dans le but d'atteindre un but commun.

4.2.4 Modélisation du domaine

HATP utilise son propre langage orienté objet pour la modélisation du domaine. Ce langage a au moins le même pouvoir d'expressivité et les mêmes fonctionnalités que SHOP2 [40].

Représentation orientée objet

Le monde est représenté par un ensemble d'entités. Chaque entité est unique et est définie par un ensemble d'attributs. Les attributs sont soit statiques (*static*), soit dynamiques (*dynamic*) et ont le type *atom* ou *vector*. Un attribut statique représente une information non modifiable tandis qu'un attribut dynamique peut être mis à jour. Un attribut atomique (*atom*) ne peut contenir qu'une seule valeur alors qu'un attribut de type *vector* est utilisé pour stocker des listes de valeurs.

Représentation des agents

Dans HATP, les agents sont considérés comme étant des objets. Cependant, comme la sortie du planificateur est un plan sous la forme de séquences d'actions par agent, le type d'entité *Agent* est prédéfini et au moins une entité *Agent* doit être initialisée.

Définition du domaine

Le domaine de planification, appelé base de faits (*fact database*) dans HATP, est défini en quatre étapes telles qu'illustrées par la figure 4.3. Premièrement, les différents types d'entités sont définis (excepté pour le type *Agent* qui est implicite). Puis, les attributs de chaque entité sont définis. Dans la troisième étape, les objets et agents présents dans l'environnement sont créés. Finalement, des valeurs initiales sont attribuées aux attributs de chaque entité créée.

4.3 Gestion des buts et des plans

La figure 4.4 résume les mécanismes de gestion des buts et des plans tels qu'ils existent actuellement dans le système. Quand on récupère un événement correspondant à une nouvelle requête de but de la part de l'homme, la validité de ce but est d'abord testée. Est-ce que ce but est déjà atteint ? Existe-t-il un plan ? Le but est considéré comme atteignable tant qu'il n'est pas atteint et qu'HATP produit un plan à partir de l'état courant.

L'exécution du plan consiste en la réalisation de chaque action du plan. En l'état actuel du système, on dispose de deux stratégies. La première stratégie consiste en une exécution séquentielle de toutes les actions. Dans la première stratégie, l'homme et le robot n'agissent pas en même temps. Le robot réalise ses actions ou suit celles attribuées à l'homme. Cette stratégie est assez rigide et lente. Néanmoins elle permet au robot de communiquer le plan dans son intégralité et d'observer précisément l'action de l'homme. C'est particulièrement adapté si l'homme n'est pas un expert et a besoin d'être surveillé. Dans la seconde stratégie, le robot ne s'occupe que de la réalisation de ses actions. Cette stratégie peut s'avérer plus rapide comme l'homme et le robot travaillent en parallèle. Elle est fondée sur l'hypothèse que l'homme est en mesure de planifier lui même sa part du travail et qu'il n'a pas besoin que le robot lui dicte la marche à suivre. En cas d'échec du plan, on replanifie.

Le pilotage des actions est réalisé en trois étapes. On vérifie d'abord que leurs préconditions sont vérifiées. Puis l'action est exécutée ou suivie (elle

```
factdatabase {
        //step 1: Definition of entity types
        define entityType Container;
        define entityType GameArtifact;
        //step 2: Definition of attributes
        define entityAttributes Agent {
                static atom string type;
                dynamic atom GameArtifact hasInRightHand;
        }
        define entityAttributes Container {
                dynamic set Agent isReachableBy;
        }
        define entityAttributes GameArtifact {
                dynamic set Agent isReachableBy;
                dynamic atom Container location;
        }
        //step 3: Creation of entities
        JIDO = new Agent;
        PINK_TRASHBIN = new Container;
        WHITE_TAPE = new GameArtifact;
        //step 4: Attributes initialization
        JIDO.type = "robot";
        PINK_TRASHBIN.isReachableBy <<= JIDO;
        WHITE_TAPE.isReachableBy <<= JIDO;
        WHITE_TAPE.location = PINK_TRASHBIN;
}
```

FIG. 4.3 – Exemple de définition d'un domaine de planification

est uniquement suivie s'il s'agit d'une action de l'homme). Enfin, les effets attendus sont testés pour valider le résultat de l'action.

Le robot informe l'homme sur les points suivants au cours de l'exécution du plan.

- Existence et statut du but.
- Existence et statut du plan.
- Action en cours.
- Échec de l'action
- Faits non vérifiés lors d'un test des préconditions ou des effets.

La figure 4.4 illustre les processus décrits ci dessus.

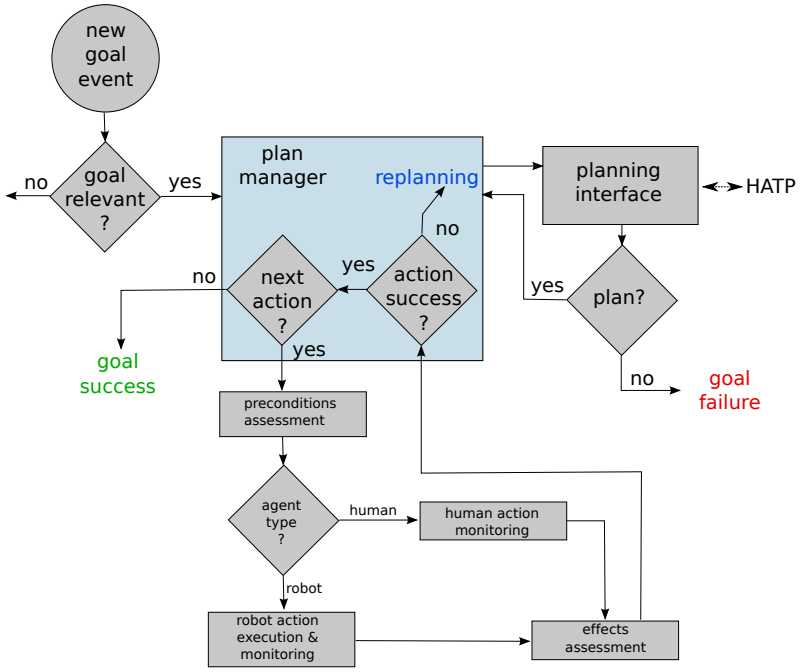


FIG. 4.4 – Automate de gestion des buts et des plans

Chapitre 5

Exécution et monitoring des actions du plan partagé

5.1 Contexte

Dans ce chapitre nous nous intéressons à la réalisation effective des actions dans le contexte de la réalisation coopérative du plan partagé. Jusque-là, les actions étaient définies de manière abstraite, par la définition de leurs préconditions et de leurs effets, au moyen d'un ensemble de prédicats sur l'état du monde. Les actions sont le point de passage entre les états mentaux et le monde physique. Les actions se traduisent le plus souvent par une séquence de mouvements dans le monde physique, même si on envisage également des actions de communication. Le robot doit réaliser ses propres actions et suivre les actions de l'homme pour évaluer la progression dans le plan et assurer la coordination avec l'homme. Pour les actions du robot, les mouvements associés à l'action doivent être planifiés et exécutés. Pour les actions de l'homme les mouvements doivent être reconnus et suivis par le robot. Les effets des actions sur le monde physique doivent être suivis ou inférés si leur perception n'est pas garantie.

Dans la figure 2.1, on peut identifier les processus associés : "Action Management", "Motion Planning", "Motion Execution". Nous nous intéressons d'abord à la réalisation des actions par le robot. Nous abordons ensuite le "Monitoring" des actions de l'homme.

5.2 Exécution des actions du robot

5.2.1 Typologie des mouvements

Dans nos scénarios, le robot réalise essentiellement des actions de prise, déplacement, pose ou lâchage d'objet. Dans ce contexte nous avons défini une typologie des différents mouvements élémentaires nécessaires.

La figure 5.1 présente la typologie des états à partir du triplet objet / support / pince.

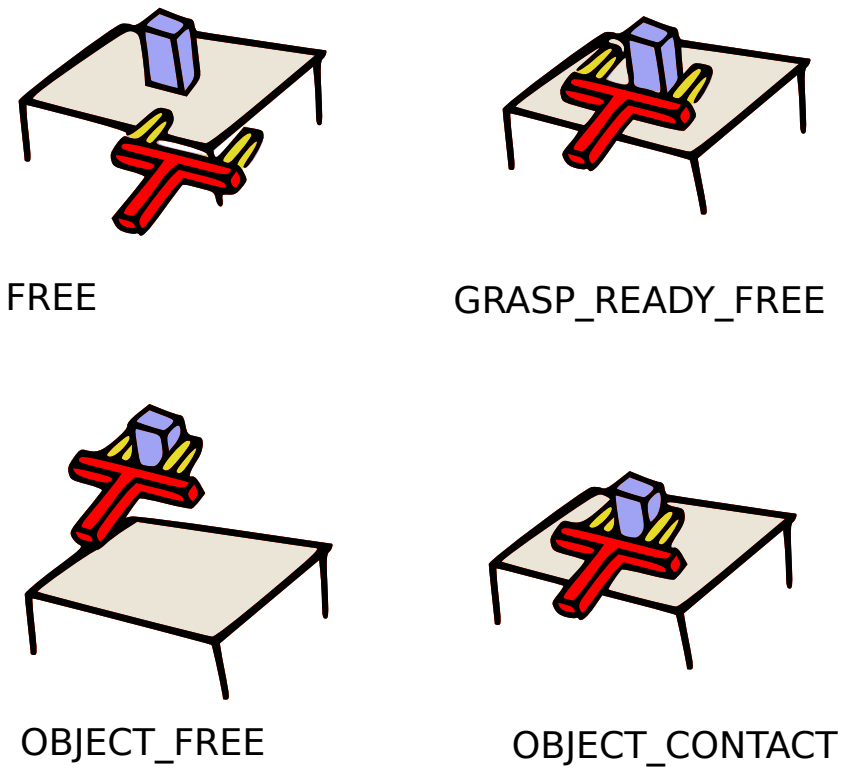


Fig. 5.1 – *
Typologie des états

- FREE : pince vide loin de tout objet.
- OBJECT_FREE : objet dans la pince loin de tout support.
- GRASP_READY_FREE : pince ouvert prêt à saisir un objet.
- OBJECT_CONTACT : objet dans la pince au contact d'un support.

Le tableau 5.1 et la figure 5.2 présentent la typologie des mouvements planifiables compte tenu du triplet objet / support / pince. Ceci correspond

MHP_ARM_FREE : mouvement libre. FREE -> FREE OBJECT_FREE -> OBJECT_FREE
MHP_ARM_PICK_GOTO : approche et prise. FREE -> OBJECT_CONTACT
MHP_ARM_TAKE_TO_FREE : éloignement d'un support. OBJECT_CONTACT -> OBJECT_FREE
MHP_ARM_TAKE_TO_PLACE : éloignement d'un support puis approche d'un autre support. OBJECT_CONTACT -> OBJECT_CONTACT
MHP_ARM_PLACE_FROM_FREE : approche d'un support OBJECT_FREE -> OBJECT_CONTACT
MHP_ARM_ESCAPE_OBJECT : éloignement d'un objet. OBJECT_CONTACT (pince ouverte) -> FREE

Tab. 5.1 – Typologie des mouvements. Pour chaque type de mouvement on décrit rapidement ce qui le caractérise, puis on précise les transitions entre états permises par ce mouvement.

à des instances de problème de planification géométrique différentes. Ils se traduisent par une finesse de planification variable.

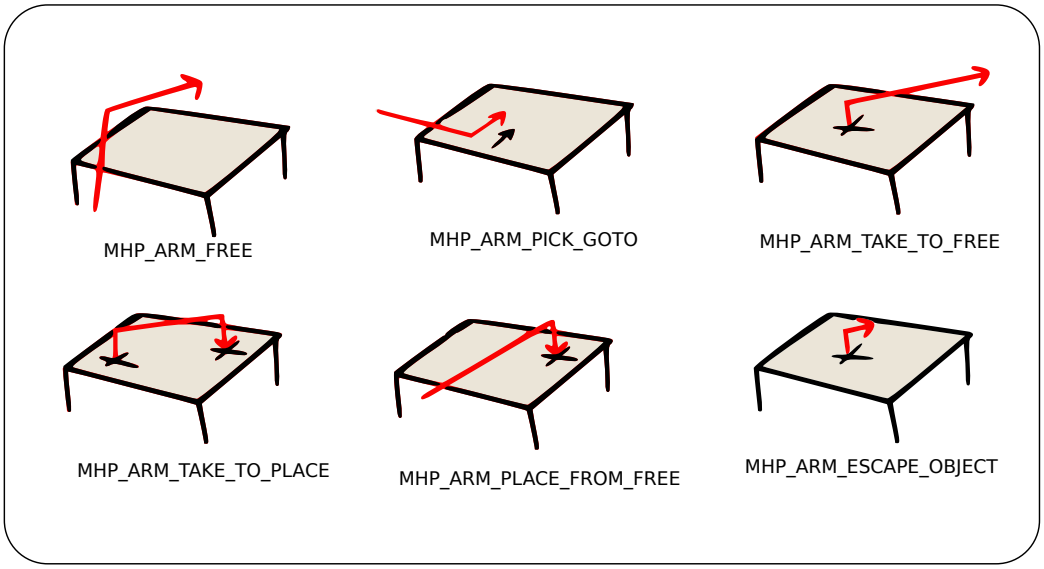


Fig. 5.2 – *
Typologie des mouvements

La figure 5.3 illustre l'automate complet qui propose toutes les transitions possibles à partir de la typologie des trajectoires et des états. L'état courant peut être déterminé à partir du calcul si l'objet est perçu, ou par la propagation des effets des mouvements et saisies si l'objet ne peut être perçu. Pour tout état cible que l'on souhaite atteindre, on en déduit donc facilement un chemin en termes de séquence de trajectoires d'un type donné.

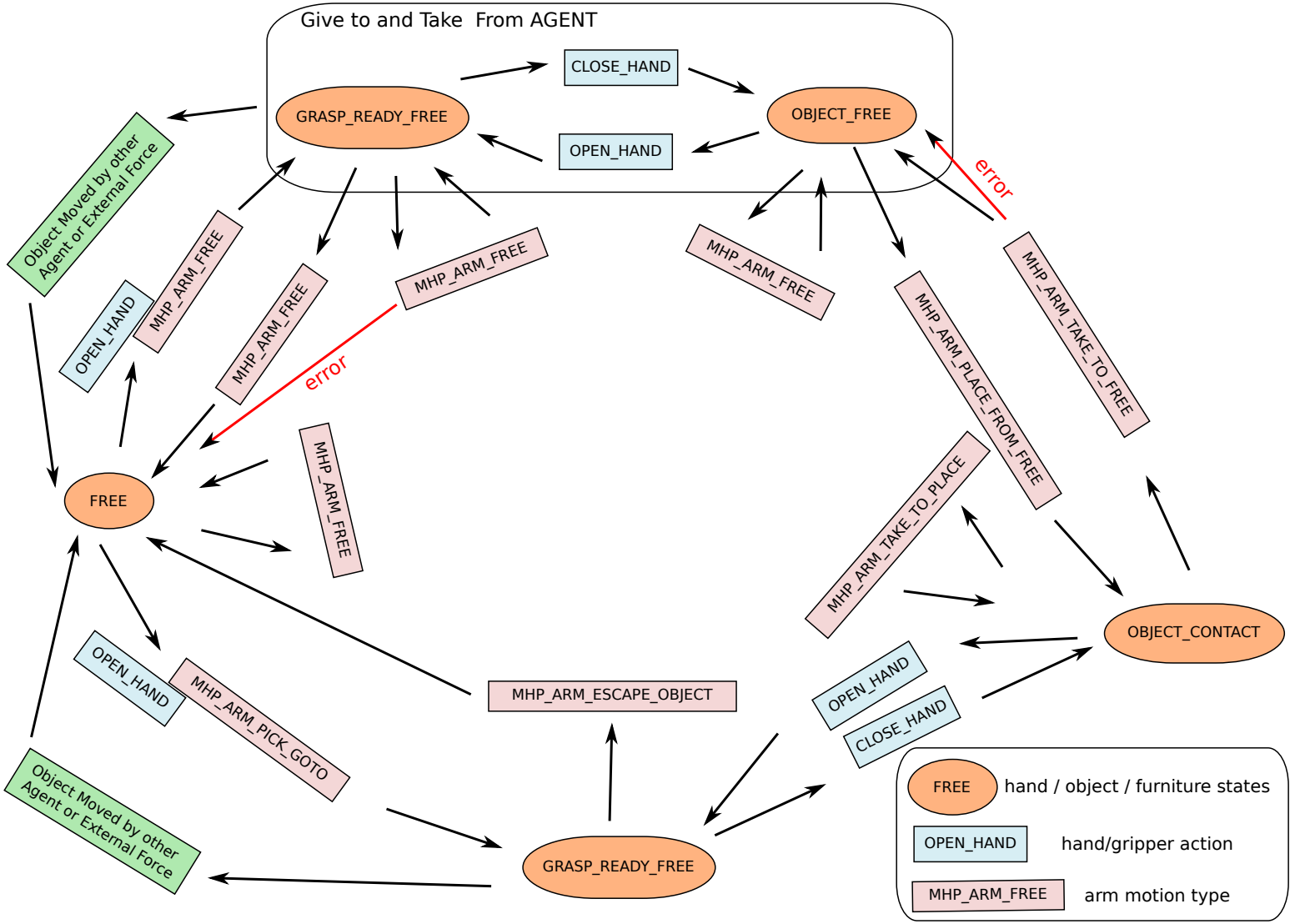


FIG. 5.3 – *
Automate complet

La planification géométrique de mouvements et de tâches permet de calculer la position finale de l'objet, la saisie et la trajectoire, en prenant en compte des contraintes spécifiques à la tâche et des contraintes liées à l'homme telles que sa posture, ses compétences et ses préférences. On pourra consulter les travaux de Sisbot, Mainprice et Pandey [55, 36, 41] pour les détails.

La position finale de l'objet ou de la pince dépend bien sûr de l'action. Nous présentons un certain nombre d'exemples d'actions et leur position correspondante.

- Jeter dans la poubelle : position à quelques centimètres au-dessus de la poubelle.
- Retour du bras à une position de repos ou de retrait : position de repos confortable pour les articulations et d'apparence naturelle.
- Donner à l'homme : position accessible, confortable et compréhensible par l'homme.
- Cacher à l'homme : position invisible pour l'homme.
- ....

Les "Mightabilities Maps" proposées par Pandey dans [41] permettent de calculer des positions finales en prenant en compte les capacités de l'homme pour par exemple donner à l'homme ou cacher à l'homme.

La figure 5.4 décrit les automates d'exécution des actions symboliques. On considère les quatre actions symboliques suivantes : TakeObject, ThrowObject, GiveObject, PutObjectReachable. TakeObject consiste à prendre un objet et revenir à une position de repos. ThrowObject consiste à lâcher un objet qui était déjà dans la pince dans un conteneur et revenir à une position de repos. GiveObject consiste à donner un objet qui était déjà dans la pince à un autre agent. PutObjectReachable consiste à poser un objet qui était déjà dans la main à un endroit tel qu'il soit atteignable par l'homme. Ces automates précisent quelles sont les positions à calculer, les mouvements à planifier, les ouvertures ou fermetures de pince à réaliser et les hypothèses de positionnement à ajouter ou supprimer pour réaliser l'action symbolique. ACTION_POSITION correspond à une position adaptée à l'action comme précisée plus haut. REST_POSITION correspond à une position de repos.

L'algorithme 6 permet d'exécuter une action à partir de ces automates. Prenons l'exemple du robot qui doit jeter la cassette grise *grey_tape* dans la corbeille *trashbin*. ThrowObject(robot,grey_tape,trashbin). La fonction

action.getCurrentStateAndPosition(state,position)

permet de récupérer l'état et la position courante. Cela nous permet de nous

situer dans l'automate de l'action. Imaginons que grey_tape soit dans la pince du robot et que le bras du robot est au repos. On en déduit que le triplet (pince/objet/support) est dans l'état OBJECT_FREE selon la classification présentée dans la figure 5.1. La position ne correspond pas à une position juste au-dessus de la corbeille. On est donc dans le premier état OBJECT_FREE en haut à gauche de l'automate de ThrowObject. La fonction

informAttentionAboutAction(action)

prévient le système attentionnel de l'action en cours. S'il n'y a pas de requête de priorité supérieure à l'action en cours (c'est-à-dire si le robot n'est pas en train de verbaliser une indication à l'homme d'après la section 3.4), cela se traduira pour ThrowObject par une focalisation des caméras du robot sur le dessus de la corbeille. La fonction

action.isFinalStateAndPosition(state,position)

renvoie "true" si l'état de l'automate caractérisé par "state" qui correspond à l'état du triplet (pince/objet/support) et la position est le dernier état de l'automate, et "false" sinon. Si la fonction renvoie true, l'algorithme termine avec succès. Si la fonction précédente renvoie false, la fonction

action.getNextTransition(transition)

est appelée. Elle renvoie la "transition" qui est soit de type mouvement si la fonction

transition.isMotion()

renvoie true, soit de type ouverture ou fermeture de la pince. Si l'on revient à l'exemple de ThrowObject(robot,grey_tape,trashbin), la première transition est de type mouvement. Le type de trajectoire est MHP_ARM_FREE selon la typologie décrite dans 5.2. S'il s'agit d'une transition de type mouvement, la fonction

action.getNextPosition(state,position,newPosition)

permet de calculer la nouvelle position. Dans notre exemple, il s'agit d'une position libre au-dessus de la corbeille. Si cette fonction échoue, l'algorithme termine sur un échec. La fonction

computeMotion(transition,newPosition,motion)

permet de planifier le mouvement *motion* de type *transition* jusqu'à la position *newPosition*. Dans notre exemple il s'agit de planifier le mouvement de type MHP_ARM_FREE qui permet au robot de déplacer l'objet de la position courante à la position libre au-dessus de la corbeille calculée précédemment. De même si cette fonction échoue, l'algorithme échoue. La fonction

executeMotion(motion)

exécute ensuite le mouvement planifié. S'il s'agit d'une transition de type ouverture ou fermeture de la pince, la fonction

executeGripperTransition(transition)

exécute cette opération. C'est le cas de l'étape suivante dans notre exemple. Une fois que l'objet est au-dessus de la corbeille, le robot ouvre la pince. La fonction

action.getHypothesisTransition(state,position,hypothesisTransition)

récupère les opérations d'ajout ou de suppression d'hypothèses de position. La fonction

applyHypothesisTransitions(hypothesisTransitions)

applique ces opérations. Dans notre exemple, il n'y a aucune opération pour la première étape. Concernant l'étape suivante, on supprime l'hypothèse selon laquelle l'objet est dans la pince. On ajoute l'hypothèse selon laquelle l'objet est dans la corbeille. Enfin la fonction

executeAction(action)

permet de continuer récursivement sur l'étape suivante.

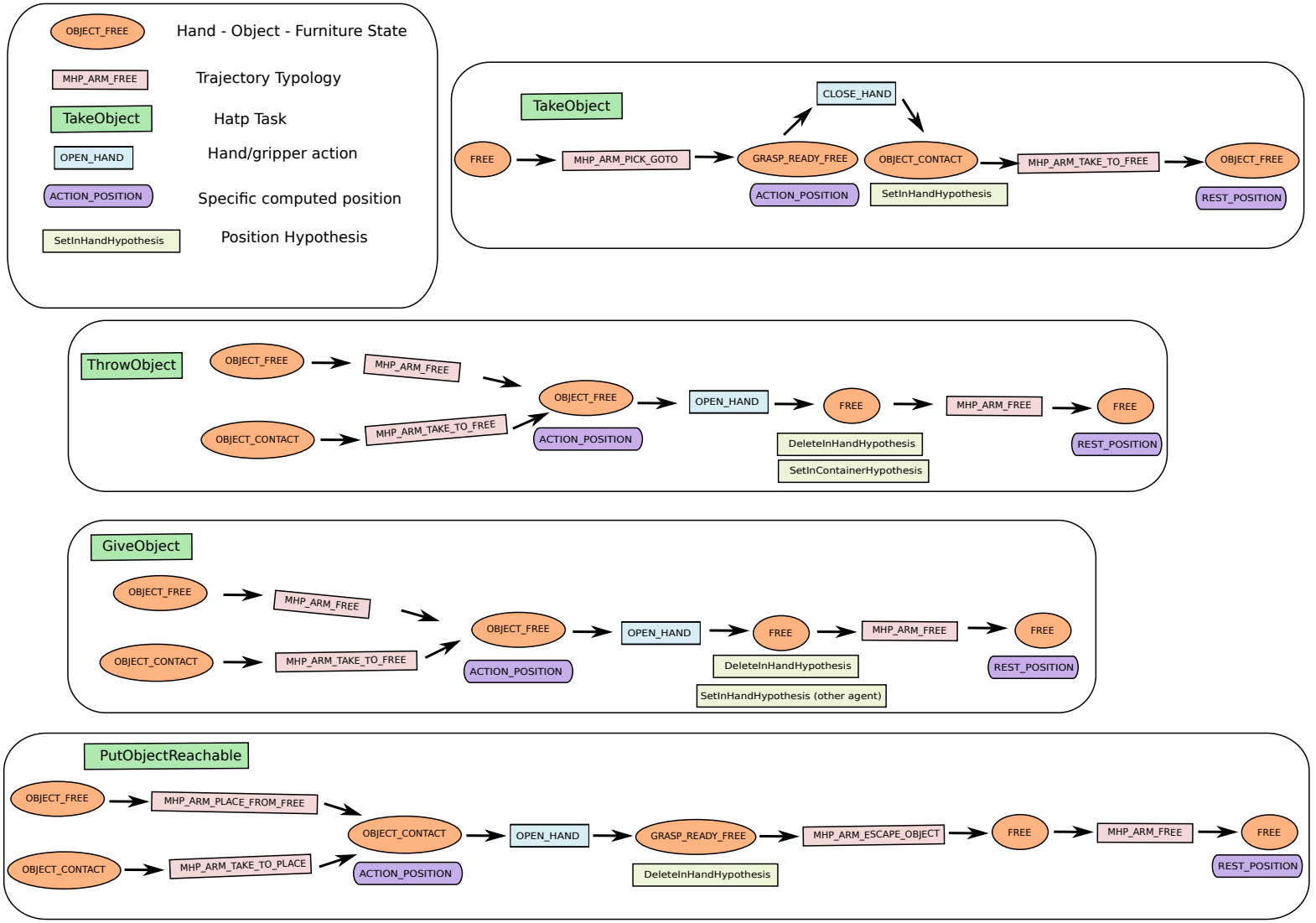


FIG. 5.4 – *
Graphe de conversion des actions en mouvements

Algorithm 6 executeAction (*action*)

```
action.getCurrentStateAndPosition(state,position)
informAttentionAboutAction(action)
if action.isFinalStateAndPosition(state,position) then
  action.isRealized = TRUE
else
  action.getNextTransition(transition)
  if transition.isMotion() then
    action.getNextPosition(state,position,newPosition)
    computeMotion(transition,newPosition,motion)
    executeMotion(motion)
  else
    executeGripperTransition(transition)
  end if
  action.getHypothesisTransitions(state,position,hypothesisTransitions)
  applyHypothesisTransitions(hypothesisTransitions)
  executeAction(action)
end if
```

5.3 Monitoring des actions

S'agissant du monitoring des actions de l'homme, le robot doit être en mesure d'identifier un mouvement de l'homme comme étant la réalisation d'une action donnée. Les "moniteurs" tels qu'évoqués dans la section 3.3 permettent d'inférer la réalisation d'actions au moyen de tests de complexité variable. On pourrait imaginer des moniteurs complexes construits à partir de modèles de Markov cachés. Dans le système actuel, on utilise un moniteur très simple de la prise d'un objet. Il est défini par une sphère virtuelle autour des objets qui sont posés sur un support. Dès qu'une main de l'homme pénètre dans cette sphère, ce moniteur déclenche. Ce déclenchement n'implique pas obligatoirement que cette action a été effectivement réalisée mais il nous permet de supposer que c'est une éventualité à considérer.

l'algorithme 7, explicite la gestion associée à la vérification par le robot de l'exécution par l'homme de ses actions. Imaginons que l'homme doive prendre la cassette grise grey_tape sur la table. TakeObject(human,grey_tape). On doit choisir le paramètre *timeStep* qui donne la fréquence avec laquelle l'algorithme examine le déclenchement des moniteurs et *waitMonitorDelay* qui donne la durée maximale d'attente de la réalisation de l'action par l'homme. On choisira *waitMonitorDelay* comme une estimation de la durée maximale

de réalisation de l'action. La fonction

action.getMonitor(actionMonitor)

permet de récupérer le "moniteur" associé à l'action s'il existe. Dans notre exemple il s'agit de la sphère autour de la cassette grise. La fonction

informAttentionAboutAction(action)

prévient le système attentionnel de l'action en cours. S'il n'y a pas de requête de priorité supérieure à l'action en cours (c'est-à-dire si le robot n'est pas en train de verbaliser une indication à l'homme d'après la section 3.4), cela se traduira pour TakeObject par une focalisation des caméras du robot sur l'objet qui doit être attrapé, c'est-à-dire grey_tape. La fonction

newTriggeredMonitor(monitor)

permet de récupérer le dernier "moniteur" *monitor* ayant déclenché depuis le dernier appel à la fonction. La fonction

getCurrentTime(curTime)

permet de récupérer la valeur de l'horloge. On boucle ensuite jusqu'à atteindre le temps d'attente maximum ou le déclenchement d'un moniteur. L'algorithme conclut au succès de la réalisation de l'action si c'est le moniteur attendu qui déclenche. L'algorithme conclut à l'échec de la réalisation de l'action s'il y avait un moniteur attendu et que, soit aucun moniteur n'a déclenché, soit un autre moniteur a déclenché. Dans le cas où il n'y a pas de moniteur attendu, l'algorithme ne peut conclure.

Algorithm 7 assessHumanActionExecution (*action*,*timeStep*,*waitMonitorDelay*)

```
action.getMonitor(actionMonitor)
informAttentionAboutAction(action)
action.isRealized = UNKNOWN
getCurrentTime(curTime)
for all timeStep before curTime + waitMonitorDelay do
  if newTriggeredMonitor(monitor) then
    if monitor == actionMonitor then
      action.isRealized = TRUE
    else
      if actionMonitor! = NULL then
        action.isRealized = FALSE
      end if
    end if
    exit FOR Loop
  end if
end for
if monitor == NULL then
  if actionMonitor! = NULL then
    action.isRealized = FALSE
  end if
end if
```

Chapitre 6

Expérimentations

Ce chapitre a pour objet de démontrer le fonctionnement réel du système présenté dans les trois chapitres précédents ainsi que dans la publication [2]. Dans la section 6.1, on décrit rapidement l'architecture LAAS qui facilite l'implémentation. Dans la section 6.2 nous décrivons de manière approfondie la réalisation d'un scénario simple pour expliciter tous les processus décrits dans les chapitres précédents. La section 6.3 décrit de manière globale le résultat d'une première campagne d'expérimentations. Enfin, nous décrivon dans la section 6.4 la migration vers le robot PR2 qui est plus facile à mettre en oeuvre que le robot Jido et qui offre un aspect extérieur plus agréable qui se prête mieux à des études utilisateurs.

6.1 L'architecture LAAS

L'architecture LAAS [1] pour systèmes autonomes, présentée figure 6.1, a été développée de manière incrémentale durant un grand nombre d'années et est mise en oeuvre sur l'ensemble des robots mobiles du LAAS. Il faut noter que les aspects généricité et programmabilité de cette architecture permettent une mise en oeuvre rapide et une bonne intégration des systèmes utilisés (GenoM ([17], [18]), OpenPRS ([26],[25]),...). Cette suite logiciel comprend trois niveaux :

- Le niveau décisionnel : Ce plus haut niveau intègre les capacités délibératives par exemple : produire des plans de tâches, reconnaître des situations, détecter des fautes, etc. Dans notre cas, il comprend :
  - un exécutif procédural OpenPRS qui est connecté au niveau inférieur auquel il envoie des requêtes qui vont lancer des actions (capteurs/actionneurs) ou démarrer des traitements. Il est responsable de la supervision des actions tout en étant réactif aux événements

provenant du niveau inférieur et aux commandes de l'opérateur. Cet exécutif a un temps de réaction garanti,
  - un planificateur (HATP),
  - une base de fait sous la forme d'une ontologie (ORO).
- Le niveau fonctionnel : Le plus bas niveau comprend toutes les actions et fonctions de perception de base de l'agent. Ces boucles de contrôle et traitements de données sont encapsulés dans des modules contrôlables (développés avec GenoM). Chaque module fournit des services et traitements accessibles par des requêtes envoyées par le niveau supérieur ou un autre module. Le module envoie en retour un bilan lorsqu'il se termine correctement ou est interrompu. Ces modules sont complètement contrôlés par le niveau supérieur et leurs contraintes temporelles dépendent du type de traitement qu'ils ont à gérer (servo- contrôle, algorithmes de localisation, etc.).
- Le niveau de contrôle des requêtes (défini dans l'architecture mais non utilisé au sein de nos instanciations sur Jido et Rackham) : Situé entre les deux niveaux précédents, le R2C « Requests and Replies Checker » [27] vérifie les requêtes envoyées aux modules fonctionnels (par l'exécutif procédural ou entre modules) et l'utilisation des ressources.

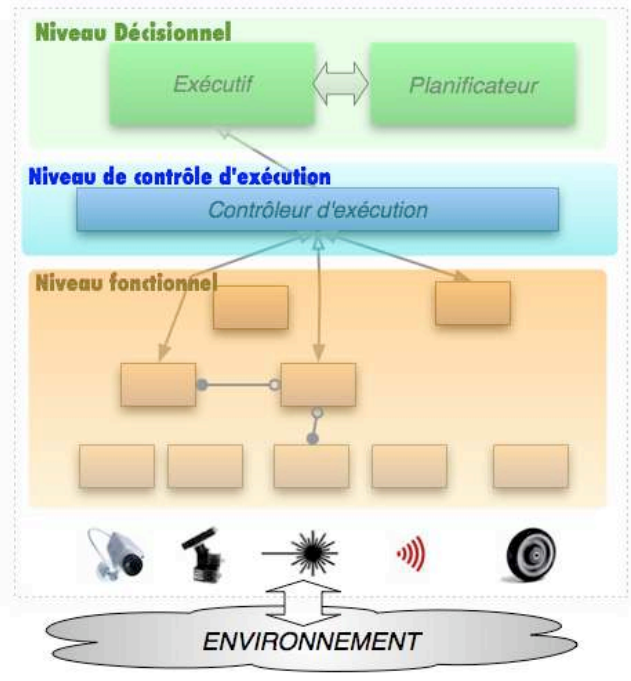


FIG. 6.1 – L'architecture LAAS

6.2 Illustration sur un exemple simple

Nous souhaitons illustrer le fonctionnement du système complet au moyen d'un scénario simple. Il y a une unique cassette sur la table. Elle est atteignable uniquement par le robot. La corbeille est quant à elle uniquement atteignable par l'homme.

La figure 6.2 illustre les principaux processus en oeuvre au cours des différentes étapes de la réalisation collaborative homme-robot du but. Le plan produit est évident. Il figure dans la troisième rangée intitulée "Goal and Plan". Il consiste en 4 actions successives qui impliquent le robot et l'homme. Le robot saisit la cassette et il la place sur la table à une position où elle est visible et atteignable par l'homme. Le robot demande alors à l'homme de saisir la cassette et de la jeter dans la corbeille. La première colonne, intitulée "Cameras", montre plusieurs photographies correspondant aux différentes étapes de l'exécution. La première image présente l'état initial. Les quatre images suivantes donnent respectivement l'état après le succès de la réalisation des 4 actions du plan. La deuxième rangée intitulée "3D Model", montre l'affichage de SPARK aux même instants. Le quatrième rang intitulé "Robot Speech Acts", présente les actes de communication produits au cours de l'exécution. Ils permettent d'informer l'homme sur la création des buts et des plans et leur statut. Ils décrivent également les actions que l'homme doit exécuter. Le cinquième rang décrit la connaissance du robot sur lui-même et les objets. Le sixième rang illustre la connaissance du robot sur les états de l'homme. Le septième rang précise l'action courante du robot et les processus associés de vérification des préconditions et des effets ainsi que la planification de mouvement. Les types de trajectoires utilisées sont décrites entre les affichages du modèle 3D. Le huitième rang donne les actions courantes de l'homme et les processus associés de vérification des préconditions et des effets ainsi que le monitoring d'action. Les types de moniteurs utilisés sont décrits entre les affichages du modèle 3D.

6.3 Premiers résultats expérimentaux

Trois jours durant, en février 2012, nous avons effectué autant d'expérimentations que possible pour différents scénarios. Un scénario est défini par le choix d'un but et d'un état initial du monde. Lors de chaque expérimentation, le robot, l'homme et les objets devaient être déjà positionnés par rapport au scénario choisi. Le robot devait alors détecter la position de tous ces éléments. Le but était alors inséré dans le système. Pour chacune de ces expérimentations, un film était produit avec une caméra haute défini-

FIG. 6.2 – Exemple de réalisation collaborative homme-robot d'un but.

tion, l'affichage du modèle 3d était aussi enregistré ainsi que les logs produits par le *robotcontroller*. Ces logs décrivent les buts, les plans symboliques, les types de plans géométriques ainsi que les étapes successives de l'exécution des plans symboliques avec leur statut. Des vidéos illustrant certaines de ces expérimentations peuvent être trouvées en utilisant l'url ci-dessous :
`http://homepages.laas.fr/mwarnier/Iros2012.html`.

6.3.1 Illustration de certaines fonctionnalités du système

J'utilise ici quelques images extraites des vidéos des expérimentations pour illustrer certaines fonctionnalités du système.

La figure 6.3 identifie les différents éléments présents sur les vidéos des expérimentations présentées sur la page dont l'url est ci-dessus. L'élément *vidéo* correspond à la vidéo de l'expérimentation réelle. L'élément *Affichage du modèle 3D du monde* donne l'état du monde courant. Les sphère vertes faiblement opaques correspondent aux "moniteurs" sur les actions de l'homme. L'élément *Affichage du planificateur de mouvement* correspond au modèle 3D utilisé par la planification de mouvement. Il est synchronisé avec l'état du monde courant au démarrage de la planification. Il donne la position finale du bras et des objets à la fin du mouvement planifié si la planification réussit. L'élément *Plan de haut niveau* donne le plan symbolique de haut niveau permettant de réaliser la tâche. L'action en cours est entourée par un cadre rouge. L'élément *Verbalisation du robot* donne le texte qui est verbalisé par le robot.

La figure 6.4 présente les trois scénarios des trois premières vidéos. Pour chacun de ces scénarios, le robot vient de calculer le plan de haut niveau et verbalise la première action. Comme décrit ci-dessus, les sphère vertes faiblement opaques correspondent aux "moniteurs" sur les actions de l'homme. Les sphères vertes autour des objets sur les tables sont les "moniteurs" de l'action prendre l'objet en question. Les sphères vertes autour des sets de tables bleu et vert sur les tables sont les "moniteurs" de l'action déposer un objet à cet endroit. La sphère verte au dessus de la corbeille est le "moniteur" de l'action "jeter un objet dans la corbeille".

La figure 6.5 illustre la réalisation de l'action "prendre la cassette blanche" par le robot. Le robot commence par planifier un premier mouvement et exécuter ce premier mouvement. Puis il ferme la pince et ajoute l'hypothèse de positionnement "objet dans la pince selon la transformation enregistrée

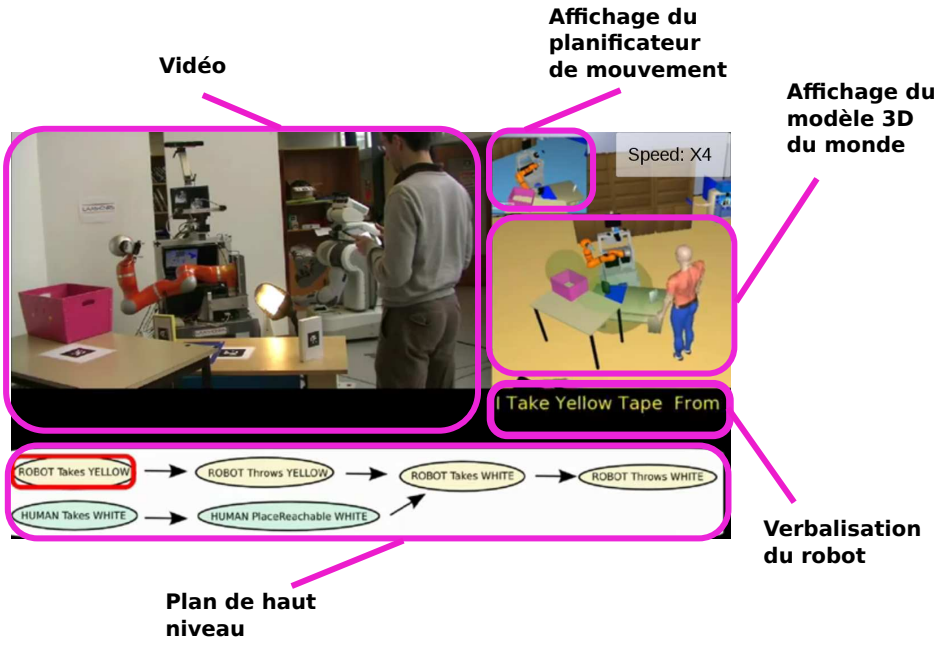


FIG. 6.3 – *
Identification des différents éléments sur les vidéos des expérimentations.

à la fin de la prise". Enfin il planifie et exécute le retour à une position de repos.

La figure 6.5 illustre une partie de la réalisation de l'action "jeter la cassette jaune (noire dans le modèle) dans la corbeille" par le robot. Le robot ouvre la pince et ajoute l'hypothèse de positionnement "objet dans la corbeille".

La figure 6.7 illustre la détection par le robot de la prise de la cassette jaune (noire dans le modèle) par l'homme. Le déclenchement du "moniteur" entraîne la création d'une hypothèse de positionnement "objet dans la main de l'homme".

La figure 6.8 illustre la détection par le robot de la pose de l'objet cassette blanche sur la table par l'homme. L'hypothèse "cassette blanche dans la main de l'homme" est cassée une fois que la main de l'homme est suffisamment loin de la position perçue de la cassette.

La figure 6.9 présente un exemple de reprise d'erreur. Initialement les trois cassettes sont sur la table. Le robot produit un plan symbolique de

nom de la tâche	distinction au sein d'une même tâche	nb scénarios
nettoyer la table	corbeille accessible aux deux	6
nettoyer la table	corbeille accessible à l'homme uniquement	11
nettoyer la table	corbeille accessible au robot uniquement et homme absent	4
nettoyer la table	corbeille accessible au robot uniquement	23
obtenir le jouet		8

TAB. 6.1 – Diversité des conditions de départ

six actions : *Prendre la cassette grise. Jeter la cassette grise. Prendre la cassette jaune (noire dans le modèle). Jeter la cassette jaune. Prendre la cassette blanche. Jeter la cassette blanche.* La troisième action échoue. Le robot déplace alors la cassette jaune mais ne la saisit pas. Le robot croyant l'avoir attrapée déplace son bras vers la position de repos. Il perçoit alors la cassette jaune. Le premier plan échoue. Un nouveau plan de quatre actions est produit : *Prendre la cassette jaune (noire dans le modèle). Jeter la cassette jaune. Prendre la cassette blanche. Jeter la cassette blanche.*

6.3.2 Analyse des résultats

Flexibilité dans le choix de l'état initial. Nous avons actuellement deux buts différents. Le but *nettoyer la table* consiste à mettre toutes les cassettes qui se trouvent sur une table dans une corbeille. Le but *obtenir le jouet* consiste à attraper un petit objet initialement recouvert par une grosse boîte à fond plat. Il y a de nombreux scénarios différents pour le but *nettoyer la table* selon le nombre de cassette et l'atteignabilité initiale par les deux agents de la corbeille et des cassettes. Si la corbeille est accessible aux deux agents, ils travaillent de manière indépendante. Si la corbeille n'est accessible qu'au robot et que certaines cassettes ne sont accessibles qu'à l'homme alors le robot requiert l'aide de l'homme. Si la corbeille n'est accessible qu'à l'homme et que certaines cassettes ne sont accessibles qu'au robot alors le robot facilite le travail de l'homme en participant à la réalisation de la tâche. Nous avons identifié les différentes situations de départ pour 52 expérimentations.

Analyse quantitative.

Des statistiques ont été récupérées et analysées pour 44 expérimentations. Le tableau 6.2 donne des statistiques générales sur ces 44 expérimentations.

Le tableau 6.3 donne des statistiques sur les scénarios pour lesquels le but n'est pas atteint. Les états bloquants correspondent à des états correctement mis à jour par le robot mais qui ne permettent pas de poursuivre la réalisation

expérimentations	44
nombre de plans symboliques	69
nombre d'actions	346
nombre de plans géométriques	150
durée moyenne	2.3 minutes
moyenne de plans symboliques par but	1.7
succès	21 (48%)

TAB. 6.2 – Statistiques générales

	nombre	pourcentage
avancement		70%
homme abandonne	3	7%
état bloquant	4	9%
état erroné	16	38%

TAB. 6.3 – Catégorisation des scénarios pour lesquels le but n'est pas atteint

du but. Dans ce cas le robot détermine à raison qu'il ne peut pas poursuivre. Les états erronés correspondent à des états qui ne sont pas correctement mis à jour par le robot mais qui ne permettent pas de poursuivre la réalisation du but. Dans ce cas le robot détermine à tort qu'il ne peut pas poursuivre.

Analyse qualitative :

La campagne d'expérimentation a démontré que dans le cadre de nos scénarios, notre robot parvenait à exécuter et suivre des actions, mettre à jour et corriger sa croyance sur l'état du monde et replanifier en cas d'échec survenant au cours de la réalisation du plan. Néanmoins le robot et l'homme échouent de manière assez fréquente (45,5%) à réaliser le but. Ce n'est pas une surprise compte tenu de la complexité du système. D'autant que ces erreurs ne sont pas toutes attribuables à un échec du robot. Il faut ditinguer les 3 cas où l'homme a abandonné et les 4 cas où le robot a constaté l'impossibilité de poursuivre la tâche compte tenu du nouvel état du monde des 16 autres cas où la non réalisation du but est attribuable à une erreur du robot.

Nous essayons de recenser les principales sources d'erreurs dans le but d'orienter certaines améliorations de notre système.

Problèmes de perception. L'humain est parfois perdu par la kinect. Lorsque la pince du robot est en mouvement et passe devant le tag d'un objet cela produit un artefact de mesure qui a pour effet de faire *sauter* l'objet. De même, le robot ne parvient pas parfois à détecter les objets petits ou éloignés

Problème du contrôleur de haut niveau. Les automates de monitoring et d'exé-

type d'erreur	nombre
non perception de l'homme ou de l'objet	4
saut de l'objet du fait d'un artefact de perception	5
échec d'automates du contrôleur de haut niveau	3
erreur de réglage des sphères de suivi d'action	3
confusion de plusieurs sphères de suivi d'action	2

Tab. 6.4 – Récurrence des causes d'échec

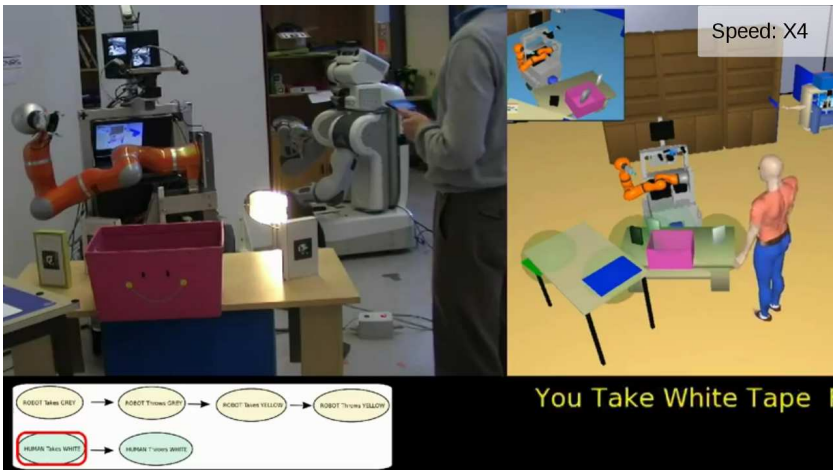
cution des actions ainsi que les processus de mise à jour du monde comportent un certain nombre de bugs résiduels qui peuvent entraîner l'échec de l'exécution.
Différence entre le calcul heuristique de l'atteignabilité et la planification de mouvement. Le calcul heuristique de l'atteignabilité est parfois trop optimiste en comparaison de l'atteignabilité vraie calculée par la planification de mouvement.
Réglage et confusion des sphères de suivi d'actions. Compte tenu d'une baisse de la précision du suivi de la position de la main de l'homme par la kinect dans certaines situation où le champ de vue du kinect est partiellement obstrué, la taille de la sphère est trop petite pour que le mouvement soit correctement repéré. De plus, en l'état du système, la superposition de plusieurs sphères de suivi d'action distinctes peut être source d'erreur dans la mesure où seule la première sphère est prise en compte.

Le tableau 6.4 identifie pour 17 scénarios (il y en a plus que les 16 considérés précédemment car même si l'on ne disposait pas de fichiers de log de contrôleur de haut niveau, la cause de l'échec a pu être déterminée par l'examen précis des vidéos) la récurrence de certaines causes d'échecs.

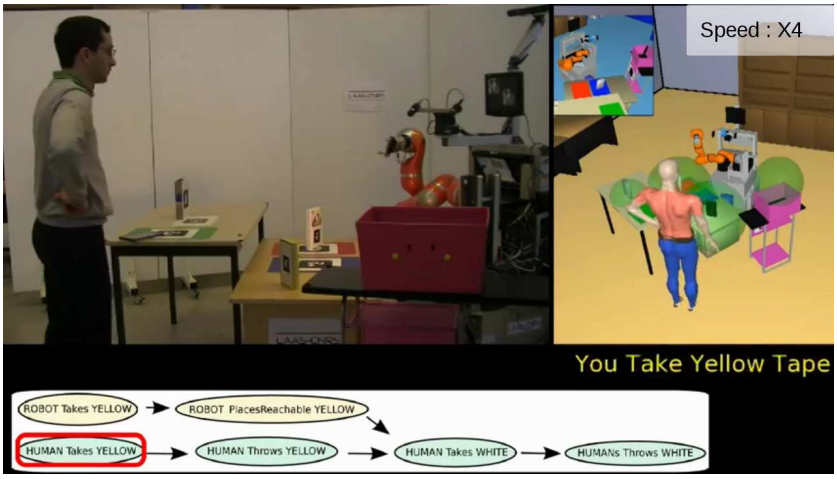
6.4 Utilisation du robot PR2

Nous avons fait le choix de conserver l'architecture de la couche décisionnelle telle que décrite dans la figure 1.4. Tous les composants de la couche décisionnelle sont conservés. Du point de vue du contrôleur d'exécution, seul les actuateurs diffèrent par rapport au robot Jido. Les requêtes de déplacement de la tête, d'ouverture et de fermeture de la pince ainsi que d'exécution des trajectoires planifiées sont différentes et l'interface a donc été modifiée en conséquence.

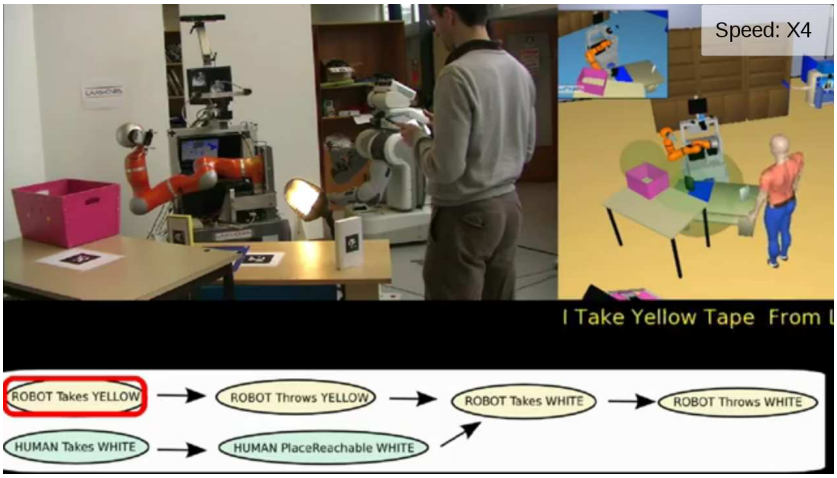
Nous sommes en mesure désormais de réaliser les mêmes tâches que sur le robot Jido. La figure 6.10 est un arrêt sur image pendant la réalisation de la tâche *nettoyerlatable* par Pr2 et un homme.



(a) Chacun travaille indépendamment. (Vidéo 1)

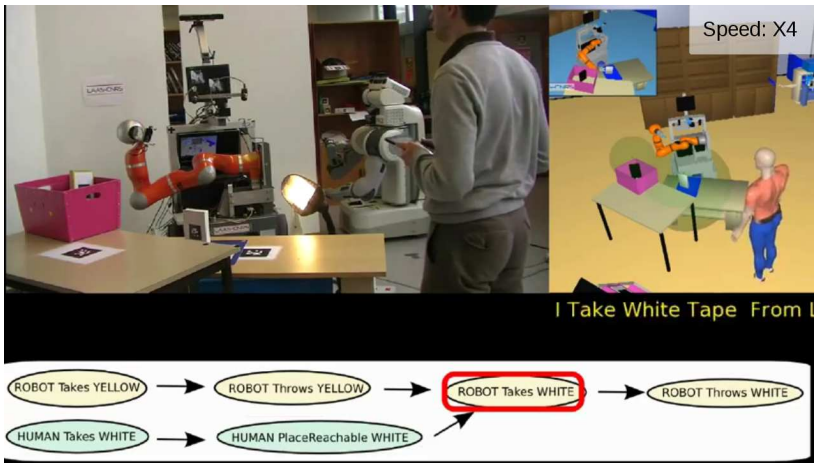


(b) Le robot aide l'homme. (Vidéo 2)

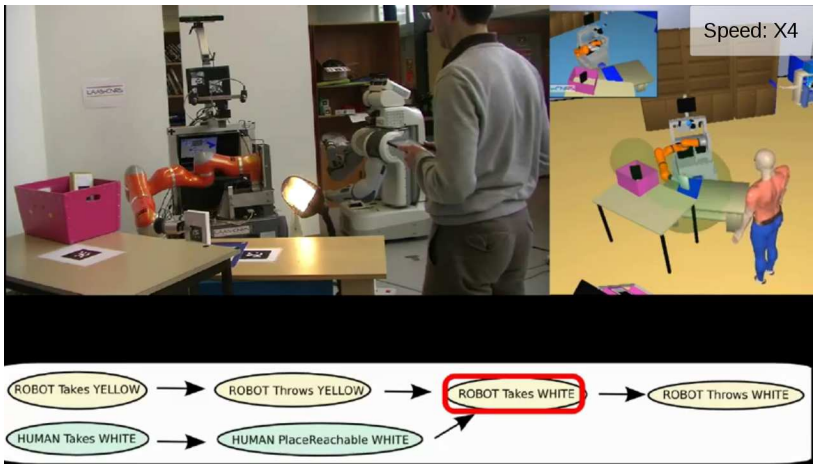


(c) L'homme aide le robot. (Vidéo 3)

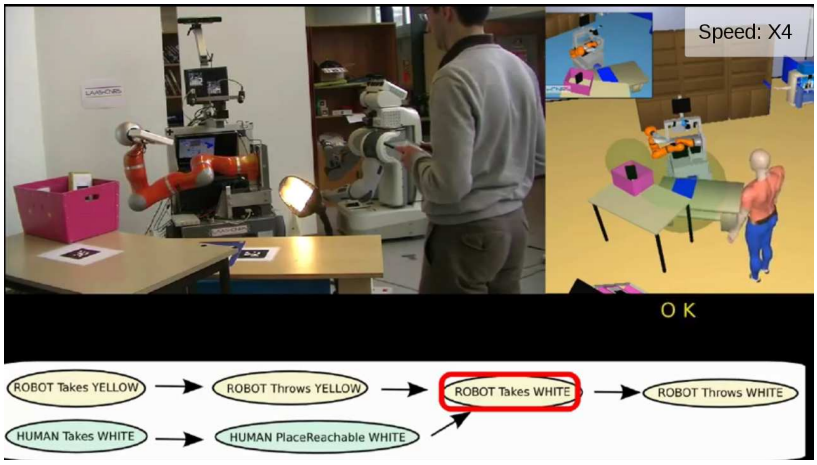
FIG. 6.4 – Trois scénarios différents pour la tâche nettoyer la table.



(a) Le premier mouvement a été planifié. L'exécution du mouvement va commencer.

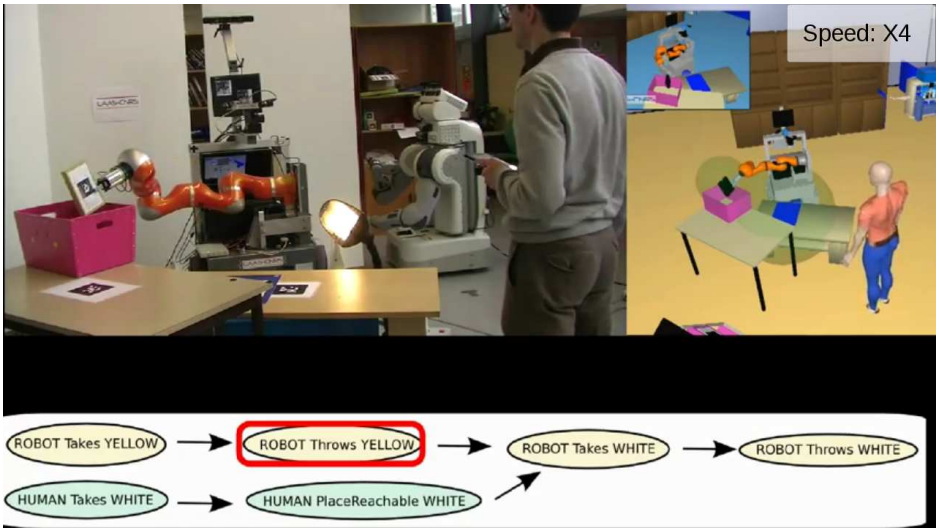


(b) Le premier mouvement a été exécuté. Le robot a fermé sa pince. Le second mouvement a été planifié.

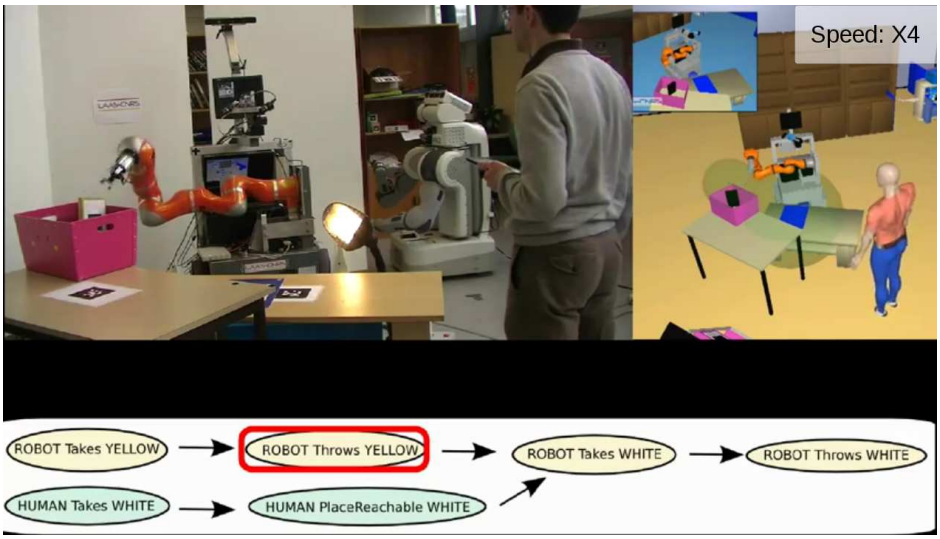


(c) Le robot a exécuté le second mouvement.

FIG. 6.5 – Le robot prend la cassette blanche.

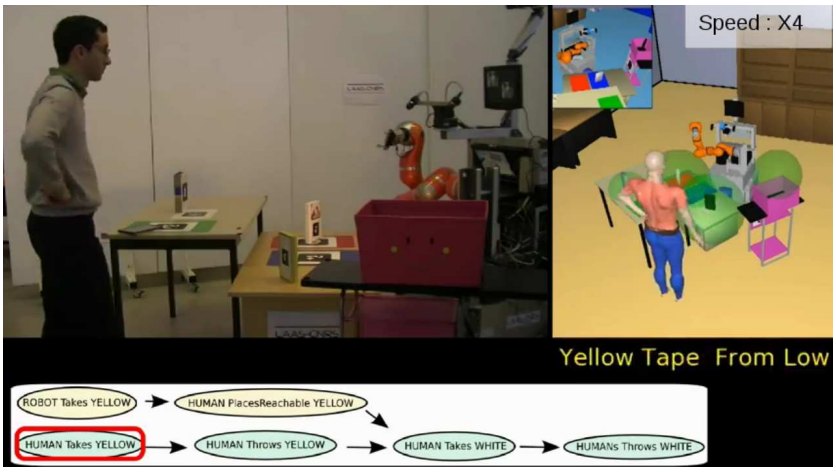


(a) Le robot a exécuté le premier mouvement.

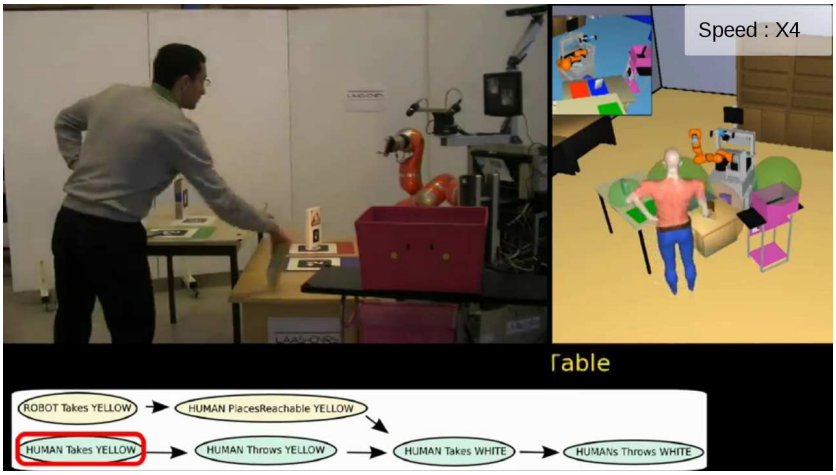


(b) Le robot a lâché l'objet.

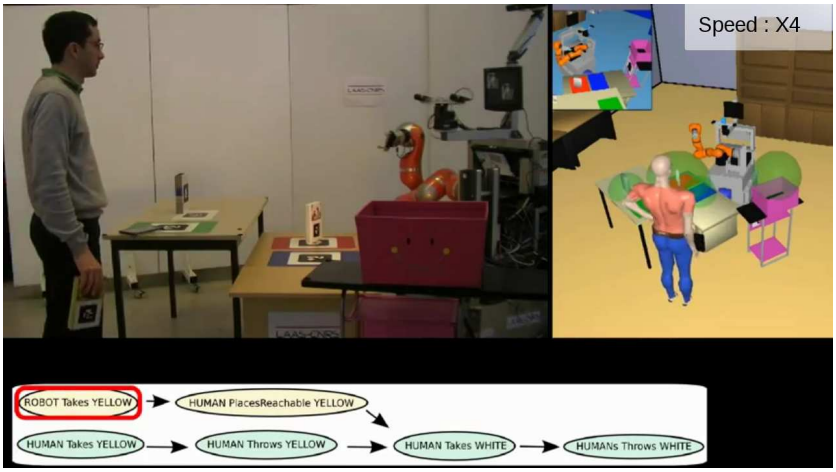
FIG. 6.6 – .



(a) Verbalisation de l'action par le robot.

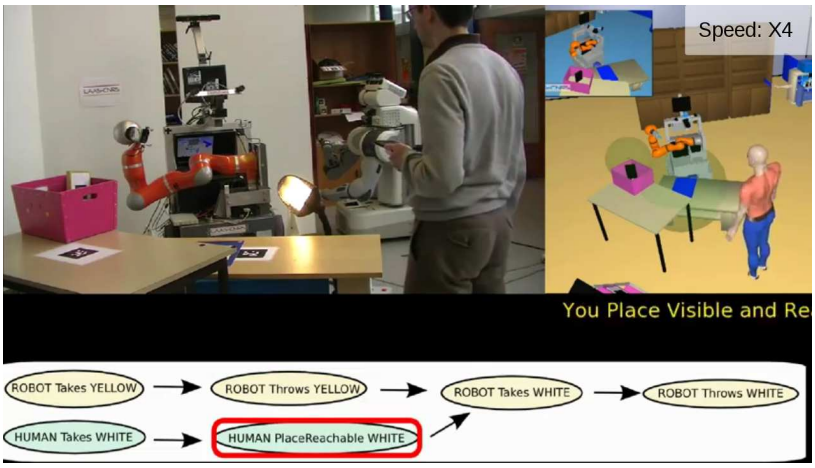


(b) L'homme dirige sa main vers l'objet pour le prendre et déclenche le "moniteur" qui se colorie alors en orange.

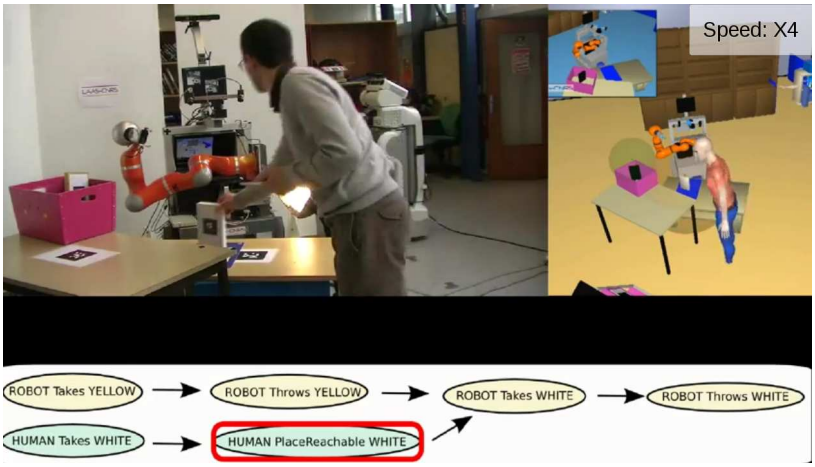


(c) L'objet est maintenant positionné dans la main de l'homme même s'il n'est pas perçu.

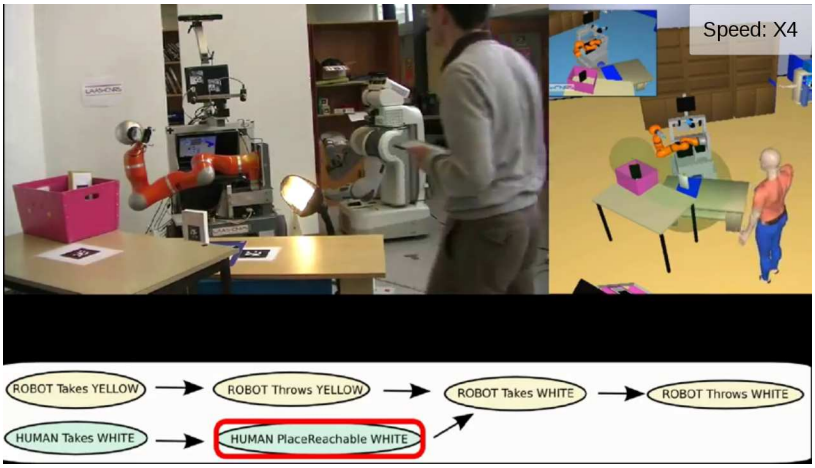
FIG. 6.7 – Détection par le robot de la prise de la cassette jaune (noire dans le modèle) par l'homme.



(a) Verbalisation de l'action par le robot.

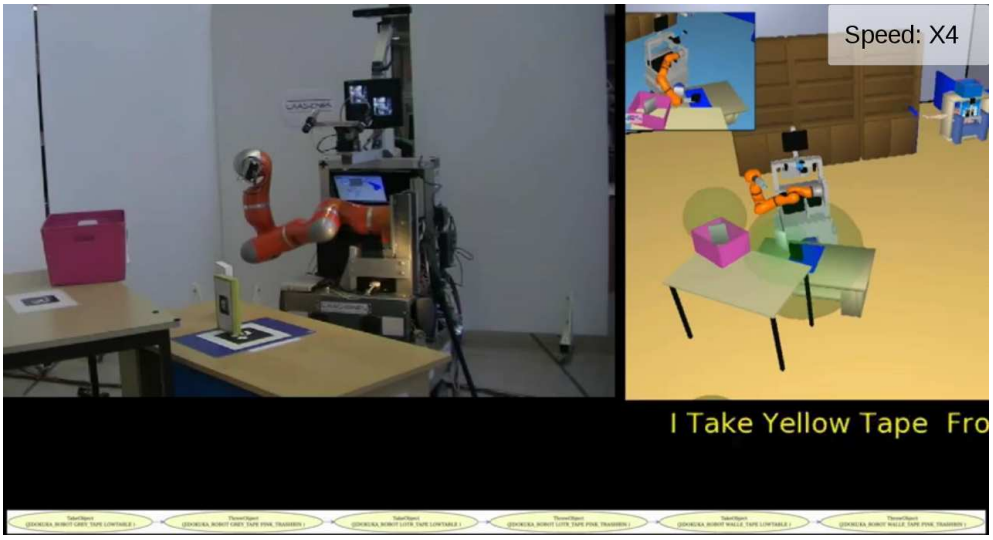


(b) L'homme pose l'objet ce qui déclenche le "moniteur". Le robot perçoit la nouvelle position de l'objet.

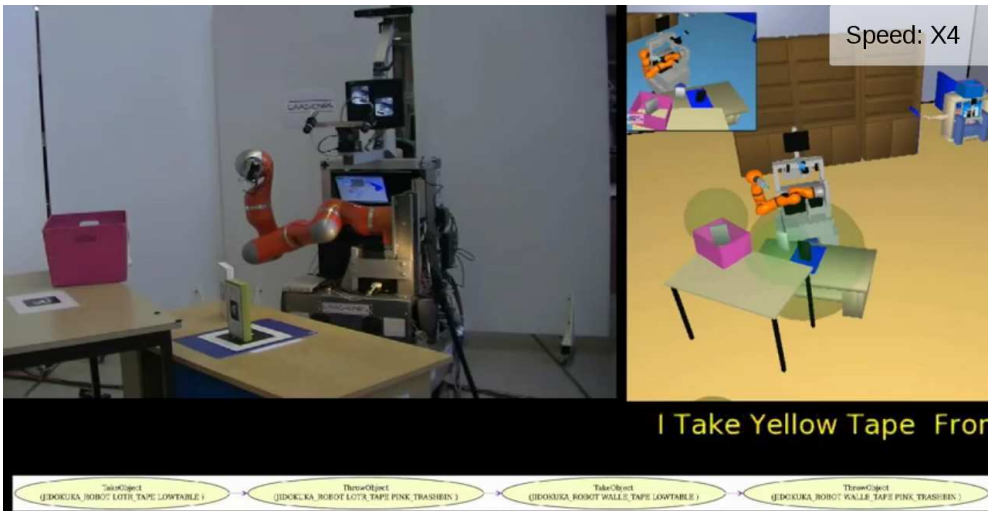


(c) Le robot casse l'hypothèse de positionnement "objet dans la main de l'homme" qui est contradictoire avec la position perçue.

FIG. 6.8 – Détection par le robot de la pose de l'objet cassette blanche sur la table par l'homme.



(a) Le robot commence à prendre la cassette jaune (noire dans le modèle). C'est la troisième action de son plan de six actions.



(b) Le robot a déplacé la cassette jaune (noire dans le modèle) en essayant de l'attraper. Il constate son échec. Un nouveau plan symbolique à quatre actions est produit. Le robot planifie le premier mouvement de la première action du plan symbolique qui consiste à prendre la cassette jaune qui est sur la table

FIG. 6.9 – Exemple de reprise d'erreur.

FIG. 6.10 – *
Implémentation du système sur le robot Pr2

Troisième partie

Gestion des croyances divergentes

Les différents agents peuvent avoir des croyances distinctes sur l'état du monde d'abord parce qu'ils n'ont pas les mêmes capacités intrinsèques d'observation et de compréhension de l'environnement mais aussi tout simplement parce que des actions sont réalisées en leur absence. Les croyances d'un agent sont déterminantes pour expliquer son comportement, ses propos et ses plans pour réaliser une tâche. Du point de vue du robot, il est donc essentiel de pouvoir identifier s'il y a des divergences entre ses croyances et celles des humains qu'il côtoie. Ensuite le robot pourrait utiliser les croyances distinctes d'un homme pour interpréter ses actions, comprendre ses propos et réaliser une tâche avec lui.

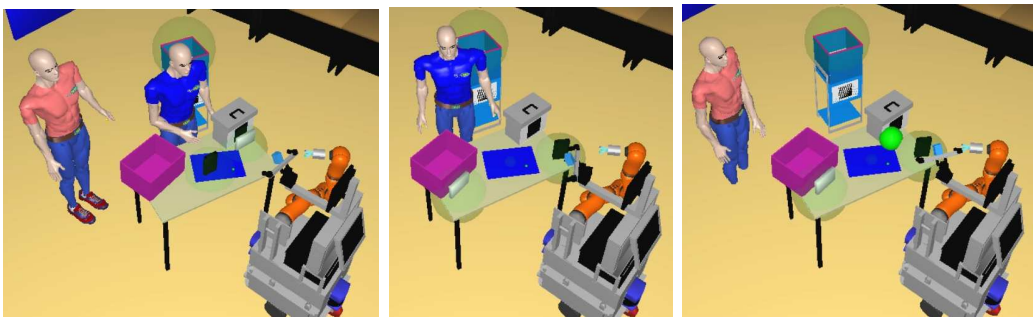
Le premier chapitre de cette partie (7) décrit le raisonnement que le robot utilise pour construire et mettre à jour les croyances de tous les agents présents. Le deuxième chapitre (8) présente l'extension du planificateur HATP qui permet de prendre en compte les croyances de chaque agent. Le troisième chapitre (9) présente des résultats d'utilisation de ces croyances divergentes.

Chapitre 7

Construction de croyances divergentes

Il faut d'abord préciser ce qu'on entend par croyances divergentes.

7.1 Deux exemples illustratifs



(a) Robert (Rose) et Bob (Bleu) sont présents. Ils connaissent la position de chaque objet.

(b) Robert part et Bob déplace deux objets.

(c) Robert revient et pense que la boîte blanche est toujours derrière la grosse boîte grise.

FIG. 7.1 – "False/Distinct belief" scenario.

Les figures 7.1 et 7.2 montrent deux exemples illustratifs. Ce sont des captures d'écrans de l'affichage du module de raisonnement spatial, enregistrées lors d'expérimentations réelles. Elles mettent en scène notre robot cognitif et deux humains, Robert et Bob. Robert porte un tee-shirt rose et Bob porte un tee-shirt bleu. Les petites boîtes noire et blanche peuvent être

(a) Robert et Bob sont présents. Ils connaissent la position de chaque objet.

(b) Robert part et Bob déplace un objet.

(c) Robert revient et ne sait pas où est la boîte noire.

FIG. 7.2 – "Lack of belief" scenario.

déplacées aisément. La corbeille rose et la boîte grise sont des objets plus volumineux dont la position est fixe durant l'expérimentation. Tous ces objets sont sur une table.

Les figures 7.1 illustrent le concept de *False/Distinct belief on object position* : dans la fig. 7.1(a), on peut supposer que Robert, Bob et le robot partagent la même croyance sur la position de tous les objets. Dans la figure 7.1(b) Robert est absent et Bob déplace deux boîtes. Dans la figure 7.1(c) Robert revient. Il observe la nouvelle position de la boîte noire mais il ne peut observer la nouvelle position de la boîte blanche et il ne pourrait pas la voir non plus si elle occupait son ancienne position. En conséquence, on peut supposer qu'il se souvient toujours que cet objet est proche de la boîte grise. Sa croyance sur la position de cet objet est différente de la position réelle de cet objet tel que perçue par le robot. Nous représentons cela au moyen d'une sphère verte complètement opaque localisée là où l'humain croit que l'objet se trouve.

La figure 7.2 illustre le concept de *Lack of belief on object position* : dans la figure 7.2(a), Robert, Bob et le robot partagent la même croyance sur la position de tous les objets. Dans la figure 7.2(b), Robert est absent et Bob déplace une des boîtes. Dans la figure 7.2(c), Robert ne peut pas voir la nouvelle position de la boîte mais il constate que l'objet n'est plus là où il l'avait vu. Le robot en conclut que Robert ne sait plus où se trouve l'objet. Nous représentons cela au moyen d'une sphère verte partiellement opaque localisée là où l'humain considérait que l'objet était jusque là.

Sans le nouveau raisonnement présenté dans ce chapitre, le robot considérerait que Robert partage les même croyances que lui même. (i.e. Robert connaît les nouvelles positions des cassettes).

7.2 Raisonnement sur les croyances distinctes

Nous présentons ci dessous les algorithmes de gestion des croyances.

7.3 Calcul des faits sans gestion explicite des croyances distinctes

Avant de gérer les croyances pour chaque agent, nous utilisions l'algorithme 8 pour construire une représentation symbolique du monde. Cette algorithme est une boucle simple, qui appelle la fonction *agent.computeFacts(objects)* qui calcule les faits symboliques décrits plus haut dans le paragraphe 3.1 pour chaque agent présent dans la scène.

Algorithm 8 computeFactsSimple (agents,objects)

```
for all agent in agents do
  if agent.isPresent then
    agent.computeFacts(objects)
  end if
end for
```

Dans cette algorithme, les nouvelles croyances ne sont pas calculées si l'agent a quitté la scène. Quand il revient, ses croyances sont mises à jour à partir de l'état courant du monde. Ceci repose sur l'hypothèse forte que les agents prennent connaissance immédiatement et complètement de tous les changements perçus par le robot. En d'autres termes, tous les agents présents partagent le même état symbolique du monde.

7.4 Gestion explicite des croyances distinctes

Le nouvel algorithme de gestion des croyances distinctes pour chaque agent correspond à l'algorithme 9. Avant d'entrer dans le détail de l'algorithme, nous présentons l'ensemble des variables et des fonctions utilisées. Toutes ces croyances sont du point de vue du robot même si elles sont relatives à un autre agent. Il s'agit alors du point de vue du robot sur la croyance d'un autre agent.

Variables utilisées dans l'algorithme :

Pour chaque objet et agent, nous utilisons les variables *positionKnown*, *hasDistinctPosition* et *distinctPosition* pour décrirent les croyances relatives à la position de l'objet.

La variable booléenne *agent.positionKnown(object)* indique si un agent connait la position d'un objet.

agent.hasDistinctPosition(object) est une variable booléenne qui indique si la croyance courante d'un agent sur la position d'un objet est distincte ou non de celle du robot.

agent.distinctPosition(object) contient la croyance distincte de l'agent sur la position d'un objet si l'agent a une croyance distincte pour cette objet. Elle est vide autrement.

La croyance du robot sur la position de l'objet est stockée dans une autre variable.

Ces variables permettent de décrire aisément les croyances de chaque agent à propos des positions d'un objet. Par exemple :

Robert ne sait pas où est la boîte grise.

```
robot.positionKnown(greyBox) = true
robert.positionKnown(greyBox) = false
robert.hasDistinctPosition(greyBox) = false
robert.distinctPosition(greyBox) = empty
```

Robert et le robot croient tous les deux que la boîte grise est à une position donnée.

```
robot.positionKnown(greyBox) = true
robert.positionKnown(greyBox) = true
robert.hasDistinctPosition(greyBox) = false
robert.distinctPosition(greyBox) = empty.
```

Robert croit que la boîte grise est à une position *someDistinctPosition*1 dans l'espace qui est connue du robot mais différente de la croyance actuelle du robot sur la position de la boîte grise.

```
robot.positionKnown(greyBox) = true
robert.positionKnown(greyBox) = true
robert.hasDistinctPosition(greyBox) = true
robert.distinctPosition(greyBox) = someDistinctPosition1
```

Robert croit que la boîte grise est à une position inconnue du robot.

```
robot.positionKnown(greyBox) = false
robert.positionKnown(greyBox) = true
robert.hasDistinctPosition(greyBox) = false
robert.distinctPosition(greyBox) = empty
```

Le tableau 7.1 présente les fonctions utilisées par l'algorithme 9.

agent.SaveCurrentSharedPositions(objects) : Cette fonction est appelée quand un *agent* quitte la scène. Elle enregistre pour chaque objet dans la liste *objects* la position courante dans *distinctPosition* sauf si cet agent a déjà une position distincte pour cet objet. Elle a également pour effet de mettre la variable agent.hasDistinctPosition(object) à true pour chacun de ces objets.
objects.previouslyUnseenAreUnknown(agent) : cette fonction est appliquée une seule fois au moment de l'apparition de l'agent *agent*. Elle assigne la variable *positionKnown* à false pour tous les objets de la liste *objects* qui n'avaient jamais été vus par cet agent jusque là.
object.isSeenAtRobotPosition(agent) : cette fonctions indique si un *agent* peut voir un objet *object* à la position qu'il a dans le modèle du robot.
agent.deleteDistinctPosition(object) : cette fonction efface la variable *distinctPosition* d'un *agent* pour un objet *object*.
objects.setDistinctPositionsInModel(agent) : cette fonction déplace dans le modèle 3D chacun des objets de la liste *objects* vers la position *distinctPosition* d'un *agent* si elle est distincte de celle du robot.
agent.computeFacts(objects) : calcule les faits pour tous les objets de la liste *objects* en utilisant les positions actuelles dans le modèle 3D qui sont normalement celle d'un *agent* spécifique.
object.isSeenAtDistinctPosition(agent) : cette fonction détermine si un *agent* peut voir un objet *object* là où il croit que l'objet est.
objects.resetRobotPositionsInModel() : cette fonction réinitialise dans le modèle 3D les positions de tous les objets de la liste *objects* sur les croyances du robot.

TAB. 7.1 – Fonctions utilisées par l'algorithme 9.

L'algorithme 9 de calcul des faits avec gestion des croyances distinctes est présenté ci dessous en pseudo-code. Lorsqu'un agent est absent, des objets peuvent être déplacés. Quand l'agent revient, il y a trois cas distincts si un objet a bougé :

- L'agent voit la position courante de l'objet et perçoit donc le changement. Il n'aura pas de croyance distincte sur la position de cet objet.
- L'agent ne voit pas la nouvelle position mais il perçoit que l'objet n'est plus là où il était. Dans ce cas, l'agent sait qu'il ne sait plus où est cet objet.
- L'agent n'a aucun moyen de percevoir le déplacement car il ne peut voir ni la nouvelle position, ni l'absence de l'objet de la position précédente. Dans ce cas, l'agent a une croyance distincte de celle du robot tant que ces deux conditions restent vrai ou si un autre agent l'informe de la nouvelle position.

Nous nous appuyons sur l'hypothèse simplificatrice suivante : le robot considère que les humains présents dans la scène perçoivent et interprètent correctement toutes les actions sur les objets et connaissent donc toutes les nouvelles positions des objets même si certains de ces objets ne sont pas visibles dans ces positions. L'agent peut alors imaginer cette position compte tenu de l'action. Les positions distinctes sont autorisées uniquement pour les objets qui peuvent être déplacés aisément et cachés derrière d'autres objets plus gros. Cette gestion des positions distinctes peut augmenter les calculs nécessaires. Les faits relatifs à un objet pour lequel un agent possède une position distincte doivent être recalculés pour cet agent. Dans le pire des cas, le nombre de calculs supplémentaires est le produit du nombre d'objets pouvant avoir une position distincte que multiplie le nombre de faits par objet que multiplie le nombre d'agents autres que le robot. Dans nos scénarios avec au maximum trois agents et cinq objets pouvant être déplacés, les calculs supplémentaires restent raisonnables.

7.5 Généralisation

Dans cette section nous voulons généraliser le concept de croyances distinctes à d'autres attributs que la position des objets. Nous nous limitons à des propriétés physiques des objets. On pourrait considérer la température, la couleur, le poids, le contenu liquide ou solide ... Prenons l'exemple de la température d'une théière, de la couleur d'une balle, du poids d'une brique de lait, de la nature du contenu d'un verre, du contenu d'une boîte.

Algorithm 9 computeFactsWithDistinctBeliefs (*agents*,*objects*)

```
Require: computeFactsSimple (robot,objects)
  for all agent in agents except robot do
    if agent.hasJustDisappeared then
      agent.SaveCurrentSharedPositions(objects)
    else if agent.isPresent then
      if agent.hasJustAppeared then
        objects.previouslyUnseenAreUnknown(agent)
      end if
      for all object in objects do
        if agent.hasDistinctPosition(object) or
        Not(agent.positionKnown(object)) then
          if object.isSeenAtRobotPosition(agent) then
            agent.deleteDistinctPosition(object)
            agent.positionKnown(object) := true
          end if
        end if
      end for
      objects.setDistinctPositionsInModel(agent)
    end if
    if agent.isPresent then
      agent.computeFacts(objects)
      for all object in objects do
        if agent.hasDistinctPosition(object) then
          if object.isSeenAtDistinctPosition(agent) then
            agent.positionKnown(object) := false
            agent.deleteDistinctPosition(object)
          end if
        end if
      end for
      objects.resetRobotPositionsInModel()
    end if
  end for
```

Pour chacun de ces attributs, le robot doit pouvoir déterminer lui-même sa valeur et estimer la croyance de l'homme sur cette valeur. On identifie ci-dessous les connaissances et raisonnements dont le robot a besoin pour cela.

- L'ensemble des états/valeurs possibles.
- La capacité à percevoir l'attribut courant.
- La capacité à percevoir ou inférer la non-validité de la croyance sur l'état courant.
- La capacité à percevoir ou inférer l'action qui fait évoluer la propriété.
- L'hypothèse initiale.

Nous allons expliciter ces différents éléments et les décliner pour tous les attributs considérés ci-dessous :

- La position d'un objet.
- La température d'une théière.
- La couleur d'une balle.
- Le poids d'une brique de lait.
- La nature du contenu d'un verre.
- Le contenu d'une boîte.

L'ensemble des états/valeurs possibles. Le robot doit avoir un modèle des différentes valeurs/états atteignables pour un attribut. Cela peut être une valeur numérique sur une échelle objective. Cela peut être également une valeur symbolique à partir de classes dont l'interprétation peut dépendre de l'agent considéré. La valeur est généralement incluse dans un sous-ensemble lié au contexte. Pour la position d'un objet il s'agit de la position dans l'espace qui peut être décrite de manière symbolique ou dans l'espace cartésien. La température de la théière peut être exprimée de manière numérique sur l'échelle Celsius ou alors de manière symbolique par des classes de température : froid, tiède, chaud, brûlant. La notion de brûlant pouvant être propre à un agent en fonction de la tolérance à la température. La pince d'un robot pourrait supporter des températures plus importantes que la main de l'homme. De même la couleur peut être exprimée dans une échelle chromatique telle que l'échelle RGB ou en utilisant les couleurs usuelles. Le poids (masse) de la brique de lait peut être exprimé en kilogramme ou au moyen des classes léger et lourd qui dépendent bien entendu également de l'agent. Le contenu d'un verre sera le plus souvent interprété par l'homme comme appartenant à un certain nombre de liquides connus (jus, vin, eau, lait ...) Il pourrait être aussi décrit par son acidité, son amertume, sa couleur... Le contenu d'une boîte sera décrit par la liste des objets qu'elle contient.

La capacité à percevoir l'état courant. Le robot doit avoir un modèle de sa propre capacité et de celle des autres agents à percevoir l'état courant. Les différents sens et capteurs dont disposent les agents leur permettent de percevoir directement ou indirectement la valeur de l'état courant de certains attributs. La position d'un objet ou la couleur d'une balle peuvent être perçues directement par la vision. La température de la théière peut être estimée directement par le toucher si l'agent dispose de récepteur de chaleur ou indirectement par la vue de buée sur la théière ou de vapeur. Le poids de la brique de lait peut être directement estimée en déplaçant l'objet si l'agent dispose de capteur d'effort ou déduit de l'observation du fait que la brique n'a pas été encore ouverte. La nature du contenu d'un verre peut être estimée en partie par l'aspect visuel du liquide, mais aussi par son "goût" qui est une notion dépendant fortement de l'agent.

La capacité à percevoir ou inférer la non-validité de la croyance sur état courant. Parfois, il est possible d'infirmer la croyance courante sur un état du monde sans pour autant parvenir à percevoir la nouvelle valeur. C'est le cas pour la position d'un objet lorsque l'agent ne perçoit plus l'objet là où l'agent croit qu'il était. Cette notion est plus difficilement généralisable aux autres attributs. En effet le plus souvent la mesure du fait que l'attribut n'a plus une valeur donnée est déduite de la mesure de la nouvelle valeur. La balle n'est pas jaune parce qu'elle est orange. La brique de lait n'est pas lourde (pleine) parce qu'on la soupèse et qu'elle pèse peu. Pour que cette notion est un sens, il faut d'abord que l'on perçoive un non-état et non un état différent, il faut aussi que l'espace des états atteignables soient suffisamment grand pour que le fait qu'on ne soit pas dans un état donné ne permettent pas d'en déduire directement l'état courant.

La capacité à percevoir, inférer l'action qui fait évoluer la propriété. Bien souvent, le nouvel état n'est pas du tout ou du moins n'est pas aisément perceptible alors que la séquence d'actions dont il résulte est elle-même perceptible. Le robot doit donc avoir un modèle de sa propre capacité et de celle des autres agents à percevoir et à interpréter les actions qui changent ces attributs. Ces séquences d'actions sont de complexité très variables. Un pick & place sur une table seront assez facilement perçus en terme de mouvement du bras et du corps. Aller faire chauffer la théière, verser le contenu de la brique de lait dans un bol, remplir le verre à partir d'une brique de jus d'orange ou d'une théière sont des activités plus étalées dans le temps et dont la reconnaissance nécessite l'identification d'autres objets et de lieux de manière précise.

L'hypothèse initiale. Le robot doit être en mesure d'initialiser la valeur de sa croyance et de celle des autres agents. Pour cela il doit se fonder sur l'historique récent ou sur le sens commun qui est le plus souvent une distribution initiale des probabilités d'être dans certains états compte tenu d'un contexte donné. La position d'un objet peut être par exemple initialisée à la position où on l'avait observé pour la dernière fois. Pour les autres attributs, on utilise le sens commun. Le beurre sera sur la table de la salle à manger pendant le petit déjeuner et dans le réfrigérateur sinon. Une balle de tennis sera a priori jaune et une balle de ping pong blanche. La théière sera très certainement froide en milieu de nuit alors qu'elle peut être chaude pendant le petit déjeuner. Le brique de lait sera pleine et lourde dans le placard et en partie vide dans le réfrigérateur. L'hypothèse initiale sur le contenu du verre dépendra fortement du type de verre (verre à vin, flûte de champagne, tasse de café) du lieu (grand restaurant le soir, salle à manger au moment du petit déjeuner).

Les croyances des agents sur les attributs d'objets dans le monde reposent sur le raisonnement du robot sur les différents états atteignables, la perception de l'état actuel, la perception de l'action qui a conduit à cet état et les hypothèses initiales sur cet état. Pour la localisation, toutes ces notions sont relativement claires et explicites, liées à la géométrie et la vision. Il s'agit de la position dans l'espace, de la vision, du déplacement des objets et de la position la plus probable compte tenu du contexte. On peut généraliser à d'autres attributs. Néanmoins cela nécessite des capacités de perception plus variées et spécialisées et des capacités de perception et d'interprétation d'actions plus complexes.

Chapitre 8

HATP pour la gestion des croyances

La description du monde telle qu'elle a été précédemment présentée dans la section 4.2 suppose que l'état courant est entièrement connu et que tous les agents partagent la même vision du monde. Pour les applications réelles, et plus particulièrement pour les problèmes d'interaction homme-Robot, ces assertions peuvent mener à des solutions impossibles ou à des plans incompréhensibles pour le partenaire humain.

Afin d'éviter ce genre de problèmes, nous proposons d'étendre cette représentation en ajoutant, d'une part, la possibilité de modéliser des croyances différentes pour chaque agent et, d'autre part, la possibilité de considérer qu'un agent connaît ou ne connaît pas certaines informations.

8.1 Modélisation des états de croyances

Dans nos expérimentations d'interaction homme-robot, la solution à un but donné est entièrement calculée par le robot. La base de faits doit modéliser l'état du monde du point de vue du robot ainsi que les croyances du robot sur les croyances de l'homme (et plus généralement, des autres agents). Nous avons expliqué dans le chapitre 7 comment le robots pouvaient déterminer si les croyances de l'homme sont distinctes des siennes.

8.1.1 L'agent myself

Pour spécifier quel agent est le robot, *i.e.,* l'agent pour lequel le système planifie, nous utilisons le mot-clé **myself** au lieu de déclarer un nouvel agent. Par exemple, si JIDO est le robot et ACHILE est un humain, l'initialisation

des agents sera :

```
JIDO = myself;
ACHILE = new Agent;
```

8.1.2 Représentation des croyances

Afin de modéliser des croyances différentes pour les agents, le formalisme utilisé par HATP est étendu en utilisant la notion de variable d'état à valeurs multiples (MVSV). Une variable d'état à valeurs multiples V est instanciée à partir d'un domaine Dom et pour chaque agent $a \in A$ la variable V a une instance $V(a) \in Dom$. Par exemple, si les agents ont une croyance différente de la position de l'objet WHITE_TAPE :

```
WHITE_TAPE(JIDO).location = PINK_TRASHBIN;
WHITE_TAPE(ACHILE).location = BLUE_TRASHBIN;
```

Par défaut, afin de clarifier le domaine de planification, seuls les attributs des entités pour lesquelles les agents ont une croyance divergente sont modélisés en utilisant le formalisme MVSV.

8.1.3 Informations connues et inconnues

Le modèle de croyances des agents inclut les notions d'information connue et d'information inconnue. Lorsqu'un agent n'a pas d'information sur la valeur d'un attribut, la variable associée reçoit l'instance *unknown*. Lorsqu'un agent différent de l'agent **myself** connaît une information qui est inconnue du robot, la valeur de l'attribut correspondant est *known*.

La représentation précédente des croyances des agents est augmentée afin de prendre en compte ces valeurs spécifiques :

$$V(a) \in \begin{cases} Dom_v \sqcup \{unknown\} & \text{si} \quad a = \text{myself} \\ Dom_v \sqcup \{unknown\} \sqcup \{known\} & \text{autrement} \end{cases}$$

Par exemple, si l'homme ne connaît pas la position de l'objet WHITE_TAPE :

```
WHITE_TAPE(JIDO).location = PINK_TRASHBIN;
WHITE_TAPE(ACHILE).location = unknown;
```

Lorsqu'un agent n'a aucune information sur un objet, tous les attributs de l'entité correspondante devraient prendre la valeur *unknown*. Nous simplifions cette représentation par :

```
WHITE_TAPE(ACHILE) = unknown;
```

8.1.4 Consistance des croyances

Pour être consistante, une variable d'état V requiert que l'union des croyances des agents sur la propriété associée forme un ensemble de dimension 1. C'est-à-dire, une croyance est consistante si l'ensemble des agents ont la même croyance sur la propriété du monde concernée.

$$\| \bigcup_{\forall a \in Ag} V(a) \| = 1$$

Cette formule est utilisée durant le processus de planification afin d'assurer que le plan est correct du point de vue des croyances des agents. Ainsi, à la fin de l'exécution du plan, les agents doivent avoir les mêmes croyances sur les objets manipulés.

8.2 Mise à jour des croyances et communication

En planification classique, une action est définie par un ensemble de préconditions représentant les conditions nécessaires à sa réalisation et, un ensemble d'effets modélisant les changements du monde résultant de l'exécution de l'action.

Planifier pour plusieurs agents qui ont leur propres croyances soulève un ensemble de questions :

- Les croyances de quel agent doit utiliser le système, surtout dans le cas d'une action jointe ?
- Les croyances des agents doivent-elles être consistantes avant l'exécution d'une action ? comment ?
- Comment évoluent les croyances des agents impliqués dans la réalisation d'une action jointe ?
- Comment évoluent les croyances des autres agents ?

8.2.1 Croyances et préconditions des actions

Afin de réaliser une action, l'acteur doit avoir la "bonne" croyance sur les objets manipulés durant cette action. Si l'acteur a une croyance divergente de l'agent principal (l'agent **myself**), le planificateur produit des actions de communication entre l'agent principal (l'agent **myself**) et les autres participants à l'action pour rétablir la consistance entre les différentes croyances.

8.2.2 Croyances et effets des actions

Comme les croyances des agents sur les objets manipulés doivent être consistantes avant d'exécuter correctement une action, les effets des actions sont appliqués sur les croyances de tous les participants. C'est-à-dire que les croyances des agents sur les objets manipulés restent consistantes après l'exécution de l'action. Concernant les croyances des agents ne participant pas à l'action mais pouvant être présents dans la scène, la mise à jour de leurs croyances n'est pas automatique et est de la responsabilité du concepteur du domaine de planification.

8.2.3 Actions de communication

Une action de communication est une action particulière qui a comme paramètres deux agents, l'émetteur et le récepteur, et un sujet qui est représenté par une entité et un attribut. Ainsi, le prototype d'une action de communication est le suivant :

```
commAction name(Agent A, Agent B, Entity E, Attribute T){
        preconditions {...};
        effects {...};
        cost{...};
        duration{...};
}
```

Le but d'une action de communication est de transmettre la valeur d'un attribut d'une entité de l'agent émetteur à l'agent récepteur. Ce qui correspond à l'effet suivant :

```
E(B).T = E(A).T;
```

Cet effet est implicite pour ce type d'action, *i.e.,* le concepteur du domaine de planification n'a pas besoin de le spécifier pour chaque action de communication.

De la même façon qu'une action classique, une action de communication est définie par un ensemble de préconditions exprimant les conditions nécessaires à la communication (*e.g.,* les agents doivent être dans la même pièce), et un ensemble d'effets additionnels à l'effet implicite résultant de la communication. Avec ces effets, il est par exemple possible de modéliser le concept de co-présence, *i.e.,* la communication n'affecte pas seulement les croyances de l'agent récepteur mais également les croyances de tous les agents présents aux alentours.

Afin d'être cohérent avec la modélisation du domaine, pour transmettre l'ensemble des informations concernant une entité de l'agent A à l'agent B,

myself	Agent_B	type de communication
v	*unknown*	information
v	v'	contradiction
unknown	*known*	question

TAB. 8.1 – Types de communication en fonction des croyances

le paramètre attribut peut prendre la valeur *null*. Dans tous les autres cas, seule la valeur de l'attribut spécifié est transmise.

8.2.4 Types de communication

En fonction des croyances des agents, les actes de communication ne seront pas traités de la même manière au moment de l'exécution. Nous choisissons de faire cette distinction dès la planification par la définition de trois types d'actions de communication : *information*, *contradiction* et *question*.

Les actions de communication de type *information* ont pour objectif de transmettre une information de l'agent **myself** à un autre agent lorsque celui-ci ne connaît pas cette information, *i.e.,* la valeur de l'attribut correspondant est *unknown* dans sa base de croyances.

Quand la valeur associée à un attribut pour un agent est différente de la valeur du même attribut pour l'agent **myself**, le planificateur produit une action de communication de type *contradiction*.

Le type *question* est utilisé lorsque l'agent **myself** ne connaît pas une donnée et qu'il existe un autre agent qui a cette connaissance (modélisée par la valeur *known*).

Le tableau 8.1 résume ces différents types de communication en fonction des croyances d'un agent comparées aux croyances de l'agent **myself**.

8.3 Mise en œuvre et adaptation du planificateur

Pour pouvoir gérer les croyances des agents, l'algorithme de planification doit être adapté afin de prendre en compte le nouveau formalisme pour la modélisation du domaine ainsi que les actions de communication.

8.3.1 Bases de faits

Afin de stocker les croyances des différents agents, nous créons une base de faits pour chaque agent. Durant la phase d'initialisation, les entités re-

présentant les agents et objets présents dans l'environnement sont créées et stockées dans la base de faits de l'agent **myself**. Pour les autres agents, afin de limiter l'espace mémoire utilisé, nous décidons de stocker dans les bases de faits additionnelles uniquement les valeurs qui sont inconsistantes avec les croyances de l'agent principal.

8.3.2 Méthode générale de communication

Pour permettre au concepteur du domaine de planification de nommer les actions de communication de la même manière qu'il le ferait pour les actions classiques, celles-ci sont liées aux concepts *information*, *contradiction* et *question* au travers d'une méthode générale de communication appelée **beliefManagement**.

```
beliefManagement {
        information { GiveInformationAbout; };
        contradiction { ForceInformation; };
        question { AskForInformation; };
}
```

Dans cet exemple, le type de communication *information* se réfère à l'action de communication appelée GiveInformationAbout par le concepteur du domaine. Chaque action de communication doit être définie précédemment dans le domaine de planification.

Chapitre 9

Utilisation des croyances divergentes

Comme énoncé précédemment, la gestion des croyances distinctes est très utile pour mieux comprendre ce que l'homme fait, ce qu'il dit et sur quoi il porte son attention. Concernant la planification de tâches, cela permet au robot de planifier des actes de communication permettant de corriger une croyance absente ou erronée de l'homme si et quand elle impacte sur la réalisation du plan. Dans ce chapitre je décris d'abord rapidement comment le robot utilise les croyances distinctes pour mieux comprendre ce que dit l'homme. Puis je développe plus longuement nos premières expérimentations où le robot tient compte des croyances de l'homme pour mieux planifier. L'url ci-dessous permet de visualiser quelques vidéos de ces expérimentations. [1]. La construction et l'utilisation des croyances divergentes ont été présentées dans la publication suivante [62].

9.1 Amélioration de l'interprétation du dialogue

Le robot peut utiliser son raisonnement sur la croyance des autres agents pour répondre à des questions ou informer l'homme de manière proactive, Dans la section §7.1 j'ai présenté deux scénarios pour illustrer les concepts de croyances distinctes et l'absence de croyance. À la fin du second scénario, l'état du monde du robot (ROBOT) est illustré par la fig. 7.2(c). Patrick (PATRICK) ne sait pas où est la boîte noire (B_BOX) mais le robot s'aperçoit qu'il voit la boîte blanche (W_BOX). Nous présentons un extrait des

[1] `http://homepages.laas.fr/mwarnier/Roman2012.html`

deux modèles symboliques du robot et de Patrick ci-dessous.

```
          ROBOT
W_BOX isVisibleBy ROBOT
W_BOX notVisibleBy PATRICK
W_BOX isOn TABLE
B_BOX isVisibleBy ROBOT
B_BOX isVisibleBy PATRICK
B_BOX isOn TABLE
B_BOX isNextTo PINK_TRASHBIN
```

```
          PATRICK
W_BOX isVisibleBy ROBOT
W_BOX notVisibleBy PATRICK
W_BOX isOn TABLE
B_BOX hasKnownLocation false
```

Patrick demande au robot : "Où est la boîte ?" Le robot peut facilement comprendre que Patrick parle de la boîte noire (B_BOX) car c'est la seule dont il ne connaît pas la position. Le robot peut répondre : "Elle est sur la table, proche de la corbeille rose". Cela enrichit l'algorithme de catégorisation proposé par Ros en 2010 dans [46].

À la fin du premier scénario, l'état du monde du robot (ROBOT) correspond à la figure 7.1(c). Patrick (PATRICK) a une croyance fausse sur la position de la boîte blanche (W_BOX). Patrick essaye ensuite de regarder derrière la boîte grise là où était la boîte blanche initialement. Nous présentons ci-dessous un extrait des deux modèles symboliques du robot et de Patrick dans ce nouvel état.

```
          ROBOT
W_BOX isNextTo PINK_TRASHBIN
W_BOX isOn TABLE

B_BOX isOn TABLE
B_BOX isNextTo BOX
```

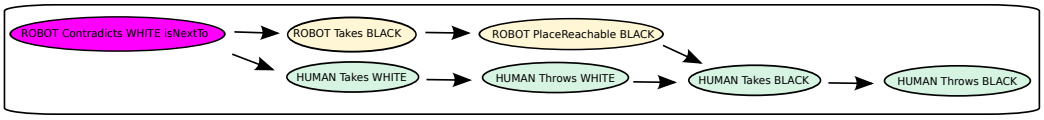
```
          PATRICK
W_BOX isNextTo BOX
W_BOX isOn TABLE
W_BOX isLookedAtBy PATRICK
B_BOX isOn TABLE
B_BOX isNextTo BOX
```

Dans le modèle de Patrick, il regarde en direction de la boîte blanche (W_BOX). Le robot peu en déduire que l'humain cherche peut-être la boîte blanche (W_BOX). Le robot peut dire de manière proactive : "L'objet que vous cherchez est proche de la corbeille rose"

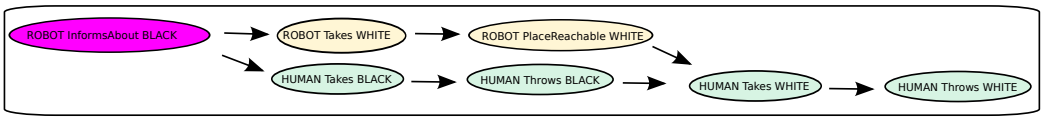
9.2 Utilisation du planificateur avec croyances divergentes

De nouveau, j'utilise les deux scénarios présentés dans la section §7.1. Patrick et le robot doivent nettoyer la table ensemble. La figure 9.1(a) donne le plan produit à partir de l'état du monde visible dans la figure 7.1(c). La première action du plan est une action de communication. Le robot contredit Patrick sur sa croyance sur la position de la boîte blanche. Cela met à jour la croyance de Patrick sur cette position. Patrick va donc prendre la boîte

blanche là où elle est effectivement alors qu'il aurait d'abord essayé de l'attraper là où il pensait initialement qu'elle était, derrière la grosse boîte grise. La figure 9.1(b) donne le plan produit à partir de l'état du monde présenté dans la figure 7.2(c). La première action du plan est également une action de communication. Le robot informe Patrick de la position de la boîte noire. Cela met à jour la croyance de Patrick sur cette position. Patrick peut alors attraper la boîte noire sans hésitation. Sans la première action du robot, Patrick n'aurait pas su directement où saisir l'objet. Il y a deux zones de la table qui lui sont invisibles. On peut penser qu'il en aurait choisi au hasard l'une des deux. Il aurait donc commencé à réaliser la mauvaise action une fois sur deux.



(a) Plan avec croyance divergente



(b) Plan avec absence de croyance

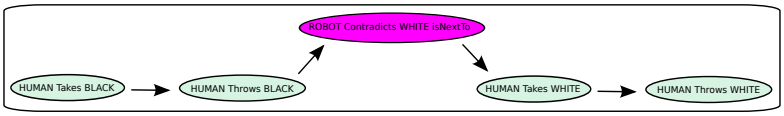
FIG. 9.1 – Plans produits

Dans la dernière expérimentation, on illustre la capacité du contrôleur de haut niveau du robot à adapter l'exécution du plan selon l'évolution de l'état du monde. Dans la figure 9.2(a), Patrick pense que la boîte blanche est toujours proche de la corbeille rose. La figure 9.3(a) montre le plan produit. L'action de communication est insérée uniquement quand c'est nécessaire. C'est-à-dire avant la première action de l'homme sur la boîte blanche. La figure 9.2(b) montre comment l'homme découvre la nouvelle position de la boîte blanche après sa première action. Cela a pour effet de mettre à jour la position de la boîte blanche dans le modèle de l'homme. Du coup, l'action de communication n'est plus nécessaire. La figure 9.3(b) montre le plan qui est en fait exécuté. On saute l'action de communication du fait que ses préconditions ne sont pas respectées alors que ses effets le sont.

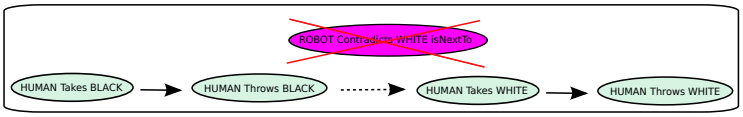
Ces premières expérimentations démontrent que le système est fonctionnel. Nous croyons fermement que ce nouveau raisonnement est utile. Au cours de ces expérimentations, nous avons pu parfois nous apercevoir que lorsque le robot échouait dans sa détection des croyances divergentes, l'homme ne savait pas directement comment agir. Cependant, nous n'avons pas évalué de manière rigoureuse et quantitative l'impact de cette nouvelle fonction-

(a) État du monde avant la planification (b) L'homme regarde en direction de l'objet après sa première action

FIG. 9.2 – États du monde avant et pendant l'observation de la nouvelle position



(a) Plan produit



(b) Plan exécuté

FIG. 9.3 – Saut de l'action de communication

nalité. Nous souhaitons réaliser des études utilisateurs dans lesquelles nous comparerions la performance avec ou sans la gestion des croyances distinctes.

Quatrième partie

Quelques perspectives

Chapitre 10

Perspectives d'amélioration et d'évaluation du système

Dans cette partie nous souhaitons présenter un certain nombre de réflexions que nous avons menées dans l'idée d'améliorer et d'évaluer le système existant. Pour l'essentiel, ces réflexions portent sur l'amélioration de la construction et de la mise à jour de l'état du monde. J'aborde aussi vers la fin du chapitre la gestion de la planification de mouvement. Ces propositions répondent à une exigence d'amélioration de la rapidité et de la précision. On cherche à construire et mettre à jour l'état du monde plus rapidement et de manière plus fidèle à la réalité par rapport à ce que fait le système aujourd'hui. De même on voudrait mieux utiliser la planification de mouvement qui ralentit le système actuellement. Pour cela on propose d'anticiper autant que possible la planification pour pouvoir exécuter au plus vite le mouvement quand il doit être réalisé. Ces propositions n'ont pas donné lieu à une implémentation concrète sur le robot et n'ont pas non plus été poussées très loin d'un point de vue théorique faute de temps. Néanmoins elles apparaissent comme une perspective intéressante pour le futur. Ce chapitre se clotûre par une section qui ouvre vers de nouvelles expérimentations qui nous permettraient d'évaluer et critiquer le fonctionnement du système.

10.1 Optimisation et rationalisation des calculs géométriques

La sous-partie 3.1 explicite la synthèse de la représentation du monde géométrique cartésien sous la forme de faits symboliques. Si le monde géométrique cartésien évolue, alors sa représentation simplifiée sous la forme de faits symboliques doit également évoluer. En l'état actuel du système, ces

calculs et mises à jour sont très loin d'être optimaux tant au niveau du calcul de chaque fait que dans les règles de mise à jour des faits.

Calcul d'un fait

Un fait est calculé sur la seule base de la géométrie et ce quel que soit l'état symbolique courant de ce fait. Par exemple, même si un objet est actuellement dans la main de l'homme ou de la pince, le fait qu'il soit atteignable par l'agent en question sera tout de même calculé géométriquement par une distance ou pire au moyen de la cinématique inverse. Il est impératif de moduler le calcul du fait selon l'état symbolique des entités incriminées. Si un objet est dans une main ou une pince alors il peut être directement considéré comme atteignable sans besoin de calculs géométriques supplémentaires. De manière plus générale, tout raisonnement logique permettant d'établir un fait plus rapidement que le calcul géométrique serait à privilégier.

Décision de mettre à jour le fait.

Tout déplacement significatif d'un objet ou d'un agent est interprété comme un changement de l'état géométrique cartésien du monde. Aujourd'hui, cela entraîne la mise à jour de tous les faits sans exception. Imaginons que notre environnement soit un bâtiment avec 100 pièces et que dans chacune de ces pièces il y ait une table avec des objets dessus. Le déplacement d'un objet dans une des pièces entraînerait entre autres le recalcul de la relation isOn de chaque objet sur chaque table. C'est bien évidemment absurde. Il faudrait être en mesure de pouvoir estimer et circonscrire l'influence d'un déplacement d'un agent ou d'un objet au moins de manière approchée. Quels sont les faits qui doivent être recalculés suite à une modification de l'état géométrique cartésien du monde ? Une hiérarchisation ou au moins une segmentation spatiale sous la forme, par exemple de salle, serait déjà une première étape.

Aujourd'hui le faible nombre d'objets et d'agents impliqués ne nous a pas obligés à considérer cette problématique.

10.2 Réflexion sur les états et la discrétisation de l'espace

En l'état actuel du système, la plupart des faits symboliques sont générés par le calcul à partir de la représentation cartésienne de la géométrie. Pour reprendre la terminologie de la figure 2.1, les processus de types mis à jour du niveau 2 depuis le niveau 1 sont bien plus présents que les processus en sens inverse. Seul l'instanciation géométrique à partir d'hypothèse

de positionnement explicitée dans la partie donne un exemple de processus allant en sens inverse. Dans cette sous-partie nous souhaitons introduire plus de richesse dans la représentation du niveau géométrique symbolique et les possibles répercussions sur la représentation cartésienne.

Nous introduisons tout d'abord la notion de zone géographique symbolique discrète. Nous prenons l'exemple d'une boîte grise GREY_TAPE qui peut être attrapée et déplacée par un robot JIDOKUKA_ROBOT qui dispose d'une pince RH et un homme HERAKLES_HUMAN qui a deux mains HHl (gauche) et HHr (droite). La table HRP2_TABLE, un conteneur ouvert en haut de type boîte ouverte par le dessus PINK_TRASHBIN et un conteneur de type boîte à l'envers ouverte en bas SURPRISE_BOX. Ces deux types de conteneurs sont très différents du point de vue de la manipulation et de la perception. En effet, l'objet doit être posé sur un autre support s'il est recouvert par une boîte à l'envers. Il ne peut pas être manipulé ni perçu tant que la boîte le recouvre. À l'inverse l'objet qui est dans la boîte ouverte par le dessus repose sur le fond de la boîte et peut être perçu et manipulé même si cela peut être difficile. On a également deux sets de table qui définissent des zones précises sur la table PLACEMAT_BLUE et PLACEMAT_GREEN. Dans la partie gauche de la figure 10.1 on peut voir une photographie d'un tel scénario. La partie droite de la figure propose une discrétisation possible. Il y a les deux mains de l'homme HHr et HHl, la pince du robot RH, l'intérieur de la poubelle PINK_TRASHBIN, la partie de la table recouverte par la cloche HRP2_TABLEh*SURPRISE_BOX et toutes les zones correspondant à une portion de la table non recouverte par un conteneur ou une cloche indexées sous la forme de HRP2_TABLES[a,b,c,d,e,f,g,i,j,k,l,m].

Ces zones permettent d'introduire de nouveaux raisonnements par rapport à une modélisation uniquement basée objet. L'exploration peut être rationalisée par l'observation successive de ces zones en commençant par celles les plus facilement observables. Un certain nombre de prédicats pourraient également être calculés pour ces zones isOn, nextTo, isVisible, isReachable. Tout objet se situant dans une de ces zones se verrait alors attribué directement les prédicats correspondants. À l'inverse, un objet non observé pourrait être positionné dans les zones non observables ce qui permettrait de déterminer un certain nombre de prédicats le concernant même si l'on n'est pas sûr de sa position exacte.

Les dimensions des zones considérées peuvent dépendre de l'objectif de la représentation. Si le but est l'exploration, on dimensionnera de telle sorte que les objets considérés soient perçus s'ils sont dans la zone, que l'objet peut être perçu et que le robot regarde au centre de la zone. Plus les caméras ont un petit angle, plus la discrétisation devra être précise. Il peut être intéressant de qualifier la probabilité de perception effective d'un objet s'il

est dans la zone que le robot observe. Cela sera développé dans la sous-partie suivante où l'on s'intéresse à l'introduction de raisonnement probabiliste. Si le but est la spatialisation d'un objet, on considérera une dimension relative à l'échantillonnage voulu.

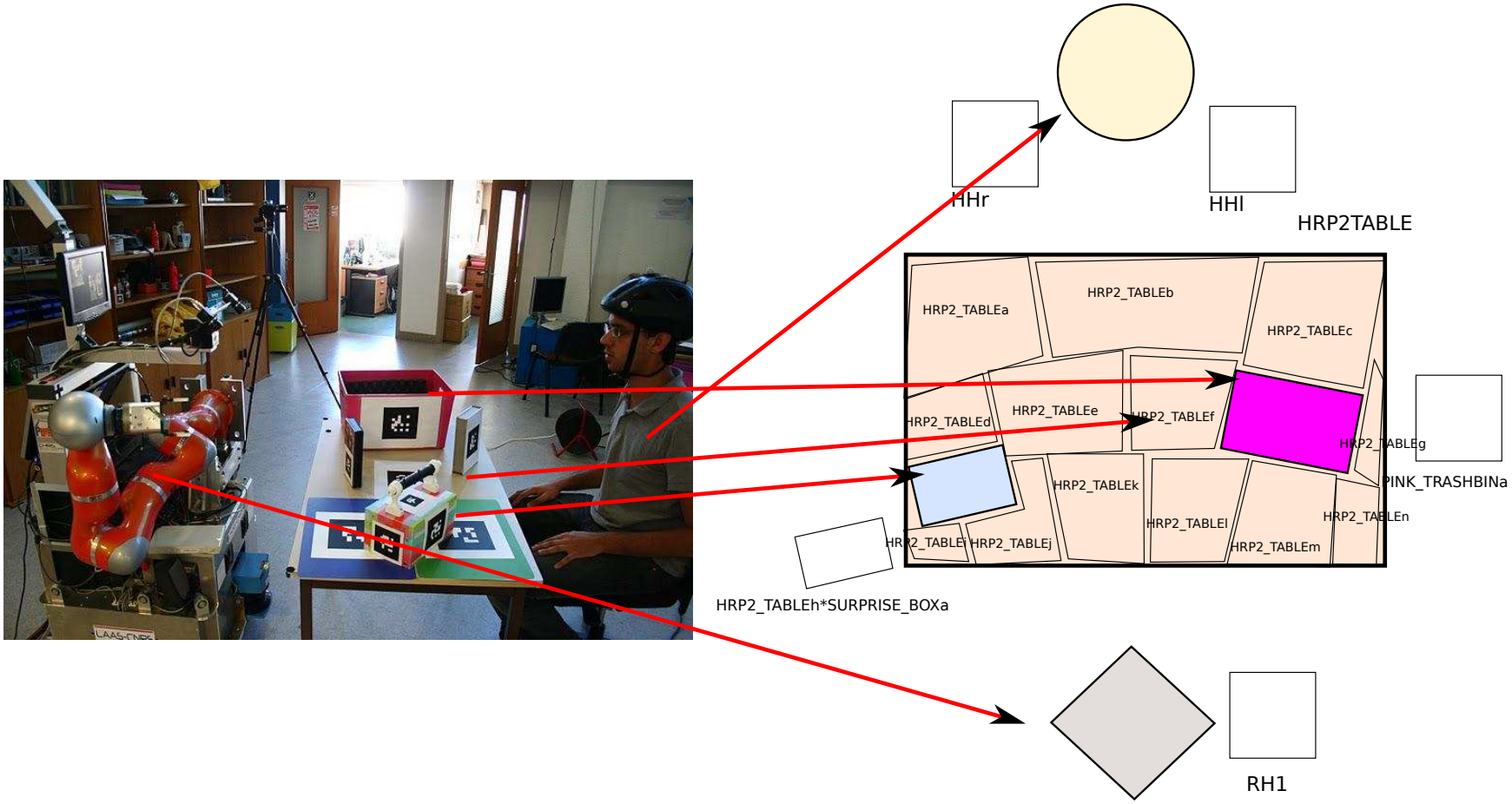


FIG. 10.1 – *
Discrétisation de l'espace

La figure 10.2 poursuit dans le sens de la figure précédente. On présente de manière exhaustive l'ensemble des états atteignables sous la forme d'un "espace vectoriel" adapté au contexte. Par simplicité on discrétise la table uniquement au moyen des deux sets de table et du reste de la table. Les transitions possibles entre états sont explicitées. Pour chaque état on précise également les processus de raisonnements à mettre en oeuvre ainsi que la qualité à priori de la perception.

Explicitons d'abord l'*espace vectoriel* en question. Nous définissons deux dimensions par rapport à notre contexte applicatif. Il s'agit d'une simplification qu'on pourrait assez facilement remettre en cause dans certains cas précis.

- Support : Sur quoi repose l'objet s'il s'agit d'un objet manipulable ? Il peut s'agir d'un support plat ou un support avec des bords de type conteneur ouvert en haut. A priori, on considère que l'objet ne repose sur qu'un support à la fois.
- Saisie, englobage : Qu'est-ce qui entoure l'objet. Avec contact s'il s'agit de la saisie par une main ou une pince de robot. Sans contact s'il s'agit d'une boîte englobant l'objet. Un objet peut être saisi par plusieurs mains ou pinces en même temps dans la limite d'existence de prises multiples. Par contre l'objet ne peut pas être à la fois saisi et englobé dans une boîte.

Dans notre exemple il y a trois valeurs possibles pour le support :

- vide : l'objet ne repose sur rien.
- HRP2TABLE : l'objet repose sur la table. À noter que dans ce cas-là on a une subdivision possible en zone comme indiquée dans la figure 10.1 ou une subdivision plus simple au moyen des sets de table PLACEMAT_BLUE et PLACEMAT_GREEN comme visible dans le schéma de la figure 10.2.
- PINK_TRASHBIN : l'objet est dans la poubelle. Il repose donc sur le fond de la poubelle.

Dans notre exemple il y a trois types de valeurs possibles pour la dimension saisie / "englobage" :

- vide : l'objet n'est ni saisi ni englobé par une boîte.
- SURPRISE_BOX : l'objet est recouvert par la boîte englobante à fond creux SURPRISE_BOX.
- saisie : l'objet est saisi par au moins une des trois mains ou pinces sui-

vantes RH, HHr, HHl.

La figure 10.2 représente de manière schématique les différentes configurations au moyen de ces deux dimensions. Les deux schémas respectivement en haut à gauche et en haut au milieu permettent d'introduire les codes couleurs et formes utilisés pour représenter les objets et les mains ou les pinces des agents. Le schéma en haut à droite représente la légende des processus mis en oeuvre dans chaque état. Les pentagones roses correspondent à l'ajout d'une correction sur une position perçue. On fait l'hypothèse que la perception est biaisée et qu'il faut donc légèrement corriger par exemple en déplaçant légèrement cet objet qui semble flotter dans l'aire de manière à ce qu'il soit en fait sur la table. Les étoiles vertes correspondent à l'estimation de la validité d'une telle correction : compte tenu de la nouvelle position perçue, est-il toujours raisonnable d'appliquer cette correction. Les pentagones turquoises correspondent aux hypothèses de positionnement explicités dans la partie . Il s'agit de positionner géométriquement un objet par rapport à un état géométrique symbolique donné dans lequel l'objet ne peut généralement pas être perçu, par exemple si l'objet est dans la poubelle ou dans la main de l'homme. Les étoiles jaunes correspondent au processus d'estimation de la validité de l'hypothèse en cas de perception de l'objet. Le rectangle gris avec un point d'exclamation et un signe pourcentage correspond à l'application du calcul géométrique des faits. Les yeux ouverts, ouverts à moitié ou fermés indiquent l'estimation a priori de la capacité à percevoir un objet dans un état donné.

Le reste de la figure permet de représenter les différentes classes d'états valides ou au contraire impossibles. Voici les quatre classes d'états atteignables :

Free on flat Support : L'objet est libre (non saisi et non recouvert par une boîte) sur un support plat (Typiquement une table). Cet état est obtenu par calcul géométrique sur la position perçue soit par l'application d'une correction sur cette position perçue dont la validité doit être alors continument estimée. L'observabilité dépend de la position de l'objet sur le support.

Under covering Box : L'objet est sur un support plat et est recouvert par une boîte englobante à fond vide. L'objet n'est pas perceptible. Cet état est calculé géométriquement si l'on connaît la position des objets ou il est instancié géométriquement si l'on suppose que l'état est atteint sans connaître la position exacte de l'objet. La validité de l'hypothèse de positionnement doit alors être estimée en cas de perception.

Free in supporting Container : L'objet est au fond d'un conteneur. L'observabilité dépend de la dimension du conteneur et de la position de l'agent. Cet état est calculé géométriquement si l'on perçoit la position des objets ou il est instancié géométriquement si l'on suppose que l'état est atteint sans connaître la position exacte de l'objet. La validité de l'hypothèse de positionnement doit alors être estimée en cas de perception.

In Hands or Gripper : L'objet est dans une ou plusieurs mains ou pinces. L'observabilité est partielle voir impossible en l'état du système. Cet état est instancié géométriquement. La validité de l'hypothèse de positionnement doit alors être estimée en cas de perception.

Les états valides sont reliés entre eux par des actions.

Voici les quatre classes d'états non valides :

Objet flottant : Un objet doit être saisi s'il n'est pas sur un support.

Objet flottant recouvert par une boîte à fond plat : Un objet ne peut être englobé par une boîte recouvrante sans fond s'il ne repose pas sur un support.

Objet dans une boîte recouvert par une boîte englobante à fond creux : Un objet ne peut être à la fois dans un conteneur à fond et sous une boite sans fond.

Objet saisi et recouvert par une boîte englobante à fond creux : Un objet ne peut être à la fois saisi et englobé par une boîte à fond creux.

Pour chacun de ces ensembles d'états, il faut ajouter ou supprimer des hypothèses ou des corrections pour atteindre l'état acceptable le plus proche.

Schematic representation of reachable states, allowed transitions and geometric processes.

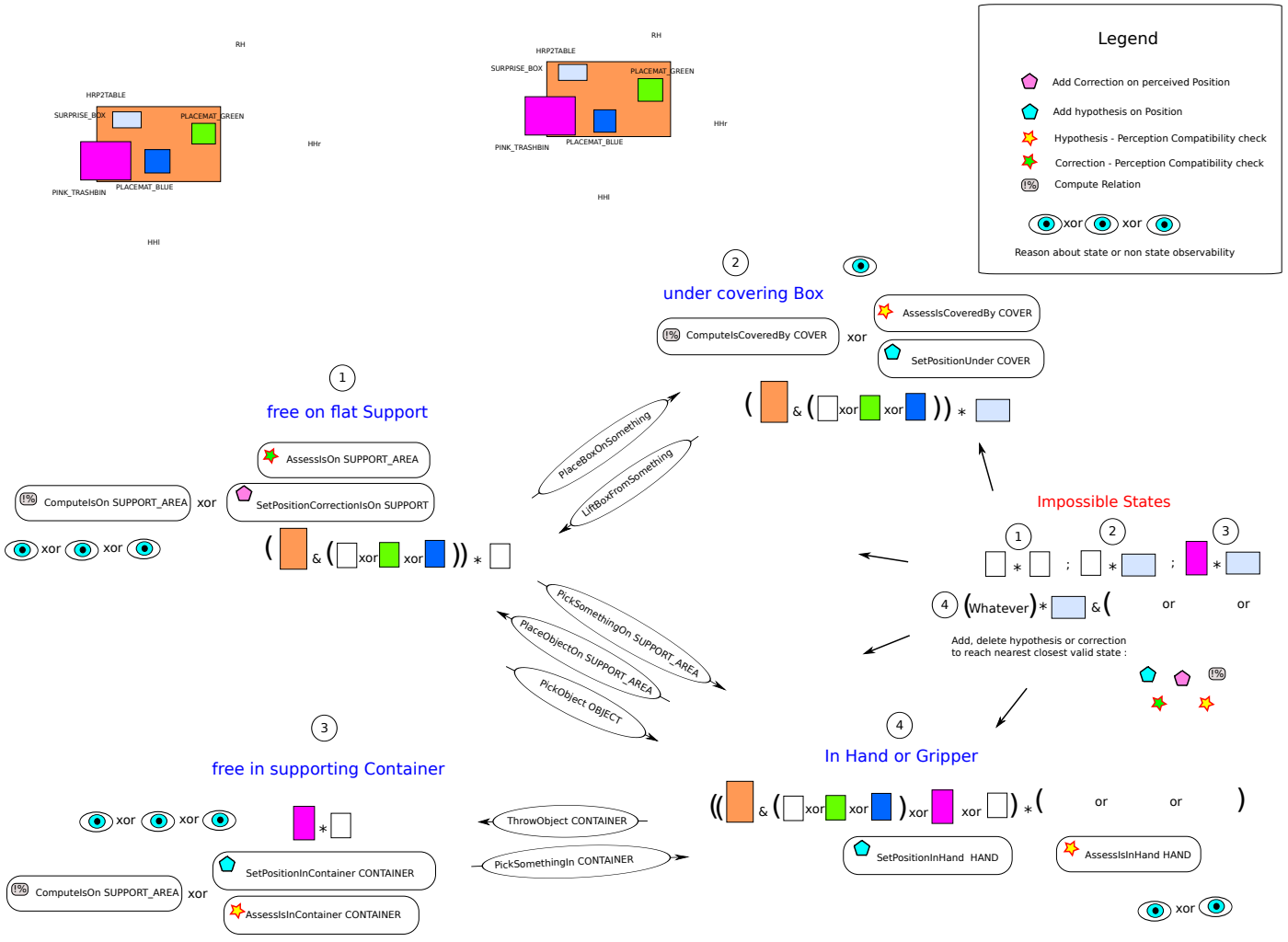


FIG. 10.2 – *
États atteignables, transitions, processus

Tout ceci permet la "vibration" entre les deux niveaux de représentation de la géométrie. Si la position géométrique courante se traduit par un état géométrique symbolique impossible, on choisira l'état atteignable qui se traduit par une correction géométrique minimum. Inversement, l'instanciation géométrique cartésienne de l'état symbolique géométrique courant peut permettre de remettre en cause l'état symbolique courant si elle est radicalement incompatible avec la position géométrique perçue.

10.3 Utilisation d'un cadre probabiliste

Je présente une brève introduction à l'utilisation d'un cadre probabiliste et très rapidement et très schématiquement deux exemples d'application possible de la modélisation probabiliste des états et de la perception des positions et des actions à nos scénarios. Notre contexte explicatif précis nous permet de proposer une modélisation des états et des actions. Notre savoir faire en terme de raisonnements géométriques devrait nous permet de moduler l'estimation de l'observation des états et des actions compte tenu de l'état du monde.

10.3.1 Introduction générale et bibliographie

Dans [58] Sebastian Thrun défend l'importance de l'application des méthodes probabilistes à la robotique. Il utilise même le terme de robotique probabiliste pour regrouper ces différentes approches. Ces méthodes probabilistes s'appliquent particulièrement à la perception et au contrôle. L'essor des méthodes probabilistes a été particulièrement important pour des applications telles que le S.L.A.M. et le filtrage.

La perception probabiliste. Les robots ne connaissent pas l'état de leur environnement avec certitude. Cette incertitude est liée aux limitations des capteurs, au bruit et au fait que les environnements les plus intéressants sont dans une certaine mesure imprévisibles. Dans l'approche probabiliste, les données capteurs sont interprétées comme une distribution de probabilité sur les états du monde possibles au lieu de ne considérer qu'un état le plus probable. En conséquence, un robot probabiliste peut se reprendre de manière efficace de ses erreurs, prendre en compte les ambiguïtés et intégrer les données capteurs de manière consistante. De plus le robot probabiliste estime sa propre ignorance.

Le contrôle probabiliste. Les robots autonomes doivent agir malgré les incertitudes du fait de leur incapacité à déterminer l'état du monde de manière certaine. Dans les approches probabilistes la prise de décision tient compte de l'incertitude du robot. Certaines approches considèrent uniquement l'incertitude courante alors que d'autres anticipent également les incertitudes futures. Au lieu de prendre uniquement en compte les situations (courantes ou futures) les plus probables, de nombreuses approches cherchent à calculer un optimum en terme de théorie de la décision. Les décisions optimales tiennent alors compte de toutes les contingences possibles. C'est le cas des processus de décision markovien partiellement observable (POMDP en anglais). Simmons dans [52] et Cassandra [12] les appliquent à la navigation d'un robot autonome. Ali dans [38] utilise un POMDP pour un robot donnant un objet à un homme.

Les modèles graphiques probabilistes. Les modèles graphiques probabilistes permettent de représenter les variables aléatoires sous la forme d'un graphe dont les noeuds sont les variables aléatoires et les arrêtes ou arcs représentent la corrélation entre ces variables aléatoires. Dans sa thèse de doctorat [39], Murphy décrit plus en détail les réseaux bayésiens dynamiques. Il explique comment les représenter, comment les utiliser pour l'inférence qui consiste à mettre à jour l'état d'un sous ensemble des variables quand d'autres variables sont observées.

10.3.2 Un réseau bayésien dynamique adapté à notre contexte

Si l'on prend le cas particulier du robot manipulateur qui agit de manière collaborative avec l'homme, il doit construire et mettre à jour l'état du monde en terme de position des objets et des agents. Il doit également suivre les actions de l'homme pour mieux suivre l'évolution de l'état du monde, vérifier que ces actions correspondent à celles attendues si l'homme et le robot sont en train de réaliser un but partagé. Si le robot était en mesure de détecter le but de l'homme il pourrait également participer de manière proactive sans attendre une requête explicite de l'homme.

Alors que nous les considérons dans le système actuel de manière relativement indépendante, buts, plans, actions et postions sont bien sûr reliés entre eux par des liens causaux. L'homme peut avoir un but. Il réalise alors un certain nombres d'actions pour atteindre ce but compte tenu de l'état du monde. Ces actions changent l'état du monde.

Le modèle graphique probabiliste présenté dans la figure suivante 10.3 permet de représenter les liens de causalité entre tous ces éléments. On fait l'hypothèse que l'homme a un but et un plan lui permettant de réaliser ce but. Si l'homme change de but ou de plan, on doit réinitialiser le modèle en question à l'instant correspondant à cette transition.

Les observations correspondent à ObsPos_t qui est l'observation sur la position et ObsAct_t qui est l'observation sur les actions. Tous les autres états du système ne sont pas directement observables sauf si l'homme verbalise directement des états. Ils correspondent à la positions des objets et des agents autour de l'instant t Positions_t, les actions de l'homme entre t et t+1 Actions_t, au but de l'homme depuis l'instant 0 Goal_0 et à Plan_0 le plan de l'homme qui permet d'atteindre le but Goal_0 à partir de l'état du monde Positions_0. Encore une fois ce modèle graphique probabiliste est valable tant que le plan se déroule avec succès.

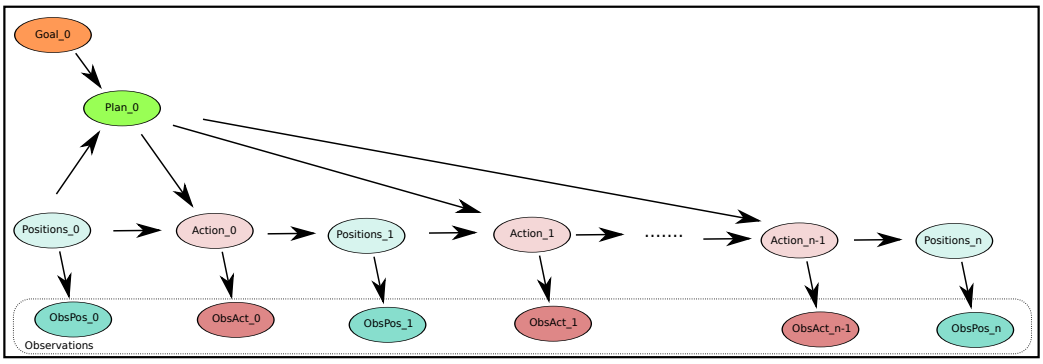


FIG. 10.3 – *
Modèle graphique probabiliste

10.3.3 Bénéfices attendus

Une meilleure interprétation des observations des positions et des actions.

Comme expliqué dans [58], l'approche probabiliste permet de mieux interpréter les données capteurs. Dans le modèle graphique probabiliste cela correspond aux arcs entre Positions_t et ObsPos_t et aux arcs entre Actions_t et ObsAct_t. Cela nécessite d'avoir un modèle des observations compte tenu de l'état du monde pour pouvoir l'inverser au moyen de la règle de Bayes.

Une estimation naturelle de l'incertitude sur la connaissance de l'état du monde du robot.

Du fait qu'il construit sa représentation du monde sous la forme d'une distribution de probabilité sur les états, le robot évalue donc le niveau d'incertitude de sa propre connaissance du monde au contraire du cas déterministe. C'est très important pour doser l'effort d'observation nécessaire avant de pouvoir raisonnablement évaluer la réussite d'une action.

Une meilleure estimation des actions et des positions

Le modèle 10.3 décrit bien que les positions à t+1 (Positions_t+1) dépendent des actions à t Actions_t et que les actions à t Actions_t ne sont possible que pour certaines valeurs des positions à t Positions_t. Un robot capable d'interpréter des actions d'un homme sur des objets doit à la fois analyser le mouvement de l'homme, reconnaître les objets qui sont impliqués dans l'action et observer l'effet de l'action sur ces objets. Alors que chacune de ces tâches perceptuelles peuvent être accomplie indépendamment l'une de l'autre, le taux de reconnaissance augmente si l'on considère les interactions qu'elles ont entre elles. Gupta dans [21] intègre ces différentes tâches perceptuelles en ajoutant des contraintes spatiales et fonctionnelles dans un cadre bayésien. C'est un exemple d'inférence qui permet aux auteurs de ces travaux d'améliorer significativement la reconnaissance des objets et la reconnaissance des actions.

La capacité à déterminer le but de l'homme.

Le plan à l'instant 0 Plan_0 qui se traduit directement par la séquence (Actions_0, Actions_1, ... , Actions_t) tant que le plan est valide et en cours est lié causalement au but à l'instant Goal_0 et à l'état du monde à l'instant 0 Positions_0. Théoriquement, il est possible de déduire le but à l'instant 0 du plan et de l'état initial. La reconnaissance du but est importante dans le cadre de la coopération homme robot. Elle permet au robot d'agir de manière proactive sans forcément attendre que l'homme requière son aide.

Malheureusement, l'espace des plans étant potentiellement très grand, l'inférence du but à partir du plan n'est à priori pas faisable dans un temps raisonnable. Schrempf dans [49] essaye de trouver des modèles reliant les buts et les actions avec un espace des états réduits pour permettre de réaliser cette inférence en ligne.

Un moyen d'optimiser le choix des observations

A tout instant le robot doit décider de la direction d'orientation de ses capteurs. Pour cela il doit déterminer quelle est la direction qui devrait lui apporter le plus d'information. Ce raisonnement se base sur la distribution courante de probabilité sur les positions mais aussi sur l'estimation de la dynamique liée à la probabilité que telle action soit réalisée.

On peut s'inspirer des travaux sur la sélection des trois paramètres pan, tilt et zoom d'une caméra de surveillance. Dans [56] Sommerlade, affirme que les paramètres des caméras devraient être choisis de manière à maximiser le gain d'information ou de manière équivalente à minimiser l'entropie conditionnelle du modèle de la scène qui consiste dans son cas à un certains nombres de cibles qui sont suivies et à une qui ne l'est pas encore.

10.3.4 Deux exemples schématiques

Modélisation des états et de leur perception.

On utilise la discrétisation de l'espace des états proposée dans la section 10.2. La distribution de probabilité de l'état courant d'un objet est caractérisée par la valeur de la probabilité que l'objet soit dans l'un de ces sous-ensembles disjoints de l'ensemble des états atteignables. La somme de ces probabilités est égale à 1. On doit pouvoir estimer l'observabilité d'un sous-ensemble en fonction de l'estimation courante de l'état du monde. Cette estimation se ferait d'abord à priori sur la base de critères liés à la géométrie comme la distance et les occlusions. Une zone de la table qui est derrière un gros objet par rapport au robot aura une observabilité plus faible. L'observation d'objet dans la main de l'homme est difficile en l'état du système. L'observabilité des mains de l'homme est donc faible en comparaison des zones de la table non cachées. Cette observabilité peut être encore plus faible si l'homme et le robot sont très éloignés l'un de l'autre. Cette observabilité pourrait être ensuite affinée et mise à jour compte tenue des observations réelles.

Modélisation des actions et de leur perception.

Les actions possibles sont connues. Leurs préconditions et effets sont bien identifiés. On peut pour chacune de ces actions estimer la probabilité de succès et les probabilités des différents types d'échecs ainsi que les effets pour chacun de ces types d'échecs considérés. L'observabilité de l'action peut être aussi modélisée. Une observation d'un "moniteur" doit nous donner une distribution de probabilités sur les actions qui ont pu provoquer le déclenchement de ce "moniteur". L'effet des actions peut être propagé sur la distribution de

l'état courant pour obtenir l'estimation de l'état suivant et réciproquement. Les valeurs d'estimation des distributions de probabilité sur les actions impactent sur les valeurs d'estimation des distributions de probabilité des états et réciproquement. A un instant donné, compte tenu de toutes les observations des états et des actions ont doit pouvoir faire converger toutes les estimations d'états et d'actions vers celles qui donnent la probabilité jointe la plus élevée.

La figure 10.4 permet d'illustrer ce concept. Le schéma à gauche de la figure décrit la connaissance à priori du système. Le robot n'a aucune connaissance sur la position de l'objet si ce n'est qu'il n'est pas dans la pince. On estime que tous les états théoriquement atteignables peuvent correspondre à l'état actuel avec une probabilité égale. En haut à gauche on présente la nouvelle estimation de l'état suite à l'observation de l'objet proche de la main droite. Compte tenu de la position de l'homme par rapport à la table, l'objet est presque sûrement dans la main droite avec une probabilité forte ou dans la main gauche avec une probabilité moindre. En bas à gauche on présente la nouvelle estimation de l'état suite à l'exploration exhaustive et infructueuse de la table et des mains. L'objet est alors presque sûrement sur les zones cachées de la table ou dans la poubelle. Il peut aussi être avec une probabilité moindre dans les mains de l'homme compte tenu de la difficulté à observer un objet dans la main de l'homme qui ne permet pas d'exclure qu'on soit dans cet état malgré l'observation infructueuse de cet état. Même si l'objet n'a pas été perçu, la connaissance sur son état s'est grandement améliorée. Le dialogue permettrait alors d'affiner encore un peu plus la connaissance. Le robot pourrait par exemple demander à l'homme : "Est-ce que vous voyez l'objet ?". Si la réponse est non, l'objet serait alors soit recouvert par la boîte englobante à fond vide SURPRISE_BOX, soit dans la corbeille PINK_TRASHBIN.

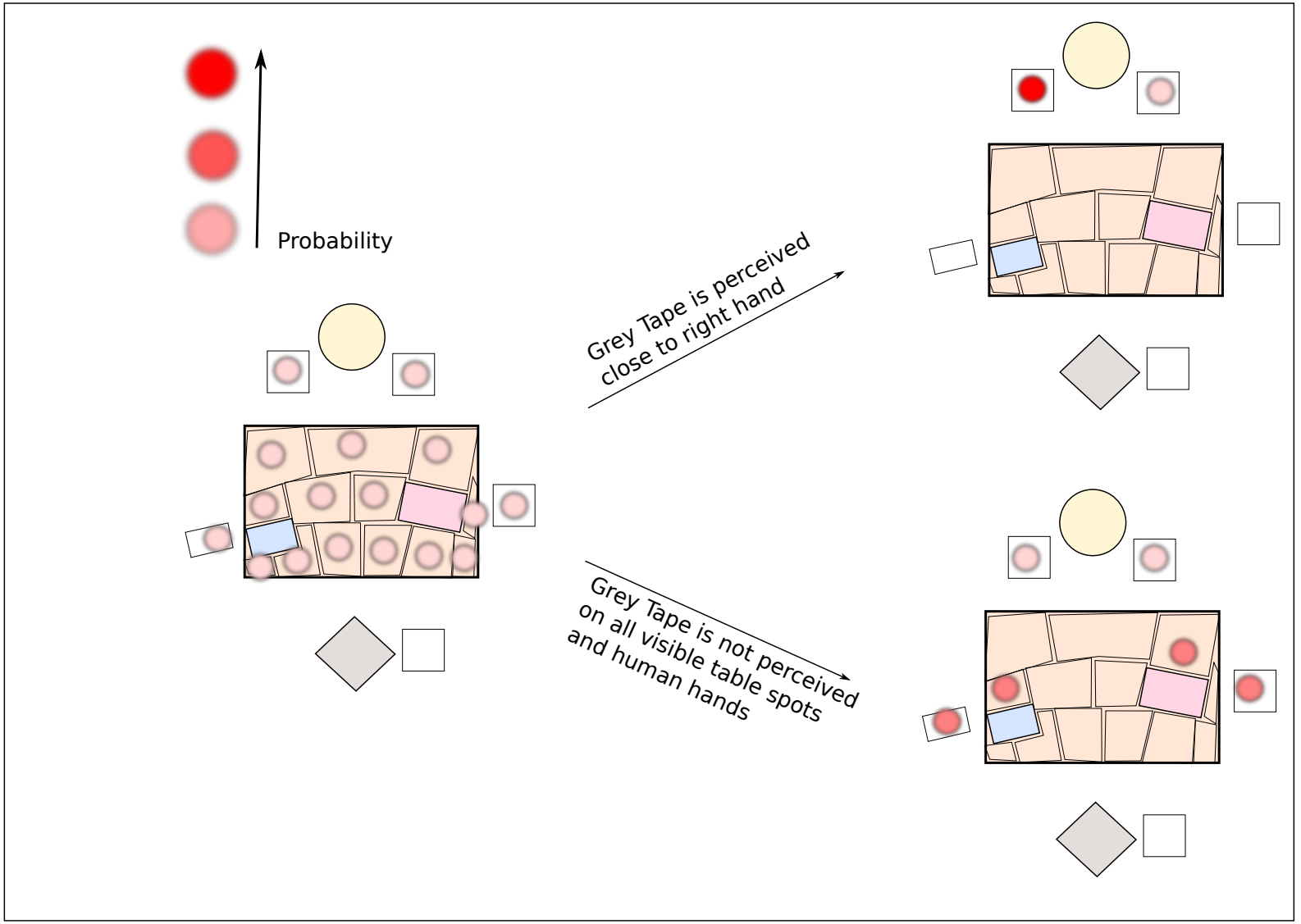


FIG. 10.4 – *
Impact de la perception sur la spatialisation de la distribution de probabilité.

La figure 10.5 permet d'introduire la réflexion sur l'utilité des aspects probabilistes dans le cadre de la reconnaissance d'action. Le schéma de gauche (A) représente l'estimation initiale sur la position des boîtes noires et blanches. L'homme effectue ensuite un mouvement qui peut être interprété par 4 actions distinctes selon la distribution de probabilité suivante.

- Pick White : 40 %
- Show White : 35 %
- Show Black : 15 %
- Pick Black : 10 %

Les actions pick ont une probabilité d'être réalisées avec succès de 90 % et 10 % de n'avoir aucun effet. Les distributions de probabilité associées au succès de chaque action sont représentées dans la partie (B) du schéma. La partie (C) du schéma permet d'estimer l'état du monde compte tenu des différentes actions possibles et de la probabilité qu'elles aient été réalisées et réussies. Cela permet de choisir l'observation la plus pertinente dans le sens de la maximisation du gain d'information sur l'estimation de l'état du monde si l'on choisit d'observer dans cette direction. Du fait de l'état du monde (plus d'incertitude sur l'objet blanc) et de l'observabilité (l'observation d'un objet sur la table est plus discriminante que l'observation dans la main qui est très incertaine) on choisit donc d'observer l'objet blanc sur la table. Il n'est pas perçu ce qui donne l'estimation numérotée (D). On ne précise pas les probabilitées dans la mesure où l'on ne fait pas le calcul de manière exacte. L'objet blanc est désormais dans la main gauche avec une forte probabilité. La probabilité que l'action qui avait eu lieu est PICK WHITE s'en trouve augmentée. Du fait du nouvel état du monde (plus d'incertitude sur l'objet noire) et de l'observabilité (l'observation d'un objet sur la table est plus discriminante que l'observation dans la main qui est très incertaine) on choisit donc d'observer l'objet noire sur la table. Il est perçu ce qui donne la nouvelle estimation notée (E). De même, on ne précise pas les probabilitées dans la mesure où l'on ne fait pas le calcul de manière exacte. La probabilité que l'objet noire soit sur la table s'en trouve augmentée. La probabilité que l'action qui avait eu lieu est PICK WHITE s'en trouve encore augmentée. Le robot peut alors conclure avec une grande confiance dans le fait que l'action réalisée était bien PICK WHITE.

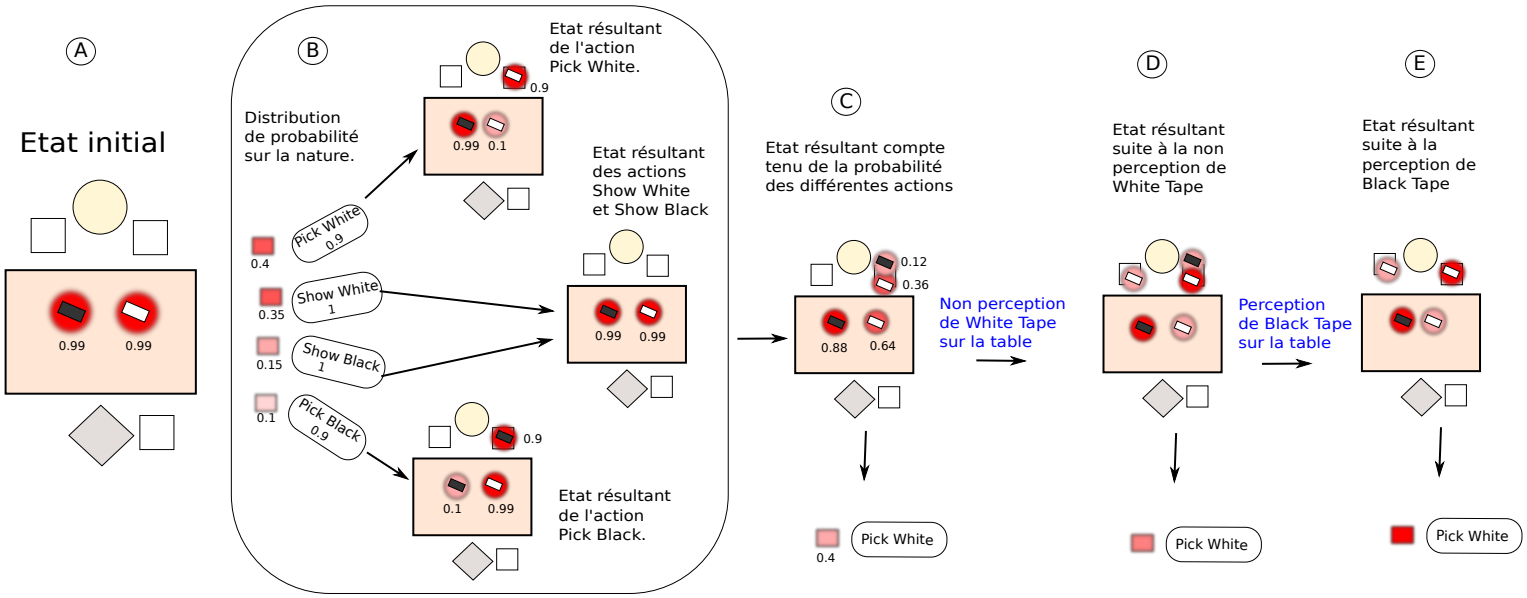


FIG. 10.5 – *
Application à la reconnaissance d'action

10.4 Prospective sur la gestion de la planification de mouvement

La planification de mouvement reste une activité coûteuse qui peut être responsable d'une partie importante de la lenteur du robot. Une gestion fine d'un portefeuille de plans pré-calculés permettrait d'optimiser grandement la vitesse d'exécution du robot. Dans l'état actuel du système, la planification est réalisée en ligne au moment où l'on décide d'exécuter l'action. Le temps de planification pèse alors lourd dans le temps d'exécution globale. Si l'on parvient à estimer les futurs mouvements les plus probables compte tenu de l'état courant du monde et éventuellement du plan en cours, il est alors possible de pré-planifier des mouvements dans la mesure d'une estimation de la disponibilité du planificateur. Si l'on est en mesure de pouvoir vérifier de manière rapide la validité d'un plan géométrique existant, alors, on peut valider à l'exécution rapidement un mouvement qui pourrait être dans la bibliothèque de plan actuel et gagner la différence entre planification et validation d'un plan existant en espérant que le plan soit toujours valable.

La figure 10.6 permet d'illustrer cela sur un exemple. Nous nous plaçons dans un cas très optimiste où il y a un but existant et un plan associé (empiler les 3 cubes). Du coup on commence à planifier en fonction du plan existant. À partir du moment où le robot décide réellement d'exécuter le mouvement, le plan correspondant existe et le robot n'a alors qu'à vérifier la validité. Sur cette séquence, le gain est égal à la somme des intervalles rouges moins la somme des intervalles bleus. Aujourd'hui, la planification nécessite quelques secondes de calcul. L'exécution prend aussi quelques secondes pour le robot. On pourrait donc au moins planifier l'action suivante pendant l'exécution de l'action en cours.

Pour résumer, il faudrait être en mesure de réaliser les 3 tâches suivantes.

- Estimer en temps réel les mouvements futurs les plus probables.
- Estimer la disponibilité du planificateur compte tenu du contexte courant d'exécution de manière à ce que l'utilisation du planificateur pour pré-planifier ne vienne pas en concurrence de la planification d'un mouvement à exécuter immédiatement.
- Estimer la validité à un instant donné d'un mouvement planifié précédemment. (Le plus rapidement possible relativement au temps de planification.)

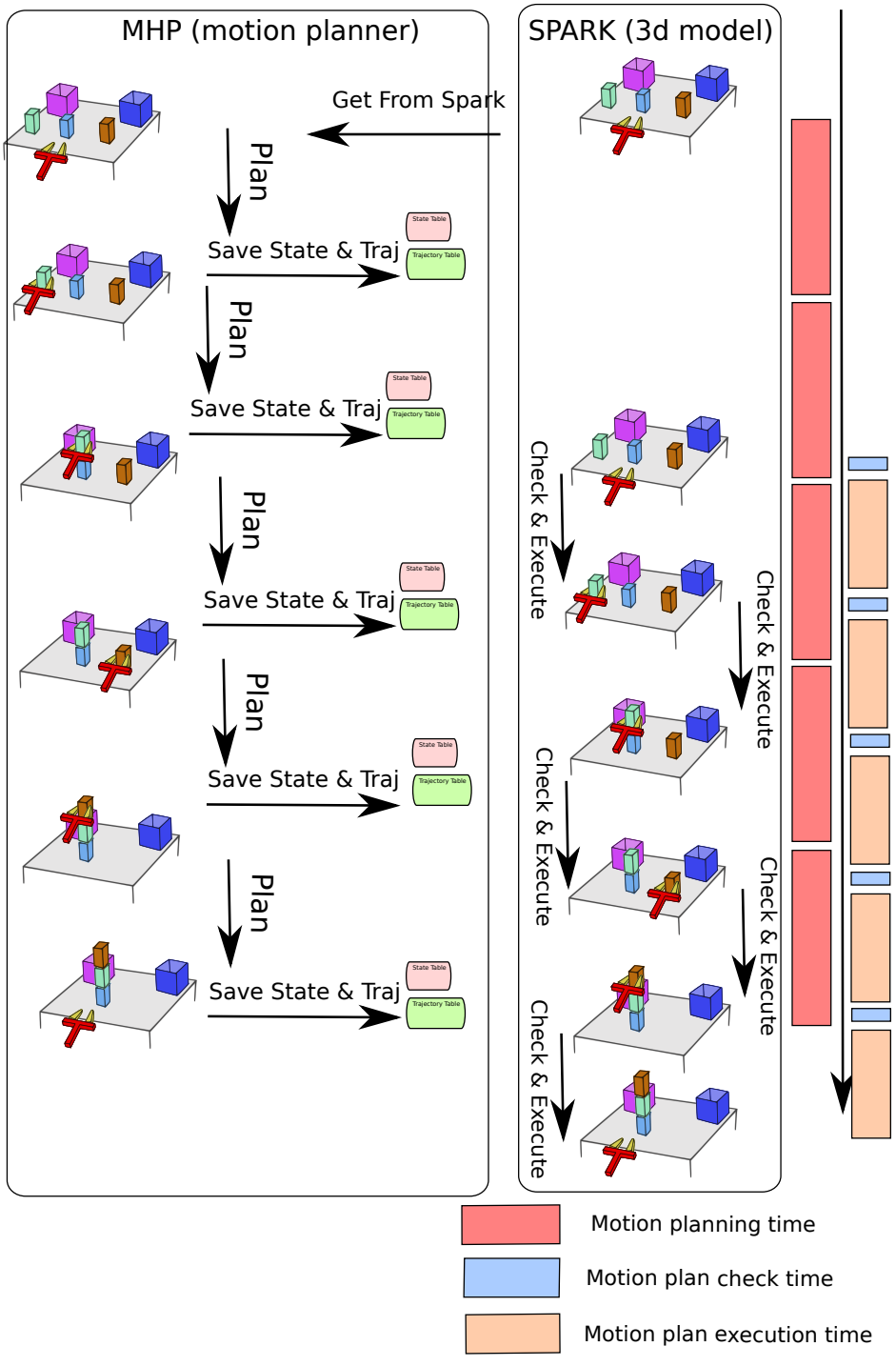


FIG. 10.6 – *
Anticipation de la planification de mouvement. On planifie avec anticipation les actions dont on pense qu'elles vont être réalisées. Au moment de l'exécution, on peut directement exécuter le plan s'il est toujours valide

10.5 Nouvelles expérimentations

Cette section introduit des nouvelles campagnes d'expérimentations que nous souhaitons mener à l'avenir. La première campagne introduite dans la section 10.5.1 vise à évaluer la capacité du robot à suivre son environnement. La deuxième campagne introduite dans la section 10.5.2 se propose d'évaluer l'impact du changement de planificateur ou de la politique d'exécution du plan.

10.5.1 Vers la mesure de la capacité du robot à suivre son environnement

Nous n'avons tenté de doter notre robot d'un arsenal complet lui permettant d'explorer, représenter et mettre à jour son environnement. Nous souhaitons donc estimer la capacité du robot à effectivement construire et mettre à jour une représentation de son environnement. Pour cela nous souhaitons réaliser un certain nombre d'expérimentations pour évaluer la performance du système pour les trois contextes ci-dessous :

- Construction de l'état initial.
- Suivi de l'évolution du monde quand le robot agit.
- Suivi de l'évolution du monde quand l'homme agit.

10.5.2 Étude de l'impact du planificateur et de la politique d'exécution du plan

Notre équipe travaille depuis longtemps sur la planification symbolique. Depuis bientôt dix ans, un planificateur adapté à la réalisation collaborative homme-robot de buts a été développé. C'est le planificateur Hatp. Il a été récemment adapté pour gérer la possibilité d'avoir des croyances différentes pour les différents agents. En l'état actuel du système qui a été utilisé pour la plupart des expérimentations conduites depuis un an, le robot utilise hatp pour obtenir un plan qui est ensuite exécuté de manière séquentielle action après action. Le robot exécute les actions ou suit les actions dont le robot est responsable. Il ne fait pas les deux de manière concurrente.

Nous souhaitons estimer et éventuellement remettre en cause ce choix. Pour cela, nous nous proposons de confronter des utilisateurs à différentes versions de notre système en jouant sur les 3 critères suivants :

– Choix d'utiliser le planificateur : Planificateur vs Pilotage par l'homme action par action.
– Choix de la version du planificateur : Hatp vs Hatp Belief Management.
– Choix de la politique d'exécution du plan : exécution et suivi vs exécution seule.

Nous distinguerons deux types de scénario pour ce qui est du critère des croyances divergentes :

– Scénario classique : les croyances des agents sont homogènes.
– Scénario avec croyances divergentes : l'homme n'a pas observé certaines transitions si bien que sa croyance sur l'état du monde diverge de celle du robot.

Étude de l'impact de l'utilisation du planificateur et du choix de la politique d'exécution du plan

Nous souhaitons savoir s'il est pertinent d'utiliser le planificateur et s'il est utilisé quelle politique d'exécution du plan est préférable.

Pilotage action par action : L'homme pilote le robot action par action. Les actions que l'homme peut demander au robot sont les suivantes : attraper un objet, jeter un objet dans un conteneur, placer un objet tel qu'il soit atteignable pour l'homme. Nous nous retrouvons dans une situation inversée par rapport au fonctionnement par défaut du système. Ici, c'est l'homme qui planifie et le robot qui suit le rythme dicté par l'homme. On considère que l'homme est en mesure de planifier et de rythmer l'exécution du plan.

Planification complète avec exécution partielle du plan. (uniquement la réalisation des actions assignées au robot) : Le robot planifie pour les deux agents. Par contre il ne s'occupe que de la réalisation de ses propres actions. Au besoin, il essaye d'attendre le résultat d'une action de l'homme. On considère que l'homme comprend, anticipe le plan vu par le robot et qu'il est en mesure de gérer lui-même l'exécution de ses actions.

Planification complète avec exécution complète non concurrente des actions : C'est le mode par défaut de notre système comme expliqué dans les parties précédentes. Le robot planifie pour les deux agents. Il procède alors à l'exécution du plan action par action. L'homme est alors guidé / surveillé comme un enfant. Cela est particulièrement adapté au cas où l'homme est dépendant, et n'est pas en mesure de se débrouiller tout seul. Cela représente

par contre un coût supérieur en terme de temps d'exécution.

La description de ces trois modes différents nous a permis de voir qu'ils sous-entendent des niveaux de compétences variables et décroissants relativement à la capacité de l'homme à planifier et gérer l'exécution du plan :

- Utilisateur expert en mesure de planifier et de gérer l'exécution collaborative du plan.
- Utilisateur partiellement expert en mesure de comprendre un plan et de gérer de manière indépendante l'exécution de sa part de travail.
- Utilisateur non expert seulement en mesure de réaliser une action à la demande.

Étude de l'impact de l'utilisation d'Hatp avec croyance divergente

Est-ce que l'utilisation du planificateur Hatp Mutual Belief est vraiment avantageuse par rapport à la version Hatp simple dans le cas de scénario avec croyance divergente ? Pour cela il faudra estimer l'impacte de ce choix sur la performance de l'humain.

On tentera d'abord d'estimer cette performance de manière quantitative en mesurant les erreurs éventuelles dans les actions de l'humain ou du moins un accroissement du temps nécessaire à la réalisation de ses actions. On pourra aussi interroger l'humain sur son ressenti en terme de confusion éventuelle.

Chapitre 11

Conclusion Générale

11.1 Leçons tirées du système

Quelles sont les leçons que nous pouvons tirer de notre système ?

Le système que nous avons construit est très ambitieux du point de vue du spectre des fonctionnalités mises en oeuvre. Nous souhaitions planifier et exécuter le but partagé dans son ensemble. Ce caractère "complet" fait émerger certaines questions qui n'apparaissent pas si l'on développe et teste les fonctions hors de ce contexte global. Par exemple, il m'est apparu qu'il y avait un écart très important entre le perfectionnement des planificateurs de mouvements et de tâches que nous développions et la faible capacité à effectivement construire et mettre à jour l'état du monde nécessaire à leur fonctionnement. Le robot pouvait se projeter de manière complexe dans l'avenir sur la base de l'état du monde courant. Par contre il pouvait difficilement construire cet état et encore plus difficilement le mettre à jour au cours de l'exécution. Le robot peinait et peine toujours à reconnaître et à comprendre ce qui se passe ou au moins à constater ou à déduire le nouvel état. C'est encore plus frappant si l'on considère des états mentaux complexes tels que les buts et les plans.

Néanmoins, nous pensons que ce travail nous a permis de valider et d'identifier un certain nombre de briques essentielles :

- La prise de perspective des agents.
- La synthèse de l'état géométrique cartésien du monde sous la forme de faits symboliques.
- La prise en compte de croyances divergentes des agents.
- La spatialisation d'états symboliques déduits du raisonnement.

– La reconnaissance basique d'action au moyen de "moniteurs" simples.
– Un mécanisme d'attention que nous définissons comme la capacité à focaliser au mieux l'acquisition d'informations.

Finalement, notre système même s'il est constitué de nombreuses briques assez simplistes produit un comportement plutôt abouti. Les erreurs assez fréquentes n'entraînent pas l'arrêt du système du fait de l'exploration continuelle et de la replanification.

11.2 Limites fortes du système

Quelles sont les limites du système ? Certaines ont déjà été évoquées dans le manuscrit.

Absence d'apprentissage. Toutes les connaissances et les processus du système sont fournis par les programmeurs : les buts, les actions que le robot peut exécuter ou suivre, la typologie des entités .... Le robot n'a aucun moyen d'augmenter ou d'améliorer au moyen de l'apprentissage ces procédures ou ces raisonnements.

Difficulté à construire et mettre à jour les états. Comme évoqué dans le manuscrit et dans la section précédente, la capacité du robot à construire et à mettre à jour l'état géométrique du monde est assez faible. C'est en partie lié aux défauts de la perception mais cela est également dû à la faiblesse des raisonnements temporels et géométriques dédiés. Notre système est très réactif. Le processus d'attention réagit à des événements qu'il oublie et ne prend pas en compte si de nouveaux événements surviennent trop tôt. Les états ne sont pas probabilistes. Nous ne pouvons donc ni représenter, ni estimer la qualité de l'état courant. Cela nous interdit de focaliser l'exploration ou le raisonnement sur les entités dont l'état est connu avec le moins de précision.

Hiérarchisation et focus spatial. Comme décrit dans 10.1, il n'y a aujourd'hui aucune différenciation ou hiérarchisation spatiale entre les différents objets, agents et localisation dans le modèle 3D. Tout changement localisé entraîne la mise à jour de tous les calculs. Cela posera inévitablement un problème si l'on choisit des scénarios avec un nombre d'agents, d'objets et de locations plus importants.

Lenteurs. Le système est aujourd'hui trop lent au point que la qualité de l'interaction s'en trouve diminuée. Une grande partie de cette lenteur est liée à la planification de mouvement. Pourtant, une meilleure utilisation de la planification pourrait permettre de réduire cette lenteur comme explicitée en 10.4.

De manière plus générale une meilleure utilisation des ressources existantes (anticipation de l'utilisation, parallélisation de l'utilisation) permettrait de réduire certaines lenteurs.

États mentaux de haut niveau. Les états mentaux de haut niveau tel que les buts, les plans, la motivation et les préférences sont encore plus difficiles à estimer que les états physiques du monde. Ils ne sont pas non plus facilement interprétables dans le discours de l'homme. Cela entraîne de fortes limites dans les processus de haut niveau. Le robot suppose l'implication de l'homme et peut difficilement se rendre compte du fait que celui-ci ne veut plus réaliser le but à moins d'un refus manifeste d'agir. Le robot planifie seul et communique le plan à l'homme au cours de son exécution en supposant que celui-ci le comprend et l'accepte. Cela rend très difficile l'exécution concurrente des actions. De même, cela limite fortement la capacité du robot à agir de manière pro-active. Enfin, c'est au détriment de la capacité du robot à se synchroniser de manière souple et rapide avec l'homme dans l'exécution de la tâche.

Bibliographie

[1] R. Alami, R. Chatila, S. Fleury, M. Ghallab, and F. Ingrand. An architecture for autonomy. *International Journal of robotics Research, Special Issue on Integrated Architectures for Robot Control and Programming*, 17(4), 1998.

[2] R. Alami, M. Warnier, J. Guitton, S. Lemaignan, and E. A. Sisbot. When the robot considers the human... In *Proceedings of the 15th International Symposium on Robotics Research*, 2011.

[3] S. Alili, V. Montreuil, and R. Alami. HATP task planner for social behavior control in autonomous robotic systems for hri. In *The 9th International Symposium on Distributed Autonomous Robotic Systems*, 2008.

[4] P. Althaus, H. Ishiguro, T. Kanda, T. Miyashita, and H. Christensen. Navigation for human-robot interaction tasks. *IEEE Int. Conf. on Robotics & Automation, New Orleans, USA*, 2004.

[5] ARToolkit. Documentation and download site. *http ://www.hitl.washington.edu/artoolkit/*.

[6] J. v. d. Berg and M. Overmars. Roadmap-based motion planning in dynamic environments. Technical report, Utrecht University, NL, 2004.

[7] W. Bluethmann, R. Ambrose, M. Diftler, S. Askew, E. Huber, M. Goza, F. Rehnmark, C. Lovchik, and D. Magruder. Robonaut : A robot designed to work with humans in space. *Autonomous Robots*, 14 :179–197, 2003.

[8] C. Breazeal. Towards sociable robots. *Robotics and Autonomous Syst.*, pages 167–175, 2003.

[9] C. Breazeal, M. Berlin, A. Brooks, J. Gray, and A. Thomaz. Using perspective taking to learn from ambiguous demonstrations. *Robotics and Autonomous Systems*, pages 385–393, 2006.

[10] C. Breazeal, J. Gray, and M. Berlin. An embodied cognition approach to mindreading skills for socially intelligent robots. *I. J. Robotic Res.*, 28(5) :656–680, 2009.

[11] J. Butterfield, O. Jenkins, D. Sobel, and J. Schwertfeger. Modeling aspects of theory of mind with markov random fields. *International Journal of Social Robotics*, 1 :41–51, 2009. 10.1007/s12369-008-0003-1.

[12] A. Cassandra, L. Kaelbling, and J. Kurien. Acting under uncertainty : discrete bayesian models for mobile-robot navigation. In *Intelligent Robots and Systems '96, IROS 96, Proceedings of the 1996 IEEE/RSJ International Conference on*, volume 2, pages 963–972 vol.2, Nov.

[13] H. H. Clark. *Using Language*. Cambridge University Press, 1996.

[14] P. R. Cohen and H. J. Levesque. Teamwork. *Nous*, 25(4) :487–512, 1991.

[15] Y. Demiris and B. Khadhouri. Hierarchical attentive multiple models for execution and recognition of actions. *Robotics and Autonomous Systems*, 54(5) :361 – 369, 2006. <ce :title>The Social Mechanisms of Robot Programming from Demonstration</ce :title> <xocs :full-name>The Social Mechanisms of Robot Programming from Demonstration</xocs :full-name>.

[16] J. Flavell. *Perspectives on Perspective Taking*, pages 107–139. L. Erlbaum Associates, 1992.

[17] S. Fleury, M. Herrb, and R. Chatila. Genom : a tool for the specification and the implementation of operating modules in a distributed robot architecture. In *IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS*, Grenoble, FR., 1997.

[18] S. Fleury, M. Herrb, and A. Mallet. *GenoM : Generator of Modules for Robots*. http ://softs.laas.fr/openrobots/tools/genom.php, 2005.

[19] J. Gray and C. Breazeal. Manipulating mental states through physical action. In S. Ge, O. Khatib, J.-J. Cabibihan, R. Simmons, and M.-A. Williams, editors, *Social Robotics*, volume 7621 of *Lecture Notes in Computer Science*, pages 1–14. Springer Berlin Heidelberg, 2012.

[20] B. J. Grosz and S. Kraus. Collaborative plans for complex group action. *Artificial Intelligence*, 86 :269–358, 1996.

[21] A. Gupta, A. Kembhavi, and L. Davis. Observing human-object interactions : Using spatial and functional compatibility for recognition. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 31(10) :1775–1789, Oct.

[22] L. M. Hiatt, A. M. Harrison, and J. G. Trafton. Accommodating human variability in human-robot teams through theory of mind. In *Proceedings of the Twenty-Second international joint conference on Artificial Intelligence - Volume Volume Three*, IJCAI'11, pages 2066–2071. AAAI Press, 2011.

[23] G. Hoffman and C. Breazeal. What lies ahead ? expectation management in human-robot collaboration. In *AAAI Spring Symposium : To Boldly Go Where No Human-Robot Team Has Gone Before*, pages 1–7, 2006.

[24] A. Holroyd, C. Rich, C. L. Sidner, and B. Ponsler. Generating connection events for human-robot collaboration. In *RO-MAN, 2011 IEEE*, pages 241 –246, 31 2011-aug. 3 2011.

[25] F. Ingrand. *OpenPRS : an open source version of PRS (Procedural Reasoning Systems).* http ://softs.laas.fr/openrobots/tools/openprs.php, 2005.

[26] F. Ingrand, R. Chatila, R. Alami, and F. Robert. Prs : a high level supervision and control language for autonomous mobile robots. *IEEE International Conference on Robotics and Automation*, 1996.

[27] F. Ingrand and F. Py. Online execution control checking for autonomous systems. *7th International Conference on Intelligent Autonomous Systems (IAS-7)*, 2002.

[28] M. Johnson and Y. Demiris. Perceptual perspective taking and action recognition. *Advanced Robotic Systems*, 2(4) :301–308, 2005.

[29] C. Karaoguz, T. Rodemann, and B. Wrede. Optimisation of gaze movement for multitasking using rewards. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2011.

[30] C. Kemp and E. Edsinger, A. Torres-Jara. Challenges for robot manipulation in human environments. *Robotics & Automation Magazine, IEEE*, 2007.

[31] W. G. Kennedy, M. D. Bugajska, A. M. Harrison, and J. G. Trafton. "like-me" simulation as an effective and cognitively plausible basis for social robotics. *I. J. Social Robotics*, 1(2) :181–194, 2009.

[32] D. Kulic and E. Croft. Pre-collision safety strategies for human-robot interaction. *Autonomous Robots*, 22(2) :149–164, 2007.

[33] S. Lemaignan, R. Ros, L. Mosenlechner, R. Alami, and M. Beetz. Oro, a knowledge management module for cognitive architectures in robotics. In *Proceedings of the 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2010.

[34] A. Leslie. Theory of mind as a mechanism of selective attention. *The new cognitive neurosciences*, pages 1235–1247, 2000.

[35] K. Madhava, R. Alami, and T. Simeon. Safe proactive plans and their execution. *Robotics and Autonomous Systems*, 54(3) :244–255, 2006.

[36] J. Mainprice, E. Sisbot, L. Jaillet, J. Cortes, R. Alami, and T. Simeon. Planning human-aware motions using a sampling-based costmap planner. In *IEEE International Conference on Robotics and Automation*, 2011.

[37] N. Mavridis and D. Roy. Grounded situation models for robots : Where words and percepts meet. In *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*, pages 4690 –4697, oct. 2006.

[38] A. Muhammad. *Contribution to decisional human-robot interaction : towards collaborative robot companions*. PhD thesis, Institut National des Sciences Appliquées, Toulouse, 2012.

[39] K. P. Murphy. *Dynamic bayesian networks : representation, inference and learning*. PhD thesis, University of California, 2002.

[40] D. Nau, T.-C. Au, O. Ilghami, U. Kuter, J. W. Murdoch, D. Wu, and F. Yaman. SHOP2 : An HTN planning system. *Journal of Artificial Intelligence Research*, 20 :380–404, 2003.

[41] A. Pandey and R. Alami. Mightability maps : A perceptual level decisional framework for co-operative and competitive human-robot interaction. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2010.

[42] C. Parlitz, W. Baum, U. Reiser, and M. Hägele. Intuitive human-machine-interaction and implementation on a household robot companion. In *Proceedings of the 2007 conference on Human interface : Part I*, pages 922–929, Berlin, Heidelberg, 2007. Springer-Verlag.

[43] J. Pineau, M. Montemerlo, M. Pollack, N. Roy, and S. Thrun. Towards robotic assistants in nursing homes : Challenges and results. *Robotics and Autonomous Systems*, 42(3-4) :271–281, 2003.

[44] C. Rich, B. Ponsler, A. Holroyd, and C. Sidner. Recognizing engagement in human-robot interaction. In *Human-Robot Interaction (HRI), 2010 5th ACM/IEEE International Conference on*, pages 375 –382, march 2010.

[45] C. Rich and C. L. Sidner. Collagen : When agents collaborate with people. *Proceedings of the first international conference on Autonomous Agents*, 1997.

[46] R. Ros, S. Lemaignan, E. Sisbot, R. Alami, J. Steinwender, K. Hamann, and F. Warneken. Which one ? grounding the referent based on efficient human-robot interaction. In *19th IEEE International Symposium in Robot and Human Interactive Communication*, 2010.

[47] R. Ros, E. A. Sisbot, R. Alami, J. Steinwender, K. Hamann, and F. Warneken. Solving ambiguities with perspective taking. In *5th ACM/IEEE International Conference on Human-Robot Interaction*, 2010.

[48] B. Scassellati. Theory of mind for a humanoid robot. *Autonomous Robots*, 12(1) :13–24, 2002.

[49] O. Schrempf, D. Albrecht, and U. Hanebeck. Tractable probabilistic models for intention recognition based on expert knowledge. In *Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on*, pages 1429–1434, 29 2007-Nov. 2.

[50] J. A. Shah, J. Wiken, B. C. Williams, and C. Breazeal. Improved human-robot team performance using chaski, a human-inspired plan execution system. In A. Billard, P. H. K. Jr., J. A. Adams, and J. G. Trafton, editors, *HRI*, pages 29–36. ACM, 2011.

[51] C. L. Sidner, C. Lee, C. Kidd, N. Lesh, and C. Rich. Explorations in engagement for humans and robots. *Artificial Intelligence*, 166(1-2) :140–164, 2005.

[52] R. Simmons and S. Koenig. Probabilistic robot navigation in partially observable environments. In *International Joint Conference on Artificial Intelligence*, volume 14, pages 1080–1087. LAWRENCE ERLBAUM ASSOCIATES LTD, 1995.

[53] E. Sisbot, L. Marin-Urias, R. Alami, and T. Simeon. A human aware mobile robot motion planner. *IEEE Transactions on Robotics*, 23(5) :874–883, 2007.

[54] E. Sisbot, R. Ros, and R. Alami. Situation assessment for human-robot interaction. In *20th IEEE International Symposium in Robot and Human Interactive Communication, (ROMAN)*, 2011.

[55] E. A. Sisbot, A. Clodic, R. Alami, and M. Ransan. Supervision and motion planning for a mobile manipulator interacting with humans. In *ACM/IEEE International Conference on Human-Robot Interaction (HRI 2008)*, 2008.

[56] E. Sommerlade and I. Reid. Information-theoretic active scene exploration. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–7, June.

[57] M. Tambe. Towards flexible teamwork. *JAIR*, 7 :83–124, 1997.

[58] S. Thrun. Probabilistic algorithms in robotics. *AI Magazine*, 21 :93–109, 2000.

[59] J. Trafton, N. Cassimatis, M. Bugajska, D. Brock, F. Mintz, and A. Schultz. Enabling effective human-robot interaction using

perspective-taking in robots. *IEEE Transactions on Systems, Man, and Cybernetics*, Part A :460–470, 2005.

[60] B. Tversky, P. Lee, and S. Mainwaring. Why do speakers mix perspectives ? *Spatial Cognition and Computation*, 1(4) :399–412, 1999.

[61] K. Wada and T. Shibata. Robot therapy in a care house - results of case studies -. In *Robot and Human Interactive Communication, 2006. ROMAN 2006. The 15th IEEE International Symposium on*, pages 581 –586, sept. 2006.

[62] M. Warnier, J. Guitton, S. Lemaignan, and R. Alami. When the robot puts itself in your shoes. managing and exploiting human and robot beliefs. In *Proceedings of the 21th IEEE International Symposium in Robot and Human Interactive Communication*, 2012.

[63] H. Wimmer and J. Perner. Beliefs about beliefs : Representation and constraining function of wrong beliefs in young children's understanding of deception. *Cognition*, 13(1) :103 – 128, 1983.