
Une architecture pour réaliser des robots autonomes

 Jean Gallantin — Emmanuel Peralta — Eric Bourreau

Adresse :

A définir sur plusieurs lignes par la commande `\address{...}`.

Email sur la dernière ligne.

Dans le cas d'adresses différentes des auteurs, indexer chaque auteur avec des astérisques comme `*` ou `**` obtenues par `\fup{*}` ou `\fup{**}`.

Voir aussi « Consignes aux auteurs »

RÉSUMÉ. Nous proposons une architecture bio-inspirée fixant un cadre au développement par un robot de son autonomie lors d'un apprentissage interactif. Les modèles psychologiques du développement qui l'inspirent portent sur les procédés de psychisation des animaux et des humains. Ce cadre permet d'aborder de manière expérimentale certaines questions liées à l'incorporation d'un esprit dans une machine. Les choix techniques pris pour le développement du démonstrateur le rendent indépendant du robot choisi. L'illustration est faite avec un robot Tribot piloté via URBI, utilisant des logiciels de planification par contrainte développés en JCHOCO, un logiciel d'apprentissage de la bibliothèque WEKA et un logiciel de contrôle normatif de son comportement INTEGRE. Une première expérimentation étudie comment un apprentissage interactif permet à un robot de contrôler un comportement réflexe.

ABSTRACT. This article presents a bio-inspired framework allowing to a robot to increase its cognitive performance during an interactive learning. The model is allow to

MOTS-CLÉS : A définir par la commande `\motscles{...}`

KEYWORDS: A définir par la commande `\keywords{...}`

Table des matières

1	Le cadre de conception de l'autonomie des machines	5
1.1	Autonomie	6
1.2	Introduction au cadre d'élaboration d'une autonomie	7
2	Le projet dubitoide	10
2.1	Expérimentation	10
2.2	Système de l'action	11
2.3	Système de perception	13
2.4	Système de diagnostic	13
2.4.1	Contraintes	14
2.4.2	Planification d'action	15
2.5	Le contrôle réactif	15
2.6	Système de décision	16
2.7	L'apprentissage	17
3	Conclusion	17

Introduction

L'autonomie est une faculté accordée aux être vivants, personnes, institutions mais pas encore aux machines. La faculté d'autonomie n'est pas pour autant définie par cette opposition. Pour avancer dans sa définition, nous proposons un langage commun et des concepts unificateurs permettant de la préciser tout autant pour des systèmes vivants qu'artificiels. Ce langage est en correspondance avec une architecture logicielle permettant à une machine, dotée de la capacité d'apprentissage, d'atteindre différents degrés d'autonomie lors d'une interaction avec des instructeurs.

La question de l'autonomie des machines provoque réactions et questionnement en Sciences Humaines et Sociales :

- 1) elle provoque des blocages culturels ;
- 2) elle pose la question de formaliser des systèmes capables d'évoluer ;
- 3) celle de formaliser le fonctionnement de l'esprit ;
- 4) et enfin celle de la substitution d'une machine au corps ;

Considérons le premier point. Pour des raisons venant de l'influence culturelle de leur religion dominante, alors que dans la culture orientale (REF LIPOUBOU), il est admis que des intelligences simples puissent animer des objets, la culture occidentale est restée longtemps réticente à admettre l'incorporation d'un esprit dans une machine. La culture occidentale oppose l'immanence d'un corps à la transcendance d'un esprit. Cette opposition se durcit quand on substitue une machine au corps. Des philosophes influents prennent position : Pascal comme Hobbes défendent que l'on peut ramener la pensée au calcul (REF parrochia).

Abordons le second point. Dans les années 70, dans son cours au collège de France sur le Neutre (REF CD BARTHES), Roland Barthes a défini la notion d'idéosphère comme la présentation des vues d'un système non dogmatique car inachevé. Pour Barthes, le marxisme, le freudisme sont des exemples d'idéosphères. Une idéosphère est initiée par un discours fondateur sur un système et elle permet d'établir des correspondances avec d'autres discours et de prendre des positions d'adhésion ou de refus. L'autonomie est une idéosphère ; les informaticiens devant spécifier, concevoir et réaliser des systèmes autonomes disposent d'exemples mais d'aucun formalisme adapté est achevé.

Abordons le point des travaux récents sur la formalisation du fonctionnement de l'esprit. S'inscrivant dans une mouvance structuraliste, Lacan (REF DOR) a posé ses bases du fonctionnement de l'esprit en considérant en particulier que l'inconscient est structuré comme un langage. Cette position rencontre celle prise en Intelligence Artificielle quand elle ramène la pensée à un langage formel en correspondance avec différentes logiques modales (REF IA). Mais maintenant de nombreuses études en pédopsychiatrie portent l'interaction précoce entre le bébé et son environnement et examine les processus de sémiotisation du nourrisson et en particulier l'usage de ses capacités de représentation à fin de communication (REF GOLSE A et B) .

Le dernier point porte sur la substitution d'une machine au corps. Nous avons maintenant de grandes réalisations de robots menées au Japon et en Corée qui ont promu la vision orientale de la machine intelligente. Ces réalisations permettent des études expérimentales sur les fonctionnalités attribuant des degrés d'autonomie à des machines. Dans les travaux menés par Oudeyer à l'INRIA (REF) portent sur une architecture respectant des principes de psychologie du développement et donnant à un robot bébé la possibilité d'être guidé dans sa construction d'un modèle du monde qui accorde ses perceptions aux actions potentielles. (REF) A Munich, il est développé un robot qui se déplace dans la ville sans GPS en demandant sa route aux piétons (?).

Présentons des perspectives plus futuristes (REF ECAP). Au Japon Ishii Kayoko s'intéresse à la co-construction de la conscience et de la schizophrénie chez les robots humanoïdes (REF Kayoko). Par ailleurs une réflexion Anthropologique (REF) renverse même les perspectives en pose les robots comme de quasi humains. La question de caractériser des états d'autonomie et d'intel-

ligence pour des machines en les associant à des capacités de pensée trouve désormais un intérêt autant conceptuel que pratique.

Après avoir présenté différents points de vue sur la question venant des SHS développons le point de vue STIC.

L'autonomie intéressant les roboticiens est celle qu'il faut donner aux systèmes pour qu'ils puissent acquérir des aptitudes d'interactivité avec d'autres systèmes, hommes et l'environnement et qu'ils acquièrent leurs propres critères d'optimisation.

Pour les informaticiens, il s'agit d'une part de définir formellement une machine autonome par un modèle indépendant de ses réalisations. Ce modèle doit permettre de définir comment passer d'une réalisation de l'autonomie à une autre, de composer des formes d'autonomies. Il s'agit d'autre part de réaliser des machines autonomes composées de machines autonomes en interaction qui s'adaptent à des environnements changeants. Il s'agit de les faire capable d'interagir avec des humains. Il nous faut faire des machines que l'on ne programme plus mais que l'on instruit car elles ont la faculté d'apprendre.

L'objectif de cet article est de présenter le cadre de conception commun pour les machines disposant de degré d'autonomie et d'intelligence. Comme on ne peut pas préjuger de ce que seront de telles machines, il s'agit ici de fixer langage assez riche pour permettre de parler des différents degrés d'autonomie de machines qui sont et seront fabriquées. Cette capacité va s'appuyer sur une architecture exprimant les invariants structurels reliant les fonctionnalités nécessaires à l'acquisition de cette autonomie. Cette architecture complète les architectures réactives à la Brooks tout comme des architectures comportementales.

Une architecture à la Brooks est fondée sur une boucle

ACTION \longrightarrow PERCEPTION \longrightarrow DIAGNOSTIC \longrightarrow ACTION

Une architecture comportementale est fondée sur une boucle

ACTION \longrightarrow PERCEPTION \longrightarrow DECISION \longrightarrow ACTION

L'architecture que nous posons intrique trois boucles

ACTION \longrightarrow PERCEPTION \longrightarrow DIAGNOSTIC \longrightarrow ACTION

ACTION \longrightarrow PERCEPTION \longrightarrow DECISION \longrightarrow ACTION

ACTION \longrightarrow PERCEPTION \longrightarrow DIAGNOSTIC \longrightarrow DECISION \longrightarrow ACTION

Les trois figures suivantes illustrent comment l'architecture proposée intègre les deux architectures.

Le plan de cet article est le suivant. Dans une première partie nous allons décrire et justifier le cadre que nous proposons pour construire l'autonomie d'une

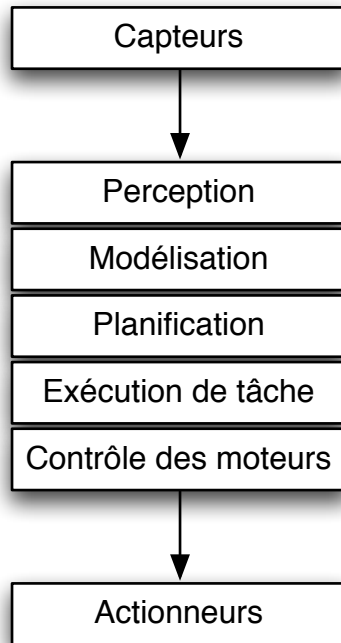


Figure 1. Schéma de plate-forme “Classique” selon Brooks

machine. Dans une seconde partie nous allons montrer comment le réaliser pratiquement et nous allons présenter quelques résultats. Dans la dernière partie, nous allons présenter les perspectives expérimentales et formelles ouvertes par ce cadre.

1. Le cadre de conception de l'autonomie des machines

Dans cette section, nous allons présenter le vocabulaire permettant de présenter comment décrire et réaliser des machines présentant différents degrés d'autonomie. Ce vocabulaire va s'établir sur une base structurelle qui a pour fonction d'organiser le traitement de l'information pour tout système ayant un degré d'intelligence. Nous allons introduire ce vocabulaire en nous appuyant sur des exemples impliquant des comportements humains mettant en évidence cette structure fondatrice.

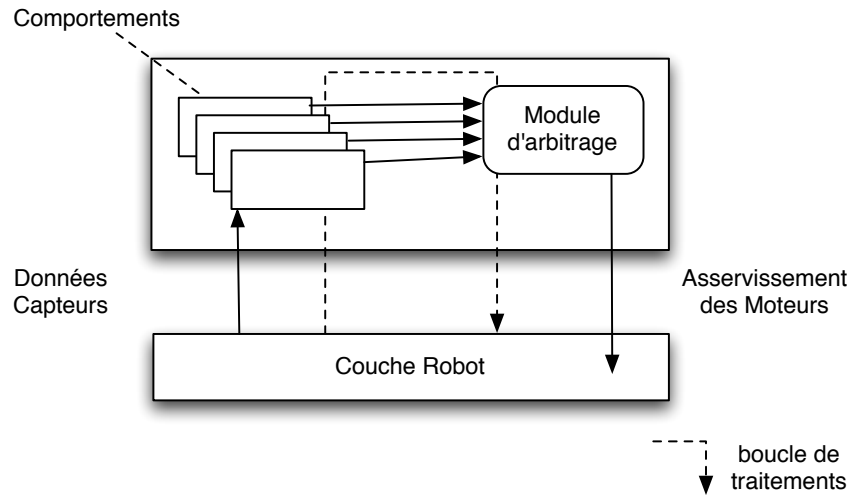


Figure 2. Schéma d'organisation des architectures comportementales.

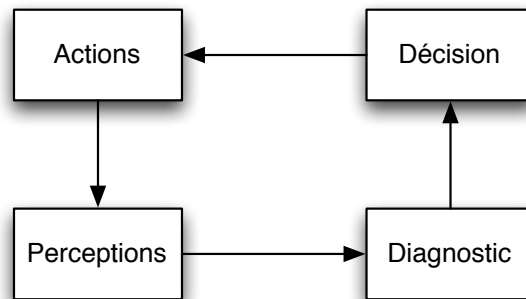


Figure 3. Liens entre Action Perception Décision et Diagnostic. ATTENTION RAJOUTER LES DIAGONALES

1.1. Autonomie

Comme nous l'avons présenté dans l'introduction, nous cherchons à préciser des invariants structurels qui président à la définition d'autonomies pour une machine. Nous allons poser deux principes.

Tout d'abord la notion d'autonomie ne doit pas dépendre de la réalisation matérielle de la machine. Comme l'a envisagé Marvin Minsky dans son livre

la société de l'esprit (REF MINSKY), une telle machine peut être vue comme une société. Cette vue est développée dans les systèmes multi-agents. On peut imaginer que cette machine a une taille astronomique, qu'elle soit à notre échelle ou encore qu'elle a une taille nanoscopique lui permettant d'intervenir dans la matière organique. Donc la notion d'autonomie ne doit pas être affectée par la matérialité de la machine.

En second lieu, une machine autonome doit avoir la possibilité d'atteindre tous les degrés d'intelligence identifiés en psychologie : stade du miroir, savoir se discerner de son environnement, savoir s'identifier comme un tout, savoir reconnaître son image, savoir s'imaginer, savoir reconnaître les autres, avoir conscience du réel, de ce qui vient, savoir dire "je". Les degrés d'intelligence sont des stades marquant une progression de la production de connaissance, sur les objets, sur soi et sur les autres. L'architecture de la machine doit permettre une telle progression. Les événements "de vie" de la machine vont motiver cette progression. Les dysfonctionnements d'une machine intelligente doivent se comprendre comme un mode de fonctionnement "dégradé" de l'architecture. L'architecture est ainsi le cadre qui précise la production d'une intelligence qui elle sera marquée par l'histoire.

Pour faire comprendre le rôle de l'architecture dans l'idéosphère de la machine autonome, considérons la différence faite en embryologie entre épigénèse et ontogénèse. L'épigénèse fixe le cadre de la croissance de l'embryon alors que l'ontogénèse en fixe les fonctionnalités. L'architecture que nous recherchons fixe un cadre épigénétique au développement étapes par étapes du système de pensée.

Notre premier principe est que la notion d'autonomie est indépendante de la nature matérielle d'une machine conçue en respectant l'architecture structurale d'une machine autonome. Notre second principe est que la notion d'intelligence, ses degrés, ses dysfonctionnements doivent s'expliquer en utilisant la description de l'architecture.

1.2. *Introduction au cadre d'élaboration d'une autonomie*

Ayant posé ces principes, nous allons présenter avec des exemples comment les sous systèmes composant cette architecture sont la condition de l'établissement de l'autonomie.

Notre premier exemple est celui de la conduite accompagnée. Considérons une voiture dans laquelle circulent l'apprenti et son parent l'instruisant. Le parent peut signaler à l'apprenti des événements signifiants (il y a un stop), il peut proférer un diagnostic qu'il infère (tu devrais freiner), il peut enjoindre de freiner (freine bon sang!!), mais il ne peut pas pour autant agir sur le frein ou sur le volant. Quatre systèmes sont en relation : le système de l'action, le système de la perception, le système du diagnostic, et le système de la supervision.

L'organisation de ces quatre systèmes est donnée sous la forme d'une boucle. Dans chaque système, sauf celui de l'action, il se passe une interaction entre l'apprenti et l'instructeur. Si nous avons pris pour exemple l'apprentissage de la marche, le parent serait intervenu également dans le système de l'action en soutenant le bébé.

Si l'apprenti est trop maladroit, inattentif, sans imagination ou sans discernement alors il sera incapable de réussir son examen de conduite. L'échec est imputé à la défaillance d'un sous système et non à l'histoire de l'apprentissage. Donc la défaillance d'un sous-système légitime la nécessité de la présence de ces quatre sous systèmes.

Changeons d'exemple et prenons celui de la découverte scientifique (REF DS). Nous la considérons comme menée par de très nombreux acteurs en interaction. Les actions sont celles des chercheurs, elles produisent des faits significatifs enregistrés comme des données dont on peut examiner la conjonction, la disjonction, la succession. Les diagnostics sont constitués par des articles exprimant des relations générales entre ces significatifs. Ces articles sont soumis à des rapporteurs qui en examinent la validité et suggère des révisions.

Dans le premier exemple, l'apprenti constructeur parvient au bout de 3000 Km à conduire de manière autonome car il ne sollicite que rarement et pour des raisons justifiées l'intervention de l'instructeur. Dans le second exemple, le jugement des pairs produit une chaîne de révision conduisant à des publications significatives.

En troisième exemple, signalons que la boucle Action, Perception, Diagnostic, Décision a été inventée pour accélérer le tempo des actions de l'armée américaine lors de la première guerre du Golfe. (REF)

Les trois exemples précédents illustrent des systèmes naturellement multi-agents dont la capacité d'intelligence vient de l'interaction entre agents humains. Il s'agit maintenant de voir comment un tel système peut rendre compte de la capacité d'acquisition d'autonomie pour un agent seul en interaction avec d'autre.

Nous allons considérer comme nouvel exemple d'un système constitué d'un bébé et de la personne qui s'en occupe. Nous allons poser que ce système est semblable à celui constitué de la voiture, l'apprenti et l'instructeur dans la situation de la conduite accompagnée. La différence essentielle et fondamentale vient de ce que la voiture est ici le corps du bébé. Dans cet exemple, l'interaction a pour fonction de permettre le développement des facultés mentales du bébé.

Développons ce dernier exemple. Tout d'abord il est avéré que l'interaction stimule un processus d'apprentissage qui perfectionne les systèmes d'Action, de Signification, de Symbolisation et de Communication du bébé. En effet, un bébé dont on ne s'occupe pas a un développement cérébral déficient. Dans le contexte du bébé, on peut distinguer deux phases de développement celle qui

précède le langage et celle qui s'appuie sur ce dernier. Les principes que nous imposons impliquent qu'il n'y a pas de changement de l'architecture structurant l'activité mentale du bébé lors du passage d'une phase à l'autre.

Cependant l'architecture précédente ne paraît pas complète pour la prise en compte d'un développement cérébral. Nous allons nous inspirer des recherches en pédopsychiatrie sur le bébé (REF GOLSE) et en psychanalyse (Ref DOR) pour la compléter. La question de fond largement développée dans ces disciplines est celle de comprendre comment une pulsion de vie illustrée par le besoin de téter du bébé doit se transformer en un désir d'interaction marqué par le désir d'être l'objet de l'attention de la mère et comment le bébé parvient à maîtriser une pulsion vitale en lui trouvant des objets de substitution.

Les modèles de fonctionnement de la pensée de Lacan sont inspirés de ceux de Freud et ils s'appuient sur les fondements de la linguistique en introduisant une relation signifiant/signifié telle qu'elle est définie par Saussure (REF DOR). Cette nouvelle relation n'est pas présente dans le diagramme du modèle précédent car elle se situe sur un axe liant le système de perception à celui de décision.

Nous venons donc de produire un vocabulaire permettant de parler de la manière dont lors d'interactions un système peut avoir la capacité de développer l'intelligence comme étant une capacité de penser. Nous allons maintenant présenter des études préliminaires menées au LIRMM sur la mise en pratique du modèle précédent.

L'informatique et la robotique cohabitent au LIRMM. Les roboticiens se préoccupent plus de mécanique et d'automatique. Dans le projet SHERPA (REF) il s'agit de fabriquer un bipède et de produire les lois de commande permettant la marche à un robot bipède. Les lois de commande recalculent à un tempo rapide les valeurs à donner aux actionneurs en fonction des valeurs des capteurs.

Le système est ainsi doté d'un contrôle réactif. Les problèmes de contrôle hybride ont été abordés dans une première thèse (?). Lors de cette thèse nous avons vérifié qu'un système à base de contrainte parvient à contrôler l'équilibre d'un robot aussi bien et dans le même tempo que les systèmes classiquement utilisés (REF). La thèse de Paulin montre comment conjuguer apprentissage d'unités de contrôle élémentaire et planification d'actions pour réaliser un contrôle réactif. Les unités de contrôle élémentaires sont les éléments du système de perception permettant de construire des plans.

La vidéo présente sur le site XXX montre comment un robot est capable de recalculer de manière compulsive des plans d'action quand son plan est contrarié. Un tel robot n'a aucune capacité de production d'une intelligence quelconque. L'étude que nous avons menée depuis s'est consacrée à la production du cadre structurel en entier de manière à tenter de reproduire les premières étapes de la réalisation d'un esprit lors d'interaction et d'apprentissage.

2. Le projet dubitoide

Le projet “dubitoide” (Ref ECAP) a pour ambition de produire un système logiciel respectant l’architecture précédente et capable de participer à l’attribution d’une certaine autonomie à une machine. Le nom dubitoide “dubito ergo sum” signale que l’activité cognitive du robot conçu dans ce cadre conceptuel va pratiquer sur un doute systématique pascalien.

Rappelons que nous avons posé deux principes : l’indépendance de l’architecture à la réalisation matérielle de la machine et l’indépendance de l’architecture aux degrés d’autonomie que la machine va pouvoir atteindre. Nous pourrions imposer un troisième principe qui serait celui de résoudre des problèmes généraux avec des outils généraux. En effet, deux sous système Diagnostic et Supervision utilisent une problématique informatique bien identifiée : la programmation par contraintes .(REF) et celle d’un système expert spécialisé dans le contrôle normatif de comportements .(REF).

Ces problématiques ont un très large champ d’application. Le système Action est réalisé avec le logiciel URBI de Gostai.(REF) Ce logiciel a pour avantage d’être utilisé pour commander de nombreux robots. Il n’est pas pleinement adapté à un contrôle réactif comme celui de l’équilibre du robot. Cependant URBI est fondé sur des principes d’ordonnancement synchrone ou asynchrone de tâches parallèles et séquentielles qui nous semblent exemplaires.

Le système de perception a été réalisé simplement par une gestion de fichiers enregistrant des séquences d’actions. Le jeu de flèches entre les sous-systèmes exprime des traductions. Certains objets d’un premier sous-système seront traduits en objets du second et certaines relations du premier seront traduites en relation du second. Par exemple les séquences d’actions sont traduites en enregistrement dans le système de perception.

La flèche diagonale qui exprime le contrôle réactif du diagnostic sur les actions a été spécialement étudiée car elle correspond à une problématique générale de génération de plan .(REF). Elle réalisée selon une problématique générique. La flèche diagonale qui exprime l’apprentissage symbolique interactif a été réalisée avec un souci de généralité. Les données du système de perception sont traduites dans les formats appropriés pour lancer les logiciels de la boîte à outil WEKA (?).

2.1. *Expérimentation*

Nous allons reprendre point par point les sous systèmes du robot. Pour chacun des systèmes nous décrirons ses objets et les relations entre ces objets. Nous donnerons le système informatique qui a servi à le réaliser et nous illustrerons avec l’expérimentation TRIBOT. Dans l’expérience TRIBOT de Mathias Paulin (?), le robot poursuit le biberon de façon réactive : tant qu’il

est capable de produire un plan pour atteindre le biberon, il replanifie sans répit. On aimerait qu'il soit capable de prendre la décision de s'arrêter en fonction d'un apprentissage du comportement de son instructeur.

Dans notre expérience, on définit deux instructeurs : le bon instructeur et le mauvais instructeur.

Le but de cette expérience est d'utiliser les fonctionnalités d'apprentissage pour arriver à distinguer le bon instructeur du mauvais, et lorsque c'est le mauvais le TRIBOT doit décider l'action d'arrêter de poursuivre son biberon et de demander de l'aide à son instructeur.

Définition 1 *Comportement du bon instructeur* Le bon instructeur ne déplace le biberon trois fois au maximum, et TRIBOT arrive à attraper le biberon.

Définition 2 *Comportement du mauvais instructeur* Le mauvais instructeur déplace sans cesse le biberon (donc plus de trois fois), et le TRIBOT est systématiquement puni après.

Les actions des instructeurs provoquent des perceptions enregistrées dans la mémoire épisodique du robot. Le but est que le robot parvienne à distinguer les deux comportements. Le robot peut ainsi apprendre à substituer au comportement compulsif de la recherche du biberon celui de la recherche d'une interaction avec son instructeur. Un tel mécanisme est pour les psychologues considéré comme une étape nécessaire à la construction d'une pensée.

2.2. Système de l'action

Les objets du système d'action sont commandées par les objets du système de diagnostic et par ceux du système de décision et elles agissent sur ceux du système de perception.

DECISION \longrightarrow ACTION

Action possible \longrightarrow Plan d'action

DIAGNOSTIC \longrightarrow ACTION

Action nécessaire \longrightarrow Plan d'action

Le diagnostic commande les actions réflexes et la décision les actions réfléchies. La décision et le diagnostic interviennent sur un système qui est en permanence en train d'agir. Le système devra décider ce qui doit se faire, continuer à se faire et ce qui peut se différer.

Le système est programmé en URBI qui gère avec une certaine autonomie l'agencement des tâches ce qui produit une certaine incertitude sur le fonctionnement du système. Comme une machine ayant une certaine autonomie n'est pas un automate cette incertitude est un phénomène obligé.

Les actions du système consomment du temps et ne peuvent se réaliser qu'après certaines actions. Les actions se composent selon deux opérations : leurs exécutions en parallèle notée AVEC ou leurs exécutions successives notées PUIS. Soit le système est informé de devoir attendre que l'une et l'autre soit en état d'être exécutées a lors on parle d'une loi de composition synchrone. Soit le système entreprend l'exécution d'une composition d'action quand l'une des actions est en état de se réaliser on parle d'un mode d'exécution asynchrone.

Deux modalités logiques président le choix des actions à faire celle de savoir si elle est possiblement à faire ou nécessairement à faire. Le contrôle réflexe venant du diagnostic propose des séquences d'action à faire nécessairement. Alors que la décision va ordonner des actions possiblement à faire. De bonnes décisions vont engendrer des actions qui vont produire des résultats qui seront des perceptions attendues.

Les actions seront décidées par le robot ou l'instructeur. Dans le cas de TRIBOT nous avons des actions de l'instructeur et des actions du robot. Les actions de l'instructeur ne sont pas codé en URBI mais elle vont être traduites en perception. L'action de souffler dans le micro produit un reset du robot.

L'action de porter permet à l'instructeur d'agir sur l'apprentissage en le facilitant et en le perturbant.

Moteurs	Objet URBI
Moteur de la roue droite	wheelR
Moteur de la roue gauche	wheelL
Moteur de la pince	claw

Figure 4. *Equivalence entre Moteurs et leurs objets Urbi.*

Les actions de l'instructeur sous forme de programme URBI

- Souffler dans le micro
- Porter.

Les actions du robot sous forme de programme URBI :

- Stopper puis alerter
- Avancer lentement puis vite
- Avancer lentement puis pincer
- Chercher
- Trouver

2.3. *Système de perception*

Le système de perception du robot est composé des signifiants et des relations entre signifiants déclenchés par les séquences d'actions qu'elles viennent du robot ou de l'instructeur.

ACTION \longrightarrow PERCEPTION

séquence d'action \longrightarrow enregistrement de séquences d'actions

séquences d'action \longrightarrow valeur des capteurs

Les signifiants pour le robot sont les enregistrements à des instants donnés les valeurs des capteurs, les valeurs des capteurs après l'action et la désignation de l'action réalisée.

Capteur	Objet URBI
Capteur de distance	sonar
Capteur de niveau sonore	sound
Capteur de luminosité	light
Indicateur de batterie	battery

Figure 5. *Equivalence entre Capteurs et leurs objets Urbi.*

Une mémoire épisodique est un cahier de labo qui enregistre les perceptions que le robot a de ses actions du robot sous la forme d'épisodes décrivant un comportement. Cette mémoire contient la trace des expériences réalisées (dans notre cas l'expérience attraper le biberon).

La mémoire épisodique se présente sous la forme d'un fichier XML il contient dans le préambule, le plan d'origine, la liste des action utilisées et la liste de descripteurs. Dans la seconde partie il contient l'exécution du plan, pour chaque action exécutée le cahier de labo contient les descripteurs pré et post action ainsi que la "prévision". Durant l'exécution du plan si une re-planification a lieu elle apparaît ainsi que le nouveau plan théorique.

Le résultat de l'expérience est + si le robot a réussi à saisir le biberon et - si il a été "puni" par l'instructeur.

2.4. *Système de diagnostic*

PERCEPTION \longrightarrow DIAGNOSTIC

valeurs des capteurs \longrightarrow Variables

Les valeurs des capteurs du robot sont reformulées sous la forme de valeurs prises par des variables. Le diagnostic va produire à partir de ces variables des plans d'actions. Le diagnostic agit sur le système d'action du robot en lui proposant des comportements nécessaires :

instance de plan d'action nécessaire \longrightarrow Séquence d'actions

et sur le système de décision en lui envoyant des comportements possibles.

Instance de plan d'action possibles \longrightarrow décision

Les relations entre les variables sont données par des contraintes. Le système va calculer en permanence des valeurs à attribuer aux actions pour permettre la réalisation de comportements possibles ou nécessaire.

Le choix de formuler par des contraintes un contrôle sur l'action ne préjuge pas que le système soit assujéti à d'autres contraintes venant de calculs numériques comme cela se fait couramment en robotique. Dans le contexte de l'expérimentation TRIBOT le diagnostic c'est uniquement réalisé par l'usage de Contrainte. Le logiciel de programmation par contrainte utilisé est JCHOCO.

2.4.1. Contraintes

La programmation par contraintes a été formalisée en 1974 par Montanari (?) et en 1977 par Mackworth (?). La programmation par contraintes est utilisée pour résoudre des problèmes d'affectation de valeurs à des variables. Ces variables sont liées entre elles par des relations que l'on nomme contraintes.

Cela revient à poser un problème sous forme de relations logiques entre plusieurs variables. Un problème à résoudre comporte un ensemble fini de variables. Chacune des variables possède un domaine et un certain nombre de contraintes, le domaine des variables peut être discret ou continu. Dans le cadre de nôtre approche on ne manipule que des réseaux de contraintes discrets.

Définition 3 (Réseau de contraintes) *Un réseau de contraintes \mathcal{R} est un triplet $(\mathcal{X}, \mathcal{D}, \mathcal{C})$ où :*

- $\mathcal{X} = \{x_1, \dots, x_n\}$ est l'ensemble des variables du problème ;
- $\mathcal{D} = \{d(x_1), \dots, d(x_n)\}$ est l'ensemble des domaines des variables du problème. Chaque variable x prends ses valeurs dans $d(x)$;
- $\mathcal{C} = \{c_1, \dots, c_m\}$ est un ensemble de contraintes sur \mathcal{X} . Une contrainte c est définie un ensemble de variables que l'on note $var(c)$, et une relation sur ses variables que l'on note $rel(c)$ qui spécifie les tuples autorisés par la contrainte.

Une contrainte est dite binaire lorsqu'elle implique exactement deux variables. Un réseau de contraintes est dit binaire lorsque l'ensemble \mathcal{C} ne contient que des contraintes binaires.

Définition 4 (Instanciation) *Soit $\mathcal{Y} = \{y_1, \dots, y_k\}$ où $\mathcal{Y} \subseteq \mathcal{X}$ une instanciation $E_{\mathcal{Y}}$ de \mathcal{Y} est un k -uplet $(v_1, \dots, v_k) \in (D_1 \times \dots \times D_k)$. On dit que l'instanciation est complète lorsque $\mathcal{Y} = \mathcal{X}$ (cette instanciation sera notée E).*

Une instantiation E_Y viole la contrainte $c \in \mathcal{C}$ si et seulement si

$$var(c) \subseteq \mathcal{Y}, E_Y[var(c)] \notin rel(c)$$

Définition 5 (Solution) Une solution est une instantiation complète telle que $\forall c \in \mathcal{C}, E[var(c)] \in rel(c)$. On note l'ensemble des solution d'un réseau $S(\mathcal{R})$, ou $S(\mathcal{X}, \mathcal{D}, \mathcal{C})$.

2.4.2. Planification d'action

STRIPS (ou STanford Research Institute Problem Solver) est un algorithme de Planification classique conçu par Richard Fikes et Nils Nilsson en 1971 (?). L'algorithme de STRIPS est assez basique, mais il est intéressant comme exemple pédagogique. On nomme aussi par ce nom le langage de représentation des données utilisée par l'algorithme. Avec le GPS (General Problem Solver) de Newell et Simon de 1961, il fait partie des premiers planificateurs utilisés en intelligence artificielle et été suivi de nombreux dérivés (GraphPlan, IPP, STAN, etc).

Définition 6 ($pre(a_i)$) $pre(a_i)$ représente l'ensemble des préconditions de l'action a_i

$$pre(a_i) = \{c_0, \dots, c_k\}$$

où k est le nombre de contraintes composant la précondition.

Définition 7 ($post(a_i)$) $post(a_i)$ représente l'ensemble des postconditions de l'action a_i

$$post(a_i) = \{c_0, \dots, c_k\}$$

où k est le nombre de contraintes composant la postcondition.

On préférera représenter les contraintes comme un ensemble de contraintes binaires plutôt que comme une conjonction afin de pouvoir plus facilement filtrer des contraintes notamment lors de la compilation de séquences d'action.

Définition 8 ($modified(a_i)$) $modified(a_i)$ représente l'ensemble des descripteurs modifiés par l'action a_i .

2.5. Le contrôle réactif

Le contrôle réactif s'effectue selon la boucle impliquant les systèmes de diagnostic, d'action et de perception qui caractérise une architecture à la Brooks..

$$DIAGNOSTIC \longrightarrow ACTION \longrightarrow PERCEPTION \longrightarrow DIAGNOSTIC$$

Le contrôle réactif est réalisé en traduisant des plans nécessaires directement en séquences d' action sans passer par la phase de décision.

Le plan prévu est réactualisé lors de cette boucle. Quand le plan est perturbé par l'instructeur ou par une erreur des capteurs, le système effectue une nouvelle passe de planification.

Deux mécanismes sont en place afin de permettre au robot d'être plus efficace :

- La parallélisation des actions. En effet la durée d'exécution du plan s'en retrouve plus courte et le robot dépense moins d'énergie ;
- La compilation d'actions fréquentes permet de diminuer la complexité de la passe de la planification ;

2.6. *Système de décision*

Le système de décision est influencé par les systèmes de perception et d'action. Il doit gérer toute la supervision du robot par lui-même comme par son instructeur.

DIAGNOSTIC \longrightarrow DECISION

variables \longrightarrow termes plan d'action possible \longrightarrow plan d'action

PERCEPTION \longrightarrow DECISION

Ensemble d'expériences \longrightarrow règles de décision

Ce système est un système expert permettant de formuler des règles sur des comportements exprimés par des séquences d'actions. Les règles du système peuvent être fournies par le constructeur du système, par les instructeurs ou encore par le robot lui-même qui les produit par apprentissage.

La logique du système intègre est inspirée par la logique juridique dont les lois, et les normes régulent des comportements tout en permettant la présence d'aléa et de défaillances de la part des acteurs. Les objets d'intègre constituent une ontologie de termes hiérarchisés et définis par des graphes conceptuels. Les relations d'intègre sont des ensembles de règles logiques. Une instance en intègre est une suite de termes instanciés. Un plan est une instance dans intègre. La supervision d'un plan va consister à en déterminer des contradictions et des incomplétudes.

Donnons un aperçu du réseau sémantique impliqué dans la prise de décision du Tribot. Quand le plan est d'attraper et que le biberon est entre 20 et 60 cm, si l'état précédent du comportement est d'approcher et que la cible est à plus 30 cm le comportement est d'approcher, si la cible est à moins de 30 cm l'état du comportement est attrapé, mais si la cible s'éloigne l'état du comportement redevient approcher.

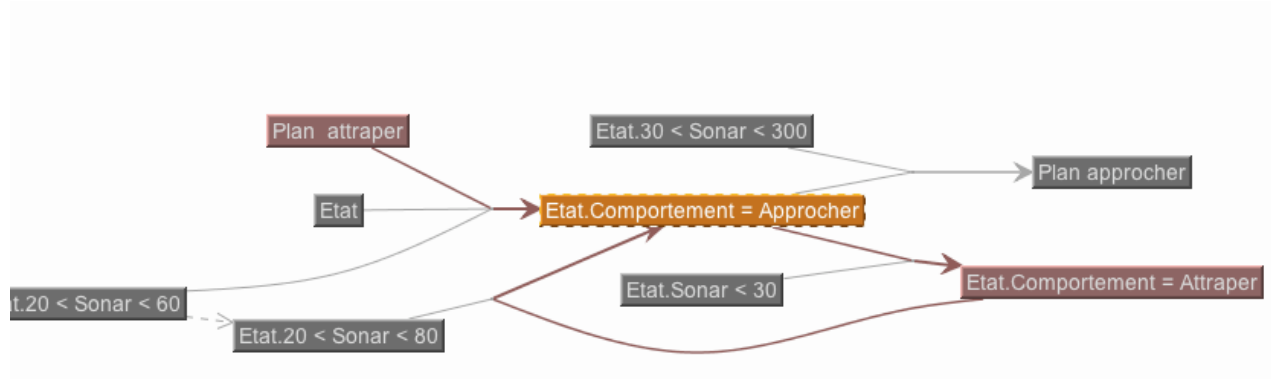


Figure 6. Un fragment du réseau sémantique de la prise de décision

2.7. L'apprentissage

L'apprentissage intervient dans la réalisation de la boucle comportementale.

ACTION \longrightarrow PERCEPTION \longrightarrow DECISION \longrightarrow ACTION

Le robot dispose de sa mémoire épisodique pour apprendre des règles de comportement. De nombreuses méthodes d'apprentissage sont candidates à réaliser cette boucle.

Par souci de simplicité, nous avons utilisé WEKA(?) afin de pouvoir tester différents classificateurs.

Comme la règle à apprendre est de type fréquentielle, (on veut que le robot stoppe sa recherche de biberon lorsqu'il l'a déjà cherché trois fois (ie. $frequency(FindTarget) > 3$)). On transforme donc chaque expérience de la mémoire épisodique en un vecteur de fréquence dont la taille est le nombre d'action du robot, et ou chaque élément est le nombre de fois ou l'action est apparue durant l'expérience.

On transforme la mémoire épisodique en un fichier ARFF dont les instances sont composées de la fréquence d'apparition de chaque action à l'intérieur d'une même expérience et du résultat de cette expérience. Le résultat de l'expérience représente la classe de l'instance.

3. Conclusion

Nous proposons une architecture générique pour caractériser un système autonome prenant en compte l'interaction, le contrôle réactif et l'apprentissage pour améliorer ses performances.

Cette architecture est indépendante de la réalisation matérielle et elle permet à un système réalisé matériellement d'accroître sa capacité de donner de la signification à ses actions.

Ce système est construit autour de trois boucles combinant quatre sous systèmes. Les deux mécanismes, l'apprentissage et le contrôle réactif, se conjuguent pour réaliser l'adéquation du système.

Cette architecture est bio-inspirée comme l'ont montré les nombreux exemples. Les outils informatiques proposés pour la réalisation du système informatique sont génériques et ils sont sous exploités dans l'application réalisée qui n'avait comme unique enjeu que de montrer que l'on pouvait réaliser effectivement cette architecture.

Ce travail ouvre de nombreuses perspectives applicatives mais surtout sur deux plans : le plan formel, le plan de sa signification en psychologie.

Sur le plan formel, le système présenté est susceptible d'une définition formelle en théorie des catégories(REF). Chaque sous système est une catégorie. Les trois boucles sont réalisées par des foncteurs. Les diagonales correspondent à des adjonctions. Une formalisation catégorielle pourrait permettre établir des correspondances avec d'autres formalisations comme cela est établi entre les différents formalismes du calcul en informatique théorique.

Dans cet état d'esprit, il est possible de regarder quelle logique intuitionniste correspond à ce modèle. Chaque système à ses propres modalités : il y aurait des modalités permettant de gérer le temps (système de l'action), le risque (système de la décision), la génération d'hypothèses (système du diagnostic) et les modalités existentielles (boîte de la perception).

Sur le plan de sa signification psychologique, nous comptons approfondir sa bonne correspondance avec les diagrammes proposés par Lacan pour expliquer les processus de construction et de fonctionnement de l'esprit. Ref DOR. En effet, Lacan dispose d'un diagramme similaire dont il explique longuement l'importance des deux flèches diagonales pour permettre à nos esprits d'organiser la relation entre le conscient et l'inconscient. Ces approfondissements permettraient une meilleure connaissance du processus de psychisation tant pour les humains que pour les machines.