

TABLE DES MATIÈRES

I L'humain virtuel comportemental	1
1 La sélection d'action	3
1.1 Introduction	3
1.2 Systèmes réactifs	3
1.2.1 Les systèmes stimuli-réponses	4
1.2.2 Les systèmes à base de règles	5
1.2.3 Les automates	6
1.2.4 Synthèse	8
1.3 Systèmes cognitifs et orientés buts	9
1.3.1 Le calcul situationnel	9
1.3.2 STRIPS : une vision simplifiée du calcul situationnel	10
1.3.3 HTN : les réseaux de tâches hiérarchiques	12
1.3.4 Les mécanismes de sélection d'actions	14
1.3.5 Les systèmes BDI	15
1.3.6 Synthèse	16
1.4 Comment choisir un modèle décisionnel ?	17
1.5 Quels sont les liens avec les modèles proposés en sciences cognitives ? .	19
1.5.1 La nature hiérarchique des niveaux de comportement	19
1.5.2 Mécanismes attentionnels pour la sélection des actions	20
1.5.3 Mécanismes d'activation et d'inhibition	20
1.5.4 Théorie de l'activité	21

1.5.5	Cognition incarnée située	23
1.5.6	Bilan	23
1.6	Conclusion	24
	Vocabulaire	29
	Table des illustrations	31

Première partie

**L'humain virtuel
comportemental**

1 LA SÉLECTION D'ACTION

Fabrice Lamarche et Stéphane Donikian

1.1 INTRODUCTION

Les mécanismes humains aboutissant à la sélection d'une action à réaliser sont complexes et pour certains encore assez mal connus. Les processus décisionnels peuvent être classés en deux grandes catégories selon qu'ils fassent intervenir ou non un processus conscient. Les processus qualifiés de *réactifs* représentent des comportements pré-cablés ne faisant pas directement intervenir de processus conscient. Pour leur part, les processus dits *cognitifs* font intervenir une représentation abstraite des connaissances et permettent de sélectionner une suite d'actions cohérentes visant à atteindre un but donné. Ces processus permettent d'effectuer une projection dans le futur afin de raisonner sur les conséquences des actions. L'objectif de ce chapitre est de présenter les grandes techniques qui ont été proposées pour modéliser la sélection d'action pour les humains virtuels ou plus généralement pour les entités autonomes dans le domaine de l'informatique. Les techniques présentées sont issues de différents domaines de recherche tels que l'éthologie, l'intelligence artificielle ou encore l'animation comportementale. Ces techniques diffèrent principalement par le niveau d'abstraction qu'elles proposent ainsi que par la typologie de comportement qu'elles permettent de décrire / générer. Certaines visent à décrire des comportements de type réactif alors que d'autres font intervenir une représentation abstraite des connaissances permettant de générer un comportement visant à réaliser un but donné. Par similitude avec les processus humains, ces techniques peuvent alors naturellement être classées dans deux catégories : les systèmes réactifs et les systèmes cognitifs et orientés but. Ce chapitre va donc s'articuler autour de la présentation de ces deux catégories de modèles puis se conclura sur quelques questions / réponses visant à guider le lecteur sur le choix d'un modèle décisionnel adapté à ses besoins.

1.2 SYSTÈMES RÉACTIFS

Les systèmes réactifs permettent de modéliser des comportements plus ou moins complexes n'utilisant pas directement de modélisation abstraite des connaissances. Ces systèmes sont caractérisés par le fait qu'ils relient la perception à l'action sans effectuer de raisonnement nécessitant une projection dans le temps. Ces comportements, qualifiés de réactifs, peuvent être classés dans deux catégories : les comportements réflexes et les comportements spécialisés. Les comportements réflexes sont des comportements pour lesquels un organisme va réagir directement à un stimulus perçu. Cette réaction est adoptée sans intervention d'un mécanisme décisionnel. A titre d'exemple, le maintien de l'équilibre entre dans ce type de comportement. La seconde catégorie, correspond à un ensemble de comportements spécialisés. La fréquence d'utilisation de tels comportements, couplée au mécanisme de renforcement, les a ancrés au plus profond du processus décisionnel de telle sorte que leur utilisation ne nécessite plus aucun effort conscient. Plusieurs typologies de méthodes peuvent être utilisées pour reproduire / générer de tels comportements. Ces méthodes diffèrent principalement par le niveau d'abstraction qu'elles proposent, leur facilité d'emploi et d'évolution.

1.2.1 LES SYSTÈMES STIMULI-RÉPONSES

Les systèmes stimuli-réponses font partie des premières générations de modèles utilisés pour modéliser les comportements réactifs. Ils sont issus des travaux développés en éthologie [Wihelms e.a. , 1989, Brooks, 1991] dans lesquels le comportement est considéré comme la réponse à une série d'interactions directes avec l'environnement. De façon générale, de tels systèmes sont représentés par un réseau de noeuds interconnectés transformant une information perçue en commandes envoyées aux effecteurs.

Le connexionnisme s'inspire des théories biologiques sur l'organisation du cerveau pour représenter un comportement sous la forme de réseaux de neurones formels. Un neurone formel est défini par l'intermédiaire d'une fonction d'activation (sigmoïde, gaussienne...) de la forme $y = f(x)$ où x représente l'excitation en entrée du neurone et y l'excitation en sortie de ce dernier. Les réseaux de neurones correspondent ensuite à l'interconnexion de ces neurones formels à l'aide de liens pondérés (corrélés à la notion de synapse en biologie) reliant la sortie d'un neurone à l'entrée d'un ou plusieurs autres neurones. L'activation en entrée de chaque neurone est ensuite calculée comme une combinaison linéaire de l'activation en sortie des neurones connectés et des pondérations associées aux liens. L'activation des neurones dépend donc de deux facteurs : la topologie du réseau créé ainsi que la valeur des pondérations associées aux liens. En jouant sur ces deux paramètres, il est possible de modifier la fonction représentée par le réseau. Il s'agit du rôle des différents algorithmes d'apprentissage (algorithmes génétiques, rétro-propagation...) qui peuvent être utilisés pour configurer ces réseaux en fonction d'un comportement attendu. Cette notion permet aux réseaux de neurones d'être adaptatifs et de généraliser les réactions apprises. Mathématiquement, les réseaux de neurones formels sont des approximateurs universels. Autrement dit, ils peuvent approximer une fonction continue à n'importe quelle précision. Cette propriété a notamment été utilisée par Grzeszczuk qui a utilisé, dans NeuroAnimator [Grzeszczuk e.a., 1998], des réseaux de neurones pour modéliser les règles de la physique d'un environnement au même titre que le comportement des entités le peuplant. Pour sa part, Van de Panne a proposé l'utilisation de telles techniques au travers des réseaux SAN (Sensor Actuator Networks) pour contrôler la locomotion [van de Panne e.a. , 1993].

La principale qualité de cette approche réside dans les processus d'apprentissage qui à partir d'une base d'exemples pertinents permettent de configurer le système de manière automatique. Cependant, cette approche possède deux inconvénients majeurs. Le premier réside dans le fait que les systèmes produits sont des boîtes noires non interprétables. Pour effectuer une modification du comportement attendu, il faut alors refaire un apprentissage complet du système. De plus, il est difficile voire impossible de prévoir les réactions du réseau. Seuls des tests permettent de vérifier la qualité de ses réactions. Le second défaut réside dans l'apprentissage lui-même. En effet ce dernier s'avère parfois difficile à paramétrer et la capacité de généralisation du système est grandement liée à l'apprentissage lui-même. Dans le cas où un tel réseau a appris des exemples par coeur, il n'est plus à même de généraliser et peut générer des réactions incohérentes face à des stimuli situés en dehors de l'ensemble d'apprentissage.

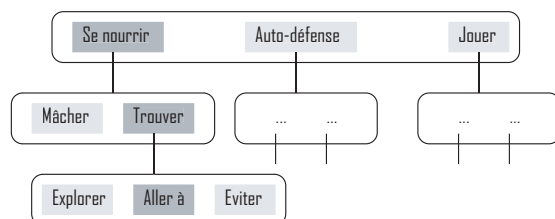


Figure 1.1 : Arbre de décision extrait des travaux de Blumberg [Blumberg, 1997].

1.2.2 LES SYSTÈMES À BASE DE RÈGLES

Tout comme les systèmes précédents, les systèmes à base de règles prennent en entrée des informations issues des capteurs et / ou de l'état interne de l'entité afin de choisir une action à réaliser. Le comportement est ici modélisé par un ensemble de règles associant une condition à une action. L'ensemble des actions dont les conditions sont satisfaites peuvent alors être mises en compétition ou encore fusionnées afin d'être réalisées. Ce type de système, de par l'utilisation de règles, offre des moyens de description plus abstraits que le précédent et s'avère interprétable et extensible.

Les systèmes à base de règles diffèrent principalement par leur mode de représentation des conditions. Ces systèmes peuvent travailler sur un ensemble de règles planes de type si alors [Laird e.a., 1987, Laird, 2001]. Dans ce cadre, la concurrence entre règles (i.e. que faire quand plusieurs règles sont applicables) doit être gérée. Plusieurs stratégies peuvent être adoptées : seule la première règle satisfaite est prise en compte, les règles sont dotées de priorités et la plus prioritaire est choisie, les résultats des règles sont fusionnés [Reynolds, 1987]... Les règles peuvent aussi être organisées sous la forme d'un arbre de décision [Coderre, 1989, Blumberg, 1997] (Cf. fig. 1.1). Dans cet arbre, les feuilles représentent des actions alors que les noeuds sont des experts en charge d'effectuer un choix entre tous leurs sous-noeuds. Les branches de l'arbre peuvent aussi être pondérées afin de faciliter le travail des experts. Sur cette base, de nombreux algorithmes de parcours d'arbres de décision peuvent être utilisés (essais successifs, séparation et évaluation progressive, glouton, parcours ascendant...). Certaines de ces approches, telle que celle de Blumberg [Blumberg, 1997], permettent de choisir des actions à exécuter en parallèle. Les actions associées à l'arbre de décision sont étiquetées par l'ensemble de ressources qu'elles utilisent. Lorsque des choix sont effectués, plusieurs actions peuvent être sélectionnées dans la mesure où elles n'utilisent pas les mêmes ressources.

Le grand avantage des systèmes à bases de règles réside dans la facilité d'interprétation et les possibilités d'extension du système (suppression/modification/ajout de nouvelles règles). Cependant, il reste parfois difficile de correctement gérer la concurrence entre plusieurs règles ainsi que les éventuels conflits liés à l'ajout de nouvelles règles, particulièrement dans les systèmes utilisant des règles planes. Dans ce cadre l'utilisation d'arbres de décision facilite la gestion des conflits. Il reste cependant un point noir. Certains comportements possèdent une logique intrinsèque qui s'exprime par l'exécution consécutive de plusieurs actions. Dans un système à base de règles, cette typologie de comportement s'avère difficile à décrire puisque toutes les actions sont exprimées sur un même plan, sans explicitation de leurs enchaînements potentiels.

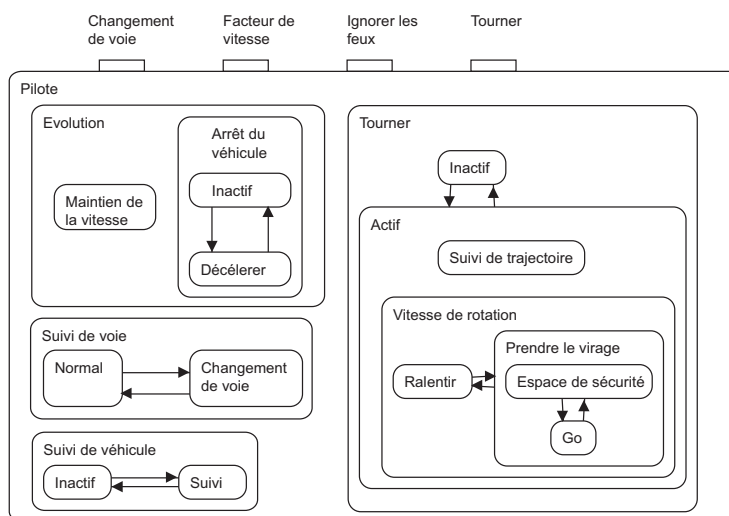


Figure 1.2 : Exemple d'automates hiérarchiques décrivant le comportement d'un conducteur de véhicule [Cremer e.a., 1995].

1.2.3 LES AUTOMATES

Les approches à base d'automates permettent de décrire des comportements sous la forme d'un enchaînement conditionnel d'actions. Prenons l'exemple du comportement adopté par une personne devant se cacher dans un jeu de cache-cache. Dans un premier temps, la personne se cache puis attend. Durant cette attente, deux cas peuvent se poser : soit la personne est touchée durant sa phase d'attente et une nouvelle partie recommence, soit la personne est vue et s'enfuit. Ici, l'enchaînement des actions dépend du contexte (touché, vu...) mais aussi de la dernière action effectuée. Pour décrire ce type de comportement, le formalisme des automates à états finis est utilisé. Un état de l'automate correspond alors à une tâche considérée comme unitaire à un certain niveau d'abstraction et les transitions entre états traduisent les conditions d'enchaînement de ces mêmes tâches.

Plusieurs typologies de systèmes permettant de décrire et gérer l'exécution de comportements via des automates ont été proposées. Les *piles d'automates* ont été originellement utilisées pour décrire le comportement d'un joueur de tennis [Noser e.a. , 1997]. Le système est basé sur l'exploitation d'une pile qui permet de transposer la notion bien connue en informatique d'appel de fonction à la gestion des automates. Lorsqu'un automate demande l'exécution d'un autre automate, le premier automate est empilé et le second est lancé, lorsque le second automate est terminé, le premier est dépilé et reprend son exécution. N. Balder a proposé l'utilisation d'*automates parallèles*, au travers des Pat-Nets [Badler e.a. , 1995], pour décrire le comportement d'humanoïdes de synthèse. Dans ce cadre, l'exécution de plusieurs automates en parallèle permet de décrire des comportements très complexes comme résultant du produit implicite d'automates représentant des comportements simples. A titre d'exemple l'exécution en parallèle de deux automates possédant 10 états simule un automate produit comportant 100 états. Pour leur part, les *automates parallèles hiérarchiques* permettent de représenter

un comportement sous la forme d'une hiérarchie d'automates fonctionnant en parallèle. A chaque niveau de cette hiérarchie, l'automate père peut posséder plusieurs rôles : un rôle de supervision de l'exécution de ses automates fils ou un rôle de synthèse d'un comportement par filtrage et / ou fusion des propositions calculées par ses automates fils. Parmi les systèmes proposant ce type de représentation, on peut citer le modèle HCMS [Cremer e.a., 1995] et le modèle HPTS [Donikian e.a. , 1995, Donikian, 2001]. La particularité du modèle HPTS est de prendre en compte des notions temporelles (permettant de modéliser des temps de réaction par exemple) ainsi que d'introduire une forme d'indéterminisme lors du choix des transitions (cela permet d'obtenir une variabilité dans les comportements). Ces modèles ont été utilisés pour décrire le comportement de conducteurs de véhicules [Cremer e.a., 1995, Moreau e.a. , 1998] (Cf. Fig. 1.2) ou encore de piétons évoluant au sein de villes virtuelles [Thomas e.a. , 2000].

Dans les modèles utilisant le parallélisme d'automates, certains aspects liés à la concurrence des comportements peuvent poser problème. Par exemple, supposons que deux comportements veuillent contrôler l'animation du bras d'un humanoïde de synthèse. Si les deux comportements prennent simultanément le contrôle de ce bras, des résultats incohérents peuvent être produits. Les mécanismes proposés par les automates hiérarchiques laissent la responsabilité, au programmeur du système de s'assurer de la cohérence du comportement généré. Cependant, dans le cadre de comportements très complexes, nécessitant une coordination fine, gérer ce type de synchronisation peut nécessiter un effort de description tel que cela devient pour ainsi dire impossible. Afin d'automatiser la gestion de ce type de synchronisation, les PaT-Nets utilisent un mécanisme de sémaphore associé à un état de l'automate pour gérer l'exclusion mutuelle [Badler e.a. , 1995]. Cependant, l'utilisation directe de ce concept issu de la programmation système pose quelques problèmes. D'une part, les comportements doivent être décrits en ayant la connaissance des autres comportements décrits afin d'être apte à éviter le traditionnel problème des inter-blocages. D'autre part, ce système ne permet pas, sans communication explicite entre les automates, de relâcher une ressource pour qu'elle soit utilisée par un comportement plus prioritaire. Afin de proposer une solution généraliste aux problématiques de l'exclusion mutuelle et de la gestion de l'importance relative des comportements, ainsi qu'à la description des adaptations potentielles en fonction des disponibilités / besoins en ressources, une extension du modèle HPTS, nommée HPTS++, a été proposée [Lamarche e.a. , 2002]. La particularité de cette approche est d'inclure des notions de priorités, de ressources et de degré de préférence. La priorité est une fonction dynamique associée à un automate. Elle permet de spécifier l'adéquation entre un comportement et un contexte. Les ressources sont associées aux états de l'automate et servent à gérer des exclusions mutuelles. Deux automates en cours d'exécution ne peuvent utiliser la même ressource en simultanée. Dans le cadre de l'animation d'humanoïde de synthèse, ces ressources peuvent correspondre aux bras, aux jambes, à la bouche ou encore à l'attention visuelle par exemple. Les degrés de préférence sont associés aux transitions des automates et décrivent les différentes possibilités d'adaptation. Ils permettent de signifier qu'une transition favorise la réalisation du comportement ou propose une adaptation visant à libérer une ressource qui pourrait être utilisée par un automate plus prioritaire. Un exemple de comportement utilisant ce formalisme est fourni figure 1.3. Cet automate décrit un comportement consistant à boire ou à fumer depuis la saisie de l'objet d'intérêt (verre, cigarette) jusqu'à la fin de la consommation qui peut être provoquée par la terminaison normale de la tâche (verre vide, fin de cigarette) ou par adaptation (degré de préférence négatif associé aux transitions). Sur la base de ces informations le système propose un algorithme d'ordonnement disposant d'un mécanisme de prévention des inter-blocages et gérant l'adaptation automatique des comportements en fonction des besoins / dispo-

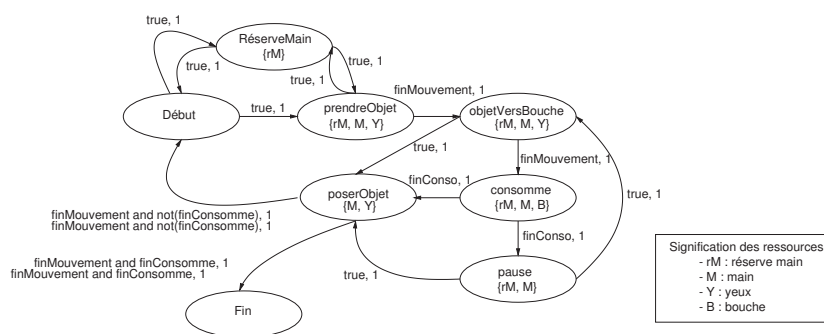


Figure 1.3 : Automate HPTS++ décrivant les comportements de boire ou fumer.

nibilités de ressources. La coordination des ensembles de comportements actifs devient alors générique et automatique. Ce système a notamment permis de reproduire automatiquement le comportement d'une personne buvant un café, fumant une cigarette, tout en lisant un livre à partir d'une description indépendante des trois comportements.

Le succès des automates dans le domaine de l'animation comportementale s'explique par la puissance d'expression du formalisme. En effet, les automates permettent de facilement modéliser des processus nécessitant la description d'un enchaînement conditionnel d'actions, enchaînement qui dépend à la fois de la dernière action exécutée (état de l'automate) et du contexte direct (conditionnelles associées aux transitions). L'aspect hiérarchique de certains de ces systèmes permet de considérer des actions à différents niveaux d'abstraction, facilitant ainsi la description de processus complexes. L'aspect concurrentiel / coopératif associé à la notion de hiérarchie d'automates, ainsi que le parallélisme, renforce le pouvoir d'expression de ce type de système. Enfin, certains algorithmes d'ordonnancement tels que ceux ajoutés dans HPTS++ facilitent la gestion du parallélisme tout en répondant à des problématiques importantes dans le cadre de la description des comportements : la gestion de la concurrence sur les ressources (corporelles et autres) et l'adaptation du déroulement des comportements en fonction de leur importance relative.

1.2.4 SYNTHÈSE

Nous venons de présenter trois typologies de systèmes réactifs. La particularité des systèmes stimuli-réponses, face aux systèmes à base de règles ou aux automates, réside dans la possibilité de modéliser un processus sans connaissance de son fonctionnement. En effet, cette approche permet de générer de manière automatisée un contrôleur à partir d'exemples caractérisant le domaine à modéliser. La contre partie réside dans l'aspect boîte noire de ces systèmes qui les rend peu interprétables et difficiles à entretenir. A contrario, les systèmes à base de règle ou les automates nécessitent la connaissance du mode de fonctionnement du processus à modéliser. Cependant, dans la plupart des cas, cette propriété est vérifiée. D'autre part, le fait que ces systèmes s'avèrent interprétables et facilement modifiables fait qu'ils sont actuellement privilégiés dans le domaine de la modélisation du comportement. La différenciation entre les systèmes à base de règles et les systèmes à base d'automates se situe principalement dans la granularité des processus à définir et dans la gestion de la cohérence temporelle

du comportement. Les automates, si l'on souhaite décrire finement des comportements nécessitant une certaine cohérence temporelle dans l'enchaînement des actions, sont à privilégier. A contrario, si la gestion d'une cohérence temporelle n'est pas une nécessité, les systèmes à base de règles s'avèrent très adaptés.

1.3 SYSTÈMES COGNITIFS ET ORIENTÉS BUTS

Contrairement aux systèmes réactifs, les systèmes cognitifs et orientés buts nécessitent une représentation abstraite du monde afin de rechercher une suite d'actions permettant de satisfaire un but donné. Le comportement n'est donc plus complètement décrit mais bel et bien calculé. La recherche des actions permettant de satisfaire un but utilise une forme de projection dans le futur permettant de raisonner sur les conséquences des actions. Dans cette section, nous allons donc présenter un ensemble de modèles permettant de gérer de tels comportements. Comme nous le verrons, ces modèles diffèrent principalement par leur coût de calcul, le niveau d'abstraction qu'ils proposent, l'exploitation d'une connaissance opératoire, leur réactivité face à des changements imprévus de l'environnement ainsi que leur capacité à exploiter des opportunités.

1.3.1 LE CALCUL SITUATIONNEL

Le calcul situationnel (*situation calculus* [McCarthy e.a. , 1969]) a été introduit par J McCarthy et P.J. Hayes vers la fin des années 60. Il s'agit d'un formalisme largement utilisé dans le domaine de l'intelligence artificielle permettant de décrire des mondes très complexes ainsi que les actions qui peuvent être effectuées par des entités le peuplant. Cette approche est basée sur six concepts permettant de représenter le monde du raisonnement :

- **Les situations** représentent l'état complet du monde à un instant donné. Elles sont généralement décrites par un ensemble de faits.
- **Les fluents** décrivent une propriété dynamique du monde. Ces derniers sont définis comme des fonctions prenant une situation en paramètre renvoyant l'état de la propriété. Généralement, ils représentent un fait atomique (il pleut) ou encore une relation entre des objets du monde (le parapluie est dans le placard).
- **Les causalités** représentent des relations de cause à effet. Elles permettent de déduire l'état de certains *fluents* en fonction de l'état d'autres *fluents*.
- **Les actions** permettent de modifier une situation. Elles sont décrites par l'intermédiaire de pré-conditions et d'effets. La pré-condition d'une action se traduit par une expression booléenne utilisant les *fluents*. Elle qualifie un sous état de la situation qui doit être satisfait pour pouvoir exécuter l'action. Les effets de l'action décrivent la manière dont la situation est modifiée par l'action. Ces effets peuvent être inconditionnels (indépendants de la situation) ou conditionnels (dépendant de la situation).
- **Les stratégies** représentent des enchaînements d'actions préconçus. Elles correspondent à des étapes de raisonnement scriptées.
- **La connaissance** permet de modéliser le fait que l'entité connaisse ou non l'état de certaines propriétés du monde. Par ce biais, il est possible de décrire des actions permettant d'acquérir de la connaissance (actions perceptives) et / ou dépendant du niveau de connaissance de l'entité sur l'état de l'environnement.

L'utilisation de ce formalisme permet de calculer des ensembles de mondes futurs, résultant de l'exécution d'une ou plusieurs actions à partir de la situation courante. Malgré son potentiel élevé, ce formalisme souffre d'un problème lié à la description des actions. Ce problème est qualifié de problème de la fenêtre [Scherl e.a. , 2003]. Il réside

dans la description de tout ce qui ne change pas dans le monde après l'exécution d'une action. Dans la pratique, cela s'avère quasiment impossible. Pour pallier ce problème, l'hypothèse du monde clos est utilisée. Cette hypothèse énonce que tout ce qui n'est pas explicitement décrit comme étant la résultante d'une action reste inchangé après la réalisation de cette même action.

Sur la base de ce formalisme, il est possible de décrire un but comme une expression booléenne portant sur des *fluents*. Ce but qualifie donc un sous état d'une situation. La planification consiste alors à trouver une suite d'actions permettant de générer une situation dans laquelle le but est satisfait. Les algorithmes utilisés pour la planification explorent l'arbre des mondes possibles qui peut être de très grande taille. Supposons qu'en moyenne, n actions puissent être réalisées à partir d'une situation donnée. De plus, supposons que le plan permettant d'atteindre le but soit constitué de k actions. Le nombre de mondes à explorer est alors de l'ordre de n^k . Certaines optimisations portant sur l'élagage de l'arbre de recherche permettent de réduire le nombre de calculs effectués mais la recherche d'un plan dans un monde complexe reste tout de même très coûteuse. Plusieurs environnements de programmation permettent d'exploiter la puissance du calcul situationnel. Parmi ceux-ci, on peut citer GOLOG (alGOL in Logic) [Levesque e.a., 1997] et plus récemment CML (Cognitive Modelling Language) [Funge e.a., 1999]. Dans CML, J. Funge a ajouté la notion d'IVE (Interval Valued Fluent) [Funge, 1999] permettant de gérer un degré d'incertitude sur une valeur numérique. Des actions de perception peuvent alors être déclenchées pour remettre à jour la connaissance. D'autre part, CML est à cheval entre un langage déclaratif et un langage impératif. L'aspect impératif permet de décrire des processus spécialisés représentant des stratégies permettant de guider les processus de planification.

Le calcul situationnel tire sa force de son grand pouvoir d'expression qui permet de modéliser des mondes très complexes. Sa plus grande faiblesse réside dans les algorithmes de planification qui peuvent s'avérer très coûteux en temps et en mémoire. D'autre part, la phase de planification suppose que l'environnement ne pourra pas changer en dehors des actions de l'entité. Si plusieurs entités peuplent le même environnement, il se peut alors que certains plans en cours de réalisation soient invalidés par les actions des autres entités. Dans ce cas, il faut effectuer une nouvelle planification. Il s'agit d'un problème récurrent associé à tous les algorithmes dont le fonctionnement réside dans la génération d'un plan d'action complet.

1.3.2 STRIPS : UNE VISION SIMPLIFIÉE DU CALCUL SITUATIONNEL

Le formalisme STRIPS (STandford Research Institute Planning System) exploite un sous ensemble du calcul situationnel [Fikes e.a., 1971] permettant ainsi de proposer des algorithmes de planification plus efficaces. Le monde est ici décrit sous la forme de faits pouvant être présents ou absents au sein d'une situation. Ces faits peuvent être corrélées à la notion de *fluent* possédant une valeur booléenne où *vrai* signifie présent et *faux* signifie absent. Ici encore, les faits représentent soit une propriété atomique, soit une relation entre des objets du monde. Les actions sont ensuite définies par l'intermédiaire d'opérateurs paramétrables possédant une pré-condition et des effets. La pré-condition regroupe un ensemble de faits qui doivent être présents dans une situation pour que l'action puisse être exécutée. Les effets d'une action se traduisent par l'intermédiaire de deux listes : une liste d'ajout et une liste de retrait de faits. La liste d'ajout contient l'ensemble des faits qui sont ajoutés lors de l'exécution de l'action tandis que la liste de retrait traduit l'ensemble des faits qui sont supprimés lors de

l'exécution de cette même action. La notion de but s'exprime alors sous la forme d'un ensemble de faits qui doivent être présents dans une situation. Un exemple d'opérateurs STRIPS est fourni figure 1.4. Ces opérateurs permettent de résoudre le problème des blocs visant à déplacer des blocs de manière à ce que ces derniers forment une pile dont l'arrangement est fourni comme but. La planification consiste alors à trouver une suite d'actions générant une situation incluant le but. Parmi les algorithmes permettant de planifier des suites d'actions au sein du formalisme STRIPS, deux algorithmes sortent du lot : GRAPHPLAN [Blum e.a. , 1997] et HSP (Heuristic Search Planning) [Bonet e.a. , 2001].

Opérateur	put(?x, ?y)
PRE	: ontable(?x), clear(?x), clear(?y)
ADD	: on(?x, ?y)
DEL	: ontable(?x), clear(?y)
Opérateur	put(?x, ?y)
PRE	: on(?x, ?z), clear(?x), clear(?y)
ADD	: on(?x, ?y), clear(?z)
DEL	: on(?x, ?z), clear(?y)
Opérateur	put-table(?x)
PRE	: clear(?x), on(?x, ?y)
ADD	: ontable(?x), clear(?y)
DEL	: on(?x, ?y)

Figure 1.4 : Exemple d'opérateurs STRIPS définissant le problème des blocs.

Le mode de fonctionnement de GRAPHPLAN [Blum e.a. , 1997] consiste à générer un graphe en couche, alternant des noeuds de type action et des noeuds de type situation. La construction du graphe part de l'état actuel du monde i.e. la situation de départ. Sur la base de cette situation, toutes les actions dont les pré-conditions sont satisfaites sont appliquées pour générer une nouvelle situation contenant les effets de toutes les actions réalisables. Cette nouvelle situation peut contenir des faits conflictuels i.e. des ensembles de faits qui ne peuvent être présents simultanément. De même parmi les actions sélectionnées, certaines peuvent aussi être conflictuelles i.e. les effets d'une action invalident les pré-conditions d'une autre action, ou les deux actions possèdent des effets contradictoires. Les couches de situation ainsi que les couches d'actions possèdent donc des marqueurs notifiant les actions et les faits conflictuels. La planification consiste alors à trouver une couche de situation dans laquelle le but est satisfait et pour lequel les faits le constituant ne sont pas marqués comme conflictuels. Dans un deuxième temps, un mécanisme de recherche avec chaînage arrière est utilisé pour trouver une suite d'actions non conflictuelles permettant de satisfaire le but. Deux cas peuvent se présenter. Si l'algorithme trouve une suite d'actions non conflictuelles alors le plan est généré. Si l'algorithme ne trouve pas de suite d'actions non conflictuelles, le graphe est étendu et le mécanisme de recherche est réutilisé jusqu'à réussite de construction du plan. Cet algorithme garantit de trouver une solution (dans la mesure où elle existe) à un problème donné en minimisant le nombre de couches de graphe utiles à la planification. La solution trouvée possède une propriété importante : lorsque qu'un ensemble d'actions constituant le plan appartiennent à la même couche du graphe, ces actions peuvent être réalisées en parallèle car elles ne sont pas conflictuelles. L'algorithme minimise donc le temps de réalisation du plan mais ne garantit pas de trouver un plan minimisant le nombre d'actions à réaliser. Le dévelop-

pement des couches d'actions et de situations est polynomial en temps et en mémoire [Blum e.a. , 1997]. L'algorithme, dans le cas de problèmes difficiles, peut donc ne pas fournir de solution par manque de mémoire.

Pour sa part, l'algorithme HSP [Bonet e.a. , 2001] utilise une heuristique pour guider la recherche d'un plan. Cette heuristique est directement extraite de la description des actions sous la forme d'opérateurs STRIPS. L'idée du calcul de cette heuristique est de considérer un sous ensemble du problème à résoudre en ne considérant pas les faits supprimés par la réalisation d'une action. Il est alors possible de définir un minorant de la distance (exprimée en nombre d'actions) entre une situation et le but à atteindre. L'algorithme HSP possède plusieurs variantes en fonction de l'algorithme de recherche utilisé (*hill climbing search*, WA^* , ...) et du mode d'évaluation de l'heuristique (de l'état courant vers le but ou du but vers l'état courant). L'un des avantages de cet algorithme réside dans le nombre de situations développées lors de la recherche qui s'avère relativement faible et permet à HSP de surpasser GRAPHPLAN dans la taille des problèmes traités. Contrairement à GRAPHPLAN, HSP ne cherche pas à paralléliser les actions mais permet de trouver des plans minimisant le nombre d'actions à effectuer pour réaliser un but.

Le formalisme STRIPS s'avère moins expressif que le calcul situationnel. Cependant, cette restriction d'expressivité permet de proposer des algorithmes de planification plus performants en termes de temps de calcul et d'occupation mémoire. Ces modèles peuvent résoudre des problèmes complexes à partir du moment où ils s'expriment au sein du formalisme. Au même titre que le calcul situationnel, la phase de planification suppose que l'environnement ne pourra pas changer en dehors des actions de l'entité. Dans le cas où les plans en cours de réalisation sont invalidés, il faut alors effectuer une nouvelle planification.

1.3.3 HTN : LES RÉSEAUX DE TÂCHES HIÉRARCHIQUES

La planification HTN (Hierarchical Tasks Networks) diffère des modes de planification présentés jusqu' alors. Plutôt que de décrire un ensemble d'actions atomiques permettant de modifier le monde et de chercher une suite linéaire d'actions satisfaisant un but, la planification par HTN fonctionne de manière hiérarchique [Erol e.a., 1994] en s'appuyant sur trois concepts : les tâches, les méthodes et les actions.

- **Les tâches** constituent ce que l'on peut qualifier de but dans les méthodes de planification qui ont été présentées précédemment. Elles peuvent correspondre à la satisfaction d'un ensemble de propriétés sur le monde (comme dans le calcul situationnel ou les domaines STRIPS), à la réalisation d'un ensemble d'actions, comme visiter une ville, ou encore à une combinaison des deux. Elles acceptent de prendre un certain nombre de paramètres, rendant ainsi possible la description de tâches générales pour lesquelles les paramètres seront instanciés lors d'une demande de planification. Le pouvoir d'expression associé à la notion de tâche est donc très grand. Elles permettent notamment d'exprimer des comportements complexes n'ayant pas forcément une influence exprimable sous la forme de faits sur l'environnement.
- **Les méthodes** correspondent à la manière de réaliser une tâche. Une méthode décrit une suite ordonnée de sous-tâches et/ou d'actions à réaliser pour remplir la tâche à laquelle elle correspond. Elle possède des pré-conditions décrivant l'état du monde nécessaire à sa réalisation. Une notion intéressante de la planification HTN apparaît alors : il est possible de décrire plusieurs méthodes permettant de réaliser la même tâche. Ces méthodes permettent donc de décrire la base de connaissance disponible

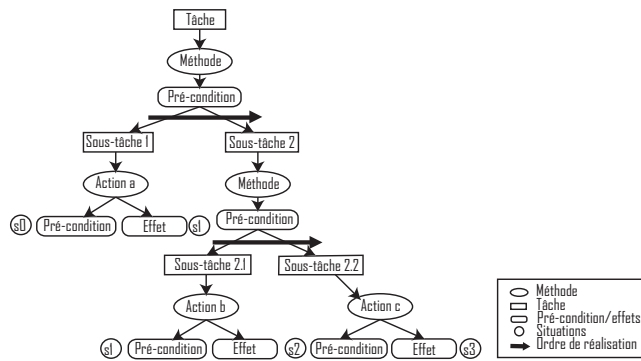


Figure 1.5 : Exemple de plan généré par planification au sein des HTN.

pour la planification. Les pré-conditions permettent d'ajouter des restrictions à l'utilisation d'une méthode, par exemple, un taxi peut être utilisé pour voyager sur des distances relativement courtes, alors qu'un avion peut être utilisé pour voyager sur de très longues distances. La liste ordonnée de sous-tâches décrit une forme de plan plus ou moins abstrait permettant de guider la recherche d'une solution et réduisant ainsi l'espace de recherche.

- **Les actions** sont représentées via le formalisme STRIPS, elles possèdent des pré-conditions et des effets. Elles sont directement réalisables, sous réserve de la véracité de leur pré-condition, sans avoir recours à une forme de décomposition.

Les concepts qui viennent d'être exposés dénotent de la conception hiérarchique de la planification HTN. Une demande de planification se traduit donc par la demande de réalisation d'une tâche. Le processus de planification consiste alors à décomposer la tâche par l'intermédiaire des différentes méthodes proposées. Ce processus de décomposition génère un arbre " et / ou " dont l'élagage est guidé par les pré-conditions associées aux méthodes. Les noeuds de type " ou " représentent le choix entre les différentes méthodes permettant de réaliser une même tâche alors que les noeuds de type " et " représentent la succession d'actions et / ou de tâches associées à une méthode. La planification consiste alors à explorer ce graphe " et / ou " en recherchant une suite d'actions (les feuilles de l'arbre " et / ou ") de telle manière que ces actions réalisent la tâche au sommet de l'arbre (Cf. figure 1.5). L'algorithme s'appuie sur une exploration d'arbre avec chaînage arrière, utilisant le formalisme STRIPS associé aux actions pour vérifier la validité du plan en cours de construction.

La propriété de la planification HTN réside dans la mise à disposition des concepts de méthode et de tâche. D'une part, les tâches permettent d'exprimer des buts qui ne se traduisent pas forcément par un ensemble de faits. D'autre part, contrairement au formalisme STRIPS, cette description permet d'introduire une connaissance experte dans le système tout en laissant une certaine latitude dans les choix par l'intermédiaire de la description de plusieurs méthodes permettant de réaliser une même tâche. Ainsi la réalisation d'une tâche peut être décrite / contrôlée finement, ce qui explique son utilisation dans le domaine de la fiction interactive pour la description de scénarios [Cavazza e.a., 2002]. Cependant, même s'il a été prouvé que ce formalisme est plus expressif que le formalisme STRIPS, le biais introduit par la décomposition en tâches offre moins de flexibilité quant à l'expression et la réalisation des buts [Charles e.a., 2003].

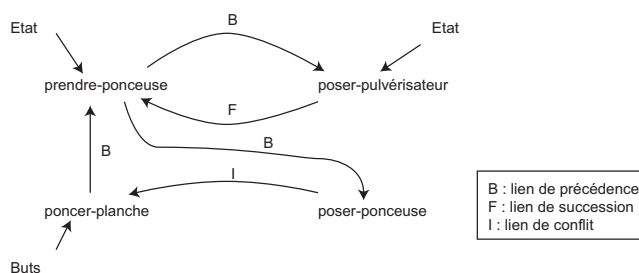


Figure 1.6 : Exemple de réseau d'action issu des travaux de P. Maes [Maes, 1990].

1.3.4 LES MÉCANISMES DE SÉLECTION D' ACTIONS

Les mécanismes de sélection d'action ont été introduits pour gérer deux aspects du comportement situé : la réactivité et l'opportunisme face aux changements de l'environnement ainsi que la recherche d'une suite d'actions visant à satisfaire un but. Contrairement aux approches précédentes, le mode de fonctionnement de ces systèmes n'est pas de générer un plan complet d'actions permettant de réaliser un but donné, mais de choisir, de manière incrémentale, des actions promettant de converger vers le but. Lorsqu'une action est choisie, elle est réalisée. Une fois cette action réalisée, un nouveau choix d'action est effectué, prenant ainsi en compte les changements de l'environnement et l'acquisition de nouvelles connaissances.

Ces mécanismes sont basés sur un graphe d'actions dont les liens définissent des canaux de propagation d'énergie. Le principe est alors de connecter un but à ce réseau et de considérer ce but comme un générateur d'énergie. L'action la plus activée et possédant sa pré-condition satisfaite sera alors l'action promettant de converger vers le but à satisfaire. Pionnière dans ce domaine, P. Maes a proposé un mécanisme de sélection d'actions basé sur une description par l'intermédiaire du formalisme STRIPS [Maes, 1990] utilisant une description non paramétrée des actions. Le graphe associé à ce mécanisme est construit de la manière suivante. Chaque nœud du graphe est une action. Entre les nœuds, trois types de liens sont utilisés : les liens de succession, des liens de précédence et les liens d'inhibition. Considérons deux actions A et B . Un lien de succession entre A et B est créé pour chaque fait appartenant à la liste d'ajouts de A et à la pré-condition B . Les liens de précédence sont les réciproques des liens de succession. Un lien d'inhibition est ajouté entre A et B pour chaque fait appartenant à la liste de retrait de A et aux pré-conditions de B . Dans ce système, les buts sont exprimés par un ensemble de faits qui doivent être présents dans la situation. Lorsqu'un but est fourni, il est lié au réseau via des liens de précédence ajoutés pour chaque action possédant l'un des faits dans sa liste d'ajout et d'inhibition pour chaque action possédant l'un des faits dans la liste de retrait. Un exemple d'un tel réseau d'action, consistant à poncer une planche, est fourni figure 1.6. L'algorithme de sélection consiste alors à propager une énergie dans le réseau d'actions. Chaque action réalisable (dont la pré-condition est satisfaite) envoie une fraction de son énergie via ses liens de succession. Chaque action dont la pré-condition n'est pas satisfaite envoie une fraction de son énergie via ses liens de précédence. Enfin, une fraction d'énergie négative est répandue via les liens d'inhibition. Les buts sont des émetteurs d'énergie. L'action alors choisie est l'action dont l'activation dépasse un certain seuil et maximisant l'activation des actions réalisables. Cependant, dans le cadre de la spécification de plusieurs buts, ce

mécanisme montre ses limites [Tyrell, 1994]. D'une part, le système peut osciller entre plusieurs actions sans jamais prendre de choix définitif. D'autre part, le mode de calcul de la dispersion d'énergie défavorise les plans ayant beaucoup de possibilités de réalisation aux profits de ceux possédant peu de degrés de liberté. B. J. Rhodes, a proposé une extension de ce système au travers des PHISH-Nets [Rhodes, 1996] permettant de contrecarrer certaines de ces limitations et ajoutant la notion de paramétrage des actions. Cependant, le problème des oscillations subsiste. Pour leur part, V. Decugis et J. Ferber [Decugis e.a. , 1998] ont proposé un mécanisme de sélection hiérarchique dans lequel les niveaux les plus bas concernent les comportements réflexes et les niveaux les plus élevés intègrent des comportements plus abstraits. A chaque niveau de la hiérarchie, un mécanisme d'arbitrage est alors utilisé pour choisir parmi les actions proposées par les mécanismes de sélection d'action sous jacents. Cette architecture possède quelques similarités avec les automates parallèles hiérarchiques et permet de rendre le processus décisionnel plus robuste.

Les mécanismes de sélection d'actions présentent de très bonnes propriétés pour l'animation comportementale. Le monde est ici considéré comme dynamique et peut donc changer sans l'intervention de l'agent. Du point de vue de l'agent, ces changements sont pris en compte à chaque nouvelle sélection d'action, lui permettant de réagir rapidement tout en exploitant des opportunités offertes par l'environnement. Ces mécanismes permettent donc d'allier réactivité et planification. Cependant, ces systèmes souffrent de deux défauts. D'une part, ils peuvent osciller dans le choix des actions. D'autre part, ils ne sont pas complets. Autrement dit, même si un plan d'action existe pour satisfaire un but, rien ne garantit que ces systèmes puissent le trouver.

1.3.5 LES SYSTÈMES BDI

Le mode de raisonnement associé aux systèmes BDI (Beliefs Desire Intentions) s'inspire du raisonnement pratique de l'être humain [Bratman, 1987]. La première implémentation de ce type de système, au travers d'une formalisation logique est attribuée à Rao [Rao e.a. , 1995]. Les initiales BDI réfèrent à trois concepts : les croyances, les désirs et les intentions.

- **Les croyances** symbolisent la connaissance que l'agent possède de l'état du monde ou plus précisément ce que l'agent croit vrai. Elles sont caractérisées par une incomplétude due aux capacités de perception réduites de l'agent et à l'hypothèse de départ sur la dynamique du monde qui rend impossible la prédiction de son état.
- **Les désirs** représentent les motivations de l'agent. Cette notion peut être assimilée à un but mais ne possède pas une formalisation telle que dans le calcul situationnel ou le formalisme STRIPS. La notion de désir se rapproche de la notion de tâche dans le formalisme HTN.
- **Les intentions** regroupent l'ensemble des plans que l'agent adopte en vue de réaliser ses désirs.

L'hypothèse de départ des systèmes BDI est que l'agent possède un certain nombre de plans, qui fournissent une méthodologie à appliquer pour satisfaire ses désirs. Le plan représente donc une forme de connaissance opératoire injectée dans le système par le concepteur. Un plan est constitué d'étapes pouvant prendre plusieurs formes : l'ajout d'un nouveau désir, la modification des croyances de l'agent ou l'exécution d'une action atomique. Ces plans possèdent aussi des conditions d'application qui doivent être satisfaites pour leurs permettre d'être adoptés comme des intentions. Ces conditions peuvent porter sur l'état du monde ou encore l'état interne de l'agent.

La sélection des actions s'effectue alors de la manière suivante. Lorsque l'agent veut satisfaire un désir, il recherche un plan le réalisant et dont les conditions d'application sont remplies. Si plusieurs plans peuvent être utilisés, un mécanisme d'arbitrage est mis en oeuvre pour choisir celui qui semblera le plus adapté. Lorsqu'un plan est sélectionné, il devient une intention. Il est alors exécuté étape par étape jusqu'à ce qu'une nouvelle sélection soit requise. Cette sélection peut être liée à l'émergence d'un nouveau désir, à la réussite ou à l'échec d'un plan en cours de réalisation ou encore à un changement observé de l'environnement. Ce système s'avère efficace car il exploite une connaissance experte fournie par l'intermédiaire des plans. L'enchaînement des actions est géré automatiquement au travers de la demande de satisfaction de désirs, qui peuvent alors être assimilés à des sous-buts. Les plans réfèrent donc à d'autres plans et ce raffinement est effectué jusqu'à l'obtention d'actions atomiques pouvant être exécutées. La rapidité du système est liée au fait qu'aucun raisonnement sur les actions n'est effectué. La première action réalisable lors de l'évaluation est exécutée, puis en fin d'exécution, la suite du plan est évaluée ou remise en cause.

Le mode de sélection d'action choisit donc les actions de manière incrémentale en prenant en compte, à chaque nouveau choix effectué, l'acquisition de nouvelles connaissances. Cela rend le système très réactif aux changements de l'environnement. Ici, la notion de projection dans le futur passe par l'évaluation des pré-conditions associées aux méthodes et par l'utilisation de plans préfabriqués. Cependant, l'agent est peu "inventif", il ne peut résoudre des situations imprévues, car il n'est pas doté de capacités de raisonnement sur ses actions. Si un désir ne dispose pas de plan pouvant directement le satisfaire, il est simplement omis. D'autre part, ce système pose le problème de la compatibilité des désirs [Rao e.a., 1991], qui par leur manque de formalisation réelle s'avère difficile à vérifier [Thanagarajah e.a., 2002].

1.3.6 SYNTHÈSE

Nous venons de présenter plusieurs systèmes permettant de décrire des comportements orientés buts. Ces modèles possèdent chacun leurs caractéristiques et diffèrent sur plusieurs points importants.

- **La réactivité et l'opportunisme.** Les systèmes générant des plans complets d'action (calcul situationnel, STRIPS, HTN) supposent que le monde ne peut être modifié que par l'agent lui-même. Cette contrainte rend ces systèmes peu réactifs face aux modifications de l'environnement et nécessitent, dans certains cas, de refaire une planification complète qui peut s'avérer très coûteuse en calculs. D'autre part, ces systèmes n'exploitent pas explicitement les opportunités offertes par l'environnement car ces dernières, pour être prises en compte, nécessitent une nouvelle planification. Sur ce point, les systèmes tels que les mécanismes de sélection d'action ou les systèmes BDI, de par le choix incrémental des actions effectué en fonction des nouvelles connaissances acquises, possèdent de meilleures propriétés.
- **La capacité à satisfaire un but.** Modulo le temps de calcul et l'occupation mémoire, les systèmes de type calcul situationnel et STRIPS garantissent de trouver un plan d'action (si ce dernier existe) satisfaisant un but donné. Les autres systèmes, même s'ils possèdent un ensemble d'actions atomiques permettant de satisfaire un but donné ne garantissent pas cette propriété.
- **Le mode d'expression du but.** Deux typologies d'expression des buts sont proposées. Les systèmes de type calcul situationnel et STRIPS gèrent des buts décrits sous la forme d'une formule logique portant sur les faits décrivant l'état de l'environnement. Les systèmes de type HTN, BDI s'appuient sur la notion de tâche ou

d'intention pour ne gérer qu'un ensemble de buts spécifiés à priori par le concepteur du système. L'avantage de ce second type de description réside dans la possibilité de décrire des buts difficiles à exprimer sous la forme de sous-situations (au sens calcul situationnel) à atteindre. Bien évidemment, le désavantage réside dans l'aspect potentiellement restrictif qui ne permet pas de considérer des buts n'ayant pas explicitement été inclus dans le système.

- **La connaissance opératoire.** Les systèmes tels que HTN et BDI utilisent une connaissance opératoire injectée lors de la description des modèles. Si cette connaissance opératoire limite l'inventivité des modèles en contraignant leur sélection d'action, elle peut aussi être vue comme un atout lorsqu'il s'agit de modéliser des procédures. De plus, dans le cadre de HTN, cette connaissance opératoire accélère le processus de planification. Dans les systèmes de type BDI, elle permet d'effectuer très rapidement un choix d'action sans utiliser de projection dans le futur. Les autres systèmes créent un mode opératoire à partir d'une description d'actions atomiques. Le biais qui peut alors être introduit dans les plans générés est de ne pas respecter une forme de " logique humaine " qui se traduit par un manque de crédibilité des comportements exhibés. Il est à noter que le calcul situationnel, par l'intermédiaire des stratégies permet aussi de prendre en compte une forme de connaissance opératoire, tout en incluant cette connaissance dans un schéma de raisonnement plus généraliste garantissant la complétude du modèle.

Ces différents points permettent de caractériser et de différencier les différents modèles cognitifs orientés buts présents dans la littérature. Il n'existe pas réellement de modèle disposant des caractéristiques optimales mais plutôt un ensemble de modèles proposant des solutions à diverses problématiques liées à la modélisation de la sélection d'action. Nous pouvons cependant observer que les modèles les plus généralistes deviennent rapidement coûteux en calculs alors que les modèles utilisant une connaissance opératoire, même s'ils sont moins inventifs, disposent d'une meilleure réactivité.

1.4 COMMENT CHOISIR UN MODÈLE DÉCISIONNEL ?

Dans les sections précédentes, nous avons présenté un ensemble de modèles utilisés pour effectuer une sélection d'action. Ces modèles sont assez variés, possèdent des caractéristiques différentes et modélisent différents concepts utiles à la description / gestion du comportement d'un humanoïde virtuel autonome. Cependant, au-delà de la variété des modèles, une question importante se pose : comment choisir un modèle décisionnel lorsque l'on souhaite modéliser le comportement d'un humanoïde autonome ? Il n'existe malheureusement pas de recette miracle et les choix doivent être effectués en fonction des besoins liés au domaine applicatif. Afin de guider le lecteur dans le choix d'un tel modèle, nous allons poser un certain nombre de questions et fournir quelques éléments de réponse.

1. *L'utilisation d'un modèle cognitif / orienté but est-elle nécessaire ou un modèle réactif est-il suffisant ?* Il faut être conscient qu'un modèle cognitif / orienté but peut avoir un coût de calcul important et que son utilisation implique de lourdes contraintes de modélisation au niveau de l'environnement. En effet, nombre de modèles orientés but fonctionnent sur une abstraction de l'environnement. Il faut alors être à même de gérer le lien entre les données numériques décrivant un environnement de simulation et la représentation abstraite des connaissances nécessaire aux modèles de raisonnement. Les modèles de type réactifs ainsi que le modèle BDI n'utilisent pas forcément une représentation abstraite de l'environnement et peuvent donner des résultats très concluants si l'environnement de simulation est suffisamment contraint pour ne pas nécessiter une forme d'inven-

tivité sur l'enchaînement des actions d'un humanoïde.

2. *Lors de l'utilisation d'un modèle orienté but, l'ensemble des buts à satisfaire est-il connu à l'avance ou est-il généré en cours d'application ?* Le grand atout des approches de type calcul situationnel, STRIPS et de certains mécanismes de sélection d'action, est de proposer des techniques très généralistes permettant de résoudre des problèmes exprimés sous la forme de formules logiques décrivant un sous état de l'environnement à atteindre. Cette propriété permet d'imaginer une application générant automatiquement des formes de buts, en fonction d'un contexte et / ou en fonction d'interactions entre les humanoïdes de synthèse. Cependant, l'utilisation de tels systèmes a un coût : le calcul situationnel et le STRIPS sont consommateurs de temps et de mémoire, les mécanismes de sélection d'action pour leur part peuvent osciller et ne pas atteindre les buts alors qu'un plan existe. Dans un grand nombre d'applications, le domaine d'expression des buts peut être connu à l'avance de par l'aspect contraint de la simulation. Dans ce cadre, il est préférable d'utiliser des systèmes de type HTN ou BDI, voire des processus réactifs jouant le rôle de scripts, qui offriront des calculs plus rapides et moins consommateurs de mémoire.
3. *Le domaine modélisé nécessite-t-il la description de procédures ?* Si la réponse à cette question est positive, il faut considérer l'utilisation de systèmes de type automates, HTN ou BDI. En effet, ces systèmes permettent de décrire le comportement d'humanoïdes dotés de la capacité à adapter (modulo description) le cours de leurs actions à des modifications de l'environnement tout en restant dans un cadre contraint permettant de respecter des procédures décrites. Cependant, ces systèmes sont peu inventifs et peuvent donc ne pas trouver de solutions face à une situation imprévue. Leur description nécessite donc une attention toute particulière.
4. *L'environnement de simulation est-il hautement dynamique ?* Dans le cadre d'environnements hautement dynamiques où l'état du monde peut changer très rapidement, il faut considérer l'utilisation de systèmes ayant un temps de réponse rapide et ayant peu de présupposés sur le fait que l'environnement ne change pas en dehors des actions de l'agent. Les systèmes alors à privilégier sont ceux de type réactif ainsi que les systèmes BDI et les mécanismes de sélection d'actions. En effet, ces derniers ont été conçus pour répondre à cette problématique.
5. *Les temps de réaction des modèles sont-ils importants ?* Dans le cadre d'applications interactives, telles que celles réalisées en réalité virtuelle, le temps de réaction des humanoïdes, pouvant être en interaction avec un humain réel est primordial. Il faut donc concevoir des systèmes aptes à réagir dans des temps paraissant raisonnables pour un être humain. Nous en venons ici encore à privilégier des systèmes de type réactif, BDI ou éventuellement HTN pour leur capacité à prendre rapidement une décision.
6. *Souhaite-t-on exploiter le parallélisme des actions ?* Le comportement humain est caractérisé par la capacité à réaliser plusieurs tâches en simultané, modulo la disponibilité de certaines ressources. Les systèmes à base de règles permettent de sélectionner des actions à effectuer en parallèle mais dans ce cadre, la granularité du parallélisme reste grossière. Les systèmes à base d'automates permettent de décrire des systèmes hautement parallèles et certains de ces systèmes, tels que le modèle HPTS++, fournissent tous les outils nécessaires à la gestion automatisée d'un parallélisme fin entre comportements. Au niveau des algorithmes de planification, un algorithme tel que GRAPHPLAN, basé sur le formalisme STRIPS, permet de générer des plans contenant des actions à réaliser en parallèle.

Ces différentes questions ne se veulent pas exhaustives. Il faut bien voir que le choix d'un modèle donné possède toujours des avantages et des inconvénients. Seule une identification précise des besoins liés au domaine applicatif peut permettre de prendre une décision raisonnable. Il est cependant utile de noter que dans un grand nombre de cas, un humanoïde autonome n'a pas besoin de savoir jouer aux échecs ou encore de savoir résoudre le problème des tours de Hanoï pour exhiber un comportement crédible au sein d'un environnement virtuel.

1.5 QUELS SONT LES LIENS AVEC LES MODÈLES PROPOSÉS EN SCIENCES COGNITIVES ?

Une autre façon d'analyser les modèles décrits précédemment peut être celle de leur adéquation avec les modèles introduits en sciences cognitives. Nous allons ainsi balayer quelques notions importantes qu'il nous semble nécessaire de voir présentes au sein de ces modèles. Pour chaque notion, nous tenterons d'identifier la liste des modèles de sélection d'action l'incluant.

1.5.1 LA NATURE HIÉRARCHIQUE DES NIVEAUX DE COMPORTEMENT

La nature hiérarchique des niveaux de comportement est aujourd'hui communément admise. Pour A. Berthoz [Berthoz, 2003], "Le problème complexe du contrôle a été résolu par une hiérarchie de niveaux de contrôle qui, chacun, s'exerce sur des modèles internes du système qui le précède". La théorie du contrôle en psychologie comportementale, telle que la décrivent R. Lord et P. Levy [Lord e.a. , 1994], reprend le principe des boucles de rétro-action, tout en l'étendant à l'ensemble des processus comportementaux de la simple tâche d'asservissement à la régulation des comportements sociaux. Pour Lord et Levy, la généralité des boucles de rétro-action pour la description du comportement provient de la nature hiérarchique des systèmes de contrôle, même si la nature des activités de contrôle peut être très différente en fonction des niveaux. Le point commun entre tous ces niveaux réside dans la comparaison entre un état perçu et un état attendu et dans le maintien de l'erreur dans des limites acceptables. Chaque niveau ne connaît que le niveau inférieur et ne reçoit que des erreurs déjà élaborées par des modèles internes qui comparent un état désiré avec un état réel. Le cerveau contient plusieurs schèmes (mécanismes de simulation d'action) du corps indépendants du corps réel [Berthoz, 2003]. Les organes supérieurs qui prennent les décisions ne fonctionnent pas nécessairement en disposant directement des informations sensorielles. Ces centres ne connaissent que l'état des niveaux d'exécution inférieurs qui contiennent des modèles des niveaux qu'ils contrôlent et surtout qui évaluent les erreurs entre ce qu'ils ont imposé et ce qui est exécuté. Lord et Lévy [Lord e.a. , 1994] émettent l'hypothèse que le contrôle des processus humains est produit par une interaction réciproque entre deux mécanismes *top-down* et *bottom-up* que nous nommerons respectivement descendant et ascendant. Le contrôle descendant est conceptuel et est très fortement lié aux intentions courantes et aux plans établis tandis que le contrôle ascendant est plus guidé par les données, relayant ainsi les informations fournies par le système perceptif pour reconnaître les discordances. La régulation ascendante est un complément nécessaire du mécanisme de contrôle descendant, assurant ainsi que le système cognitif notera et répondra correctement aux demandes physiologiques et psychologiques. Néanmoins, la détection de discordances au niveau hiérarchique correspondant à la gestion des tâches ne doit pas créer trop de surcharge cognitive à celui

correspondant au raisonnement symbolique. Par contre, les discordances sont capables d'interrompre la pensée pour rediriger l'attention et sont génératrices de nouveaux modèles aptes à les surmonter.

Cette nature hiérarchique des comportements est prise en compte dans les modèles de type automates parallèles hiérarchiques avec une gestion des échanges bidirectionnels (contrôle descendant puis remontée d'information ascendants) entre niveaux. Les modèles de type ASM hiérarchique et PHISH-Nets intègrent aussi une structuration hiérarchique mais avec seulement un mécanisme de communication ascendant permettant au niveau supérieur de faire un choix parmi les actions proposées au niveau inférieur.

1.5.2 MÉCANISMES ATTENTIONNELS POUR LA SÉLECTION DES ACTIONS

Shallice [Shallice, 1982] a proposé deux types de mécanismes attentionnels pour la sélection des actions : le « système attentionnel de supervision » (*Supervisory Attentional System*) et le « pilote automatique » (*Contention Scheduling*). Ce système de « pilotage automatique » intervient pour sélectionner de manière automatique les actions à effectuer lorsque la situation est routinière. L'exemple typique est celui de la conduite automobile : une fois la période d'apprentissage effectuée, le conducteur peut changer de vitesse tout en discutant avec ses passagers ou en écoutant la radio. Par exemple, dans la conduite automobile, du moins après la période d'apprentissage, la sélection des différentes actions s'effectue le plus généralement de manière automatique. Le sujet peut effectuer un changement de vitesse, qui implique une manipulation simultanée de la pédale d'embrayage et du levier de vitesse, tout en conversant avec ses passagers, écoutant la radio ou encore pensant à programmer ses activités du week-end. Lorsque le sujet est confronté à une situation nouvelle ou lorsque des paramètres d'une situation routinière changent, le « pilote automatique » ne peut plus intervenir, puisqu'il ne peut sélectionner lui-même les actions adéquates. Dans ce cas, c'est le système attentionnel de supervision qui contraint les choix effectués par le « pilote automatique ».

Dans l'ensemble des modèles présentés précédemment, seul HPTS++ peut gérer de façon automatique cette coordination de plusieurs fils d'activité grâce à sa prise en compte de la gestion des ressources corporelles.

1.5.3 MÉCANISMES D'ACTIVATION ET D'INHIBITION

Le cerveau est un simulateur d'action, un générateur d'hypothèses. Anticiper et prédire les conséquences des actions en fonction de la mémoire du passé est l'une de ses propriétés fondamentales. Il n'y a pas de mécanisme de la perception séparé de l'action, ni de mécanisme de l'attention séparé de la sélection de l'action. Décider, ce n'est pas seulement choisir entre plusieurs solutions, c'est aussi réussir à bloquer des comportements non désirés, c'est donc aussi inhiber [Berthoz, 2003]. Pour J. Kuhl [Kuhl, 1984], le traitement des tâches doit être protégé des interruptions, surtout si les tâches sont complexes, d'où l'importance des mécanismes inhibiteurs qui empêchent ce qui n'est pas réellement utilisé de pénétrer dans la mémoire de travail¹. Lord et Lévy [Lord e.a. , 1994] postulent qu'il existe un mécanisme de la volonté qui, afin de

¹Système de mémoire active qui s'occupe à la fois du traitement et du maintien des informations à court terme

protéger un comportement en cours des interruptions, va inhiber directement tous les comportements compétiteurs. Ils émettent plusieurs propositions :

Proposition 1 : l'instanciation d'un but va privilégier l'information catégoriellement proche :

- (a) en augmentant la vitesse à laquelle on peut accéder à cette information ;
- (b) en augmentant l'envie d'accéder à une telle information.

Proposition 2 : l'activation d'un but va supprimer l'instanciation de buts compétiteurs :

- (a) en augmentant la latence de leur activation ;
- (b) en réduisant l'envie d'accéder à de telles informations ;
- (c) en produisant des effets primaires négatifs.

Proposition 3 : la réalisation normale d'un but désactive les structures référentes, en libérant le système cognitif des effets positifs et négatifs.

Proposition 4 : l'échec répété d'un but peut désactiver les structures référentes, en libérant le système cognitif des effets positifs et négatifs.

Proposition 5 : le suivi et la détection automatiques de discordances est un mécanisme de contrôle ascendant important qui intègre des besoins biologiques ainsi que des traitements au niveau symbolique.

Alain Berthoz [Berthoz, 2003] présente trois types d'architectures qui ont été proposées pour la sélection de l'action :

1. la subsumption : actions dotées d'un indice hiérarchique qui permet de les sélectionner automatiquement suivant un ordre de priorité fixé ;
2. l'organisation des actions en réseau : connexions inhibitrices réciproques entre toutes les actions au sein d'un réseau d'actions distribuées et connectées aux capteurs et effecteurs ;
3. le superviseur ou interrupteur central qui active sélectivement les circuits. Cette approche nécessite beaucoup moins de connexions, et offre aussi plus de flexibilité. Il combine les avantages de la modularité et de la centralisation.

On peut faire le lien entre la première approche et les systèmes à base de règles présentés en 1.2.2 et les automates d'états finis simples. La seconde approche est ainsi représentée par les systèmes stimuli-réponse présentés en 1.2.1 et les mécanismes de sélection de l'action présentés en 1.3.4. Le seul représentant de la troisième approche est à notre connaissance le modèle HPTS++ présenté en 1.2.3.

1.5.4 THÉORIE DE L'ACTIVITÉ

Selon W.J. Clancey [Clancey, 2002], la théorie de l'activité est une forme antérieure de la cognition située. Les trois niveaux de la théorie de l'activité sont :

- l'activité (motivations) : les forces agissant sur la prise de décision ;
- l'action (buts) : ce qui doit être fait ;
- l'opération (conditions) : comment cela doit être effectué.

M. Sierhuis [Sierhuis, 2001] définit l'activité de la manière suivante :

« Une activité est une collection d'actions réalisées par un individu, construite socialement, située dans un monde physique, prenant du temps, demandant des efforts et

nécessitant l'application de connaissances. Une activité possède des début et fin bien déterminés, mais peut être interrompue. »

Pour comprendre l'activité il faut prendre en compte l'aspect fondamentalement social de l'action humaine. Notre activité en tant qu'être humain est toujours forgée, contrainte et est rendue signifiante par nos interactions continues avec les mondes du travail, de la famille et d'autres communautés auxquelles nous appartenons. Une activité est ainsi non seulement une chose que nous faisons, mais une manière d'interagir. Toute activité humaine est réfléchie, mais un but n'est pas forcément un problème qui doit être résolu et chaque action ne doit pas forcément être motivée par une tâche à accomplir. W.J. Clancey [Clancey, 1997] prend l'exemple d'une personne qui écoute de la musique tout en conduisant sa voiture sur le trajet la ramenant à son domicile. Cette activité fait partie de la pratique de la conduite pour beaucoup de personnes mais n'est pas du tout un sous-but nécessaire pour atteindre sa destination. Selon Clancey, les motifs sous-tendant les comportements humains ont été imparfaitement caractérisés au travers de la théorie de la résolution de problème introduite par Allan Newell dans sa théorie unifiée de la cognition [Newell, 1990]. Tous les comportements orientés buts ne sont pas obtenus par inférence ou compilation. Certaines actions reproduisent simplement des motifs culturels, tandis que d'autres sont coordonnées sans délibération moyennant des mécanismes d'attention et d'adaptation. Clancey discute de la notion de parallélisme de tâches. Une personne n'effectue pas du multitâche en parallèle, mais plusieurs tâches vont se dérouler en fusionnant attentivement plusieurs intérêts parallèles. Il dit que la conceptualisation de cette notion est encore primitive et que sa nature n'est pas développée dans les théories neuropsychologiques. Dans [Berthoz e.a. , 2006] A. Berthoz et J.L. Petit évoquent l'existence d'au moins cinq boucles de circuits neuronaux (thalamus-cortex-ganglions de la base) qui fonctionnent de façon autonome et en parallèle permettant ainsi de contrôler les mouvements des membres des yeux, de la mémoire et des émotions, ces systèmes bouclés se coordonnant entre eux de façon dynamique.

A l'inverse de l'approche *résolution de problème*, Clancey défend qu'une activité n'est pas forcément interrompue lorsque le besoin s'en fait sentir comme, par exemple, si une autre activité devient pressante ou si une condition externe vient interrompre ce que la personne était en train d'effectuer. Un mécanisme d'activation compétitive est impliqué. La terminaison et le processus de démarrage d'une activité sont plus subtils qu'une décision purement orientée but. Clancey dit encore qu'il n'y a pas d'opposition séquentiel/parallèle mais un couplage. Les sous-systèmes couplés vont s'organiser ensemble en temps réel. Le parallélisme permet de combiner plusieurs activités en même temps, tandis que le sérialisme contraint le traitement de formes ordonnées de séquences d'action. Le parallélisme est fondamental pour coupler des comportements et les ordonner dans le temps, notamment lorsque cela implique plusieurs modalités sensorimotrices.

Dans l'ensemble des modèles comportementaux, seuls les automates parallèles hiérarchiques permettent cette combinaison des aspects séquentiel et parallèle, et au sein de ces modèles seul HPTS++ peut prendre en compte les différentes modalités sensorimotrices.

1.5.5 COGNITION INCARNÉE SITUÉE

Il est courant de considérer que seule l'approche symbolique peut être utilisée pour modéliser les comportements cognitifs, car ces derniers ont souvent été introduits comme étant des problèmes à résoudre formulés sous la forme condition/actions. Cependant, cette approche traditionnelle des sciences cognitives et de l'intelligence artificielle a été confrontée à des problèmes tels que celui de la chambre chinoise [Searle, 1980] : le fait de disposer de symboles et de règles permettant de les manipuler ne définit en rien une quelconque intelligence, car à aucun moment il n'y a l'usage de leur signification. L'intelligence peut être décrite comme la somme des expériences corporelles antérieures de l'agent acquises au travers de son interaction avec son environnement. En d'autres termes, l'intelligence d'un agent est fondée sur ses interactions antérieures avec le monde physique. Brooks [Brooks, 1990] a ainsi introduit l'hypothèse de l'ancrage physique qui postule que l'intelligence d'un agent doit être fondée sur l'interaction entre l'agent physique et son environnement.

D'après Harnad [Harnad, 1990], les symboles doivent acquérir leur sens de la réalité, c'est ce qu'il appelle le problème de l'ancrage des symboles :

«Un système artificiel fondé entièrement sur la manipulation de symboles n'entrevoit jamais la sémantique qui leur est associée.»

Afin de trouver une solution au problème, il propose que les symboles soient fondés sur un processus d'extraction d'invariants à partir de signaux sensori-moteurs, défini en trois étapes :

Iconisation : la transformation des signaux en représentations iconiques ou icônes ;

Discrimination : la capacité de juger si deux entrées sont identiques ou différentes et si elles sont différentes, en quoi elles diffèrent ;

Identification : la capacité à assigner une réponse unique (un nom) à une classe d'entrées, les traitant toutes comme équivalentes d'une certaine manière.

Tandis qu'il y a un consensus dans la communauté dite de la cognition située et incarnée pour dire que l'approche traditionnelle de la cognition en tant que processus calculatoire est incomplète voire erronée, il n'y a par contre pas de consensus sur ce que sont les fondements de cette nouvelle approche.

Aucun des modèles présentés en 1.3 ne prend en compte l'incarnation des symboles. Ils fonctionnent tous dans des mondes abstraits sans accès au sens des symboles manipulés.

1.5.6 BILAN

Le bilan principal qui peut être tiré des liens existant entre les modèles de sélection de l'action et les travaux issus des sciences cognitives sur les mécanismes de sélection de l'action est qu'il existe très peu de relations. Il n'existe pas ou peu de discussion dans la littérature en animation comportementale ou en intelligence artificielle sur les fondements cognitivistes des modèles proposés, mis à part le courant récent de la science cognitive incarnée située qui se trouve être vraiment pluridisciplinaire et qui regroupe des chercheurs sciences cognitives, en robotique et en intelligence artificielle. Au sein des modèles réactifs, seul HPTS++ intègre l'ensemble des notions présentées.

1.6 CONCLUSION

Nous avons présenté, tout au long de ce chapitre, diverses techniques permettant de modéliser le comportement d'humanoïdes virtuels ou plus généralement d'entités autonomes évoluant au sein d'environnements virtuels. Un dernier point reste cependant à évoquer et à considérer : il s'agit du couplage des modèles de sélection d'action. En effet, les modèles réactifs et les modèles cognitifs et orientés but peuvent être couplés au sein d'une architecture permettant d'exploiter le meilleur de chacun d'entre eux pour la modélisation du comportement. Cette notion est discutée dans le chapitre " modélisation de l'activité humaine finalisée ". D'autre part, une entité n'évolue pas seule dans les environnements et peut interagir avec d'autres entités. Cela implique alors de prendre en compte ces interactions dans les processus décisionnels modélisés. Cette notion est discutée dans le chapitre " Les comportements collectifs et sociaux ".

Il ressort par ailleurs de la discussion sur le lien avec les sciences cognitives qu'un grand nombre de points restent à résoudre pour doter les humains virtuels de mécanismes proches de ceux de l'être humain.

RÉFÉRENCES BIBLIOGRAPHIQUES

- [Badler e.a. , 1995] N. I. Badler and B. L. Webber. Planning and parallel transition networks : Animation's new frontiers. In *Pacific Graphics '95* (1995).
- [Berthoz, 2003] A. Berthoz. *La décision*. Odile Jacob (2003).
- [Berthoz e.a. , 2006] A. Berthoz and J. Petit. *Physiologie de l'action et Phénoménologie*. Odile Jacob (2006).
- [Blum e.a. , 1997] A. L. Blum and M. L. Furst. Fast planning through planning graph analysis. *Artificial Intelligence*, 90(1/2) :281–300 (1997).
- [Blumberg, 1997] B. M. Blumberg (1997). *Olds Tricks, New Dogs : Ethology and Interactive Creatures*. PhD thesis, Massachusetts Institute of Technology.
- [Bonet e.a. , 2001] B. Bonet and H. Geffner. Planning as heuristic search. *Artificial Intelligence*, 129(1/2) :5–33 (2001).
- [Bratman, 1987] M. E. Bratman. *Intentions, Plans, and Practical Reason*. Harvard University Press (1987).
- [Brooks, 1990] R. Brooks. Elephants don't play chess. *Robotics and Autonomous Systems*, 6 :3–15 (1990).
- [Brooks, 1991] R. A. Brooks. Intelligence without reason. In J. Myopoulos and R. Reiter, editors, *Proceedings of the 12th International Joint Conference on Artificial Intelligence (IJCAI-91)* (1991), pages 569–595, Sydney, Australia. Morgan Kaufmann publishers Inc. : San Mateo, CA, USA.
- [Cavazza e.a., 2002] M. Cavazza, F. Charles, and S. J. Mead. Planning characters' behaviour in interactive storytelling. *The Journal of Visualization and Computer Animation*, 13 :121–131 (2002).
- [Charles e.a., 2003] F. Charles, M. Lozano, S. J. Mead, A. F. Bisquerra, and M. Cavazza. Planning formalisms and authoring in interactive storytelling. In *First International Conference on Technologies for Interactive Digital Storytelling and Entertainment* (2003), Darmstadt, Germany.
- [Clancey, 1997] W. J. Clancey. *Situated Cognition : On Human Knowledge and Computer Representations*. Cambridge University Press (1997).
- [Clancey, 2002] W. J. Clancey. Simulating activities : Relating motives, deliberation, and attentive coordination. *Cognitive Systems Research*, 3(3) :471–499 (2002).
- [Coderre, 1989] B. Coderre. Modeling behavior in petworld. In *Interdisciplinary Workshop on the Synthesis and Simulation of Living Systems* (1989).
- [Cremer e.a., 1995] J. Cremer, J. Kearney, and Y. E. Papeilis. HCSM : A framework for behavior and scenario in virtual environments. *Modeling and Computer Simulation*, 5(3) :242–267 (1995).

- [Decugis e.a. , 1998] V. Decugis and J. Ferber. Action selection in an autonomous agent with a hierarchical distributed reactive planning architecture. In K. P. Sycara and M. Wooldridge, editors, *Proceedings of the 2nd International Conference on Autonomous Agents (Agents'98)* (1998), pages 354–361, New York. ACM Press.
- [Donikian, 2001] S. Donikian. Hpts : a behaviour modelling language for autonomous agents. In J. P. Müller, E. Andre, S. Sen, and C. Frasson, editors, *Proceedings of the Fifth International Conference on Autonomous Agents* (2001), pages 401–408, Montreal, Canada. ACM Press.
- [Donikian e.a. , 1995] S. Donikian and E. Rutten. Reactivity, concurrency, data-flow and hierarchical preemption for behavioural animation. In *Eurographics Workshop on Programming Paradigms in Graphics* (1995), pages 137–153.
- [Erol e.a., 1994] K. Erol, J. A. Hendler, and D. S. Nau. UMCP : A sound and complete procedure for hierarchical task-network planning. In *Artificial Intelligence Planning Systems* (1994), pages 249–254.
- [Fikes e.a. , 1971] R. E. Fikes and N. J. Nilsson. Strips : A new approach to the application of theorem proving to problem solving. *Artificial Intelligence*, 2(3/4) (1971).
- [Funge, 1999] J. Funge. Representing knowledge within the situation calculus using interval-valued epistemic fluents. *Journal of Reliable Computing*, 5(1) (1999).
- [Funge e.a., 1999] J. Funge, X. Tu, and D. Terzopoulos. Cognitive modeling : Knowledge, reasoning and planning for intelligent characters. In A. Rockwood, editor, *Proceedings of the Conference on Computer Graphics (Siggraph99)* (1999), pages 29–38, N.Y. ACM Press.
- [Grzeszczuk e.a., 1998] R. Grzeszczuk, D. Terzopoulos, and G. Hinton. Neuroanimator : Fast neural network emulation and control of physics-based models. In M. Cohen, editor, *SIGGRAPH 98 Conference Proceedings*, Annual Conference Series (1998), pages 9–20. Addison Wesley.
- [Harnad, 1990] S. Harnad. The symbol grounding problem. *Physica D*, 42 :335–346 (1990).
- [Kuhl, 1984] J. Kuhl (1984). Volitional aspects of achievement motivation and learned helplessness : towards a comprehensive theory of action control. In B. A. Maher, editor, *Progress in experimental personality research*, number 13, pages 99–171. Academic Press, New York ,USA.
- [Laird, 2001] J. E. Laird. It knows what you are going to do : Adding anticipation to a quakebot. In *Fifth International Conference on Autonomous Agents* (2001).
- [Laird e.a., 1987] J. E. Laird, A. Newell, and P. S. Rosenbloom. Soar : An architecture for general intelligence. *Artificial Intelligence*, 33(3) :1–64 (1987).
- [Lamarche e.a. , 2002] F. Lamarche and S. Donikian. Automatic orchestration of behaviours through the management of resources and priority levels. In M. Gini, T. Ishida, C. Castelfranchi, and W. L. Johnson, editors, *Proceedings of the First International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS'02)* (2002), pages 1309–1316. ACM Press.
- [Levesque e.a., 1997] H. Levesque, R. Reiter, Y. Lesperance, F. Lin, and R. Scherl. Golog : A logic programming language for dynamic domains. *Journal of Logic Programming*, 31 :59–84 (1997).
- [Lord e.a. , 1994] R. G. Lord and P. E. Levy. Moving from cognition to action : A control theory perspective. *Applied Psychology : an international review*, 43 (3) :335–398 (1994).

- [Maes, 1990] P. Maes. Situated agents can have goals. In P. Maes, editor, *Designing Autonomous Agents* (1990), pages 49–70. MIT Press.
- [McCarthy e.a. , 1969] J. McCarthy and P. J. Hayes (1969). Some philosophical problems from the standpoint of artificial intelligence. In B. Meltzer and D. Michie, editors, *Machine Intelligence 4*, pages 463–502. Edinburgh University Press.
- [Moreau e.a. , 1998] G. Moreau and S. Donikian. From psychological an real-time interaction requirements to behavioural simulation. In B. Arnaldi and G. Hegron, editors, *Computer Animation and Simulation '98*, SpringerComputerScience (1998), pages 29–44. Springer-Verlag Wien New York.
- [Newell, 1990] A. Newell. *Unified theories of cognition*. Harward University Press (1990).
- [Noser e.a. , 1997] H. Noser and D. Thalmann. Sensor based synthetic actors in a tennis game simulation. In *Computer Graphics International '97* (1997), pages 189–198. IEEE Computer Society Press.
- [Rao e.a. , 1991] A. S. Rao and M. P. Georgeff. Modeling rational agents within a BDI-architecture. In J. Allen, R. Fikes, and E. Sandewall, editors, *Proceedings of the 2nd International Conference on Principles of Knowledge Representation and Reasoning (KR'91)* (1991), pages 473–484. Morgan Kaufmann publishers Inc. : San Mateo, CA, USA.
- [Rao e.a. , 1995] A. S. Rao and M. P. Georgeff. BDI-agents : from theory to practice. In *Proceedings of the First Intl. Conference on Multiagent Systems* (1995), San Francisco.
- [Reynolds, 1987] C. Reynolds. Flocks, herbs and schools : A distributed behavioral model. In *SIGGRAPH'87* (1987).
- [Rhodes, 1996] B. J. Rhodes (1996). Phish-nets : Planning heuristically in situated hybrid networks. Master's thesis, Massachusetts Institute of Technology.
- [Scherl e.a. , 2003] R. Scherl and H. J. Levesque. Knowledge, action, and the frame problem. *Artificial Intelligence*, (144) :1–39 (2003).
- [Searle, 1980] J. Searle. Minds, brains, and programs. *Behavioral and Brain Sciences*, 3 :417–424 (1980).
- [Shallice, 1982] T. Shallice. Specific impairments of planning. *Philosophical Transactions of the Royal Society*, 298 :199–209 (1982).
- [Sierhuis, 2001] M. Sierhuis (2001). *Modeling and Simulating Work Practice. BRAHMS : a multiagent modeling and simulation language for work system analysis and design*. PhD thesis, University of Amsterdam.
- [Thanagarajah e.a., 2002] J. Thanagarajah, L. Padgham, and J. Harland. Representation and reasoning for goals in bdi agents. In *Australasian Conference on Computer Science* (2002), Melbourne, Australie.
- [Thomas e.a. , 2000] G. Thomas and S. Donikian. Virtual humans animation in informed urban environments. In *Proceedings of the Conference on Computer Animation (CA-00)* (2000), pages 112–121, Los Alamitos, CA. IEEE Press.
- [Tyrell, 1994] T. Tyrell. An evaluation of maes's bottom-up mechanism for action selection. *Adaptive Behavior*, 2(4) :307–348 (1994).
- [van de Panne e.a. , 1993] M. van de Panne and E. Fiume. Sensor-actuator networks. In J. T. Kajiya, editor, *Computer Graphics (SIGGRAPH '93 Proceedings)*, volume 27 (1993), pages 335–342.
- [Wihelms e.a. , 1989] J. Wihelms and R. Skinner. An interactive approach to behavioral control. In *Graphics Interface* (1989).

VOCABULAIRE

TABLE DES ILLUSTRATIONS

1.1	Arbre de décision extrait des travaux de Blumberg [Blumberg, 1997].	5
1.2	Exemple d'automates hiérarchiques décrivant le comportement d'un conducteur de véhicule [Cremer e.a., 1995].	6
1.3	Automate HPTS++ décrivant les comportements de boire ou fumer.	8
1.4	Exemple d'opérateurs STRIPS définissant le problème des blocs. . .	11
1.5	Exemple de plan généré par planification au sein des HTN.	13
1.6	Exemple de réseau d'action issu des travaux de P. Maes [Maes, 1990].	14