Comportements perceptifs d'acteurs virtuels autonomes

Une application des cartes cognitives

Jacques Tisseau* — **Marc Parenthoën*** — **Cédric Buche*** **et Patrick Reignier****

* *Centre Européen de Réalité Virtuelle, LISyC (UBO - EA 3883)*
25, rue Claude Chappé, BP38 — F-29280 Plouzané
{tisseau,parenthoen,buche}@enib.fr
** *INRIA / GRAVIR*
655 Avenue de l'Europe — F-38330 Montbonnot Saint Martin
patrick.reignier@imag.fr

RÉSUMÉ. *Nous nous intéressons ici aux comportements perceptifs d'acteurs virtuels autonomes. Ces comportements doivent déterminer leurs réponses, en fonction des stimulus externes, mais aussi en fonction d'émotions internes. Nous proposons de décrire de tels comportements émotionnels à l'aide de cartes cognitives floues où ces états internes sont explicitement représentés. Nous détaillons comment les cartes cognitives floues permettent la spécification, le contrôle, la simulation interne et l'adaptation dynamique du comportement perceptif d'un animat. L'exécution parallèle et asynchrone du contrôle, de la simulation interne et de l'apprentissage débouche sur une architecture comportementale pour entités virtuelles autonomes. Nous illustrons notre démarche à travers un exemple académique non trivial de fiction interactive.*

ABSTRACT. *We are interested here in the perceptive behaviors of autonomous virtual actors. These behaviors must determine their answers, not only according to the external stimuli, but also according to internal emotions. We propose to describe such emotional behaviors using fuzzy cognitive maps, where these internal states are explicitly represented. We detail how fuzzy cognitive maps allow the specification, the control, the internal simulation and the dynamic adaptation of the perceptive behavior of an animat. Their parallel and asynchronous execution leeds to the proposal of a behavioral architecture for virtual autonomous entities. All these are illustrated by the academic example of a non-trivial interactive fiction.*

MOTS-CLÉS : *acteur virtuel, perception active, apprentissage, carte cognitive, architecture d'agent*

KEYWORDS: *virtual actor, active perception, learning, cognitive map, agent architecture*

RSTI - TSI. Volume 24 – n°10/2005, pages 1259 à 1293

1. Introduction

La majorité des travaux en réalité virtuelle concerne l'immersion sensorimotrice de l'utilisateur humain au sein d'univers virtuels (Fuchs *et al.*, 2001) ; mais ces univers, aussi réalistes soient-ils, manqueront de crédibilité tant qu'ils ne seront pas peuplés d'acteurs autonomes (Tisseau *et al.*, 2003).

Les premiers travaux sur la modélisation des acteurs virtuels se focalisent sur leur comportement physique (Magnenat-Thalmann *et al.*, 1991), et permettent par exemple l'étude de l'ergonomie des postes de travail (*Jack* (Badler *et al.*, 1993)). En fait, dès l'origine, l'animation d'acteurs virtuels cristallise sur elle toutes sortes de techniques d'animation puisqu'elle s'intéresse aussi bien aux squelettes qu'aux formes (Chadwick *et al.*, 1989), à la marche (Boulic *et al.*, 1990) qu'à la préhension (Magnenat-Thalmann *et al.*, 1988), aux expressions faciales (Platt *et al.*, 1981) qu'aux cheveux (Rosenblum *et al.*, 1991), et même aux vêtements (Weil, 1986). Une autre catégorie de travaux abordent le problème de l'interaction entre un acteur virtuel et un opérateur humain : agent compagnon (*ALIVE* (Maes, 1995)), agent formateur (*Steve* (Rickel *et al.*, 1999)), agent animateur (Noma *et al.*, 2000). Enfin, d'autres s'intéressent à l'interaction entre acteurs virtuels ; dans de tels théâtres virtuels, le scénario peut être contrôlé au niveau global (*Oz* (Bates, 1992)), dépendre de scripts de comportements (*Improv* (Perlin *et al.*, 1996)) ou reposer sur un jeu d'improvisations (*Virtual Theater Project* (Hayes-Roth *et al.*, 1996)).

Trois grandes catégories de modèles numériques interviennent en animation : les modèles descriptifs, causaux et comportementaux (Arnaldi, 1994). Le modèle descriptif reproduit les effets du phénomène modélisé sans aucune connaissance *a priori* sur les causes (surfaces géométriques, cinématique inverse, ...). Le modèle causal décrit les causes capables de produire un effet (rigides poly-articulés, éléments finis, masses-ressorts...). Son exécution nécessite d'expliciter la solution qui est contenue implicitement dans le modèle ; le temps de calcul est donc plus long que celui d'un modèle descriptif pour lequel la solution est donnée (causal/dynamique *versus* descriptif/cinématique). Le modèle comportemental imite le fonctionnement des êtres vivants selon un cycle perception / décision / action (vie artificielle (Langton, 1986), animat (Wilson, 1985; Guillot *et al.*, 2000)). Il concerne aussi bien les comportements internes qu'externes des entités (Donikian, 1994). Les comportements internes sont relatifs aux transformations internes qui peuvent provoquer des changements perceptibles à l'extérieur de l'entité (morphologie, mouvement). Ces comportements internes sont constitutifs des entités et dépendent peu de son environnement, contrairement aux comportements externes qui traduisent l'influence de l'environnement sur le comportement de l'entité. Ces comportements externes peuvent être purement réactifs ou pulsionnels (stimulus/réponse), perceptifs ou émotionnels (la réponse au stimulus dépend d'un état émotionnel interne), cognitifs ou intentionnels (la réponse est guidée par un but), adaptatifs ou évolutifs (des mécanismes d'apprentissage permettent d'adapter la réponse au cours du temps), sociaux ou collectifs (la réponse est contrainte par des règles de société).

On peut distinguer trois grands types d'architectures comportementales. Les architectures *connexionnistes* définissent le comportement d'un acteur à partir d'un ensemble de capteurs et d'effecteurs reliés entre eux par un réseau de nœuds transformant l'information. Cette approche comprend les modèles de type réseaux de neurones (van de Panne *et al.*, 1993). Les architectures à base d'*automates* décrivent un comportement par un ensemble d'états et de transitions. Le passage d'un état à un autre s'effectue lorsqu'un événement intervient. Les comportements peuvent être assemblés en utilisant plusieurs automates (Brooks, 1986; Maes, 1991; Donikian, 2001). Enfin, les architectures utilisant des *règles* mathématiques, de scripts ou d'inférences représentent un comportement par une approche symbolique (Perlin *et al.*, 1996; Blumberg *et al.*, 2002; Sanza, 2001).

Nous nous intéressons ici aux comportements perceptifs d'acteurs virtuels autonomes. Ces comportements doivent déterminer leurs réponses, non seulement en fonction des stimulus externes, mais également en fonction d'émotions internes telles que la peur, la satisfaction, l'amour ou encore la haine.

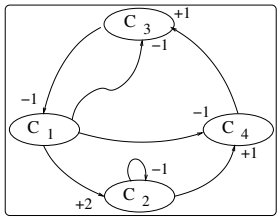
Nous proposons de décrire de tels comportements émotionnels à l'aide de cartes cognitives floues où ces états internes seront explicitement représentés. Cette représentation concilie une spécification ergonomique des comportements avec une faible complexité algorithmique pour leur exécution. En effet, les cartes cognitives floues ont une capacité à traduire visuellement une expertise du comportement sous la forme d'un graphe sémantique, et la rapidité d'exécution nécessaire à des simulations temps réel est, pour les cartes cognitives floues, du même ordre que celle offerte par l'approche connexionniste.

Dans la section 2 suivante, nous présentons le modèle de cartes cognitives que nous utilisons pour spécifier les comportements perceptifs, nous permettant ainsi de clairement différencier la perception de la sensation. La perception devient active dans la section 3 grâce à la simulation interne des comportements. L'apprentissage des cartes cognitives floues est ensuite abordé dans la section 4 à travers l'auto-apprentissage de comportements par imitation. Le contrôle, la simulation interne et l'apprentissage de comportements décisionnels nous amènent à proposer une architecture basée sur trois modes — réactif, prédictif, adaptatif — fonctionnant en parallèle et de manière asynchrone. Enfin, l'exemple académique non trivial de la section 5 — le berger, son chien et son troupeau — nous permet d'illustrer l'ensemble des possibilités des cartes cognitives floues pour la spécification, le contrôle et l'évolution du comportement perceptif d'acteurs virtuels autonomes.

2. Cartes cognitives floues

Les cartes cognitives sont issues des travaux des psychologues qui introduisirent ce concept pour décrire des comportements complexes de mémorisation topologique chez les rats (*cognitive maps* (Tolman, 1948)). Elles furent ensuite formalisées sous la forme de graphes orientés et utilisées en théorie de la décision appliquée au domaine

économique (Axelrod, 1976). Puis elles furent associées à la logique floue pour devenir les cartes cognitives floues (FCM : *Fuzzy Cognitive Maps* (Kosko, 1986)). L'utilisation de ces cartes fut même envisagée pour la modélisation globale d'un monde virtuel (Dickerson *et al.*, 1994). Nous proposons ici de délocaliser les cartes cognitives floues au niveau de chaque acteur virtuel pour modéliser leur comportement perceptif autonome au sein d'un univers virtuel. Nous n'avons pas conservé *a priori* l'idée historique qu'une carte cognitive puisse représenter une topologie spatiale de l'environnement, comme dans le modèle de connaissances SSH (*Spatial Semantic Hierarchy* (Kuiper, 2000)) où elles sont de véritables cartes de l'environnement sensorimoteur d'un système ; nous ne les utilisons que comme un outil formel, en tant que graphe d'influence entre concepts sémantiques.



La carte ci-contre est formée de 4 concepts et possède 7 arcs. Chaque concept C_i a un degré d'activation interne a_i.

$$a = \begin{pmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \end{pmatrix}, \quad L = \begin{pmatrix} 0 & +2 & -1 & -1 \\ 0 & -1 & 0 & +1 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & +1 & 0 \end{pmatrix}$$

L'absence d'arc du concept C_i vers le concept C_j, est équivalent dans la matrice des liens à $L_{ij} = 0$. Un élément non nul de la diagonale $L_{ii} \neq 0$ correspond à un arc du concept C_i sur lui-même.

Figure 1. *Exemple de carte cognitive*

A l'image des réseaux sémantiques (Sowa, 1991), les cartes cognitives sont des graphes orientés dont les nœuds sont des concepts (C_i) et les arcs des liens d'influence (L_{ij}) entre ces concepts (figure 1). Un degré d'activation interne $(a_i(t))$ associé à chaque concept est combiné par un opérateur de logique floue avec une activation externe $(f_{a_i}(t))$, alors que le poids L_{ij} d'un arc traduit pour les activations internes, une relation d'inhibition ($L_{ij} < 0$) ou d'excitation ($L_{ij} > 0$) du concept C_i vers le concept C_j. La dynamique des activations internes de la carte est calculée mathématiquement par produit matriciel normalisé comme le spécifie la définition formelle des cartes cognitives floues (Kosko, 1986).

2.1. *Définition*

On désigne par K l'un des anneaux $\mathbb{Z}$ ou $\mathbb{R}$, par δ l'un des nombres 0 ou 1, par $\mathcal{V}$ l'un des ensembles $\{0, 1\}$, $\{-1, 0, 1\}$ ou l'intervalle $[-\delta, 1]$. Soient $n \in \mathbb{N}^*$, $t_0 \in \mathbb{N}$ et $\rho \in \mathbb{R}_+^*$.

Une **carte cognitive floue** $\mathcal{F}$ est un sextuplet $< \mathcal{C}, \mathcal{A}, L, A, f_a, \mathcal{R} >$ où :

1) $\mathcal{C} = \{C_1, \cdots, C_n\}$ est l'ensemble des concepts formant les noeuds d'un graphe.

2) $\mathcal{A} \subset \mathcal{C} \times \mathcal{C}$ est l'ensemble des arcs (C_i, C_j) orientés de C_i vers C_j.

3) Chaque arc est pondéré. $L : \begin{array}{ccc} \mathcal{C}\times\mathcal{C} & \rightarrow & K \\ (C_i,\ C_j) & \mapsto & L_{ij} \end{array}$ est une fonction de $\mathcal{C}\times\mathcal{C}$ vers K associant un poids L_{ij} à un couple de concepts $(C_i,\ C_j)$, avec $L_{ij} = 0$ si $(C_i,\ C_j) \notin \mathcal{A}$, ou avec L_{ij} égal au poids de l'arc orienté de C_i vers C_j si $(C_i,\ C_j) \in \mathcal{A}$. $L(\mathcal{C}\times\mathcal{C}) = (L_{ij}) \in K^{n\times n}$ est une matrice de $\mathcal{M}_n(K)$. C'est la matrice des liens de la carte $\mathcal{F}$ que, pour simplifier, on notera L.

4) $A : \begin{array}{ccc} \mathcal{C} & \rightarrow & \mathcal{V}^{I\!N} \\ C_i & \mapsto & a_i \end{array}$ est une fonction qui à chaque concept C_i associe la suite $(a_i(t))_{t\in I\!N}$ de ses degrés d'activation telle que pour $t \in I\!N,\ a_i(t) \in \mathcal{V}$ soit son degré d'activation interne à l'instant t. On notera $a(t) = [(a_i(t))_{i\in[\![1,n]\!]}]^T$ le vecteur des activations internes à l'instant t, M^T désignant la transposée d'une matrice M.

5) $f_a \in (I\!R^n)^{I\!N}$ une suite de vecteurs d'activations externes tels que pour $i \in [\![1,n]\!]$ et $t \geq t_0,\ f_{a_i}(t)$ soit l'activation externe du concept C_i à l'instant t.

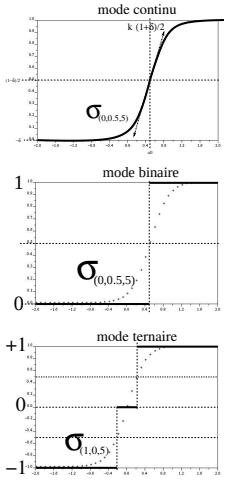
6) $(\mathcal{R})$ est une relation de récurrence sur $t \geq t_0$ entre $a_i(t+1)$, $a_i(t)$ et $f_{a_i}(t)$ pour $i \in [\![1,n]\!]$ traduisant la dynamique de la carte $\mathcal{F}$.

$$(\mathcal{R}) : \forall i \in [\![1,n]\!],\ \forall t \geq t_0, \left\{ \begin{array}{l} a_i(t_0) = 0 \\ a_i(t+1) = \sigma\left[g_i \quad f_{a_i}(t), \sum_{j\in[\![1,n]\!]} L_{ji}a_j(t) \quad \right] \end{array} \right. \qquad [1]$$

où $g_i : I\!R^2 \rightarrow I\!R$ est un opérateur de comparaison entre l'activation interne due au graphe d'influences $\sum L_{ji}a_j(t)$ et l'activation externe $f_{a_i}(t)$ du concept C_i, par exemple :

$$g_i(x,y) = \min(x,y)\ ,\ \text{ou}\ \max(x,y)\ ,\ \text{ou}\ \alpha_i x + \beta_i y\ ,\ \ldots$$

et où $\sigma : I\!R \rightarrow \mathcal{V}$ est une fonction de $I\!R$ vers l'ensemble des degrés d'activation $\mathcal{V}$ normalisant les activations comme le montre la figure 2.



(a) En mode continu, $\mathcal{V} = [-\delta, 1]$, σ est la fonction sigmoïde $\sigma_{(\delta,a_0,\rho)}$ centrée en $(a_0,\ \frac{1-\delta}{2})$, de pente $\rho \cdot \frac{1+\delta}{2}$ en a_0 et de limites en $\pm\infty$ respectivement 1 et $-\delta$:

$$\sigma_{(\delta,a_0,\rho)} : \begin{array}{ccc} I\!R & \rightarrow & [-\delta;1] \\ a & \mapsto & \dfrac{1+\delta}{1+e^{-\rho(a-a_0)}} - \delta \end{array}$$

(b) En mode binaire, $\mathcal{V} = \{0,1\}$ et $\rho > 0$ n'a pas de rôle :

$$\sigma : a \mapsto \begin{array}{lll} 0 & \text{si } \sigma_{(0,0.5,\rho)}(a) \leq 0.5, & i.e. :\ a \leq 0.5 \\ 1 & \text{si } \sigma_{(0,0.5,\rho)}(a) > 0.5, & i.e. :\ a > 0.5 \end{array}$$

(c) En mode ternaire, $\mathcal{V} = \{-1,0,1\}$. Plus $\rho > 0$ est grand, plus la plage "activation du concept non définie" correspondant à la valeur 0 est étroite :

$$\sigma : a \mapsto \begin{array}{lll} -1 & \text{si } \sigma_{(1,0,\rho)}(a) < -0.5, & i.e. : a < -\ln 3/\rho \\ 0 & \text{si } |\sigma_{(1,0,\rho)}(a)| \leq 0.5, & i.e. : |a| \leq \ln 3/\rho \\ 1 & \text{si } \sigma_{(1,0,\rho)}(a) > 0.5, & i.e. : a > \ln 3/\rho \end{array}$$

Figure 2. *Fonctions de normalisation des cartes cognitives*

Nous utilisons les cartes cognitives floues pour spécifier le comportement d'un agent (structure du graphe), et pour contrôler son activité, en particulier son déplacement (dynamique de la carte). Ainsi, une carte cognitive floue possède des concepts sensitifs dont les activations externes ($f_{a_i}(t)$) sont obtenues par *fuzzification* des données issues des capteurs de l'agent. Elle possède des concepts moteurs dont les activations internes ($a_i(t)$) sont *défuzzifiées* pour être envoyées sur les effecteurs de l'agent. Les concepts intermédiaires traduisent l'état interne de l'agent et interviennent dans le calcul de la dynamique de la carte.

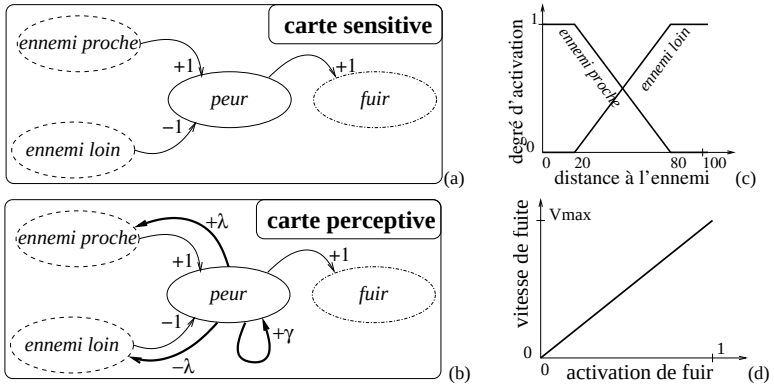
Nous définissons alors un comportement à base de cartes cognitives floues comme étant le cycle perception/décision/action (Newell *et al.*, 1976) composée des mesures de capteurs, leur fuzzification en activations externes des concepts sensitifs, la dynamique des cartes cognitives floues, la défuzzification des concepts moteurs vers des effecteurs et l'action des effecteurs sur l'environnement.

La *fuzzification* et la *défuzzification* sont obtenues conformément aux principes de la logique floue (Wenstop, 1976; Kosko, 1992) : un concept représente un sous-ensemble flou, et son degré d'activation, le degré d'appartenance au sous-ensemble flou. Une carte cognitive floue sans concepts intermédiaires, qui relie directement des concepts sensitifs à des concepts moteurs, est assimilable à un contrôleur flou (Sugeno *et al.*, 1985). Mais on peut avantageusement utiliser une carte cognitive possédant des concepts internes qui explicitent par exemple des émotions, en la dotant de liens récurrents d'un concept vers lui-même pour traduire des phénomènes de mémoire à court terme, ou de liens récurrents d'un concept interne vers un concept sensitif pour donner accès à la perception.

2.2. *Sensation* versus *perception*

A titre d'exemple, on veut modéliser un agent percevant sa distance à un ennemi. En fonction de cette distance et de sa peur, il va décider de s'enfuir ou non. Plus l'ennemi est proche, plus il est effrayé, plus il est effrayé, plus il s'enfuit rapidement et inversement.

Nous modélisons cette fuite par la carte cognitive floue de la figure 3a. Cette carte possède quatre concepts : deux concepts sensitifs (*ennemi proche* et *ennemi loin*), un concept moteur (*fuir*) et un concept interne (*peur*). Trois liens traduisent les influences entre ces concepts : la proximité de l'ennemi excite la peur (*ennemi proche* → *peur*), la peur provoque la fuite (*peur* → *fuir*), et l'éloignement de l'ennemi inhibe la peur (*ennemi loin* → *peur*). On choisit le mode continu ($\mathcal{V} = [0, 1], \delta = 0, \rho = 5$) pour la fonction de normalisation σ. Hormis l'activation externe des concepts sensitifs *ennemi proche* et *ennemi loin* qui est réalisée par *fuzzification* du capteur de la distance à l'ennemi (figure 3c), l'activation externe des autres concepts est nulle. Tandis que la *défuzzification* de l'activation interne de l'envie de fuir donne une vitesse de fuite à cet agent (figure 3d).



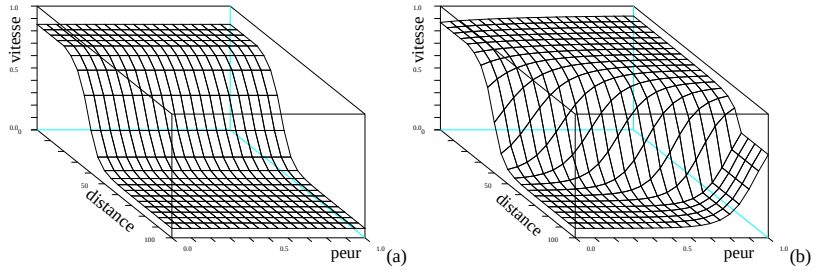
Les concepts sensitifs en tiretés sont activés par *fuzzification* de capteurs. Les concepts moteurs en tiretés-pointillés activent les effecteurs par *défuzzification*. En (a), le concept C_1 *(ennemi proche)* excite C_3 *(peur)* alors que C_2 *(ennemi loin)* l'inhibe et *peur* excite C_4 *(fuir)*. Cela définit une carte purement sensitive. En (b), la carte est perceptive : la peur peut s'auto-entretenir (mémoire : γ) et même influencer les sensations (perception : λ). En (c), la *fuzzification* de la distance à un ennemi fournit les activations externes des 2 concepts sensitifs : *ennemi proche* et *ennemi loin*. En (d), la *défuzzification* de l'activation interne de *fuir* règle linéairement la vitesse de fuite.

Figure 3. *Comportement de fuite : sensitif* versus *perceptif*

Nous distinguons la sensation de la perception : la sensation résulte des capteurs seuls, la perception est la sensation influencée par l'état interne (ici *peur*). Une carte cognitive floue permet de modéliser la perception grâce aux liens entre des concepts internes et des concepts sensitifs. Par exemple, ajoutons trois liens à la carte de fuite précédente (figure 3b). Un premier lien auto-excitateur $(\gamma \geq 0)$ sur *peur* modélise une sorte de "stress". Un deuxième lien excitateur $(\lambda \geq 0)$ de *peur* vers *ennemi proche* et un lien inhibiteur $(-\lambda \leq 0)$ de *peur* vers *ennemi loin* modélisent une sorte de "paranoïa"[1]. Une distance à l'ennemi donnée paraîtra plus ou moins grande selon l'activation du concept *peur*. L'agent devient alors perceptif selon son degré de "paranoïa" λ et son degré de "stress" γ. Ainsi, une sensation identique (même donnée capteur) peut conduire à des perceptions différentes qui déclencheront éventuellement des réactions différentes, suivant l'activation des concepts internes.

Dans l'exemple de la figure 4, la variable temporelle t intervenant dans la relation de récurrence [1] est à considérer comme un temps mathématique propre à la dynamique de la carte cognitive. Il ne faut pas le confondre avec le temps de la simulation s'écoulant dans le monde des agents. Plus précisement, à une itération du temps des agents peut correspondre N itérations de la relation de récurrence [1], où N est au moins égal à la somme, d'une part de la longueur du plus long chemin direct entre les concepts perceptifs et les concepts moteurs, d'autre part de la longueur de la plus grande boucle simple reliant des concepts ($N = 2 + 2 = 4$ dans la figure 3b). Cette manière de choisir N permet d'implémenter un comportement raisonnablement per-

1. Nous n'avons pas la prétention de fournir un modèle de stress ou de paranoïa ; nous utilisons ces termes psychologiques par simple analogie comportementale, dans un but pédagogique.



La perception de la distance d'un ennemi peut être influencée par la peur : en fonction de la proximité d'un ennemi et de la peur, la dynamique de la carte décide d'une vitesse obtenue ici au $4^{ème}$ cycle après remise à zéro des activations internes, sauf celle de la peur. Ce nombre de cycles correspond à la somme, d'une part de la longueur du plus long chemin acyclique entre les concepts perceptifs et les concepts moteurs , d'autre part de la longueur de la plus grande boucle simple reliant des concepts (*i.e. :* respectivement 2 et 2 pour la carte de la figure 3b). Un tel nombre de cycles permet de tenir compte de l'effet de toutes les influences entre concepts avant défuzzification des concepts moteurs. En (a), $\lambda = \gamma = 0$, l'agent est purement sensitif et sa perception de la distance à l'ennemi ne dépend pas de sa peur. En (b), $\lambda = \gamma = 0.6$, l'agent est perceptif : sa perception de la distance à un ennemi est modifiée par sa peur.

Figure 4. *Vitesse de fuite selon le modèle sensitif et selon le modèle perceptif*

ceptif du point de vue du temps de la simulation, dans la mesure où en un pas de temps, l'effet des activations externes a pu parcourir directement le graphe, mais également les boucles perceptives et modifier les activations internes en tenant compte de toute la structure du graphe.

Un agent autonome peut posséder un groupe de cartes cognitives indépendantes comme on le verra dans l'exemple détaillé de la section 5. Pour notre exemple, la carte de la figure 3b permet de décider de la vitesse de fuite, mais on pourrait spécifier une seconde carte permettant de décider de la direction de fuite. Ce groupe de deux cartes spécifierait un comportement prototypique de proie. Aussi, un agent peut posséder plusieurs groupes de cartes, chaque groupe étant le prototype d'un comportement particulier. L'ensemble de ces prototypes forme une bibliothèque de comportements pouvant être vue comme la culture comportementale de cet agent.

Ainsi, les cartes cognitives floues, permettent la spécification du comportement perceptif d'un agent (structure du graphe) et le contrôle du mouvement de l'agent (dynamique de la carte). La prochaine section montre qu'elles autorisent également la simulation interne du comportement (simulation dans la simulation) qui permet d'envisager une perception active.

3. Simulation interne de comportements

La simulation interne de comportement prend ses racines en neurophysiologie avec la notion de perception active. Le principe de simulation interne est corroboré expérimentalement et explique la sensation du mouvement comme une anticipation sensorimotrice (Brunia, 1999). Ainsi, un acteur virtuel possédant une bibliothèque de

comportements décisionnels et doté d'un mécanisme de simulation interne, aura des capacités de perception de soi et des autres.

3.1. *Perception active*

En psychologie, plutôt que de considérer la perception et l'action comme deux éléments distincts placés aux extrémités d'une boucle de coordination (Fechner, 1860), les stimuli et les réponses peuvent être placés au centre de la boucle de coordination en un concept couplé, définissant la réalité d'une expérience relativement aux intentions et aux possibilités d'action de l'acteur (Dewey, 1896). Cette boucle sensorimotrice, prise comme une entité physiologiquement indissociable, explique la forte inscription de l'individu dans son environnement (Bernstein, 1967). Les contraintes posées par cette boucle peuvent servir de base pour expliquer le fonctionnement de mécanismes supérieurs s'inscrivant dans l'espace représentatif, permettant à un individu d'imaginer le monde réel à partir de la simulation de perceptions et d'actions (Craik, 1943). Ainsi, voir un verre d'eau lorsqu'on a soif, c'est déjà simuler les mouvements qu'il faut réaliser pour le boire.

Cette capacité de simulation interne du mouvement a des fondements neurophysiologiques aujourd'hui bien établis (Brunia, 1999). Un individu réalise ces prédictions sensorimotrices non pas, par un raisonnement logique sur des symboles abstraits, représentants du monde réel, mais par une simulation biologique où, grace à des mécanismes inhibiteurs, *tout se passe comme si* l'individu agissait réellement (Berthoz, 1997). Par exemple, pour la vision, le cerveau dispose d'une possibilité d'imaginer des déplacements du regard sans les exécuter grâce à l'action de neurones inhibiteurs qui ferment le circuit de commande des muscles oculaires : en fixant un point devant soi, et en déplaçant son attention, sorte de *regard intérieur*, on a effectivement la sensation d'un déplacement du regard d'un point à l'autre de la pièce. Ce déplacement virtuel du regard a été simulé par le cerveau en activant les mêmes neurones, seule l'action des neurones moteurs a été inhibée.

Le principe de *perception active* résume la manière dont la perception et l'action sont intimement liés en une même entité (Paillard, 1990). Le cerveau peut ainsi être considéré comme un simulateur biologique qui prédit en utilisant sa mémoire et en faisant des hypothèses sur le modèle interne du phénomène.

3.2. *Perception de soi et des autres*

Nous proposons d'utiliser les cartes cognitives dans un but de prédiction, non pas par raisonnement formel comme cela a déjà été réalisé pour le raisonnement sur des croyances, la décision distribuée et l'organisation d'agents en interaction d'un point de vue global (modèle formel CM-RELVIEW (Chaib-Draa *et al.*, 2002), graphes conceptuels pour experts humains (Aissaoui *et al.*, 2003)), mais par simulation de comportements.

Un agent peut effectivement utiliser un groupe de cartes cognitives floues dans un espace imaginaire et simuler un comportement. Cet espace imaginaire est un monde autonome de simulation propre à cet agent, fonctionnant en parallèle avec le contrôle du mouvement et de manière asynchrone pour ne pas bloquer l'animation comportementale de l'agent perceptif. Ainsi, avec ses propres cartes, il accède à une perception de lui-même en s'observant dans son domaine imaginaire. S'il connaît les cartes d'un autre agent, il aura une perception de l'autre et pourra mimer un comportement ou une coopération. En effet, si un agent possède un groupe de cartes cognitives floues, il peut l'utiliser en simulation, en estimant les activations externes des concepts sensitifs et en faisant agir les concepts moteurs dans l'espace imaginaire comme projection du monde réel ; un tel agent est capable de prédire son propre comportement ou celui d'un agent décidant selon ce groupe de cartes.

Aussi, ce monde de simulation permet à un agent autonome de choisir une stratégie parmi plusieurs, non pas par un raisonnement logique mais par une simulation de son propre modèle de comportement, à l'instar de l'explication physiologique. Si un agent possède dans sa bibliothèque de comportements les cartes cognitives d'un autre agent, ou bien des prototypes approchant le comportement de cet autre agent, il peut prévoir le comportement de cet agent en simulant un scénario réalisé en déroulant les cycles de leurs cartes dans l'espace imaginaire. Dans le cas où il ne reconnaît pas un autre agent, il peut réaliser dans son espace imaginaire plusieurs simulations, en utilisant sa culture comportementale, afin de conserver par la suite le meilleur groupe de cartes cognitives pour la simulation du comportement de cet agent dans son monde imaginaire. Cette technique ouvre les portes à la coopération entre agents, chacun pouvant anticiper le comportement de l'autre et ainsi choisir son propre comportement.

Ainsi, les cartes cognitives floues, permettent la spécification du comportement perceptif d'un agent (structure du graphe), le contrôle du mouvement de l'agent (dynamique de la carte) et la simulation interne du mouvement (simulation dans la simulation). La section suivante montre comment elles permettent également de réaliser une adaptation dynamique du comportement, par imitation à partir de prototypes comportementaux. Toutes ces propriétés seront illustrées en détail par une fiction interactive dans la section 5.

4. Apprentissage du comportement

En ayant doté des acteurs virtuels d'une capacité de simulation interne, nous pouvons mettre en place un processus d'apprentissage autonome de comportement qui permettra de quantifier les relations entre les perceptions et les actions. Notre approche de l'auto-apprentissage s'initialise par un processus de simulation : c'est la comparaison entre la simulation d'un modèle et l'observation du réel qui permet l'apprentissage (Schultz *et al.*, 1996). Ainsi, nous proposons un auto-apprentissage qui repose sur l'imitation de comportements observés pour modifier des prototypes comportementaux prédéfinis.

4.1. *Imitation à partir de prototypes comportementaux*

Nous proposons de doter nos acteurs virtuels autonomes et perceptifs d'une capacité d'auto-apprentissage, afin qu'ils décident des modifications de leurs comportements de manière autonome. Cet apprentissage respectera les deux contraintes suivantes :

– il sera libre plutôt que supervisé ;

– il s'effectuera en temps réel plutôt qu'en temps différé.

En ce qui concerne le temps, l'apprenant doit être en mesure de s'adapter en temps réel à toutes nouvelles modifications de son environnement. Par ailleurs, l'apprentissage ne doit pas interrompre le déroulement de la simulation ; l'acteur continue d'évoluer dans le monde virtuel en même temps qu'il modifie son comportement : il apprend en faissant. En effet, classiquement en apprentissage artificiel (Cornuéjols *et al.*, 2002), l'enseignant peut jouer trois rôles bien distincts : superviseur, évaluateur ou démonstrateur. Comme superviseur, l'enseignant devra prendre en charge l'apprentissage de l'élève tout au long de la période d'apprentissage (choisir les exemples, évaluer les réponses). En tant qu'évaluateur, il ne portera un jugement que sur les actions de l'élève. Enfin, l'enseignant-démonstrateur montre juste les actions qu'il faut faire sans se préoccuper de leur assimilation par l'élève ; cette dernière méthode est plus en adéquation que les deux premières avec la nécessité d'autonomie. Dans ce cas, n'importe quel acteur (virtuel ou avatar humain) peut jouer le rôle de démonstrateur sans même le savoir puisque c'est l'élève qui décide seul d'imiter l'action de l'acteur : elle n'est démonstrative que pour l'élève. Les techniques d'apprentissage par renforcement (Sutton, 1988), permettant à un agent de découvrir par lui-même des interactions sensorimotrices en explorant l'ensemble des actions possibles sont également susceptibles d'être utilisées dans le contexte de l'auto-apprentissage. Elles ne peuvent cependant être utilisées seules : même s'il est possible d'apprendre à un robot un jeu arbitraire d'associations sensorimotrices, il n'est pas imaginable d'utiliser ce type de technique pour apprendre en temps réel un comportement élaboré, comme manipuler un outil, créer une structure complexe... (le temps d'exploration de l'ensemble des actions possibles deviendrait très rapidement inacceptable en temps réel).

Nous nous plaçons ici dans le cadre de l'auto-apprentissage par imitation (Meltzoff, 1995; Gallese, 2000) et distinguons quatre grands types d'approche pour l'imitation : les approches logique, connexioniste, probabiliste et prototypique.

1) *approche logique* : l'apprentissage consiste à générer un ensemble de règles de logique formelle sur les observations sensorimotrices décrivant l'exemple observé (*c.f. :* logique XSTL (Del Bimbo *et al.*, 1995)). Cette approche très formelle est difficilement adaptable à notre modélisation de comportements perceptifs ; il faudrait pouvoir lier la pondération des arcs d'une carte cognitive avec de telles règles sensorimotrices.

2) *approche connexionniste* : les actions sont corrélées aux sensations par un réseau de neurones adaptatif, éventuellement inhibé par un mécanisme de reconnais-

sance d'exceptions (*c.f. :* architecture *PerArc* (Gaussier *et al.*, 1998)) ; la modélisation par réseaux de neurones permet d'obtenir un comportement statistiquement satisfaisant, mais n'est pas explicite sémantiquement au contraire des cartes cognitives floues.

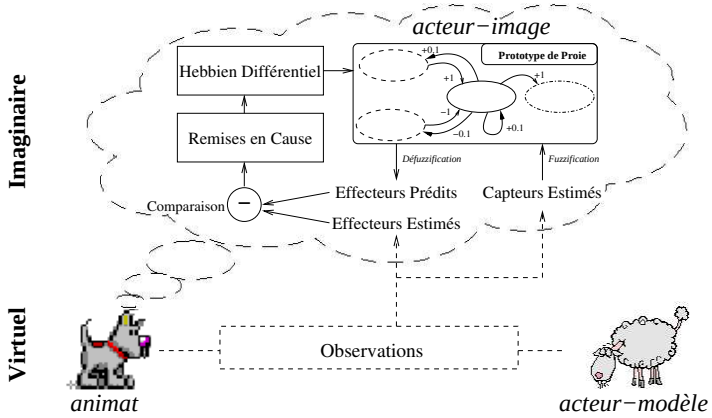
3) *approche probabiliste* : l'approche probabiliste dite approche *F+D (Système Formel + Description)* reprend l'approche *F+I (Système Formel + Interprétation)* des systèmes sensorimoteurs mais en transformant la notion d'interprétation pour la rendre plus proche de la réalité sensorimotrice d'un robot (Bessière *et al.*, 1999a; Bessière *et al.*, 1999b) ; le système utilisé possède bien des variables internes, mais elles ne modélisent pas les émotions et ne traduisent pas notre notion de la perception, ces variables internes ne modifiant pas les variables sensitives par des effets de retour.

4) *approche prototypique* : l'apprentissage à partir de prototypes permet de réaliser l'animation d'un animat, en retrouvant les primitives générant le mouvement à imiter (Voyles *et al.*, 1997; Mataric, 2002). Nos animats disposent eux aussi d'une bibliothèque de comportements prototypiques, mais à la différence de M.J. Mataric ou R. Voyles, nos primitives se situent au niveau de la décision des mouvements, et non pas au niveau des mouvements eux-mêmes. La capacité d'apprentissage dont nous voulons les doter doit leur permettre d'identifier ces prototypes à un environnement donné.

L'acteur virtuel (ou animat) doit pouvoir modifier ses primitives comportementales pour mimer dans son monde imaginaire le comportement observé d'un modèle, qui peut être un autre acteur virtuel ou un avatar contrôlé par un opérateur humain. Puisque nous utilisons des cartes cognitives pour spécifier le comportement perceptif d'animats, nous proposons dans la sous-section suivante un algorithme pour l'apprentissage *temps réel* des poids des liens entre les concepts d'une carte prototypique, afin d'imiter un comportement donné. Cet apprentissage ne modifie ni la structure du graphe d'influence pour conserver la cohérence du comportement vu par un observateur humain, ni la fuzzification des capteurs ou la défuzzification des concepts moteurs qui restent propres à l'animat : l'animat considère que les autres agents perçoivent le monde comme lui-même le perçoit. Puisque les concepts sont associés à une description sémantique, la modification des liens de causalité entre concepts peut mettre en œuvre des méta-connaissances sur l'apprentissage. La dernière sous-section de cette section 4 montre comment notre animat peut sélectionner ses périodes d'apprentissage de manière autonome grâce à la perception active.

4.2. *Algorithme d'auto-apprentissage par imitation*

Cette sous-section décrit une méthode permettant de doter un acteur virtuel autonome et perceptif d'un auto-apprentissage par imitation en temps réel. Nous y présentons l'algorithme d'apprentissage d'un animat contrôlable par des cartes cognitives floues et possédant une culture comportementale composée d'une librairie de comportements perceptifs prototypiques spécifiés par des groupes de cartes cognitives floues. L'animat observe un acteur-modèle et simule le comportement d'un acteur-image qui doit imiter l'acteur-modèle (figure 5).



L'animat "chien" et l'acteur-modèle "mouton" évoluent dans le monde virtuel. Le chien possède un monde imaginaire qui lui est propre, dans lequel il peut réaliser les simulations des comportement des prototypes de sa librairie de comportements, dont celui de "proie". Le chien cherche à faire imiter dans son monde imaginaire le mouton par un acteur-image avec ce prototype de proie. L'imitation est réalisée *en temps réel* en fonction des événements arrivant dans le monde virtuel, en comparant les effecteurs observés du mouton dans le monde virtuel avec les effecteurs prédits par le prototype de proie dans le monde imaginaire de simulation à partir de l'estimation des capteurs du mouton. Si nécessaire, des remises en cause discrètes sont réalisées au niveau des activations internes de la proie afin de diminuer cette différence. Puis le prototype de proie est mis à jour par hebbien différentiel.

Figure 5. *L'animat imitateur, l'acteur-modèle et l'acteur-image*

L'acteur-modèle peut être un acteur virtuel autonome éventuellement mais pas nécessairement contrôlé par des cartes cognitives, ou un avatar contrôlé par un opérateur humain. L'acteur-image est simulé par l'un des prototypes de la librairie de comportements de l'animat. L'animat s'efforce d'imiter l'acteur-modèle dans son propre monde de simulation en modifiant les cartes cognitives floues qui contrôlent le comportement de l'acteur-image. L'animat ne peut avoir accès qu'à une estimation des capteurs et des effecteurs de l'acteur-modèle par observation. L'animat doit posséder les capteurs lui permettant cette estimation. Il fait l'hypothèse que l'acteur-image peut imiter le comportement de l'acteur-modèle en modifiant le prototype comportemental qui le contrôle. L'animat va simuler une image du comportement de l'acteur-modèle dans son monde imaginaire en utilisant des cartes cognitives adaptatives, et va comparer les effecteurs simulés de l'acteur-image avec l'estimation des effecteurs de l'acteur-modèle pour mettre à jour les cartes cognitives floues qui contrôlent l'acteur-image. Il doit identifier le comportement de l'acteur-image à la personnalité de l'acteur-modèle en adaptant les matrices des liens des cartes cognitives floues concernées. A la fin de la période d'apprentissage, les cartes cognitives floues modifiées remplacent ou s'ajoutent aux prototypes initiaux dans la librairie.

L'apprentissage de liens excitateurs et inhibiteurs est caractéristique d'un apprentissage hebbien (Hebb, 1949). Il existe deux méthodes principales d'apprentissage hebbien par une carte cognitive floue d'un cycle limite des activations internes fourni par un expert humain (Kosko, 1988). L'une est basée sur les corrélations entres les activations (Kosko, 1992), l'autre sur les corrélations de leurs variations (hebbien dif-

férentiel) (Dickerson *et al.*, 1994). L'avantage de l'apprentissage différentiel est que des relations de causalité de plus haut degré peuvent être apprises si elles mettent en corrélation des écarts sur les effets avec des écarts sur de multiples causes. L'algorithme d'apprentissage hebbien différentiel est d'une part basé sur la connaissance d'un cycle limite fourni par un expert humain et comprenant les activations internes de tous les concepts, et d'autre part il s'appuie sur l'hypothèse que les activations externes sont constantes. Dans notre cas, les activations externes évoluent dans le temps et nous n'avons pas accès à un tel cycle limite sur les activations internes car seuls les capteurs et les effecteurs estimés de l'acteur-modèle peuvent être observés : un opérateur humain pouvant conduire cet acteur, une carte cognitive les ayant générés n'est pas disponible. Une extention de l'algorithme hebbien différentiel a été développée très récemment (Koulouriotis *et al.*, 2003) : l'apprentissage d'appariements de paires d'entrées/sorties dans une carte cognitive floue est bien réalisé sans l'aide d'un expert, par une méthode évolutive (Holland, 1975), mais la complexité de l'algorithme génétique proposé empêche son utilisation en temps réel. Nous proposons ici un algorithme d'apprentissage hebbien différentiel avec remises en cause discrètes dont la complexité est en $\mathcal{O}(n)$.

Une fois choisi par l'animat imitateur un prototype de comportement pour l'acteur-image, *i.e.* un groupe de cartes cognitives floues reliant les capteurs-image aux effecteurs-image, l'algorithme d'adaptation que nous proposons pour chaque carte cognitive est un cycle itératif en quatre étapes entre t et $t+1$ (t étant le temps des agents) schématisé sur la figure 6 et explicité dans les paragraphes suivants.

Tant que l'apprentissage est activé :

1) **Observation** : les valeurs des capteurs et des effecteurs de l'acteur-modèle sont estimés par observation directe de l'acteur-modèle par l'animat imitateur.

2) **Prédiction** : les capteurs-image (simple copie des capteurs estimés de l'acteur-modèle) sont fuzzifiés en des activations externes des concepts perceptifs. Le calcul de la dynamique de chaque carte cognitive du comportement choisi pour l'acteur-image est alors effectué. Puis les effecteurs-image sont obtenus par défuzzification des activations internes des concepts moteur.

3) **Remises en cause** : les effecteurs-image et les effecteurs estimés de l'acteur-modèle sont comparés et si nécessaire des remises en cause sont déterminées pas à pas en remontant le long du graphe d'influence des concepts moteurs vers les concepts perceptifs, sans modifier les liens et en utilisant éventuellement des méta-connaissances sur l'apprentissage. A chaque étape, la meilleure combinaison d'un nombre fini de pseudo-activations pour ces concepts est choisie.

4) **Mise à jour** : la matrice des liens est mise à jour en appliquant un apprentissage hebbien différentiel discret à la séquence correspondant au passage des activations internes calculées lors de la prédiction vers les pseudo-activations calculées lors des remises en cause.

Figure 6. *Etapes de l'algorithme d'auto-apprentissage*

4.2.1. *Observation*

L'imitateur mesure les caractéristiques de l'acteur-modèle nécessaires aux estimations des capteurs et des effecteurs de l'acteur-modèle (*c.f.* : figure 5). Nous faisons l'hypothèse que ces caractéristiques sont accessibles à l'imitateur. Par exemple, l'animat "chien" estime la distance entre le mouton et un prédateur par la différence de positions à l'instant t, et il estime la vitesse et direction du mouton par la différence

des positions du mouton aux instants t et $t-1$. A l'issue de cette étape d'observation, nous disposons d'une estimation des capteurs et des effecteurs de l'acteur-modèle vu par l'animat imitateur.

4.2.2. *Prédiction*

Elle correspond simplement à l'utilisation habituelle d'un groupe de cartes cognitives pour contrôler un acteur virtuel, ici l'acteur-image. Cette étape détermine les activations internes des cartes cognitives de l'acteur-image à l'instant $t + \delta t$ dans le monde imaginaire ($\delta t < 1$ étant un temps logique permettant ce calcul), selon l'estimation des capteurs de l'acteur-modèle issue de l'étape précédente et la dynamique d'une carte cognitive :

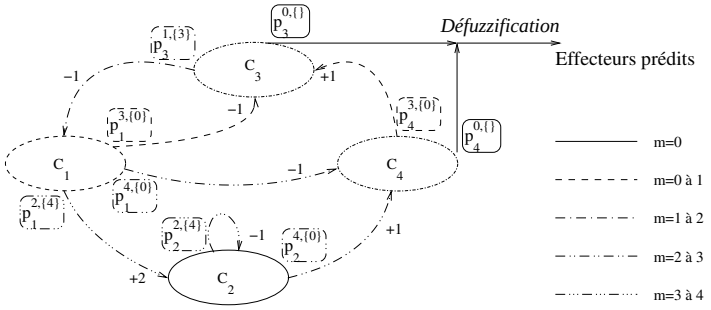
$$\begin{array}{l} \text{Pour } \tau = 1, \cdots N \ , \\ \forall i \in [\![1, n]\!] \ , \ a_i(t + \frac{\tau}{N}\delta t) = \sigma(g_i \left(f_i(t), \sum_{j=1}^{n} L_{ji} a_i(t + \frac{\tau-1}{N}\delta t) \right)) \end{array} \quad [2]$$

où N est un nombre d'itérations correspondant à la somme de la longueur du plus long chemin acyclique et de la longueur du plus long cycle dans le graphe de la carte cognitive comme expliqué en fin de section 2, n est le nombre de concepts de la carte, $f_{i \in [\![1,n]\!]}$ les activations externes provenant de la fuzzification des capteurs-image identifiés aux estimations des capteurs de l'acteur-modèle, $a_{i \in [\![1,n]\!]}$ les activations internes, $L_{(i,j) \in [\![1,n]\!]^2}$ la matrice des liens, $g_i : I\!R^2 \rightarrow I\!R$ un opérateur de comparaison et $\sigma(x) = \frac{1+\delta}{1+e^{-\rho(x-a_0)}} - \delta$ la fonction de normalisation, de paramètres $\delta \in \{0, 1\}, \rho > 0$ et a_0 réel. La défuzzification des concepts moteur à l'instant $t + \delta t$ fournit les effecteurs-image prédits par ce prototype de comportement. Pour alléger le formalisme, on note dorénavant a_i les degrés d'activations internes $a_i(t + \delta t)$ obtenus après l'application de l'équation [2]. A l'issue de cette étape de prédiction, nous disposons des activations internes pour les concepts des cartes cognitives de l'acteur-image et de ses effecteurs, tels qu'ils seraient pour l'animat si l'acteur-image évoluait à la place de l'acteur-modèle.

4.2.3. *Remises en cause*

Les effecteurs-image précédemment calculés ne sont pas forcément conformes à ce qui est mesuré lors de la première étape. Si tel est le cas, cela signifie que les cartes prototypiques ne sont pas adaptées à l'individu que l'on veut imiter. Il faut alors modifier les valeurs des poids des connexions de chaque carte cognitive. Les remises en cause des activations internes préparent le calcul de cette modification, afin d'obtenir un meilleur ajustement du comportement prédit avec le comportement observé. Elles sont déterminées pas à pas en remontant le long du graphe d'influence des concepts moteurs vers les concepts perceptifs, sans modifier les liens et en utilisant éventuellement des méta-connaissances sur l'apprentissage. A chaque étape, la meilleure combinaison d'un nombre fini de pseudo-activations pour ces concepts est choisie. La figure 7 illustre ces remises en cause sur l'exemple de carte de la figure 1. Détaillons ce processus récursif, sachant que les choix aléatoires qui interviennent dans le processus peuvent suivre une distribution de probabilité non-uniforme si des

éléments supplémentaires associés à des connaissances sur l'apprentissage des relations entre des concepts mettent en avant par exemple des notions de priorité entre les relations sémantiques.



Pour cette carte, les concepts moteurs sont C_3 et C_4 et le seul concept sensitif est C_1 (*i.e.* : $f_1 \neq 0,\ f_{2,3 \text{ et } 4} = 0$).
Initialisation $m = 0 : I_0 = \{3, 4\}$. Quatre pseudo-activations $p_3^{\pm}$ et $p_4^{\pm}$ sont calculées selon [3] en plus de $p_3^{=} = a_3$ et $p_4^{=} = a_4$. La meilleure des 9 combinaisons pour leur défuzzification est conservée : $p_3^{0,\{\}} = p_3^{+}$ et $p_4^{0,\{\}} = p_4^{=}$ par exemple. On a alors $P_1 = \emptyset,\ P_2 = \emptyset,\ P_3 = \{p_3^{0,\{\}}\}$ et $P_4 = \{p_4^{0,\{\}}\}$.
Progression de $m = 0$ **à** $m = 1$: aucun concept n'a encore été utilisé : $T_0 = \emptyset$. Un tirage aléatoire donne $3 \in I_0 \setminus T_0$. $J_3 = \{1, 4\}$ et quatre pseudo-activations internes $p_1^{\pm}$ et $p_4^{\pm}$ sont calculées selon [4] en plus de $p_1^{=} = a_1$ et $p_4^{=} = a_4$. Un tirage aléatoire donne $p_3^{0,\{\}} \in P_3$ (unique choix possible). La meilleure des 9 combinaisons des pseudo-activations (influence sur C_3 au plus près de $p_3^{0,\{\}}$) donne par exemple les remises en cause $p_1^{3,\{0\}} = p_1^{-}$ et $p_4^{3,\{0\}} = p_4^{-}$. Donc $P_1 = \{p_1^{3,\{0\}}\}$ et $P_4 = \{p_4^{0,\{\}}, p_4^{3,\{0\}}\}$. Aussi, $I_1 = I_0 \cup J_3 = \{1, 3, 4\}$, $T_1 = T_0 \cup \{3\} = \{3\}$ et $m = 0 + 1 = 1$.
Progression de $m = 1$ **à** $m = 2$: un tirage aléatoire donne $1 \in I_1 \setminus T_1$. $J_1 = \{3\}$ et deux pseudo-activations $p_3^{\pm}$ sont calculées selon [4] en plus de $p_3^{=} = a_3$. Un tirage aléatoire donne $p_1^{3,\{0\}} \in P_1$ (unique choix possible). La meilleure des trois possibilités ($p_3^{+},\ p_3^{-}$ et $p_3^{=}$) donne la remise en cause $p_3^{1,\{3\}} = p_3^{-}$ par exemple. Donc $P_3 = \{p_3^{0,\{\}}, p_3^{1,\{3\}}\}$. Aussi, $I_2 = I_1 \cup J_1 = \{1, 3, 4\}$, $T_2 = T_1 \cup \{1\} = \{1, 3\}$ et $m = 1 + 1 = 2$.
Progression de $m = 2$ **à** $m = 3$: un tirage aléatoire donne $4 \in I_2 \setminus T_2$ (unique choix possible), $J_4 = \{1, 2\}$ et quatre pseudo-activations $p_1^{\pm}$ et $p_2^{\pm}$ sont calculées selon [4] en plus de $p_1^{=} = a_1$ et $p_2^{=} = a_2$. Un tirage aléatoire donne $p_4^{0,\{\}} \in P_4$. La meilleure des 9 possibilités donne par exemple les remises en cause $p_1^{4,\{0\}} = p_1^{=}$ et $p_2^{4,\{0\}} = p_2^{+}$. Donc $P_1 = \{p_1^{3,\{0\}}, p_1^{4,\{0\}}\}$ et $P_2 = \{p_2^{4,\{0\}}\}$. Aussi, $I_3 = I_2 \cup J_4 = \{1, 2, 3, 4\}$, $T_3 = T_2 \cup \{4\} = \{1, 3, 4\}$ et $m = 2 + 1 = 3$.
Progression de $m = 3$ **à** $m = 4$: un tirage aléatoire donne $2 \in I_3 \setminus T_3$ (unique choix possible). $J_2 = \{1, 2\}$ et quatre pseudo-activations $p_1^{\pm}$ et $p_2^{\pm}$ sont calculées selon [4] en plus de $p_1^{=} = a_1$ et $p_2^{=} = a_2$. Un tirage aléatoire donne $p_2^{4,\{0\}} \in P_2$ (unique choix possible). La meilleure des 9 possibilités donne par exemple les remises en cause $p_1^{2,\{4\}} = p_1^{-}$ et $p_2^{2,\{4\}} = p_2^{-}$. Donc $P_1 = \{p_1^{3,\{0\}}, p_1^{4,\{0\}}, p_1^{2,\{4\}}\}$ et $P_2 = \{p_2^{4,\{0\}}, p_2^{2,\{4\}}\}$. Aussi, $I_3 = I_2 \cup J_2 = \{1, 2, 3, 4\}$, $T_4 = T_3 \cup \{2\} = \{1, 2, 3, 4\}$ et $m = 3 + 1 = 4$.
Terminaison : tous les nœuds disponibles dans I_4 ont été utilisés ($T_4 = I_4$), donc toutes les connexions appartenant à des chemins arrivant vers les concepts moteurs ont servi pour faire des remises en cause.

Figure 7. *Exemple complet de pseudo-activations et de remises en cause*

1) *Initialisation* ($m = 0$) : proposer deux pseudo-activations $p^{\pm}$ pour chaque concept moteur, puis choisir la remise en cause initiale réalisant le meilleur ajustement entre la défuzzification des pseudo-activations potentielles des concepts moteur et les effecteurs estimés du modèle. Chaque concept moteur est soit activé p^{+}, soit désactivé p^{-}, soit non modifié $p^{=}$. C'est par conviction sur l'apprentissage que nous faisons cette proposition de remise en cause discrète, plutôt qu'un calcul continu de type gradient (Mozer, 1995; William *et al.*, 1995). Un choix discret facilite la prise de décision par l'animat. Parmi les combinaisons possibles, l'une d'elle propose une meilleure dé-

fuzzification. Plus formellement, soit I_0 l'ensemble des indices des concepts défuzzifiés en des effecteurs-image. Pour chaque $i \in I_0$, deux pseudo-activations potentielles $p_i^{\pm}$ simulent une activation/inactivation du concept C_i, ($\alpha_i \geq 1$) traduisant la radicalité du choix :

$$p_i^{\pm} = \sigma(a_0 \pm \frac{2\alpha_i}{\rho}) \qquad [3]$$

Avec la valeur a_i, cela donne 3 pseudo-activations $p_i^{=} = a_i$, p_i^{+} ou p_i^{-} possibles pour chaque C_i. Les $n_{I_0} = 3^{\mathrm{Card}I_0}$ combinaisons sont défuzzifiées et comparées à l'estimation de l'effecteur du modèle. La comparaison est fondée sur la distance entre les effecteurs estimés de l'acteur-modèle et la défuzzification d'une combinaison de pseudo-activations (distance : $d(x, y) = \|y - x\|$). La meilleure combinaison $(p_i^{0,\{\}})_{i \in I_0}$ est conservée (le 0 est l'indice correspondant à la défuzzification, l'ensemble vide $\{\}$ sera utilisé pour les remises en cause ne correspondant pas aux défuzzifications). Réalisée pour chaque effecteur, cette procédure nous donne un premier ensemble de remises en cause pour les activations des concepts moteurs concernés par les défuzzifications : $\forall i \in I_0,\ P_i = \{p_i^{0,\{\}}\}$. Les autres ensembles de pseudo-activations sont initialisés vides : $(P_i = \emptyset)_{i \in (\llbracket 1,n \rrbracket \setminus I_0)}$.
Après avoir traité les concepts moteurs, on remonte dans la carte vers les concepts internes et les concepts sensitifs.

2) *Progression (de $m \in I\!N$ à $m + 1$)* : Soit $I_m \subset \llbracket 1, n \rrbracket$ l'ensemble des indices des concepts dont l'ensemble des remises en cause n'est pas vide. Soit $T_m \subset I_m$ l'ensemble des indices des concepts déjà utilisés pour le calcul des remises en cause dans la carte cognitive ($T_0 = \emptyset$).

- On choisit aléatoirement $i \in I_m \setminus T_m$. On note a_i (respectivement f_i) l'activation interne (resp. externe) du concept C_i, P_i son ensemble de Λ remises en cause déjà obtenues précédemment : $P_i = \{p_i^{k_1,\{l_1\}}, \cdots, p_i^{k_\Lambda,\{l_\Lambda\}}\}$, le couple $(k_\lambda, \{l_\lambda\})$ étant soit $(0, \{\})$ si la remise en cause provient de la défuzzification, soit constitué de $k_\lambda \in \llbracket 1, n \rrbracket$: indice du concept utilisé par la $\lambda^{ème}$ remise en cause de C_i (nécessairement : $L_{i,k_\lambda} \neq 0$) et de $l_\lambda \in \llbracket 0, n \rrbracket$ permettant de mémoriser le choix de la remise en cause de C_{k_λ} utilisée par la $\lambda^{ème}$ remise en cause de C_i. Soit $J_i \subset \llbracket 1, n \rrbracket$ l'ensemble des indices des concepts qui sont des causes du concept C_i (*i.e.* : $L_{ji} \neq 0$).

- Si $J_i \neq \emptyset$, nous complétons les ensembles P_j pour $j \in J_i$ comme suit :
- Pour chaque $j \in J_i$, on calcule selon la formule [4] deux pseudo-activations potentielles p_j^{+} et p_j^{-} de telle sorte que leur influence sur a_i fasse un choix clair entre un C_i actif ou inactif, en tenant compte de l'activation externe f_i de C_i, avec $\alpha_i \geq 1$ traduisant la radicalité du choix :

$$p_j^{\pm} = \left(a_0 \pm \frac{2\alpha_i}{\rho} - f_i - \sum_{l \in J_i \setminus \{j\}} L_{li} a_l \right) / L_{ji} \qquad [4]$$

- On choisit aléatoirement[2] un $\lambda \in \llbracket 1, \Lambda \rrbracket$; cela correspond au choix d'une remise en cause $p_i^{k_\lambda,\{l_\lambda\}} \in P_i$ de C_i. Parmi les $n_{J_i} = 3^{\mathrm{Card}J_i}$ combinaisons possibles, $p_j^{=} = a_j$, p_j^{+} ou p_j^{-}

2. Le choix aléatoire évite l'introduction d'un biais du à la numérotation des remises en cause. L'algorithme est ainsi insensible à leur renumérotation.

pour $j \in J_i$, on détermine la meilleure combinaison $p_j^{i,\{k_\lambda\}}$ des pseudo-activations pour les concepts d'indice $j \in J_i$ pour ce choix de λ, c'est à dire une remise en cause qui donne une activation locale de $C_i : \sigma(g_i(f_i, \sum_{j \in J_i} L_{ji} p_j^{i,\{k_\lambda\}}))$ la plus proche de $p_i^{k_\lambda,\{l_\lambda\}}$.

- On met à jour l'ensemble des indices des concepts dont l'ensemble des remises en cause est non vide : $I_{m+1} = I_m \cup J_i$, avec $P_j(m+1) = P_j(m) \cup \{p_j^{i,\{k_\lambda\}}\}$ pour $j \in J_i$.

- Sinon $J_i = \emptyset$: le concept C_i est le départ d'un chemin vers les concepts moteur ; il n'est pas possible de remonter plus loin dans le graphe.

- On mémorise l'indice utilisé pour ce calcul : $T_{m+1} = T_m \cup \{i\}$

- m est incrémenté et l'on continue la progression si $T_m \neq I_m$

3) *Terminaison* : si $T_m = I_m$, cela signifie que tous les arcs appartenant aux chemins arrivant en l'un des concepts moteurs $(C_i)_{i \in I_0}$ ont été étudiés.

On dispose alors de remises en cause pour tous les concepts appartenant à des chemins arrivant aux concepts moteurs. Il reste à mettre à jour les poids de la carte cognitive afin de mémoriser ces remises en cause.

4.2.4. *Mise à jour*

La dernière étape modifie la matrice des liens de la carte cognitive floue, de telle sorte que sa dynamique soit orientée vers un comportement résultant s'approchant de celui de l'acteur-modèle, en utilisant le hebbien différentiel discret pour l'apprentissage du passage des activations internes a vers les remises en cause p calculées à l'étape précédente. Au contraire de Dickerson qui utilisent un cycle limite et un taux d'apprentissage décroissant vers zéro avec le temps pour assurer une convergence (Dickerson *et al.*, 1994), nous ne faisons apprendre que le passage des activations internes a vers les remises en cause p pour la modification des liens, sans faire de cycles et avec un taux d'apprentissage constant $r(t) = R$. Faire un cycle ferait apprendre non seulement le passage de a à p, mais aussi celui de p à a, ce qui est plus génant. Notre taux d'apprentissage est constant afin que l'animat conserve une forte adaptativité. Formellement, on note $\mathcal{A} \subset [\![1, n]\!]^2$ l'ensemble des arcs de la carte cognitive, $\beta \in]0; 1 + \delta[$ un niveau de sensibilité pour la détection d'un écart significatif entre la remise en cause et l'activation interne d'un concept, réalisée par $s : I\!R \to \{-1, 0, 1\}$, fonction discrète définie par $s(x) = -1,\ 0$ ou 1 respectivement si $x \leq -\beta,\ -\beta < x < \beta$ ou $x \geq \beta$. La loi d'apprentissage obéit aux équations [5] pour le passage de $L(t)$ à $L(t+1)$.

$$
\begin{array}{l}
\forall (i,j) \in \mathcal{A}, \\
\text{si } \exists k \in [\![0, n]\!], \text{ tel que } : p_j^{i,\{k\}} \in P_j, \text{ alors il est unique. Et pour ce } k \text{ :} \\
\left\{ \begin{array}{l}
\Delta_j = s(p_j^{i,\{k\}} - a_j),\ \Delta_i = s(p_i^{k,\{\cdots\}} - a_i), \text{ où soit } p_i^{k,\{\cdots\}} = p_i^{0,\{\}}, \\
\text{soit } p_i^{k,\{\cdots\}} = p_i^{k,\{l\}}, \text{ avec } l : \text{remise en cause associée à } k \\
\\
L_{ij}(t+1) = \left| \begin{array}{ll}
L_{ij}(t) + R(\Delta_i \Delta_j - L_{ij}(t)) & , \text{ si } \quad \Delta_i \neq 0 \\
L_{ij}(t) & , \text{ si } \quad \Delta_i = 0
\end{array} \right.
\end{array} \right. \qquad [5] \\
\text{sinon } \mathcal{A}_{ij} \notin \{\text{chemins vers des effecteurs}\} : \ L_{ij}(t+1) = L_{ij}(t)
\end{array}
$$

De plus, certains liens peuvent être maintenus dans des bornes $\mathcal{B}_{ij} = [L_{ij}^{min}, L_{ij}^{max}]$ de telle sorte que le comportement modifié reste cohérent avec la description de l'expert : si $L_{ij}(t+1) < L_{ij}^{min}$ alors $L_{ij}(t+1) = L_{ij}^{min}$ et si $L_{ij}(t+1) > L_{ij}^{max}$ alors $L_{ij}(t+1) = L_{ij}^{max}$. Cela permet l'imitation tout en conservant la personnalité et les émotions de l'imitateur. Théoriquement, rien n'empêche de modifier le taux d'apprentissage au cours du temps, en lui faisant suivre une suite décroissante convergeant vers zéro, dont la série associée divergerait vers l'infini, comme par exemple $r(t > t_0) = \frac{R}{t-t_0}$. Cela assurerait théoriquement la convergence des poids des liens dans les cartes cognitives floues, mais l'adaptativité serait de plus en plus faible avec le temps. Empiriquement, les expériences réalisées sur le chien de berger (voir section 5) montre qu'il y a convergence avec un taux d'apprentissage constant $r(t) = R$ et que le chien de berger est capable d'adapter en temps réel ses groupes de cartes prototypiques de proie et de prédateur à des chiens et des moutons spécifiques.

4.2.5. *Complexité*

Pragmatiquement, la complexité de cet algorithme d'auto-apprentissage, repris dans ses grandes lignes sur la figure 8, est une fonction polynomiale du nombre de concepts n dans chaque carte cognitive prototypique fournie par un expert humain, et même un $\mathcal{O}(n)$. Sans faire attention, on pourrait croire que cet algorithme a une complexité exponentielle à cause de la troisième étape, mais pragmatiquement il n'en est rien car les cartes cognitives, générées par des experts humains, sont de connectivité faible. En effet, il nous semble que pour un expert, les causes **directes** d'un concept sont toujours en nombre très limité (rarement plus de cinq en pratique). Par conséquent, nous considérons pour évaluer la complexité de l'algorithme que le nombre d'arcs arrivant sur chaque concepts est borné par un nombre M ($M \approx 5$) indépendant du nombre n de concepts dans une carte cognitive, *i.e. :* $\forall i, \mathrm{Card} J_i \leq M$. Le nombre de combinaisons $n_{J_i} = 3^{\mathrm{Card} J_i}$ lors de la troisième étape est donc majoré dans la pratique par un nombre indépendant de n. Les mêmes remarques s'appliquent au calcul de la dynamique de la carte lors de la deuxième étape, dont la complexité est un $\mathcal{O}(n)$ alors qu'elle apparaît comme un $\mathcal{O}(n^2)$, grâce au grand nombre de zéros dans la matrice des liens, le nombre de liens non nuls dans une colonne étant majoré par M, quel que soit n. Evidemment, la complexité du calcul du N (*c.f.* : figure 4, N est le nombre d'itérations de la relation [1] par cycle de vie d'un animat. C'est la somme de la longueur de la plus longue boucle avec la longueur du plus long chemin sans boucle des concepts perceptifs vers les concepts moteurs) n'est pas polynomiale, car il s'agit de trouver des chemins dans un graphe. Cependant, la structure du graphe n'étant pas modifiée par l'apprentissage, ce N est calculé une fois pour toute à l'initialisation. L'hypothèse comme quoi ce N est majorable par un nombre indépendant de n est raisonnable dans la mesure où les cartes cognitives sont spécifiées par des experts humains. Quant à la quatrième étape : la mise à jour des poids des liens n'est réalisée que pour les liens appartenant à des chemins menant aux concepts moteurs, tous les indices dont on a besoin sont déjà déterminés dans les ensembles P_i, la complexité est alors en $\mathcal{O}(n)$. Cet algorithme d'auto-apprentissage peut donc être implémenté pour une utilisation en temps réel.

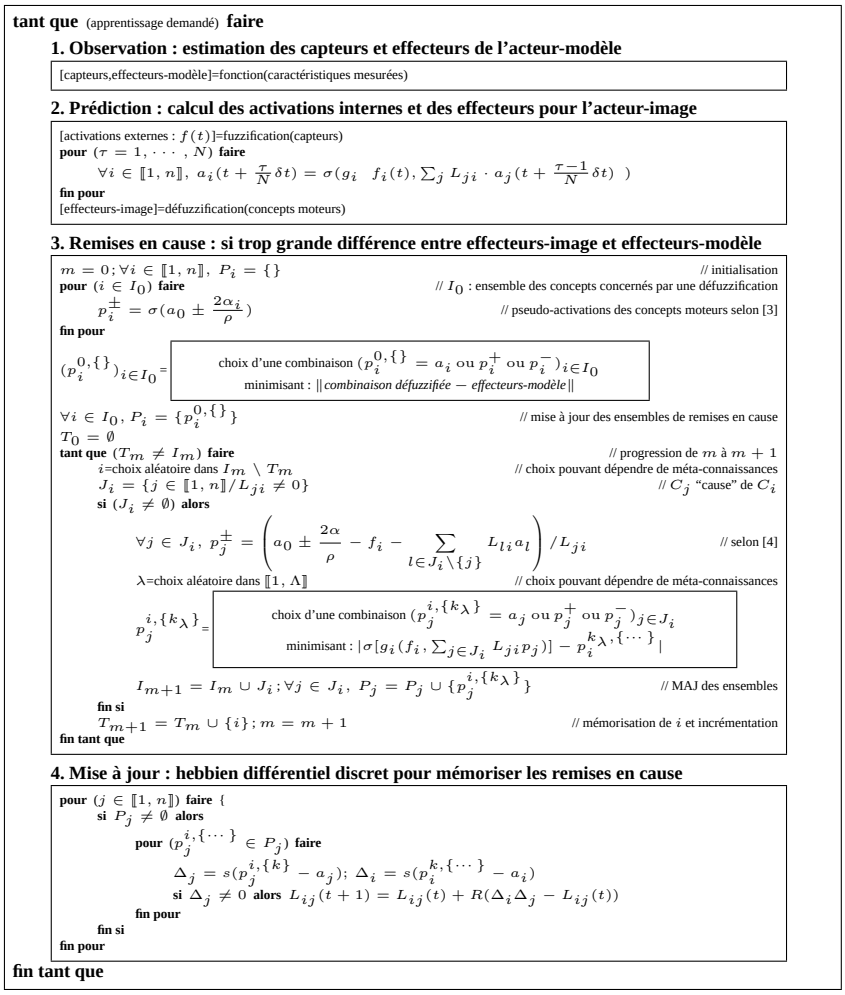


Figure 8. *Algorithme d'auto-apprentissage pour chaque carte du prototype*

Ainsi, à l'issu de l'apprentissage qui s'est déroulé sans interrompre l'activité sensorimotrice de l'animat dans le monde virtuel (décrit dans les sections 2 et 3), la dynamique de la carte est mieux ajustée aux observations réalisées durant la période d'apprentissage. Le comportement de l'acteur-image déterminé dans l'espace imaginaire de l'animat par le prototype modifié permet alors à l'animat de réaliser des simulations plus pertinentes.

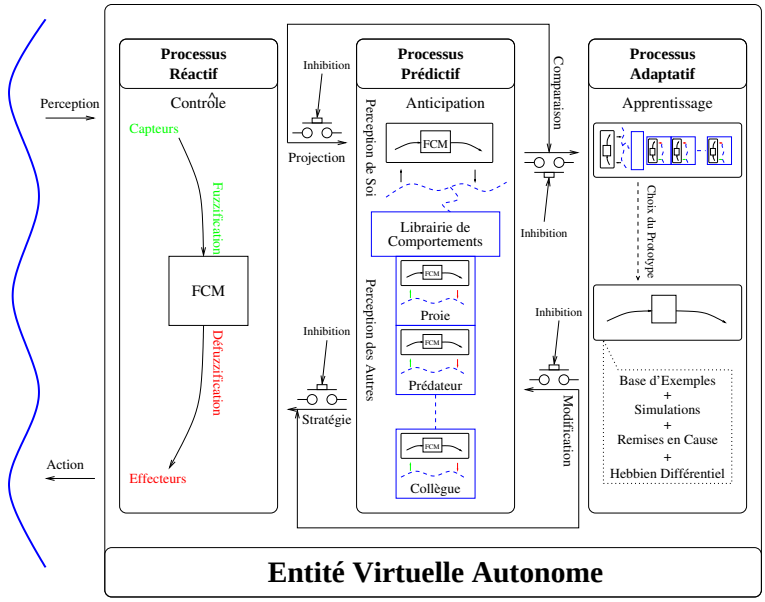
4.3. *Perception active pour une autonomisation de l'apprentissage*

L'algorithme précédent est actif durant les périodes d'apprentissage. L'exécution non bloquante de cet algorithme s'effectue en parallèle et en asynchronie avec le comportement de l'agent perceptif et ses simulations internes. Le choix des périodes d'apprentissage peut être inné (préprogrammé), pris en charge par un opérateur humain (interactif), mais aussi déterminé par la perception active (autonomie).

Un acteur virtuel autonome doit pouvoir choisir seul les moments où il doit apprendre. L'approche *perception active* (Berthoz, 1997) fusionne le cycle perception / décision / action en un unique processus neurophysiologique. Les processus de synchronisation neurophysiologique entre l'anticipation et la réalisation effective du mouvement ont été largement étudiés et mettent en cause l'hippocampe comme lieu priviligié de la synchronisation (Buzsaki *et al.*, 1992) ; ce processus serait à la base d'un mécanisme neurophysiologique d'apprentissage (Lisman *et al.*, 1995), qui trouverait ses origines dans une oscillation de la concentration d'un neuromédiateur synaptique au niveau de neurones impliqués dans le contrôle de la synchronisation, ce neuromédiateur inhibant l'apprentissage au fur et à mesure que le temps passe après la synchronisation des informations sensorielles avec l'anticipation d'action (Kesner *et al.*, 2001). De tels processus artificiels ont été appliqués aux animats (Trullier *et al.*, 2000; Heinze *et al.*, 2001; Gaussier *et al.*, 2002). Nous indiquons ici comment un acteur virtuel peut sélectionner ses propres périodes d'apprentissage de manière autonome grâce à la perception active, le choix de la carte prototypique restant *ad hoc*.

Des recherches en ergonomie cognitive ont montré l'existence de plusieurs niveaux cognitifs fonctionnant en parallèle lors de l'exécution de tâches de planification de trajectoires par les êtres humains (Morineau, 2004). Les deux premiers niveaux peuvent correspondre aux modes de contrôle réactifs et prédictifs de la perception active décrits à la section 3. Le troisième niveau, plus abstrait, serait le siège des remises en cause et de l'apprentissage.

Inspirés par ces idées provenant de l'ergonomie cognitive et de la neurophysiologie, nous proposons l'architecture de la figure 9, comme structure permettant l'implémentation d'un acteur virtuel autonome, prédictif et adaptatif. L'acteur virtuel agit selon le comportement perceptif spécifié par le groupe de cartes cognitives sélectionné pour le processus réactif. Il réalise ainsi l'un des comportements de sa bibliothèque en fonction d'une stratégie sensorimotrice. Le processus réactif vérifie aussi que certaines variables sensorimotrices restent dans des bornes déterminées par le mode prédictif (perception active), sinon il envoie un message de synchronisation vers le mode prédictif. Parallèlement, l'acteur virtuel simule des scénarii dans son mode prédictif selon des stratégies sensorimotrices différentes. Ces stratégies se différencient soit par le choix d'autres prototypes de comportement dans sa biliothèque pour se simuler lui-même et simuler les autres agents, soit par le choix d'une autre manière de concentrer son attention : par exemple en tenant compte de tous les agents visibles, ou seulement de ceux qui sont dans un certain voisinage. Le résultat de ces simulations permet au



Ce schéma présente le principe de fonctionnement d'un acteur virtuel autonome, prédictif et adaptatif. Trois processus cognitifs fonctionnent en parallèle et ne se synchronisent que par des messages intermittants levant les inhibitions. Grossièrement, ils correspondent de gauche à droite à des cognitions à court ($< 1s$), moyen ($0,1s < 100s$) et long terme ($> 10s$). Le premier, réactif, fonctionne à haute fréquence pour l'acquisition de certains capteurs et est associé à une stratégie sensorimotrice spécifiée par un groupe de cartes cognitives et des bornes pour certaines variables sensorimotrices. Le deuxième, prédictif, réalise des simulations internes à moyenne fréquence, qui peuvent être synchronisées épisodiquement avec les informations perceptives. Ce mode prédictif peut modifier le mode réactif en lui fournissant une nouvelle stratégie sensorimotrice. Le troisième, adaptatif, compare les prédictions avec les perceptions pour modifier les modèles de comportement utilisés par les deux autres modes, et quelles simulations doivent être effectuées par le processus prédictif. Mis en œuvre lorsqu'une erreur de prédiction est manifeste, le troisième mode peut fonctionner en arrière plan, à basse fréquence.

Figure 9. *Implémentation en parallèle de 3 modes cognitifs asynchrones*

mode prédictif de fournir une nouvelle stratégie au mode réactif. Avec ces deux seuls modes, l'acteur virtuel peut réaliser des prédictions d'après sa culture comportementale et enchaîner diverses combinaisons de comportements primitifs décrits dans sa librairie. Avec le troisième mode, il peut aussi comparer ses prédictions avec ce qui se passe et décider de déclencher un apprentissage lorsqu'une trop mauvaise prédiction est observée par le processus réactif grâce à la perception active. Par exemple, la trajectoire prévue de tel agent pour tel horizon temporel est trop éloignée de la trajectoire observée. Un message contenant les informations permettant d'estimer les capteurs et les effecteurs de l'agent-modèle lors de la dernière période générant une erreur de prédiction est alors envoyé vers le processus adaptatif. Parallèlement, l'acteur virtuel met en œuvre l'algorithme d'apprentissage décrit précédemment sur les bases d'exemples fournies par la perception active afin d'imiter au mieux les comportements observés. Un exemple concret est fourni dans le cas d'école du chien de berger.

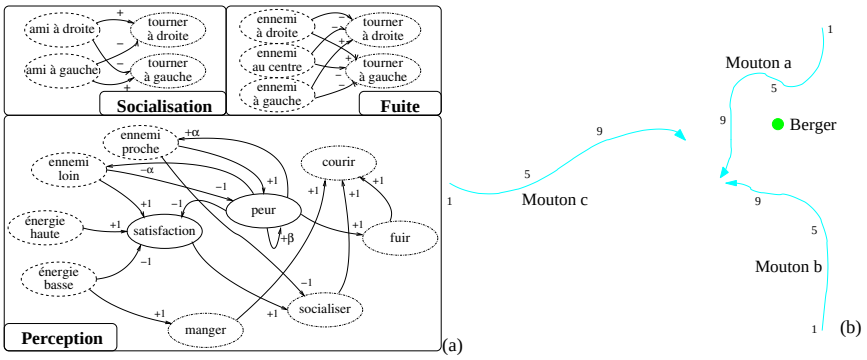
5. Le berger, son chien et son troupeau

Nous proposons ici d'utiliser les cartes cognitives floues pour caractériser des rôles d'agents crédibles dans des fictions interactives. Illustrons ceci par une histoire d'alpages. *Il était une fois un berger, son chien et son troupeau de moutons . . .* Cet exemple a déjà été utilisé comme métaphore de comportement collectif complexe au sein d'un groupe de robots mobiles (RoboShepherd (Schultz *et al.*, 1996)), comme exemple de robot gardien d'oies réelles (Sheepdog Robot (Vaughan *et al.*, 2000)), ou encore comme exemple de scènes d'improvisation (Black Sheep (Klesen *et al.*, 2000)). L'implémentation de nos modèles est réalisée avec le langage `oRis` (Harrouet *et al.*, 2002) et le support de leurs activités est un PC linux, avec un cpu à 400MHz et une RAM de 128Mo.

5.1. *Spécifications et contrôles de comportements perceptifs*

Le berger (avatar joué par un opérateur) se déplace dans son pâturage, peut parler au chien pour lui donner des ordres et/ou des savoir-faire (communication par envoi de messages). Il veut regrouper ses moutons dans une zone qu'il détermine à mesure des besoins. Par défaut il reste assis ; le berger est un avatar d'un acteur humain qui prend toutes les décisions à sa place.

Chaque mouton distingue un ennemi (un chien ou un homme) d'un mouton et d'une herbe comestible. Il capte la distance et la direction relative (gauche ou droite) d'un agent qu'il voit dans son champ de vision. Il sait identifier l'ennemi le plus proche. Il sait tourner à gauche ou à droite et courir sans dépasser une certaine vitesse maximale. Il possède une réserve d'énergie qu'il regénère en mangeant, et dépense en courant. Par défaut, il va tout droit et finit par s'épuiser. On veut que les mou-

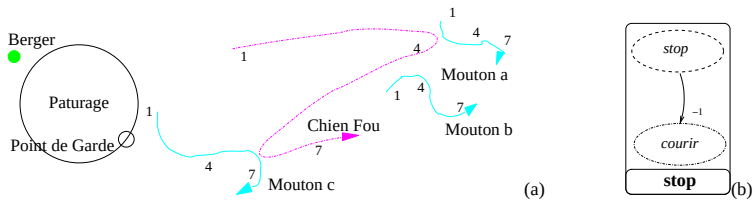


En (a), les cartes cognitives floues définissent le rôle d'un mouton. Les deux du haut déterminent sa direction à un but ; celle du bas lui donne son caractère selon sont degré de paranoïa (α) et de stress (β). En (b), les moutons cherchent à se rejoindre tout en évitant le berger. Les 3 moutons agissent en laissant des empreintes datées sur leurs trajectoires tous les quatre cycles (ici : 1, 5, 9).

Figure 10. *Peur et socialisation chez les moutons*

tons mangent de l'herbe (répartie aléatoirement), aient peur des chiens et des hommes quand ils sont trop près et, selon Panurge, qu'ils cherchent à socialiser. Conformément à la section 2, on choisit alors une carte cognitive principale (figure 10) comportant des concepts sensitifs (*ennemi proche*, *ennemi loin*, *energie haute*, *energie basse*), des concepts moteurs (*manger*, *socialiser*, *fuir*, *courir*) et des concepts internes (*satisfaction*, *peur*). Cette carte calcule la vitesse de déplacement par *défuzzification* du concept *courir*, et la direction du déplacement par défuzzification des trois concepts *manger*, *socialiser* et *fuir*, l'activation interne de chaque concept étant utilisée comme une pondération sur la direction relative à suivre (à gauche ou à droite), respectivement pour aller vers l'herbe, rejoindre un mouton et fuir un ennemi.

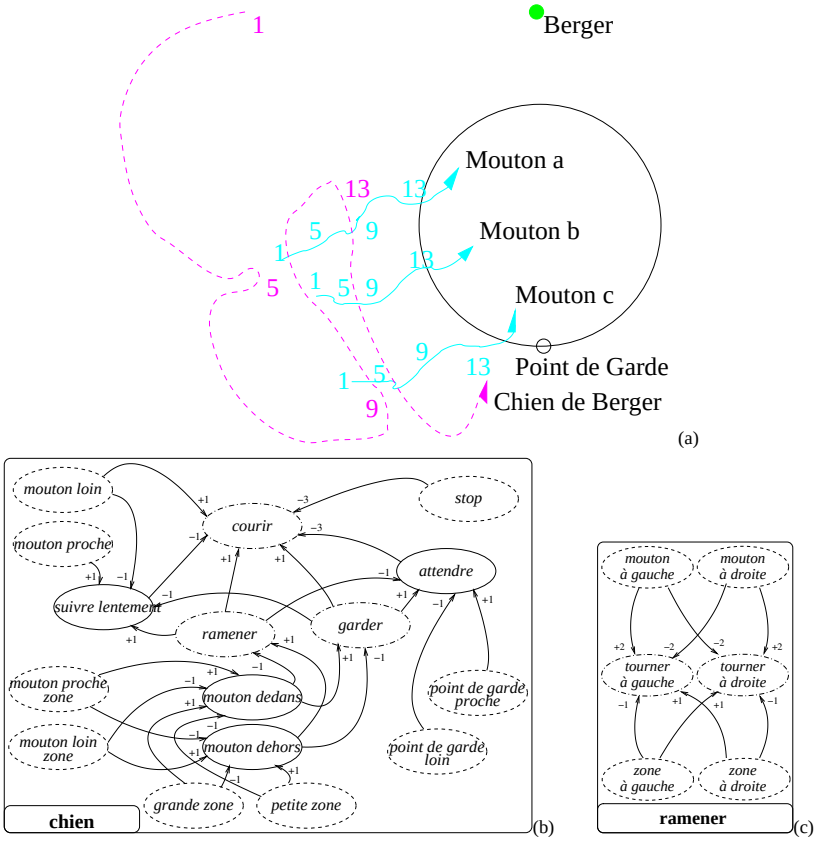
Le chien est capable d'identifier les hommes, les moutons, la zone du pâturage et le point de garde. Il distingue visuellement et auditivement son berger d'un autre homme et sait repérer le mouton le plus loin de la zone parmi un groupe de moutons. Il sait tourner à gauche ou à droite et courir jusqu'à une vitesse maximale. Initialement jeune et fou, il a un comportement consistant à courir après les moutons ; ce qui a pour effet de les disperser rapidement (figure 11a). Dans un premier temps, le berger veut que le chien obéisse à l'ordre de ne pas bouger, ce qui aura comme conséquence une socialisation des moutons. Cela est réalisé en donnant au chien une carte sensible au message du berger et inhibant l'envie de courir (figure 11b). Le comportement du chien est décidé par la carte et le chien s'immobilise quand son maître le lui demande (diffusion du message *stop*).



En (a), le berger est immobile. Une zone circulaire représente la zone de parcage. Sur cette zone est attaché un point de garde toujours situé diamétralement opposé au berger : c'est au point de garde que doit se mettre un chien de berger lorsque tous les moutons sont à l'intérieur de la zone. Initialement, le comportement d'un chien sans carte cognitive floue est de courir après les moutons qui se dispersent rapidement et hors de la zone de regroupement. En (b), la carte élémentaire réalise l'obéissance d'un chien au message *stop* de son berger en inhibant l'envie de courir.

Figure 11. *Le chien fou et les moutons*

Ensuite, le berger donne au chien une carte structurée d'après les concepts associés à la zone du troupeau, au fait qu'un mouton soit hors ou dans cette zone, ainsi que les concepts permettant de ramener un mouton (figure 12) et de maintenir le troupeau dans la zone en se plaçant au point de garde, c'est à dire sur le bord et à l'opposée du berger. Il est remarquable d'observer que la trajectoire en S du chien de berger virtuel émergeant de cette simulation *in virtuo* (figure 12a), et non explicitement programmée, est une stratégie adoptée et observable *in vivo* par un chien de berger réel qui ramène un groupe de moutons.



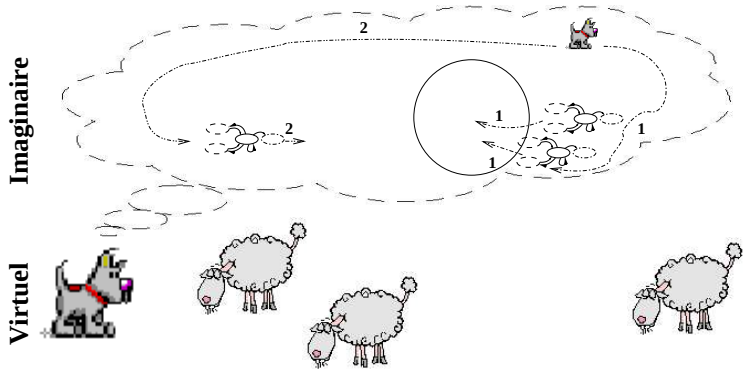
Pour cette simulation, la socialisation des moutons est inhibée. En (a), le film du chien de berger ramenant 3 moutons s'est déroulé avec, pour les moutons, leurs cartes de la figure 10 et pour le chien, les cartes représentées en (b) et (c). En (b), la carte principale du chien traduit le rôle consistant à ramener un mouton dans la zone et à maintenir une position relative au berger lorsque tous les moutons sont dans la zone souhaitée. En (c), la carte décide de l'angle d'approche du mouton pour le ramener vers la zone : aller vers le mouton, mais en l'abordant par le coté opposé à la zone.

Figure 12. *Le chien de berger et les moutons*

5.2. *Simulation dans la simulation*

L'expérience précédente menée avec plusieurs dizaines de moutons montre que, lorsque les moutons se sont regroupés en plusieurs groupes à des distances similaires du paturage, le chien passe son temps à courir d'un groupe à l'autre sans être vraiment efficace, car à peine a-t-il commencé à ramener le groupe le plus éloigné, qu'il passe à un autre groupe du coup plus éloigné. Pour remédier à cela, nous donnons au chien deux manières de concentrer son attention : *vision illimitée* et *vision rapprochée*. La première consiste à tenir compte de tous les moutons visibles, la seconde à ne tenir compte que des moutons visibles dans un voisinage limité (par exemple à moins de $10\,m$). Le berger (l'opérateur humain) peut donner l'ordre au chien (par envoi de mes-

sage) d'adopter l'une ou l'autre de ces deux stratégies, mais afin de rendre le chien de berger plus autonome, nous l'avons doté de la capacité de perception de lui-même et des autres. Le chien possède un monde autonome de simulation dans la simulation, dans lequel il peut se simuler lui-même en utilisant son propre modèle de comportement et simuler les moutons en utilisant le prototype de proie avec la carte de la figure 3(b). Le chien possède donc deux prototypes dans sa bibliothèque de comportements : lui-même et celui d'une proie. Le berger sera statique dans les simulations internes au chien qui ne possède pas de modèle de comportement du berger. Le chien évolue dans un monde virtuel peuplé du même berger que précédemment, et de plusieurs centaines de moutons de "personnalités" différentes grâce à une génération aléatoire des taux de "paranoïa" α et de "stress" β pour chaque mouton (voir figure 10a). Le scénario qu'il réalise dans son monde prédictif de simulation consiste à synchroniser initialement sa propre position et la position de trois proies avec les positions observées dans le monde virtuel (le monde réel de l'animat *chien*), puis de simuler en parallèle dans son monde imaginaire le regroupement des trois proies pendant un certain nombre de cycles (≈ 100) avec chacune des deux stratégies perceptives : en *vision illimitée*, et en *vision rapprochée* (figure 13). Pour similer les trois proies, le chien utilise trois cartes cognitives égales à son prototype de proie, chaque mouton étant représenté par une proie dans les simulations internes du chien. A l'issue de ces deux simulations (*illimitée* et *rapprochée*), un critère de meilleur regroupement des proies (somme des distances des proies à la zone de garde) lui permet de choisir la stratégie de perception la plus adaptée, que le mode prédictif transmet alors au mode réactif pour appliquer la meilleure stratégie. Le chien commence alors une nouvelle simulation interne.



L'animat *chien* vit dans le monde virtuel avec des moutons. Parallèlement, il simule des scénarii dans son propre monde imaginaire en utilisant sa culture comportementale. Il utilise son propre modèle de comportement pour se simuler lui-même et des prototypes du type proie pour simuler les moutons. Le chien peut utiliser deux stratégies d'attention visuelle différentes : vision rapprochée ou vision illimitée. Pour choisir tout seul la stratégie perceptive la mieux adaptée à la situation, il simule régulièrement dans son propre monde imaginaire l'application de chaque stratégie avec les conditions initiales de cette situation (trajectoires prédites en vision rapprochée : tiret-pointillés **1** (la proie éloignée n'a presque pas bougé), en vision illimitée : tiret-double pointillés **2** (les proies moins éloignées n'ont presque pas bougé)), puis adopte dans le monde virtuel la stratégie la plus efficace pour regrouper les proies, ici la **1**, d'après les prédictions réalisées.

Figure 13. *Choix de stratégie par simulations de comportements perceptifs*

Implémentées dans l'environnement de simulation interactive multi-agents *oRis*, les expériences réalisées montrent que le chien virtuel alterne alors les stratégies selon les configurations rencontrées lors de son improvisation libre avec les moutons et le berger : lorsque les moutons forment des groupes situés à égale distance de la zone de garde, le chien de berger passe en mode *vision rapprochée* et ne fait plus d'allers-retours entre les groupes ; il sait également reprendre la stratégie *vision illimitée* pour aller chercher le groupe le plus éloigné.

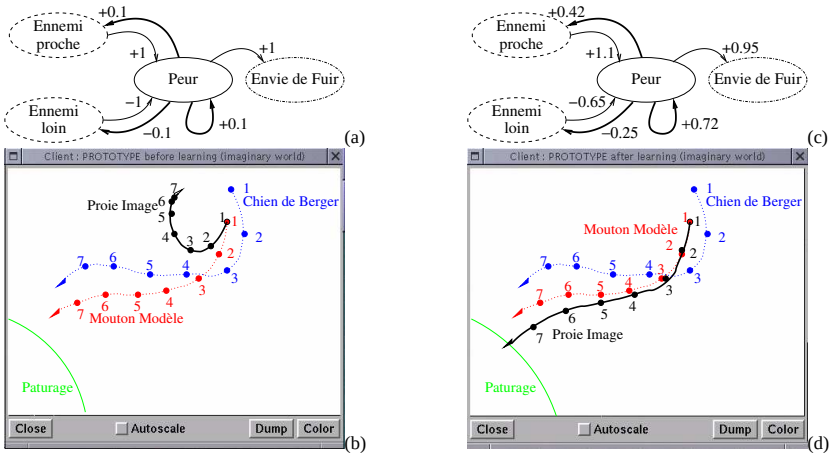
5.3. *Adaptation de prototypes comportementaux*

Nous avons doté le chien de berger de la capacité d'apprentissage par imitation en implémentant l'algorithme de la section 4 dans un processus adaptatif autonome propre au chien, fonctionnant en parallèle avec les deux autres processus cognitifs (réactif et prédictif). Le chien possède dans sa bibliothèque de comportements son propre comportement et le prototype d'une proie. Nous avons alors réalisé trois scénarii utilisant les capacités d'imitation du chien : l'imitation d'un mouton stressé pour améliorer la pertinence du mode prédictif, l'imitation d'un autre chien au comportement plus satisfaisant (contrôlé soit par une autre carte cognitive, soit par un opérateur humain) pour faciliter le réglage des poids d'une carte cognitive lors de la conception d'un comportement, enfin nous avons repris le premier scénario mais en ayant automatisé les périodes d'apprentissage par la perception active.

5.3.1. *Amélioration de la pertinence des prédictions*

Tout en agissant dans le monde virtuel selon les modes réactif et prédictif explicités au paragraphe précédent, le chien modifie son prototype de proie pour imiter au mieux dans son monde imaginaire un mouton donné (sur la figure 14, le mouton modèle est contrôlé par la carte de la figure 10a avec un taux de paranoïa $\alpha = 0.5$ et de stress $\beta = 0.6$). C'est un opérateur humain qui décide du début et de la fin des périodes d'apprentissage. L'expérience acquise par le chien pour réussir l'imitation présentée à la figure 14 représente une centaine de cycles, pendant lesquels le chien s'approche puis s'éloigne deux fois du mouton. Si l'apprentissage devient permanent, on peut observer un stress γ dans le prototype de proie pouvant momentannément diminuer jusqu'à 0,3 lorsque le mouton est resté longtemps (plus de 1000 cycles) loin du chien, mais qui reprend très rapidement (en moins de 10 cycles) une valeur de l'ordre de 0,7 dès que le chien s'est approché à nouveau du mouton. On voit alors l'intérêt d'un taux d'apprentissage constant : l'adaptation reste très réactive quelle que soit la durée de l'apprentissage. L'imitation d'un mouton par une proie modifie le temps de simulation de l'ensemble du monde virtuel comme si l'on y avait ajouté quatre proies.

Nous avons ensuite généré un troupeau de 30 moutons de personnalités différentes (différents α et β dans la carte "perception" du mouton). Le chien de berger ne possédant qu'un unique prototype de proie dans sa bibliothèque de comportements, à chaque fois que le chien poursuivait un nouveau mouton, les poids du prototype se modifiaient et la proie prenait approximativement la personnalité du dernier mouton



Un chien de berger poursuit un mouton particulier, possède dans sa bibliothèque de comportements le prototype de proie et essaie d'imiter le mouton avec son prototype de proie. La carte principale du prototype de proie est décrite en (a) avant apprentissage, en (c) après apprentissage. Dans les mondes imaginaires avant (b) et après (d) apprentissage, seul le comportement de la proie vue par le chien est simulé en trait noir plein ; les trajectoires en pointillés du chien de berger en bleu et du mouton modèle en rouge sont les estimations du chien obtenues par observation du monde virtuel. La proie est simulée comme si elle percevait le chien, en étant synchronisée avec le mouton modèle à $t = 1$. A l'issue de l'apprentissage, le chien de berger voit ce mouton comme une proie stressée ($\gamma = 0,72$) et légèrement paranoïaque ($\lambda = 0,42$). L'enrichissement de la bibliothèque de comportements du chien avec ce prototype de proie identifié à ce mouton permet au chien de réaliser des prédictions plus pertinentes dans son monde de simulation, lorsqu'il s'agit de simuler ce mouton particulier.

Figure 14. *Imitation d'un mouton stressé par un chien de berger*

poursuivi. Ainsi, les prédictions réalisées avec cet unique prototype n'étaient pas toujours pertinentes, notamment lorsque le dernier mouton poursuivi avait une personnalité extrême : une proie très paranoïaque voulait s'enfuir en permanence.
Aussi, nous avons donné au chien un prototype de proie par mouton et avons demandé autant de processus d'apprentissage en parallèle que de moutons dans le troupeau. Nous avons alors obtenu des prédictions pertinentes, chaque proie s'étant rapidement adaptée à chaque mouton (relative stabilité des coefficients en moins de 1500 cycles, temps nécessaire pour que le chien se soit approché au moins une fois de chacun des moutons). Cependant, une telle technique d'apprentissage simultané n'est pas envisageable pour des troupeaux beaucoup plus gros (en utilisant le langage `oRis` (Harrouet *et al.*, 2002) sur notre PC bas de game, à partir de 300 moutons le chien n'a plus le temps d'apprendre en temps réel). Nous verrons que la perception active pour le choix des périodes d'apprentissage pour chacun de ces processus adaptatifs résout en partie ce problème de complexité.

5.3.2. *Aide à la spécification de comportement*

Nous avons également pu faire imiter un chien de berger virtuel spécifique par un prototype de chien ayant les mêmes cartes, mais avec des poids initiaux différents. Après adaptation du prototype, chaque poids était déterminé à 0,1 près sur une base expérimentale de quelques centaines de cycles, avec une convergence très nette des

coefficients (il n'y a pas de connexions récurrentes dans les cartes cognitives du chien de berger). A l'issu de l'apprentissage, le chien imitateur remplace son propre modèle de comportement par le prototype modifié de sa librairie. Son comportement est alors très similaire à celui du chien-modèle, les deux chiens n'étant jamais distants de plus d'un mètre.

Aussi, la même expérience a été reprise, mais le chien-modèle était contrôlé par un opérateur humain et non par des cartes cognitives. Les coefficients convergent relativement bien si l'opérateur humain imite les comportements générés par les cartes cognitives, c'est à dire qu'il respecte la sémantique spécifiée dans les cartes prototypiques. Bien sûr, si l'être humain génère un comportement non "compréhensible" par le prototype de l'imitateur, les coefficients ne convergent pas avec un taux d'apprentissage constant $r(t) = R$. On peut les forcer à converger en choisissant $r(t) = 1/t$. Cependant, même s'il y a convergence, le comportement de l'acteur-image ne correspond bien sûr pas au comportement de l'acteur-modèle humain, dans le cas où celui-ci a montré le "mauvais exemple".

5.3.3. *Perception active pour le choix des périodes d'apprentissage*

Dans le premier scénario adaptatif, la période d'apprentissage est déterminée par l'opérateur humain. Il s'agit ici de ne déclancher les processus d'apprentissage chez le chien qu'en cas d'erreur de prédiction importante entre son monde interne de simulation et le monde effectivement observé pendant qu'il regroupe les moutons.

Ce troisième scénario illustre l'utilisation de la perception active pour l'auto-détermination de la période d'apprentissage. L'autonomisation de l'apprentissage a été implémentée, avec la stratégie de perception active suivante : dès que la distance du chien au mouton-modèle est inférieure à deux fois la distance critique d'éloignement, et qu'elle est non-constante à un seuil près, alors les dix cycles suivants la détection de ce critère sont mémorisés, et la proie est simulée comme dans la figure 14(b ou d) pendant dix cycles. Si, à la fin de la simulation, les positions de la proie et du mouton-modèle sont à une distance supérieure à ce que peut parcourir la proie en un cycle, alors les cycles mémorisés sont envoyés au processus adaptatif.

On observe alors expérimentalement[3] que, lorsque le comportement de la proie est très différent de celui du mouton-modèle, dès que le chien passe à proximité du mouton-modèle, un processus d'apprentissage est déclenché. Ce phénomène d'apprentissage s'arrête lorsque le chien s'est approché ou éloigné du mouton-modèle une quinzaine de fois. Le comportement de la proie, c'est à dire le mouton vu par le chien, imite alors très bien celui du mouton-modèle, lorsque celui-ci est poursuivi par le chien. Le chien peut donc enrichir sa bibliothèque de comportements par ce prototype modifié, afin de simuler ce mouton particulier dans son monde imaginaire.

Lorsque l'on utilise cet auto-apprentissage pour un chien gardant tout un troupeau et cherchant à identifier des prototypes de proie à chacun des moutons du troupeau,

3. Par expérience, nous entendons une simulation des modèles par un système de réalité virtuelle à laquelle participent des opérateurs humains

on se rend compte que ne deviennent actifs que les processus adaptatifs et prédictifs associés aux quelques rares moutons proches du chien. Le chien apprend alors la personnalité de chaque mouton en temps réel, même pour un troupeau de 300 moutons (en utilisant le langage `oRis` et un PC bas de game), et de manière plus pertinente que lors de la dernière expérience du premier scénario.

6. Conclusion

Nous nous sommes intéressés aux comportements perceptifs d'acteurs virtuels autonomes. Nous avons décrit comment utiliser les cartes cognitives floues où les états internes sont explicitement représentés, pour spécifier, contrôler, prédire et adapter de tels comportements, dont l'exécution nécessite une architecture faisant fonctionner les trois modes — réactif, prédictif, adaptatif — en parallèle et de manière asynchrone.

Nous utilisons les cartes cognitives floues comme un outil formel déterminant un comportement défini par une expertise, et non pas dans l'idée qu'elles puissent représenter un plan topologique de l'environnement. Avec la structure de la carte cognitive floue, nous spécifions des comportements pouvant être perceptifs ; la perception, à la différence de la sensation, dépendant des états internes de l'acteur virtuel. Avec la fuzzification des capteurs vers les concepts perceptifs, la dynamique, puis la défuzzification des concepts moteurs, les propriétés asymptotiques de la dynamique de la carte cognitive floue assurent le contrôle d'un processus réactif par des décisions crédibles, au sens où elles obéissent à la sémantique explicitement représentée dans la carte.

Un mécanisme de simulation dans la simulation implémenté au niveau de chaque agent autonome donne accès à la perception active pour nos acteurs virtuels. En même temps qu'il agit dans le monde virtuel, chaque agent peut simuler dans son monde imaginaire son propre comportement ou celui d'autres entités, à partir de sa culture comportementale composée de groupes de cartes cognitives floues, chacun d'eux spécifiant un prototype de comportement (lui-même, proie, prédateur, ...), le choix du prototype restant *ad hoc*. Cette simulation dans la simulation lui donne la capacité de perception de soi et des autres en prédisant l'évolution du monde virtuel, et lui permet notamment de choisir une stratégie sensorimotrice parmi plusieurs ou de coopérer avec d'autres entités, non pas par un raisonnement logique, mais par une simulation de comportements, tout en continuant d'agir selon son mode réactif.

Nous avons ensuite abordé l'auto-apprentissage par imitation à partir des cartes cognitives d'une bibliothèque de comportements prototypiques, sans étudier le problème du choix du prototype. L'algorithme proposé utilise un principe de remises en cause discrètes pouvant intégrer des méta-connaissances sur l'apprentissage, plutôt qu'une approche numérique continue de type gradient ou génétique. Sa complexité polynomiale en $\mathcal{O}(n)$ permet d'envisager son utilisation en temps réel, même pour des comportements complexes faisant intervenir un nombre n important de concepts. Ce processus adaptatif peut fonctionner en parallèle au processus réactif de contrôle et au processus prédictif de simulation ; il est non bloquant ni au niveau du mode

réactif, ni au niveau du mode prédictif. La perception active générée par ces deux derniers modes est utilisée pour l'autonomisation du choix des périodes d'apprentissage : l'apprentissage a lieu lorsqu'une erreur de prédiction est détectée.

Enfin, l'implémentation de la fiction intéractive non triviale — le berger, son chien et son troupeau — nous a permis d'illustrer l'ensemble de ces possibilités offertes par les cartes cognitives floues pour la spécification, le contrôle, la simulation et l'adaptation du comportement perceptif d'acteurs virtuels autonomes.

A l'heure actuelle, c'est l'opérateur humain qui fourni le choix des prototypes pour les simulation internes et pour l'imitation des acteurs virtuels. Transférer cette compétence au niveau des acteurs virtuels augmentera leur autonomie, en automatisant l'ensemble du processus.

Nous utiliserons les cartes cognitives pour spécifier des comportements d'humains virtuels. Les futurs travaux se feront en collaboration avec des psychologues-ergonomes pour la spécification de cartes cognitives d'humains virtuels, dans le cadre de la fiction interactive. Parallèlement à ces recherches sur la spécification de cartes cognitives pour humains virtuels, l'architecture proposée combine des aspects cognitifs avec des aspects physiologiques, et l'on peut envisager qu'en tant qu'espace d'expérimentation, les simulations puissent servir de support à des travaux articulant l'approche symbolique (Shepard, 1984) avec l'approche écologique (Morineau, 2004) pour l'étude de l'adaptation de l'Homme à son environnement naturel ou artificiel.

7. Bibliographie

Aissaoui G., Genest D., Loiseau S., « Le modèle des cartes cognitives de graphes conceptuels : un modèle graphique d'aide à la prise de décision », *in* A. Herzig, B. Chaib-draa, P. Mathieu (eds), *Modèles Formels de l'Interaction (MFI)*, Cépaduès, Lille, p. 243-248, 2003.

Arnaldi B., « Modèles physiques pour l'animation », , Habilitation à Diriger des Recherches (HDR), Université de Rennes 1, Rennes, France, 1994.

Axelrod R., *Structure of decision*, Princeton University Press, USA, 1976.

Badler N., Phillips C., Webber B., *Simulating humans : computer graphics animation and control*, Oxford University Press, New York, 1993.

Bates J., « Virtual reality, art, and entertainment », *Presence*, vol. 1, n° 1, p. 133-138, 1992.

Bernstein N., *The coordination and regulation of movement*, Pergamon Press, New York, 1967.

Berthoz A., *Le sens du mouvement*, Odile Jacob, Paris, 1997.

Bessière P., Dedieu E., Lebeltel O., Mazer E., Mekhnacha K., « Interprétation ou description (I) : proposition pour une théorie probabiliste des systèmes cognitifs sensori-moteurs », *Intellectica*, vol. 26-27, n° 1-2, p. 257-311, 1999a.

Bessière P., Dedieu E., Lebeltel O., Mazer E., Mekhnacha K., « Interprétation ou description (II) : fondements mathématiques de l'approche F+D », *Intellectica*, vol. 26-27, n° 1-2, p. 313-336, 1999b.

Blumberg B., Downie M., Ivanov Y., Berlin M., Johnson M. P., Tomlinson B., « Integrated learning for interactive synthetic characters », *SIGGRAPH '02 : Proceedings of the 29th annual*

conference on Computer graphics and interactive techniques, ACM Press, New York, NY, USA, p. 417-426, 2002.

Boulic R., Magnenat-Thalmann N., Thalmann D., « A global human walking model with real-time kinematic personification », *Visual Computer*, vol. 6, n° 6, p. 344-358, 1990.

Brooks R., « A Robust Layered Control System For A Mobile Robot », *itra*, vol. RA-2, n° 1, p. 14-23, March, 1986. ISSN 0882-4967.

Brunia C., « Neural aspects of anticipatory behavior », *Acta Psychologica*, vol. 101, p. 213-242, 1999.

Buzsaki G., Horvath Z., Urioste R., Hetke J., Wise K., « High frequency network oscillations in the hippocampus », *Science*, vol. 256, p. 1025-1027, 1992.

Chadwick J., Haumann D., Parent R., « Layered construction for deformable animated characters », *Computer Graphics*, vol. 23, n° 3, p. 243-252, 1989.

Chaib-Draa B., Demolombe R., « L'interaction comme champ de recherche », *Information-Interaction-Intelligence*, 2002. Numéro hors série sur l'interaction.

Cornuéjols A., Miclet L., Kodratoff Y., *Apprentissage artificiel : concepts et algorithmes*, Eyrolles, Paris, 2002.

Craik K., *The nature of explanation*, Cambridge University Press, 1943.

Del Bimbo A., Vicario E., « Specification by example of virtual agents behavior », *IEEE Transactions on Visualization and Computer Graphics*, vol. 1, n° 4, p. 350-360, 1995.

Dewey J., « The reflex arc concept in psychology », *Psychological Review*, vol. 3, p. 357-370, 1896.

Dickerson J., Kosko B., « Virtual worlds as fuzzy cognitive maps », *Presence*, vol. 3, n° 2, p. 173-189, 1994.

Donikian S., « Les modèles comportementaux pour la génération du mouvement d'objets dans une scène », *Revue internationale de CFAO et d'infographie*, vol. 9, n° 6, p. 847-871, 1994.

Donikian S., « HPTS : a behaviour modelling language for autonomous agents », *AGENTS '01 : Proceedings of the fifth international conference on Autonomous agents*, ACM Press, New York, NY, USA, p. 401-408, 2001.

Fechner G., *Elemente der psychophysik*, Leipzig : Breitkopf und Härtel, 1860.

Fuchs P., Moreau G., Papin J., *Le traité de la réalité virtuelle*, Presses de l'Ecole des Mines, Paris, 2001.

Gallese V., « The inner sense of action : agency and motor representations », *Journal of Consciousness Studies*, vol. 7, n° 10, p. 23-40, 2000.

Gaussier P., Moga S., Quoy M., Banquet J., « From perception-action loops to imitation process : a bottom-up approach of learning by imitation », *Applied Artificial Intelligence*, vol. 12, p. 701-727, 1998.

Gaussier P., Revel A., Banquet J., Babeau V., « From view cells and place cells to cognitive map learning : processing stages of the hippocampal system », *Biological Cybernetics*, vol. 86, p. 15-28, 2002.

Guillot A., Meyer J., « From SAB94 to SAB2000 : what's new, animat ? », *From Animals to Animats (SAB)*, vol. 6, p. 1-10, 2000.

Harrouet F., Tisseau J., Reignier P., Chevailler P., « oRis : un environnement de simulation interactive multi-agents », *Technique et Science Informatiques (RSTI-TSI)*, vol. 21, n° 4, p. 499-524, 2002.

Hayes-Roth B., Van Gent R., Story-making with improvisational puppets and actors, Technical Report n° KSL-96-05, Stanford University, Stanford, CA, USA, 1996.

Hebb D., *The organization of behaviour*, John Wiley and Sons, New York, USA, 1949.

Heinze A., Gross H., « Anticipation-based control architecture for a mobile robot », *International Conference on Artificial Neural Networks (ICANN)*, p. 899-905, 2001.

Holland J., *Adaptation in natural and artificial systems*, University of Michigan Press, Ann Arbor, 1975.

Kesner R., Rolls E., « Role of long-term synaptic modification in short-term memory », *Hippocampus*, vol. 11, p. 240-250, 2001.

Klesen M., Szatkowski J., Lehmann N., « The black sheep : interactive improvisation in a 3D virtual world », *I3*, p. 77-80, 2000.

Kosko B., « Fuzzy cognitive maps », *International Journal of Man-Machine Studies*, vol. 24, p. 65-75, 1986.

Kosko B., « Hidden patterns in combined and adaptative knowledge networks », *International Journal of Approximate Reasoning*, vol. 2, p. 337-393, 1988.

Kosko B., *Neural networks and fuzzy systems : a dynamical systems approach to machine intelligence*, prentice-hall edn, Engelwood Cliffs, 1992.

Koulouriotis D., Diakoulakis I., Emiris D. M., Antonidakis E., Kaliakatsos I., « Efficiently modeling and controlling complex dynamic systems using evolutionary fuzzy cognitive maps », *International Journal of Computational Cognition*, vol. 1, n° 2, p. 41-65, 2003. http://www.YangSky.com/yangijcc.htm.

Kuiper B., « The spatial semantic hierarchy », *Artificial Intelligence*, vol. 119, p. 191-233, 2000.

Langton C., « Studying artificial life with cellular automata », *Physica*, vol. D22, p. 120-149, 1986.

Lisman J., Idiart M., « Storage of short term memories in oscillatory subcycles », *Science*, vol. 267, p. 1512-1515, 1995.

Maes P., « The agent network architecture (ANA) », *SIGART Bull.*, vol. 2, n° 4, p. 115-120, 1991.

Maes P., « Artificial life meets entertainment : interacting with lifelike autonomous agents », *Communications of the ACM*, vol. 38, n° 11, p. 108-114, 1995.

Magnenat-Thalmann N., Laperriere R., Thalmann D., « Joint-dependent local deformations for hand animation and object grasping », *Graphics Interface*, p. 26-33, 1988.

Magnenat-Thalmann N., Thalmann D., « Complex models for animating synthetic actors », *IEEE Computer Graphics and Applications*, vol. 11, n° 5, p. 32-44, 1991.

Mataric M., « Visuo-motor primitives as a basis for learning by imitation : linking perception to action and biology to robotics », *in* K. Dautenhahn, C. Nehaniv (eds), *Imitation in Animals and Artifacts*, MIT Press, p. 392-422, 2002.

Meltzoff A., « Understanding the intentions of others : re-enactement of intended acts by 18-month-old children », *Developmental Psychology*, vol. 31, p. 838-850, 1995.

Morineau T., « L'émergence d'une perspective écologique en psychologie ergonomique, à travers la distinction entre différents niveaux de contrôle cognitif dans l'activité », *Colloque de l'Association pour la Recherche Cognitive (ARCo)*, Compiègne, France, p. J3-S1, 2004.

Mozer M., « A focused BACKPROPAGATION algorithm for temporal pattern recognition », *in* Y. Chauvin, D. Rumelhart (eds), *Backpropagation : Theory, architechtures and applications*, Lawrence Erlbaum Associates, p. 137-169, 1995.

Newell A., Simon H., « Computer science as empirical enquiry », *Communications of the ACM*, vol. 19, p. 113-126, 1976.

Noma T., Zhao L., Badler N., « Design of a virtual human presenter », *Computer Graphics*, vol. 20, n° 4, p. 79-85, 2000.

Paillard J., « Réactif et prédictif : deux modes de gestion de la motricité », *Pratiques sportives et modélisation du geste*, Nougier, V. and Bianch, J.P., Grenoble, UFRAPS, p. 13-56, 1990.

Perlin K., Goldberg A., « Improv : a system for scripting interactive actors in virtual worlds », *Computer Graphics*, vol. 30, p. 205-216, 1996. http://www.mrl.nyu.edu/projects/improv/.

Platt S., Badler N., « Animating facial expressions », *Computer Graphics*, vol. 15, n° 3, p. 245-252, 1981.

Rickel J., Johnson W., « Animated agents for procedural training in virtual reality : perception, cognition, and motor control », *Applied Artificial Intelligence*, vol. 13, p. 343-382, 1999.

Rosenblum R., Carlson W., Tripp E., « Simulating the structure and dynamics of human hair : modelling, rendering and animation », *Journal of Visualization and Computer Animation*, vol. 2, n° 4, p. 141-148, 1991.

Sanza C., Evolution d'Entités Virtuelles Coopératives par Système de Classifieurs, PhD thesis, Université Paul Sabatier, 2001.

Schultz A., Grefenstette J., Adams W., « RoboShepherd : learning a complex behavior », *RoboLearn*, p. 105-113, 1996.

Shepard R., « Ecological constraints on internal representation : resonant, kinematics of perceiving, imagining, thinking and dreaming », *Psychological Review*, vol. 91, p. 417-447, 1984.

Sowa J., *Principles of semantic network : exploration in the representation of knowledge*, Morgan Kaufmann, San Mateo, CA, USA, 1991.

Sugeno M., Nishida M., « Fuzzy control of a model car », *Fuzzy Sets and Systems*, vol. 16, p. 103-113, 1985.

Sutton R., « Learning to predict by the methods of temporal differences », *Machine Learning*, vol. 3, p. 9-44, 1988.

Tisseau J., Harrouet F., *Le traité de la réalité virtuelle*, vol. 2, Presses de l'Ecole des Mines, Paris, chapter Autonomie des entités virtuelles, 2003.

Tolman E., « Cognitive maps in rats and men », *Psychological Review*, vol. 55, n° 4, p. 189-208, 1948.

Trullier O., Meyer J., « Animat navigation using a cognitive graph », *Biological Cybernetics*, vol. 83, p. 271-285, 2000.

van de Panne M., Fiume E., « Sensor-Actuator Networks », *Proc. of SIGGRAPH-93 : Computer Graphics*, Anaheim, CA, p. 335-342, 1993.

Vaughan R., Sumpter N., Henderson J., Frost A., Cameron S., « Experiments in automatic flock control », *Journal of Robotics and Autonomous Systems*, vol. 31, p. 109-117, 2000.

Voyles R., Morrow J., P. K., « Towards gesture-based programming : Shape from motion primordial learning of sensorimotor primitives », *Robotics and Autonomous Systems*, vol. 22, n° 3-4, p. 361-375, 1997.

Weil J., « The synthesis of cloth objects », *Computer Graphics*, vol. 20, n° 4, p. 49-54, 1986.

Wenstop F., « Deductive verbal models of organizations », *International Journal of Man-Machine Studies*, vol. 8, p. 293-311, 1976.

William R., Zipser D., « Gradient-based learning algorithms for recurrent networks and their computational complexity », *in* Y. Chauvin, D. Rumelhart (eds), *Backpropagation : Theory, architechtures and applications*, Lawrence Erlbaum Associates, p. 433-486, 1995.

Wilson S., « Knowledge growth in an artificial animal », *Genetic Algorithms and their Applications*, p. 16-23, 1985.

Jacques Tisseau *est Professeur des Universités en Informatique à l'Ecole Nationale d'Ingénieurs de Brest (ENIB) où il dirige le Centre Européen de Réalité Virtuelle (CERV). Ses recherches portent sur l'autonomisation des entités virtuelles, l'interaction avec ces entités autonomes et l'épistémologie de la réalité virtuelle.*

Marc Parenthoën *est professeur agrégé à l'ENIB et chercheur au CERV. Ses travaux portent sur la modélisation des phénomènes naturels pour une instrumentation en réalité virtuelle.*

Cédric Buche *est post-doctorant au CERV. Ses travaux concernent les environnements virtuels de formation et les techniques d'apprentissage artificiel.*

Patrick Reignier *est Maître de Conférences à l'Université Joseph Fourier. Sa recherche concerne l'apprentissage pour les assistants virtuels et la reconnaissance de contexte en informatique pervasive.*