

Rapport de stage pour l'obtention du DEA IARFA de l'université Paris 6:

Structures auto-organisées : modélisation et simulation d'agents constructeurs de ponts

Directeur de stage : Professeur A. Drogoul.
Laboratoire d'accueil : LIP6.

Abstract :

Dans ce rapport, nous décrivons en détail un modèle qui représente des agents capables de s'accrocher les uns aux autres pour former des « ponts » ; qu'ils soient vivants comme les fourmis *oecophylla* ou mécaniques comme les robots reconfigurables. Notre modèle est un automate fini déterministe à sept états et quatorze transitions. Nous illustrons son fonctionnement en présentant trois simulations de différentes stratégies de formation de pont par des agents artificiels.

I. Introduction :

Nous allons, dans ce rapport, nous intéresser à des agents artificiels capables de s'accrocher les uns aux autres pour former dynamiquement des ponts "émergents". Bien que dans la nature ce phénomène se rencontre assez peu, on peut donner l'exemple des fourmis *oecophylla*. Cette espèce est en effet capable de former des chaînes de plusieurs dizaines d'individus afin de passer d'une branche à l'autre d'un arbre. Cela représente un avantage considérable sur les autres insectes.



Figure Erreur! Argument de commutateur inconnu. - Pont de fourmis

Malheureusement, à part des photographies et quelques constatations de ce phénomène [cf. 8], il n'existe aucune description précise de la construction de ces ponts par les fourmis. En particulier, les comportements individuels qui permettent l'émergence d'une structure aussi évoluée ne sont absolument pas connus.

Lorsque des agents, notamment des insectes, s'accrochent les uns aux autres pour créer ensemble une structure ayant un rôle bien défini, on qualifie cela d'auto-assemblage. Le terme a été employé pour la première fois par A. Lioni dans sa thèse [cf. 9]. Les auto-assemblages sont des cas particuliers des phénomènes d'agrégation d'individus. L'article suivant de C. Anderson [cf. 2] fait une liste exhaustive des différents auto-assemblages qu'on peut trouver chez les insectes et qui ont chacun une utilité et une forme propre.

A l'origine, le sujet de stage prévoyait que nous partions de l'observation des fourmis bâtisseuses de pont et que grâce à une description précise de leur activité constructrice nous transposions leurs comportements à des robots simulés. Après une recherche bibliographique sur le sujet et la constatation qu'il n'existait pas de base suffisante pour réaliser ce travail, nous avons décidé de faire évoluer le sujet du stage vers la découverte d'un modèle capable de représenter des agents effectuant un pont par auto-assemblage et de valider ce modèle par la simulation de plusieurs types d'agents mettant en jeu différentes stratégies.

Un pont permet à ceux qui l'empruntent d'éviter un grand détour, voire d'être définitivement bloqués. L'importance stratégique de pouvoir franchir les obstacles tels que les fossés a motivé depuis longtemps les militaires dans l'élaboration de véhicules poseurs de ponts. En dotant un groupe d'individus de la capacité de franchir les fossés on augmente grandement sa marge de manoeuvre et surtout on lui donne la possibilité d'évoluer en terrain accidenté. Cette propriété est essentielle pour les robots mobiles modernes et bon nombre de roboticiens ont travaillé sur ce problème. Cependant, en robotique traditionnelle, où la notion de « collectif »

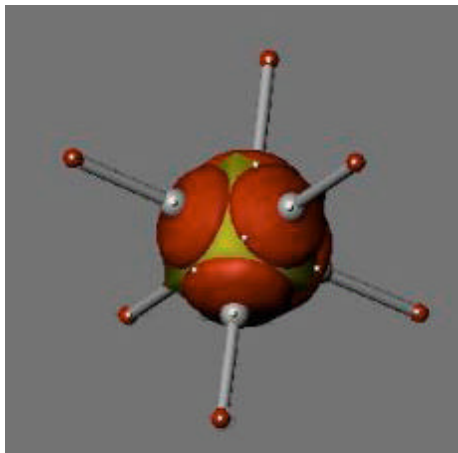
n'est qu'à peine abordée, la construction d'un pont par accrochage n'est pas envisageable. Par contre, c'est un champ d'investigation qui recèle beaucoup de potentiel dans le domaine appelé la robotique reconfigurable.

La robotique reconfigurable est une branche de la robotique qui, au lieu d'utiliser un seul "gros" robot difficile à concevoir, prône l'utilisation d'un grand nombre d'entités plus petites, assez simples individuellement, et capables de s'auto-organiser. Cette approche promet, à terme, de minimiser les coûts de production et d'offrir une très grande tolérance aux pannes. En effet, dans le cas d'un robot unique, la panne d'un composant peut engendrer le blocage total alors qu'en robotique reconfigurable (du moins en théorie) la perte de quelques unités ne change en rien le déroulement de l'action.

La programmation de ces robots utilise des concepts massivement multi-agents : A partir de robots simples ayant chacun une vision locale de la situation, on obtient un "meta-robot" capable d'accomplir parfaitement la tâche souhaitée.

Il existe à l'heure actuelle beaucoup de projets de robotique reconfigurable. [Par exemple **3, 4, 6, 10, 11, 12, 14**]. Mon stage se déroule dans le cadre de l'un d'eux : Le projet MAAM [cf. **10**]. C'est un projet qui regroupe six laboratoires (dont le LIP6) et qui est financé par le CNRS dans le cadre du programme ROBEA. MAAM signifie "Molecule is Atom or Atom and Molecule" ce qui souligne le côté modulaire de cette approche.

Son but est de définir, spécifier, concevoir et réaliser un ensemble d'atomes robotiques capables de s'assembler en une molécule qui pourra, par reconfigurations successives, réaliser une tâche donnée.

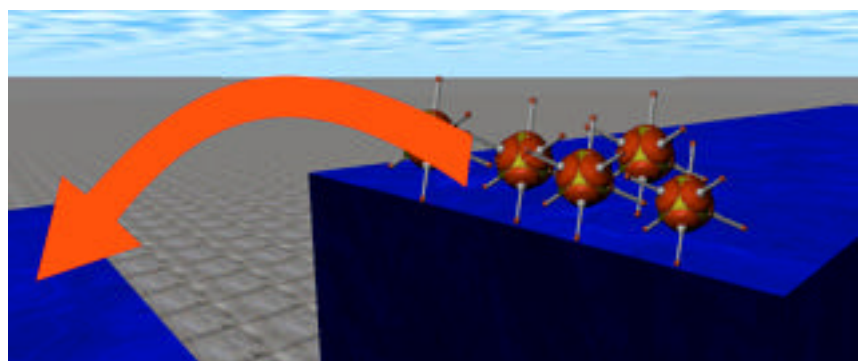


Les robots ont une géométrie "en oursin". Chaque patte est munie, à son extrémité, d'un dispositif d'accrochage par aimantation et peut bouger dans tout l'hémisphère qui la contient ; ce qui donne six points d'accrochage et permet donc un très grand nombre de combinaisons.

De plus, les pattes sont munies de détecteurs infrarouges pour se trouver les uns les autres et permettre à deux robots de s'accrocher.

Figure Erreur! Argument de commutateur inconnu. - **Un atome MAAM.**

Le projet MAAM s'est fixé comme objectif au bout de trois années de recherche la réalisation de la tâche suivante : 10 robots sont répartis aléatoirement sur le sol. Ils doivent se trouver et s'assembler. Puis ils doivent monter ensemble sur un premier bloc et passer sur un deuxième en formant un pont. Enfin, leur dernier but est de descendre ensemble.



Notre travail de stage s'intègre donc assez naturellement dans ce projet.

Nous allons commencer par une présentation détaillée du modèle générique (i.e., indépendant des contraintes physiques auxquelles sont soumis les agents) que nous avons développé : nous donnerons une description de l'automate et de ses états, puis nous nous intéresserons à son fonctionnement et nous justifierons nos choix. Suite à cela, nous présenterons trois études de cas. Enfin, nous tenterons de faire une critique objective de notre stage avant de conclure et de montrer que notre travail ouvre la voie à des recherches passionnantes.

II. Présentation du modèle :

Dans le but de modéliser des agents capables de former dynamiquement des ponts nous avons choisi d'utiliser une architecture d'automate fini déterministe. Un automate fini déterministe est un objet mathématique qui peut être défini par un quintuplet $M = (Q, S, T, I, F)$, où

- ? Q est un ensemble fini d'états,
- ? S est un alphabet
- ? $T : Q \times S \rightarrow Q$ est la fonction de transition,
- ? I appartient à Q et est l'état initial,
- ? F est contenu dans Q et est l'ensemble des états accepteurs.

De plus un automate est capable de lire sur son entrée des informations séquentielles qui appartiennent à S .

Un automate est une machine qui résout un problème, c'est à dire qui reconnaît (accepte) un certain langage. La machine démarre à l'état I et lit sur son entrée une information s . En fonction de son état courant et de s , la fonction de transition T nous permet de savoir quel sera le nouvel état de l'automate.

Dans notre modèle les états représentent des "comportements" bien spécifiques. Par "comportement" nous entendons une sélection de l'action mais également une subjectivité vis à vis de l'appréhension de l'environnement. Ainsi deux jeux d'informations identiques transmis par les capteurs n'auront pas la même valeur sémantique suivant l'état dans lequel se trouve l'automate. L'état conditionne la compréhension de l'agent du monde qui l'entoure et son action dans celui-ci. Un état tend à réaliser un morceau de la solution du problème global.

Les informations que l'automate peut lire sont les données transmises par les capteurs et représentent donc l'environnement à un certain instant.

Nous supposons l'exemple suivant : un groupe d'agents devant atteindre un but qui se trouve de l'autre côté d'un fossé. Le franchir grâce à un auto-assemblage peut se faire de cette façon: Les agents détectent le but et se dirigent dans sa direction. Quand ils rencontrent le fossé ils se reconfigurent puis se déploient pour construire le pont. Quand des agents touchent l'autre côté, le pont se résorbe et tous reprennent leur déplacement vers le but.

Nous nous sommes demandés quel était le nombre minimal d'états que notre modèle devait avoir pour autoriser le franchissement d'un fossé. Pour cela nous avons commencé par faire

l'automate le plus simple possible. Puis nous avons complexifié notre modèle pour lui permettre de réaliser des tâches de plus en plus évoluées.

Voici quel a été notre parcours :

Un agent inactif ne comporte qu'un état. Notre premier souci a été d'avoir un agent capable de détecter l'attractant et de se diriger vers lui. L'attractant est ce qui motive les agents, leur récompense, l'endroit où ils doivent se rendre. Sans attractant il n'y a aucune raison de faire un pont. Pour cela nous avons eu besoin de trois états. Puis nous avons voulu que les agents réagissent aux obstacles et surtout aux fossés. Nous avons donc ajouté un état de réaction. Ensuite, nous avons cherché à amorcer une construction du pont par déplacement, nous avons eu besoin d'un état supplémentaire pour cela. L'étape suivante a consisté à permettre au début de pont d'avoir une stabilité suffisante et donc à faire passer les agents dans un état autre que celui de construction active. Enfin un dernier état nous a permis de faire "prendre conscience" aux agents qu'ils touchaient l'autre bord et que donc le pont était terminé, qu'ils pouvaient envisager de le résorber.

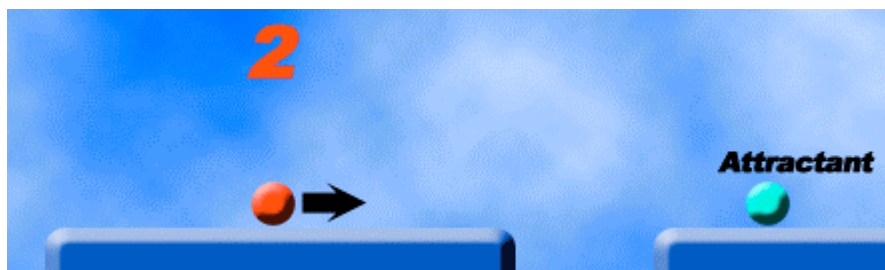
Cette façon de procéder nous a donc donné la possibilité de trouver un modèle à sept états. Chaque état a un ou plusieurs rôles bien définis que nous allons maintenant détailler.

1) Attente active :



La première action de nos agents est de trouver l'attractant. Grâce à des capteurs adéquats l'agent scanne son environnement et cherche son but. Le fait de le trouver le fait passer dans l'état de déplacement.

2) Déplacement :



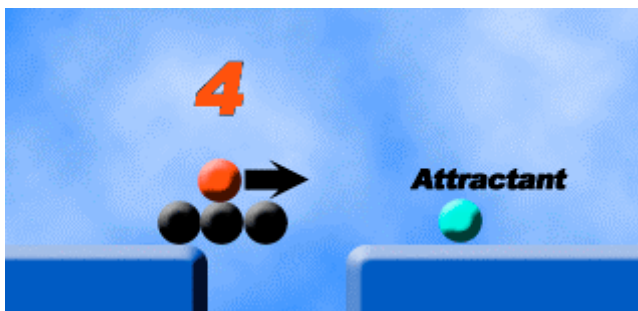
C'est dans cet état que l'agent se déplace vers son but. Suivant sa géométrie il pourra rouler, ramper, glisser, etc. Ce déplacement peut être individuel ou collectif.

3) Reconfiguration :



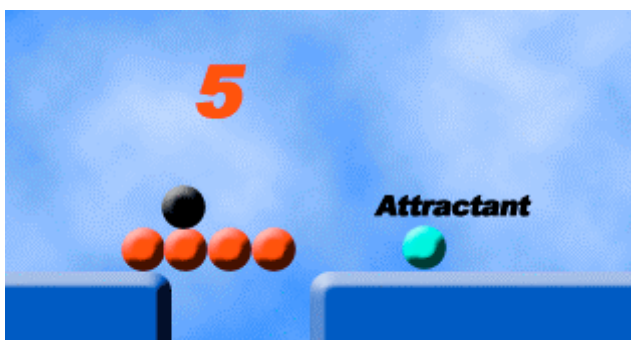
Quand survient un obstacle les agents doivent prendre une configuration adaptée à la situation (en accord avec leur morphologie et leurs comportements potentiels). Cette reconfiguration peut avoir divers degrés de complexité ou être complètement absente de la stratégie de construction pour un type d'agent donné. Dans le cadre général que nous souhaitons établir, cette phase est très importante car elle permet, pour les agents qui en sont capables, une anticipation vis à vis de la construction qui permettra un déploiement utilisant toutes les ressources disponibles.

4) Déplacement sur les autres (Parcours de la chaîne ou "grimpe") :



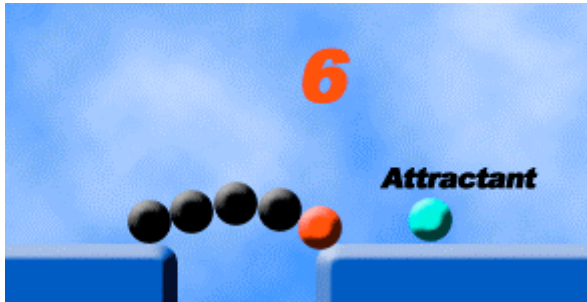
La construction d'un pont par auto-assemblage implique dans la grande majorité des cas le fait que les agents se déplacent les uns sur les autres. Un agent dans cet état parcourra le pont (ou le début de pont) déjà construit ou bien aidera à sa résorption. On peut remarquer que dans certaines circonstances les ébauches de ponts doivent être désassemblées avant d'avoir pu jouer leur rôle. Le parcours d'une chaîne ne doit donc pas être considéré comme étant à sens unique. De plus cet état contient également le fait de s'accrocher en bout de chaîne dans le but d'allonger le pont déjà formé.

5) Dans la chaîne :



Dans cet état les agents sont immobiles. Ils font partie de l'ébauche de pont (la chaîne) ou du pont déjà formé. Cela assure une relative stabilité à l'édifice. De plus, dans cet état, un agent n'est pas passif ; il détecte s'il touche de nouveau le sol et si, par conséquent, il a atteint l'autre bord.

6) Dans le pont :



Cet état ressemble au précédent à la différence que l'agent a conscience du fait qu'il forme un pont. C'est à dire qu'il touche le bord opposé ou qu'un autre agent lui a transmis l'information.

7) Attractant atteint :

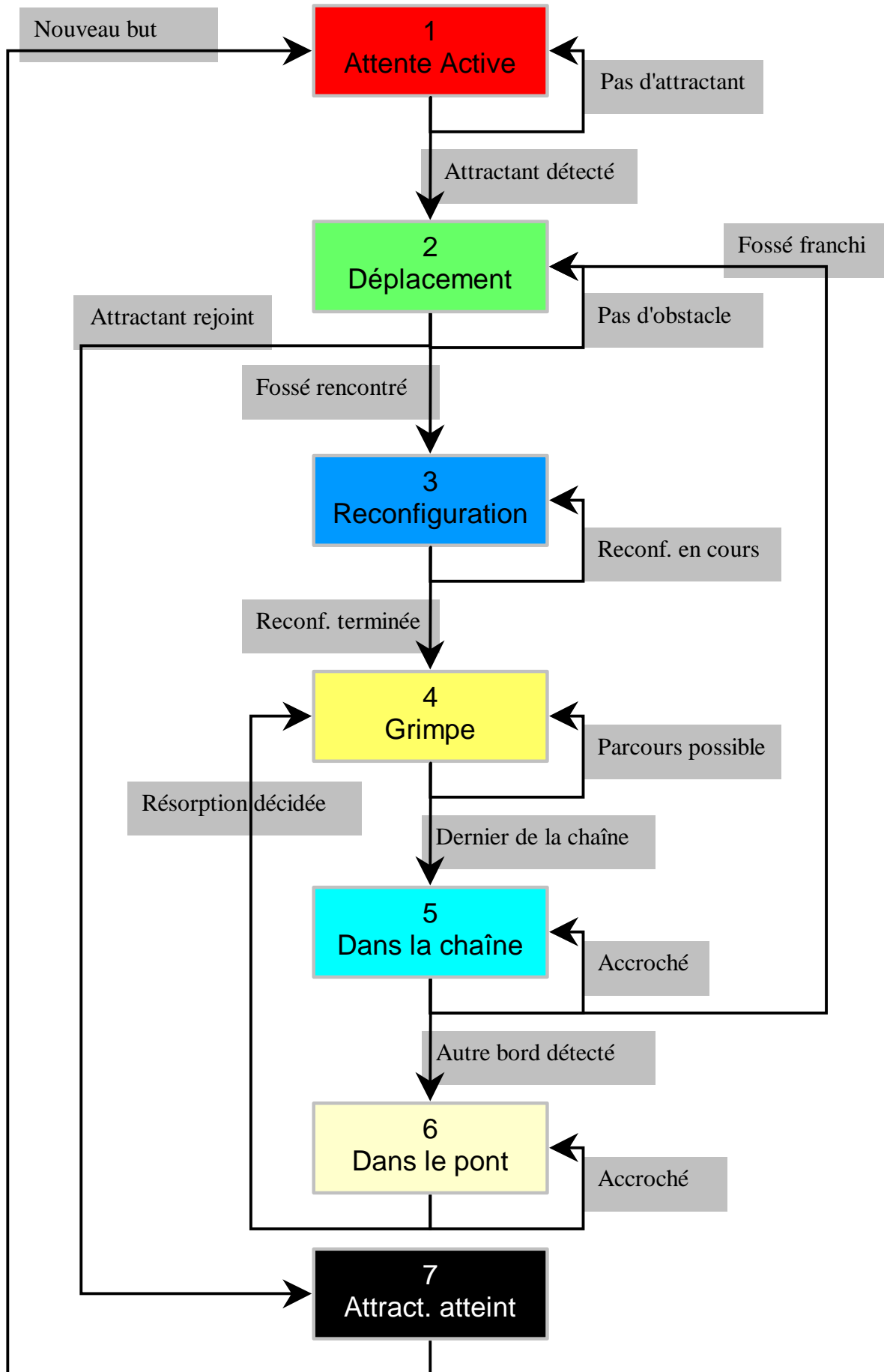


C'est le seul état final de l'automate. Quand l'agent touche l'attractant il s'arrête : son but est atteint.

L'automate :

Nous allons maintenant donner l'automate complet. Il s'agit des sept états vus précédemment reliés entre eux par les transitions. Celles-ci sont étiquetées par la condition nécessaire au passage.

Ajoutons que l'état initial est l'état 1: Attente active.



Transitions entre les états :

Les transitions représentent le passage d'un état à un autre. C'est à dire qu'à un instant donné l'agent a décidé qu'une, ou plusieurs, conditions étaient suffisantes pour changer de comportement et passer à une autre étape de la résolution du problème.

On remarque cependant que ce n'est pas toujours le cas car chaque état a une transition qui boucle sur lui même. En effet, si l'agent estime qu'il n'a pas accompli la sous tâche courante il continue son exécution.

Justification - Pourquoi un modèle sous la forme d'un automate :

On peut remarquer que la tâche est clairement fractionnable en plusieurs sous problèmes dont certains sont identiques. Par exemple, la phase de déplacement sur les autres agents et celle de résorption du pont utilisent toutes les deux un mécanisme de passage d'agent en agent. Il y a donc un cycle dans les comportements de l'agent.

De plus, rien n'empêche plusieurs fossés de se présenter à la suite avant d'atteindre l'attractant. Le processus de construction de pont doit donc pouvoir être répété. Un fois encore, un cycle apparaît dans le processus. Or, ces cycles sont facilement pris en compte avec un modèle en automate. Il suffit d'avoir une transition qui pointe vers un état déjà visité.

En outre, la simulation sur ordinateur ainsi que l'implémentation d'algorithmes sur de vrais robots passe par une pseudo discrétisation du temps. A chaque pas de temps l'environnement est évalué et des décisions sont prises. Etant donné que les états d'un automate peuvent boucler sur eux-mêmes on gère parfaitement ce phénomène.

III. Résultats

Nous allons voir trois illustrations de notre modèle par simulation. La première peut être comparée à la façon de faire des fourmis. Cependant nous devons rappeler que nous ne savons pas comment les fourmis font exactement. Nous ne pouvons constater qu'une ressemblance au niveau phénoménologique.

La seconde illustration utilise des agents capables de se reconfigurer mais qui peut ne pas aboutir si le fossé est trop grand.

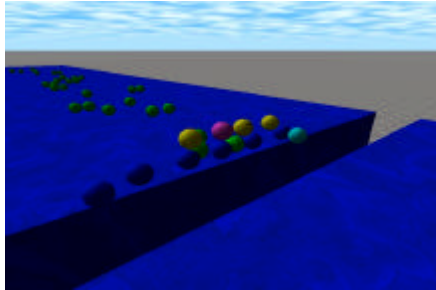
Enfin, le troisième exemple d'utilisation de notre modèle est ce qu'on peut espérer avoir avec les robots du projet MAAM. Il y a une gradation au niveau de l'efficacité dans ces trois utilisations.

Ces trois implémentations ont été réalisées grâce à un outil de simulation que nous avons conçu. Celui-ci est écrit dans le langage C et utilise une bibliothèque de fonctions gérant les interactions physiques (gravitation, collisions, joints entre objets) et la représentation en 3D des scènes : ODE (Open Dynamics Engine). Notre programme se compose de fonctions d'action et de détection qui représentent ce que les agents sont capables de faire et de percevoir et d'un codage de l'automate. Par exemple, pour un type d'agent donné, on peut trouver une fonction d'accrochage, une de communication avec un autre agent, une fonction de détection de fossé, une fonction de collision, etc...

N.B. : Pour toutes les images qui vont suivre le côté de départ est soit à gauche, soit en haut.

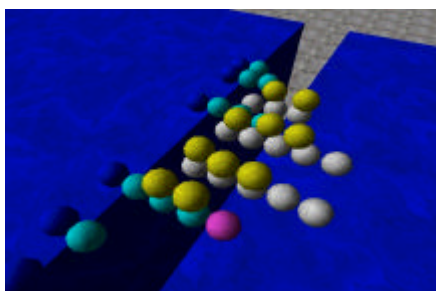
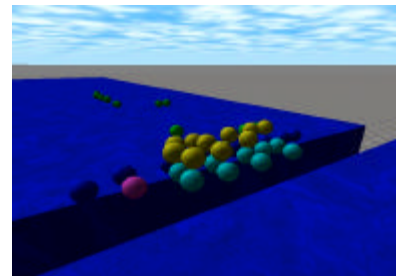
Etude de cas numéro un : pas de communication entre les agents :

Moyen de communication : aucun.
Conscience des autres : non.
Nombre de mécanisme d'accrochage : 1
Peut soulever 10 autres agents.
Peut s'accrocher au sol.
Détection des fossés.
Peut grimper sur les autres agents.



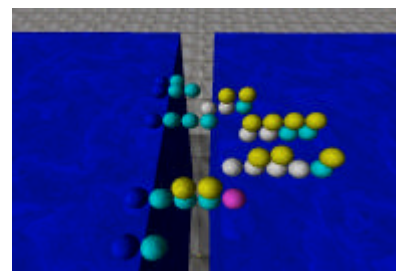
Le point le plus important de cette stratégie est que les agents n'ont pas conscience des autres. Ils démarrent et vont tous vers l'attractant. Ceux qui rencontrent le fossé s'arrêtent et se préparent à accrocher ceux qui passeront à leur portée.

Les autres leur grimpent dessus et continuent d'avancer. Ils se font alors attraper. La phase de reconfiguration est inexistante pour ce type d'agent. De cette façon, on constate l'apparition de plusieurs chaînes parallèles.



Les chaînes suffisamment longues pour atteindre l'autre bord forment des ponts valides. Quand un agent atteint l'autre côté il tente de continuer à avancer. Or, comme il est accroché la traction se propage dans la chaîne. Ainsi, celui que personne ne tient (c'est à dire le premier de la chaîne) peut emprunter le pont ; ce qui fait débiter la résorption.

Cette illustration du modèle utilise des agents capables de porter plusieurs autres agents et capables de s'accrocher au sol. De nombreuses chaînes n'atteignent pas l'autre bord. Il faut donc beaucoup d'agents pour former un pont. Les agents n'ont besoin que d'un seul point d'accrochage.



Cette méthode n'est pas la plus optimale mais elle présente une tolérance aux pannes extrêmement bonne car elle est très redondante. En moyenne, pour 40 agents disposés aléatoirement sur la surface de départ, 55% des agents passent de l'autre côté et 7 ou 8 chaînes parallèles sont créées. Pour être sûr d'avoir au moins un pont il faut un nombre d'agents supérieure à $7 \times$ (taille du fossé en nombre d'agents).

Etude de cas numéro deux : méthode alternative à l'escalade d'autres agents :

Moyen de communication : par contact.

Conscience des autres : oui.

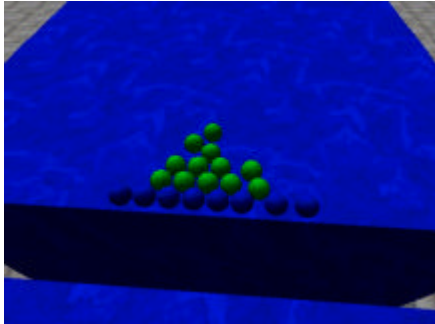
Nombre de mécanismes d'accrochage : 1

Peut soulever 1 autre agent.

Ne peut pas s'accrocher au sol.

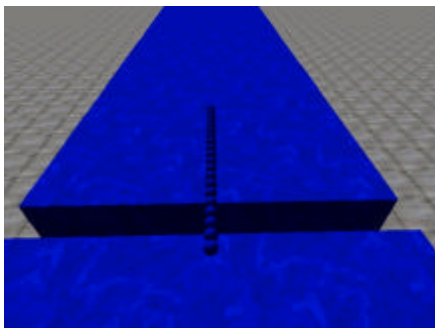
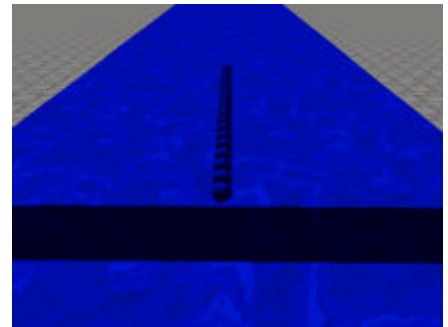
Détecte les fossés.

Ne peut pas grimper sur les autres agents.



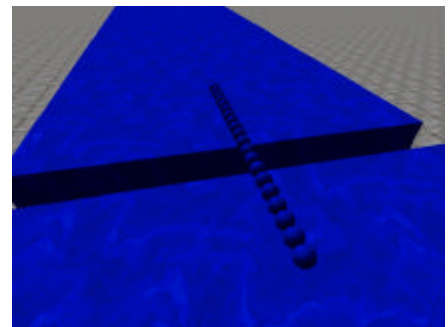
Les agents détectent l'attractant et s'y dirigent. Lorsqu'ils arrivent devant le fossé ils entament la phase de reconfiguration.

Cette reconfiguration consiste à "élire" un premier et à ce que chacun se place derrière lui. On peut constater qu'une communication entre les agents est nécessaire. On obtient de cette façon une ligne d'agent.



Dans cette illustration les agents n'utilisent pas la phase d' « escalade ». Après s'être reconfigurés ils reviennent à l'état de déplacement et avancent vers l'attractant en ligne.

Si le nombre d'agents de la ligne est suffisant alors ils passent le fossé sans encombre.



Si le nombre d'agents est au moins supérieur à $(2 \times T) + 1$ (où T est la taille du fossé en nombre d'agents pour le combler) alors il y a 100 % des agents qui passent de l'autre côté. Dans le cas contraire il y a 100 % de pertes.

(On peut signaler que cette façon de faire est réalisable avec des robots plus simples que ceux utilisés dans beaucoup de projets de robotique reconfigurable. L'équipe de Marco Dorigo du laboratoire IRIDIA travaille par exemple sur un franchissement de fossé qui utilise cette stratégie avec des robots munis de roues : c'est le projet SwarmBot.) [cf. 11]

Etude de cas numéro trois : communication par contact et connaissance du groupe :

Moyen de communication : par contact.

Conscience des autres : oui, chaque agent connaît l'effectif total et peut compter ceux qui passent sur lui.

Nombre de mécanisme d'accrochage : 6

Peut soulever 1 autre agent.

Ne peut pas s'accrocher au sol.

Détecte les fossés.

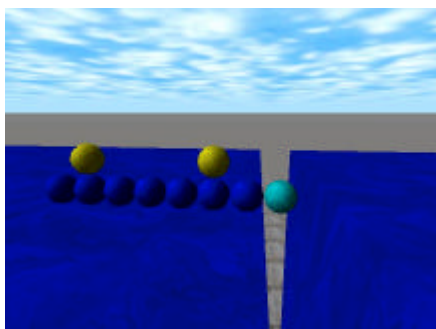
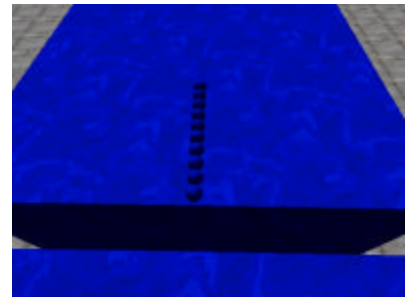
Peut grimper sur les autres agents.

Cette instance est la plus aboutie. En effet, les agents utilisent le meilleur des deux précédentes stratégies de formation de pont. Les agents perçoivent les autres et sont même capables de les compter pour s'assurer que le pont peut être créé sans qu'un déséquilibre n'apparaisse.



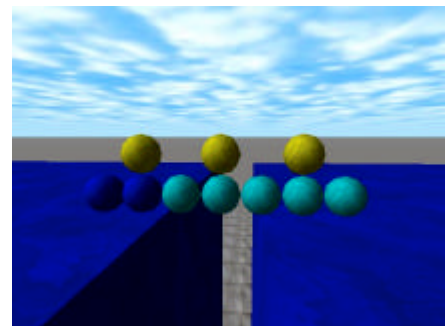
Les agents trouvent l'attractant, se déplacent vers lui et rencontrent le fossé. Ils entament alors la phase de reconfiguration qui consiste à former une ligne

Pour cela, ils utilisent la méthode de l'exemple précédent : ils élisent l'agent qui semble le plus près et vont se placer chacun leur tour dans son alignement là où il y a une place.



Le dernier de la chaîne commence alors la phase d'escalade sur les autres (en jaune sur les images). L'agent juste au bord du fossé sait à tout instant combien d'agents sont devant lui et combien sont derrière.

Ainsi il peut s'assurer qu'il n'y a pas de déséquilibre possible.



Pour cette stratégie dans 100% des cas les agents passent de l'autre coté ou bien réalisent que le franchissement est impossible. Pour un nombre d'agents supérieur à $(2 \times T) + 1$ (où T est la taille du fossé en nombre d'agents pour le combler) il y a 100 % des agents qui passent de l'autre coté.

Critiques et limites de notre modèle :

Il est temps maintenant de relativiser nos résultats et d'en faire la critique. Il est illusoire de prétendre avoir trouvé *le* modèle générique pour *tous* les types d'agents. Certes, notre travail s'efforce d'englober le plus de morphologies et le plus de stratégies de construction possibles.

La première critique est que notre modèle n'est qu'un modèle. Il ne permet pas de savoir exactement comment doter un certain type de robots reconfigurable de la capacité de faire des ponts. Il donne simplement de grandes lignes de conception.

La seconde critique que nous pouvons formuler est que pour que ce modèle fonctionne deux conditions sont nécessaires :

- Les agents sont capables de détecter le fossé de façon efficace.
- Les agents peuvent sentir l'attractant et connaître sa direction.

La première condition peut paraître évidente à énoncer mais elle est cependant non triviale pour ce qui est de la mise en pratique sur des robots réels. (Cela sort du cadre de notre travail mais peut constituer un bon sujet de recherche.)

La seconde condition est quant à elle plutôt paradoxale. Les agents ont une vision du monde très réduite. Ils ne disposent pas de carte ou d'un moyen de positionnement global. Nous partons pourtant du fait qu'ils sont capables de distinguer parfaitement la direction d'un objet et que celui ci les attire. Certes, on peut nous reprocher ces deux points ; mais sans cela nous affirmons qu'aucune construction de pont n'est possible.

Conclusion :

Dans ce travail nous avons donné un modèle qui, s'il est implémenté dans des robots reconfigurables, leur permettra de former des ponts. Nous avons montré que ce modèle fonctionne en décrivant trois simulations. Cette aptitude à construire des ponts autorisera les robots à évoluer en terrain très accidenté. C'est un pas supplémentaire pour faciliter l'apparition des robots dans des domaines trop complexes pour eux jusqu'ici tels que par exemple l'extraction minière ou l'exploration spatiale. Couplé à une miniaturisation, on obtiendrait des objets ayant la possibilité de changer de forme suivant les conditions, des coques de bateau qui boucheraient seules leur fissures, des bracelets qui si la température est trop basse se "transformeraient" en gants. Seule l'imagination limite les applications potentielles.

Ouverture du sujet :

En partant de notre modèle et des implémentations que nous en avons fait on pourrait très bien imaginer un réglage automatique des paramètres et des stratégies qui régissent la construction des ponts par algorithmes génétiques. En effet, la fonction d'évaluation est toute trouvée : il suffit de compter combien d'agents franchissent le pont indemnes. On pourrait raffiner cette fonction en mettant des points positifs pour les agents qui franchissent le fossé et des points négatifs pour ceux qui sont bloqués. Ce qui risque de poser un peu plus de problème c'est la réduction des paramètres du modèle à des nombres réels.

L'aspect biologique et éthologique est un autre point sur lequel nous aimerions orienter nos recherches futures (si l'occasion nous en est donnée). Comme nous l'avons déjà dit, il n'existe pas d'articles sur la formation des ponts par les fourmis. L'une des applications de notre travail pourrait être de donner aux biologistes un point de départ pour l'étude de ces auto-assemblages. On a déjà vu des collaborations entre informaticiens et biologistes très fructueuses. Notre intention est de faire de ces simulations des méthodes heuristiques qui permettent aux biologistes d'avancer dans la compréhension d'un phénomène qu'il est difficile d'étudier sur le terrain.

Enfin, le dernier point et, à notre sens le plus passionnant, est celui de l'apprentissage collectif. Si franchir un fossé prend un temps t alors notre modèle, pour franchir 10 fossés, mettra $10t$. Or, dans la nature, et notamment chez les fourmis, on constate que plus une tâche collective est effectuée, plus elle aura de chances d'être effectuée plus rapidement ou plus efficacement les fois suivantes. Il y a très souvent un apprentissage et obligatoirement une forme de mémorisation de ces comportements collectifs. En particulier, comme dans d'autres phénomènes auto-organisés, il est vraisemblable que les individus n'occupent pas toujours le même rôle dans le temps. L'apprentissage collectif se réalise certainement à ce niveau, utilisant la différenciation des rôles comme inscription des « connaissances » apprises au cours des expériences antérieures. Comme dans les phénomènes de morphogenèse, l'individu engrange ainsi des connaissances qui n'ont de sens que dans ses interactions avec les autres. Ce phénomène d'apprentissage collectif est un sujet qu'un stage de DEA ne peut épuiser, et qu'il nous plairait de pouvoir continuer à étudier par la suite, notamment dans le cadre du projet MAAM.

Bibliographie:

- [1] Anderson C. and D.W. McShea, 2001. "**Intermediate-level parts in insect societies: aptive structures that ants build away from the nest**" In *Insectes soc.* 48 (2001) 291–301
- [2] Anderson C., G. Theraulaz and J.L. Deneubourg, 2002. "**Self Assemblage in insect societies**" In *Insectes soc.* 49 (2002) 99-110.
- [3] Bojinov H., A. Casal, and T. Hogg. "**Multiagent control of self-reconfigurable robots**". In *Proceedings of the Fourth International Conference on MultiAgent Systems*, pages 143-150, Boston, Massachusetts, USA, 2000
- [4] Bojinov H., Casal A. and Hogg T. 2000. "**Emergent Structures in Modular Self-reconfigurable Robots**". In *Proc. IEEE International Conference on Robotics and automation*, San Francisco, CA. pp. 1734-1741
- [5] Bonabeau E., G. Theraulaz (Eds.), "**Intelligence Collective**", Hermes, Paris, 1994.
- [6] Butler Z. and D. Rus, "**Distributed Locomotion Algorithms for Self-Reconfigurable Robots Operating on Rough Terrain**", *Int'l Conf. on Computational Intelligence in Robotics and Automation (CIRA) '03*, July 2003.
- [7] Ferber J., **Les Systèmes Multi-Agents: Vers une Intelligence Collective**, Intereditions, France, 1995.

- [8] Holldobler, B. & E.O. Wilson. 1990. **The Ants**. Springer-Verlag, Berlin.
- [9] Lioni, A., 2000. “**Auto-assemblage et transport collectif chez *Oecophylla***”. Unpublished Ph.D. Thesis, Université Libre de Bruxelles, Belgium. 1395–1405.
- [10] **Projet MAAM** : www.univ-ubs.fr/valoria/Dominique.Duhaut/maam/france.html
- [11] Projet SwarmBot (description) : www.swarm-bots.org/index.php?main=3&sub=31
- [12] Støy, K. W.-M. Shen, and P. Will, "**How to Make a Self-Reconfigurable Robot Run**", In proceedings of the 1st international joint conference on autonomous agents and multiagent systems (AAMAS'02) (to appear), Bologna, Italy, July 15-19, 2002.
- [13] Suh J., Homans S. and Yin M., 2002. Telecubes: "**Mechanical design of a module for self-reconfigurable robotics**" In IEEE 2002, International Conference on Robotics & Automation, pp. 4095-4101 Mai 2002.
- [14] Theraulaz G., Lioni A., Libert F., Bonabeau E. and Deneubourg J.L., 1996. "**Dynamique de formation des grappes et des chaînes chez la fourmi d'Argentine *Linepithema Humile***" In Actes Coll. Insectes Sociaux, 10: 175-178
- [15] Yim M., Ying Z., Lampling J. and Mao E., 2001. "**Distributed control of 3D metamorphosis**" In Journal of Autonomous Robots 10, Kluwer, 2001, p. 41-56.

Annexe A – Logiciels.

I) BREVE.

Mon stage a commencé par des recherches bibliographiques. Puis, pour tester les premières idées que j'avais de la formation des ponts "émergents", j'ai fait mes premières simulations. Pour cela j'ai utilisé un environnement de simulation dédié à la vie artificielle se nommant Breve.

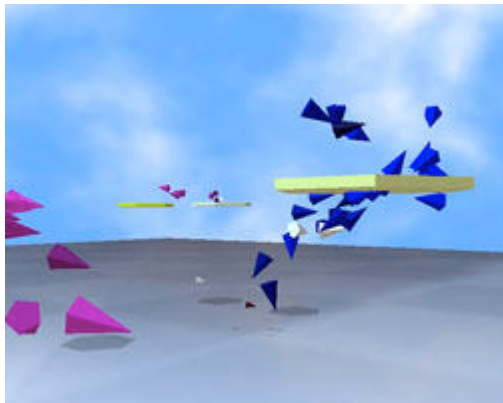


Fig 1 : Simulation en Breve des Boids.

Bien que ce soit un très bon outil de développement pour la vie artificielle et les simulations de systèmes décentralisés la version que j'ai utilisée ne me permettait pas de simuler la formation des ponts par auto-assemblages. En effet, il m'était impossible de créer des joints entre les agents puis de les détruire sans que le programme se termine de façon abrupte.

C'est pourquoi, après une correspondance électronique avec l'auteur (au demeurant fort sympathique et très ouvert – mais très pris par ses propres activités) j'ai décidé de développer ma propre plateforme de simulation.

Bibliographie :

- ? Site Internet : <http://www.spiderland.org/breve/>
- ? Présentation de Breve : Klein, J. 2002. **breve: a 3D simulation environment for the simulation of decentralized systems and artificial life.** *Proceedings of Artificial Life VIII, the 8th International Conference on the Simulation and Synthesis of Living Systems.* The MIT Press.

Breve se compose d'un langage objet (STEVE) et d'un interpréteur pour ce langage. Il peut être comparé aux plateformes de développement qu'on trouve en multi-agents comme StarLogo ou Swarm à la différence que Breve est en 3D et gère les interactions physiques.

Breve est construit sur la bibliothèque d'interactions physiques ODE et sur la bibliothèque de rendu 3D OpenGL.

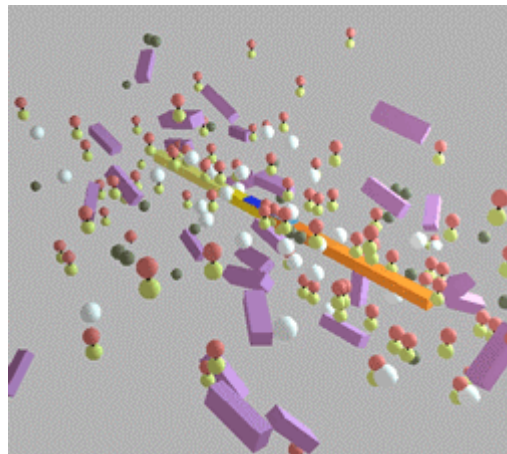
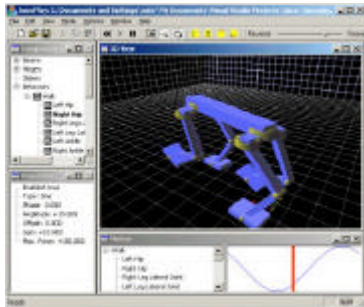


Fig 2 : Intéactions des protéines et de l'ADN en Breve.

II) Open Dynamics Engine (ODE)

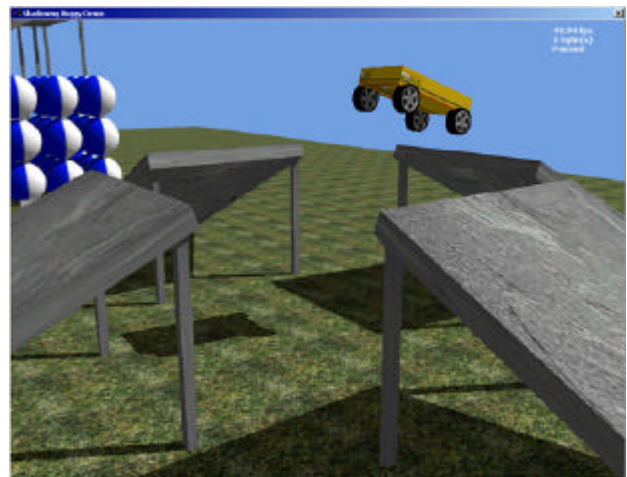
Mon but a toujours été de produire un travail réutilisable par les acteurs du projet de robotique reconfigurable MAAM. Dans cette optique, l'outil de simulation que j'allais construire devait être proche de la "réalité" et donc être en trois dimensions avec une gestion des interactions physiques assez évoluée. Comme je l'ai dit avant, j'ai commencé par utiliser Breve qui ne m'a pas apporté ce que je souhaitais. Or cet environnement est lui même construit sur la bibliothèque ODE. Je me suis par conséquent tourné vers celle-ci.



A la base, ODE a été écrit par Russel Smith, puis une grande communauté s'est développée autour du projet. Beaucoup de projets utilisent les fonctions fournies par ODE pour leurs simulations.

Comme ODE est une bibliothèque on peut s'en servir en C ou en C++. Les types de base (entier, flottants, etc.) sont redéfinis pour prendre en compte les nombres infinis (pour les coefficients de frottement par exemple).

De plus, il existe également une bibliothèque très simple d'utilisation qui permet assez rapidement de générer graphiquement les simulations. C'est la bibliothèque DrawStuff fournies en standard avec ODE.



ODE est donc un excellent outil pour aider au développement d'une plateforme de simulation. Cependant pour y parvenir il faut fournir plus d'effort de programmation qu'avec des outils comme Breve.

Bibliographie :

? Site Internet d'ODE : <http://opende.sourceforge.net/>

Annexe B – Personnes "rencontrées" pendant le stage.

I) Dominique Fresneau

Au début de mon stage, je pensais que je pouvais partir des observations que les éthologues ont fait sur les ponts "construits" par les fourmis. Dans ce but j'ai rencontré Dominique Fresneau au Laboratoire d'Ethologie Expérimentale et Comparée (LEEC) de l'université de Paris 13.

D. Fresneau travail en ce moment sur l'écologie comportementale des Fourmis néotropicales et l'analyse des règles de décisions individuelles à l'origine de l'organisation collective.

Il m'a beaucoup aidé dans mes recherches bibliographiques sur les fourmis – domaine où j'étais complètement novice. C'est grâce à lui si j'ai pu me procurer un exemplaire de la thèse d'A. Lioni sur les auto-assemblages de la fourmi *oecophylla*.

C'est également grâce à lui (entre autres) si j'ai pu, dès le début, avoir un oeil plus critique sur mon travail de stage.

Site Internet du LEEC : <http://www-leec.univ-paris13.fr/>

II) Marco Dorigo

Marco Dorigo est directeur de recherche à l'IRIDIA, un laboratoire de recherche en Intelligence Artificielle à l'université libre de Bruxelles. Il travaille sur les *Ant Algorithms*, des algorithmes tirés de l'activité des fourmis et est notamment l'un des inventeurs de l'algorithme d'optimisation par colonie de fourmis (Ant Colony Optimization).

J'ai été amené à le rencontrer lors d'une présentation de mon travail de stage devant son équipe à l'IRIDIA.

Ses critiques très constructives m'ont aidés dans mon travail de stage. Ma présentation devant son équipe était en anglais et loin d'être aussi aboutie que celle qu'un bon chercheur est capable de faire. Au delà des informations que j'ai pu obtenir pour mon stage cette rencontre m'a permis de prendre plus confiance en moi et de constater que mon niveau en anglais était meilleur que ce que je croyais.

Site Internet de Marco Dorigo : <http://iridia.ulb.ac.be/~mdorigo/>