

# Projet Warbot - DEA Info 2004

Aurélien Campéas, Samuel Poujol

19 février 2004

## 1 Introduction

Seuls les agents Explorer et RocketLauncher ont des capacités évoluées. L'agent Home est totalement passif et se contente d'appeler à l'aide en cas d'agression.

## 2 Architecture des Agents

Nous parlons surtout du RocketLauncher, qui est le modèle le plus développé. Il a une mémoire, des buts/engagements et effectue la sélection d'action en fonction des percepts, des messages reçus et de son engagement actuel.

### 2.1 Organisation interne (statique)

Sa mémoire est composée des éléments suivants :

- un Sac de Percepts : un ensemble des percepts de l'instant, partitionné selon le type des percepts,
- une Boîte aux Lettres : ensemble des messages reçus pendant le tour, partitionné d'après la sémantique des messages,
- une Liste des Tâches : ensemble ordonné des tâches à réaliser pour ce tour,
- un Gestionnaire de Buts : une pile de buts, des méthodes de sélection de but.

La mainloop de l'agent (la méthode *doIt*) consiste à :

1. mettre à jour le sac de percepts et la boîte aux lettres,
2. sélectionner un but pour ce tour (calcul de situation global réalisé par le Gestionnaire de Buts),
3. exécuter la liste des tâches courantes, et construire les tâches (continuations) du prochain tour (calcul de situation local).

Une telle description ne rend pas compte du rapport entre les tâches de l'agent et la sélection d'action. Voyons l'aspect dynamique.

## 2.2 Gestion des Buts, décision globale

L'agent a quatre buts principaux : Exploration, Attaque d'un ennemi (RocketLauncher ou Base), Défense de sa base.

1. l'exploration représente le comportement de base de l'agent, c'est aussi son activité la moins prioritaire,
2. la défense de la base constitue le premier degré d'action,
3. l'attaque d'un ennemi (RocketLauncher) est prioritaire sur l'Exploration et la Défense,
4. l'attaque de la Base ennemie (lorsqu'elle est découverte) est la priorité absolue (et cela coïncide avec l'objectif global des agents).

On a ici un ordre total statique - un peu simpliste - sur les Buts de l'agent, mais le système Warbot donne la prime à l'agressivité. Nous avons cependant envisagé (sur le papier) la possibilité de donner des priorités dynamiques à certains buts : en particulier la défense de la base, dont la priorité pourrait varier en fonction du rapport attaquants/attaqués, de la santé de la base, etc. L'implémentation du Gestionnaire de Buts permet en fait d'ajouter simplement les priorités dynamiques si besoin est.

L'engagement de l'agent est représenté comme ceci : si l'agent exécute son but courant, et qu'aucun but de priorité supérieure n'est *calculé* (d'après ses percepts et les messages reçus), il continue. Si son but courant est terminé, il revient au but de niveau inférieur. L'exploration est le but *de plus bas niveau*.

## 2.3 Des buts aux Tâches, décision locale

Lorsqu'un agent est engagé dans un but, il exécute un certain nombre de tâches élémentaires. Ces tâches sont :

1. initialisées lorsque l'agent sélectionne un nouveau But,
2. exécutées et éventuellement reconduites sinon.

A chaque But est associé un environnement de travail *global* (différent de l'environnement global de l'agent). Cet environnement contient des informations propres au But et partagées par les tâches associées au But.

Chaque tâche a son propre environnement *lexical*, c'est à dire reçoit des paramètres spécifiques à son exécution.

L'*initialisation* consiste donc à créer l'environnement global et à appeler la première tâche courante du But.

L'*exécution* consiste à :

1. exécuter les tâches de la Liste des Tâches, dans l'ordre (d'ajout, cf. point 2),
2. pour chaque tâche exécutée, contruire la prochaine Liste des Tâche, c'est à dire *éventuellement* (la **décision est purement locale**) y ajouter une continuation.

Enfin, pendant l'exécution d'une tâche particulière, l'analyse de l'environnement peut conduire à un changement de But, qui se traduit par un appel au Gestionnaire de Buts. Par exemple, en mode *Attaque de RocketLauncher*, si :

- il n'y a plus d'ennemi dans nos percepts, et
- il n'y a pas d'ennemi rapporté par nos pairs,

on demande au Gestionnaire des Tâches de revenir au but de *priorité inférieure*.

L'intérêt de cette approche est de permettre une forme de parallélisme. Ce parallélisme n'est en pratique exploité que dans la phase d'initialisation, mais pourrait être développé (par exemple, une gestion des communications indépendante de la sélection d'action).

## 2.4 Etats et Transitions

Finalement, on peut voir le modèle de comportement de l'agent (actions) comme un automate à états finis déterministe à *deux niveaux* (où les Buts et les Tâches sont des sommets) :

Le **niveau des Buts**, où les transitions sont de deux type : **interrompt** et **reprend**.

Par exemple :

- [Attaque Base Ennemie]  $\leftarrow$  (interrompt)  $\leftarrow$  [Défense Base], et
- [Exploration]  $\leftarrow$  (reprend)  $\leftarrow$  [Défense Base].

Avec l'ordre total sur les Buts, il est facile de faire un petit dessin.

Le **niveau des Tâches**, où les transitions sont du type **continue**.

Par exemple, dans le contexte du But *Attaque Ennemi* :

- [Attaque RocketLauncher]  $\leftarrow$  (continue)  $\leftarrow$  [Trouver Ennemi Distant] ou
- [Fuite]  $\leftarrow$  (continue)  $\leftarrow$  [Attaque RocketLauncher] ou finalement
- [Fuite]  $\leftarrow$  (continue)  $\leftarrow$  [Fuite].

## 2.5 Implémentation

L'implémentation de ce système est un peu en-deçà de ce que nous avons décrit. Une partie de l'infrastructure nécessaire est en place, mais les impératifs d'efficacité nous ont forcé la main... Une stratégie centrée sur l'attaque tous azimuts est donc retenue, avec seulement deux modes de base : exploration et attaque. La nature profondément réactive des agents n'en ressort que davantage.

## 3 Gestion des conflits dans l'espace

Pour des agents situés dans l'espace ayant des impératifs de coordination (en particulier par rapport à des Buts situés dans l'espace aussi), il faut un minimum de règles de savoir-vivre (ne pas se gêner ou se tirer dessus). Nous avons adopté un modèle attraction/répulsion, où la *cible courante* est attractive, à longue portée, et *tous* les objets des percepts (y compris la cible) sont répulsifs, à courte portée. Une pondération de la force de répulsion est fournie, en fonction de la nature de l'objet.

L'objectif est de fournir un comportement purement réactif et le plus efficace. C'est à peu près réussi, mais le *tuning* des paramètres qui contrôlent la répulsion et l'attraction est délicat.

A chaque tour, l'orientation de l'agent est donc calculée en fonction de son objectif et des répulsifs locaux (somme vectorielle).

Lorsque l'agent doit et peut tirer, il calcule si aucun agent ami ne se trouve dans la ligne de tir. On calcule pour cela un cône de non-agression associé à chaque agent ami des percepts.