

Modélisation multi-agents de systèmes naturels

Réflexions générales et application en biologie marine

E. RAMAT⁽¹⁾, P. PREUX⁽¹⁾, L. SEURONT⁽²⁾ et Y. LAGADEUC⁽²⁾

(1) Laboratoire d'Informatique du Littoral (LIL)
B.P. 719
62228 Calais Cedex
France

(2) Station marine de Wimereux - URA CNRS 1363
28 avenue Foch
BP 80
62930 Wimereux
France

e-mail : {ramat,preux}@lil.univ-littoral.fr

e-mail : {seuront,lagadeuc}@loalit.univ-littoral.fr

Résumé

Dans le cadre d'un travail de recherche pluridisciplinaire, nous proposons une modélisation multi-agents d'un écosystème. Nous discutons d'aspects méthodologiques pour passer d'un modèle à base de bilans à une modélisation multi-agents. Nous justifions l'utilisation d'une modélisation multi-agents pour l'étude du comportement d'un organisme vivant ce qu'un modèle analytique ne peut rendre compte d'une manière pertinente. Nous présentons également une plate-forme que nous avons développés dans un souci de généricité afin de la réutiliser dans d'autres projets en cours.

Mots-clés

SMA, modélisation d'écosystèmes, biologie marine, laboratoire virtuel

1 Introduction les SMA dans les sciences de la nature

L'objet de nos travaux est la modélisation et la simulation de systèmes naturels à l'aide de systèmes multi-agents [4]. Ce travail est pluridisciplinaire : l'informaticien apporte sa connaissance de l'informatique, développe des travaux nouveaux pour répondre à des questions que se posent des chercheurs d'une autre discipline, une discipline des sciences naturelles dans notre cas. Notre travail repose sur un dialogue constant entre les disciplines.

Le choix des SMA pour l'étude de ce type de systèmes n'est pas un choix académique : c'est un choix de raison. En effet, notre objectif n'est pas, comme cela est généralement le but dans l'utilisation de l'informatique, de calculer une fonction ; l'objectif est d'étudier la dynamique d'un système. La modélisation consiste à transcrire un système dans un certain langage pour pouvoir l'étudier plus aisément (des équations différentielles, par exemple). Cependant, nous pensons qu'il est beaucoup plus judicieux, parce qu'il se prête mieux à l'expression d'objets complexes, d'utiliser des agents dont le comportement est décrit algorithmiquement. Aussi, la modélisation multi-agents répond t-elle à nos attentes. Notons bien que nous ne cherchons pas à " rendre tout agent ", même s'il est intéressant de voir jusqu'où les SMA peuvent se substituer aux équations [7,10]. Si, dans le système que nous étudions, certaines parties de la dynamique sont parfaitement décrites par des équations que l'on sait résoudre, nous les utilisons.

D'un point de vue plus global, notre objectif est la réalisation de laboratoires virtuels [6]. Nous entendons par là un logiciel dans lequel un système est décrit et la dynamique simulée. Les paramètres définissant le système sont modifiés dynamiquement et des capteurs placés à tout endroit

du système permettent la collecte de mesures. Le but est de pouvoir tester des hypothèses en faisant jouer des scénarii et d'observer et mesurer l'évolution du système. Cela permet également la visualisation des phénomènes émergents que l'on ne peut pas imaginer/prévoir *a priori*. Cette visualisation aide à la compréhension du système et de sa dynamique.

1.1 Démarche pour la modélisation

D'une manière générale, les modèles rencontrés dans les sciences du type biologie, géologie, ... décrivent des flux de manière analytique. En modélisation MA, il faut décrire le comportement individuel des agents et leurs interactions. C'est un changement complet de perspective. Dans une modélisation multi-agents, les bilans sont émergents : ce sont des résultats, et non plus des données. Généralement, le chercheur a un modèle de type bilans et une intuition concernant le comportement à un niveau plus fin et seul l'aspect bilan est utilisé effectivement, du fait qu'il soit analytique. En renversant complètement le point de vue à l'aide d'une modélisation MA en mettant en avant le comportement individuel, nous mettons en question la cohérence des deux niveaux de "compréhension" du système : normalement, la simulation des comportements individuels doit naturellement déboucher sur l'observation des bilans censés décrire la dynamique du système (voire *infra*).

1.2 Validation

La validation du modèle est basée sur une comparaison entre les observations et la dynamique de la simulation. Après modélisation du comportement des agents et de leurs interactions, on paramètre le système et on simule des situations expérimentales connues. En fonction du décalage entre l'observation et la simulation, on cale le modèle.

Une fois calé, on extrapole en faisant jouer des scénarii nouveaux en supposant que le comportement du simulateur demeure fidèle à la réalité.

1.3 Impact des SMA dans un domaine particulier

Du point de vue de la mise en place d'un modèle multi-agents, la modélisation est intuitivement assez facile à réaliser en comparaison avec une modélisation analytique ; elle ne demande pas une aptitude pour les mathématiques et leurs applications ; elle évite les problèmes liés à la résolution effective d'un système d'équations différentielles.

Comme on l'a déjà souligné plus haut, elle permet de simuler les lois comportementales observées et d'en vérifier leur cohérence par rapport aux modèles de bilans classiques. On renverse donc la perspective entre les données et les résultats.

L'approche permet d'intégrer des modèles du comportement de plusieurs types d'agents dans une même simulation et de tester la cohérence de ces modèles entre eux. Cette possibilité est très originale et permet de se rendre compte des incohérences entre des modèles décrivant des processus différents et interagissants.

L'aspect visualisation de la dynamique du système est également fondamentale à nos yeux. Par essence, une émergence ne peut être prédite ; il est donc important de disposer d'un outil permettant de la voir, donc de mieux comprendre le système étudié.

1.4

Principes généraux de notre modélisation

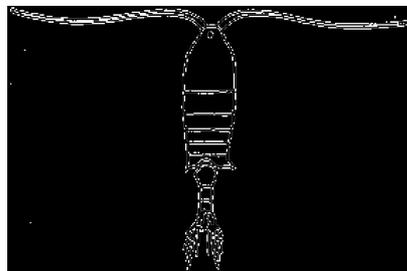
Dans le reste de cet article, nous décrivons la modélisation que nous avons mise en place dans le cadre d'une collaboration informatique/biologie marine. Nous utilisons une modélisation multi-agents pour tester une hypothèse proposée par les biologistes sur la modélisation du milieu marin et sa pertinence pour rendre compte du comportement du zoo-plancton. Les agents représentant les organismes zoo-planctoniques sont des agents situés, réactifs, et percevant leur environnement. Ainsi, le modèle de leur environnement et de leur voisinage n'est pas fourni aux agents ; au contraire, les agents le construisent à chaque instant via leurs "sens". C'est pour nous très important de modéliser la perception des agents sous cette forme "réaliste". La perception de l'agent peut varier en fonction de son état physiologique et de son environnement ; sa situation (spatiale, dans l'espace des états physiologiques...) intervient dans la perception qu'il a de son milieu. Notons également que le fait que les agents soient réactifs n'impliquent pas qu'ils n'ont pas une certaine forme de mémoire. Le comportement passé d'un agent rétro-agît sur son comportement à venir ; le comportement d'un agent à un instant est donc influencé par son histoire, qui lui est, par nature, individuelle.

Après avoir décrit le problème et le système biologique, nous présentons la modélisation qui a été réalisée. Nous montrons les résultats obtenus qui viennent confirmer les hypothèses des biologistes. Ce travail remet en cause une bonne partie des modélisations réalisées à ce jour dans ce domaine, ce qu'une approche de type analytique n'a pas permis de réaliser. Enfin, nous discutons des suites et perspectives de ce travail.

2 Modélisation multi-agents en biologie marine

L'objectif de ce travail est d'identifier les règles comportementales du copépode (cf.

Figure 1) et les influences comportement : l'hypothèse est comportement actif de (phytoplanctons). Pour cela, l'observation sur le vivant, les et la modélisation par agents. est fortement hétérogène dans montrent que cette



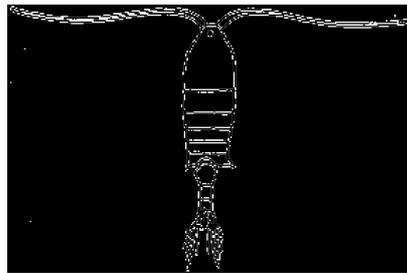
de l'environnement sur son que le copépode a un recherche de nourriture nous nous proposons de croiser résultats des modèles analytiques La distribution du phyto-plancton le milieu. Les résultats actuels [1] hétérogénéité influe sur le bilan

énergétique du copépode. En mesurant la quantité d'azote absorbée lors de la nutrition, les observations montrent que le rendement du comportement du copépode (énergie dépensée/énergie ingérée) varie en fonction du type de distribution de la nourriture [1,3]. Par exemple, un milieu perturbé, propice aux échanges, favorise le taux de rencontre du copépode avec les particules de phyto-plancton et donc augmente son rendement. Il reste maintenant à montrer l'influence du comportement du copépode dans de telles situations. L'étude de ce comportement est incompatible avec une approche analytique, comme nous allons le voir. Aussi, nous étudions une approche SMA [4].

2.1 Le système étudié

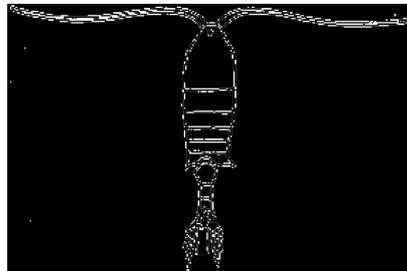
A l'heure actuelle, le copépode (cf.

Figure 1), petit organisme zoo-planctons, est représenté par des modèles de type “boîte noire” (cf. Figure 2) ou modèles cherchant à décrire dans la vie de l’organisme en flux de sortie et une fonction



marin appartenant à la famille des par des modèles de type “boîte analytiques (cf. Figure 3). Ces chaque processus intervenant identifiant des flux d’entrée, des de transfert.

Figure 1. Copépode



Centropages hamatus

Attardons nous sur le dans le cas du phytoplancton capture une proie (une particule temps de manipulation, celle-ci entre dans le processus de son contenu soit en énergie utilisable (proies assimilées) que l’on exprime en azote, ou en déchets (pelotes fécales). Cette transformation est continue : à chaque Δt , une quantité Δq transite (cette quantité est proportionnelle à la quantité stockée dans l’estomac). L’énergie utilisable est soit consommée (métabolisme, digestion ou nage) soit stockée (pour la production d’œufs chez les femelles, par exemple). Quant aux déchets, ils sont éjectés.

processus d’ingestion des proies (cf. Figure 2). Le copépode de phytoplancton). Après un est stockée dans l’estomac et digestion. L’estomac transforme

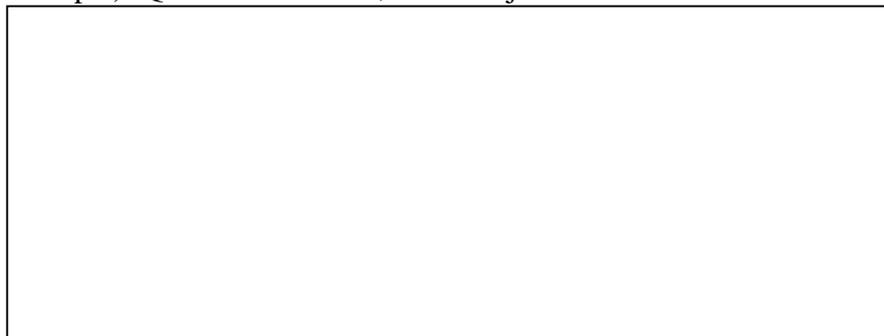


Figure 2. Modèle en boîte du processus d’ingestion

Il existe des modèles analytiques qui ont pour ambition de représenter, au mieux, les processus biologiques qui régissent les copépodes. [1] propose un modèle (cf. Figure 3) synthétisant les différents modèles développés jusqu’à présent. Il résume, à l’aide de cinq équations différentielles interdépendantes, l’activité de capture et de digestion.

$$\frac{dX_1}{dt} = I - A - F \quad (1)$$

$$\frac{dX_2}{dt} = A - \frac{C}{M_N} \quad (2)$$

$$\frac{dX_3}{dt} = F - G \quad (3)$$

$$\frac{dX_4}{dt} = G \quad (4)$$

$$\frac{dX_5}{dt} = C_{st} + C_{sda} + C_{sw} \quad (5)$$

Figure 3. Modèle analytique du processus d’ingestion

X_1	Nombre de proies dans l'estomac	
X_2	Nombre de proies assimilées	
X_3	Nombre de pelotes fécales	
X_4	Nombre de pelotes fécales évacuées	
X_5	Energie dépensée exprimée en azote	
I	Taux d'ingestion	
A	Taux d'assimilation	$A = C_a X_1 / T_g$
F	Taux de formation de pelotes fécales	$F = (1 - C_a) X_1 / T_g$
C	Energie dépensée par unité de temps	$C = C_{st} + C_{sda} + C_{sw}$
C_a	Pourcentage de proies assimilées par rapport aux proies capturées	
T_g	Temps de transit (dépendant du nombre de proies dans l'estomac)	
M_N	Masse en azote du copéode	
G	Nombre de pelotes fécales évacuées par unité de temps	
C_{st}	Energie dépensée pour le métabolisme par unité de temps	
C_{sda}	Energie dépensée pour l'ingestion par unité de temps	
C_{sw}	Energie dépensée pour le déplacement par unité de temps (nage + saut)	

Table 1. Définition des variables et constantes du modèle analytique de la Figure 3

Prenons, à titre d'illustration, l'équation (1) qui décrit mathématiquement la variation du nombre de proies dans l'estomac de l'animal (X_1). Cette variable dépend de trois autres variables :

- * le taux d'ingestion I qui mesure le nombre de proies rencontrées par le copéode en fonction de sa satiété¹ et du niveau de turbulence² du milieu,
- * le taux d'assimilation A, proportion d'azote apportée par les proies qui devient utilisable,
- * le taux de formation des pelotes fécales, proportion d'azote apportée par les proies qui n'est pas utilisée mais évacuée en pelotes fécales.

Les deux dernières variables modélisent, *a priori* de manière correcte, la digestion. En revanche, la variable I est censée décrire le processus de capture du copéode en fonction de sa satiété et de l'environnement. A aucun moment, les équations mathématiques ne font mention du comportement du copéode. Pour s'en convaincre, développons un peu ce qui se cache sous I.

$$I = (\beta_{\text{behaviour}} + \beta_{\text{turbulence}}) N_p F_A$$

I est directement liée à la capacité de rencontre du copéode. Dans le modèle analytique proposé, cette capacité de rencontre est fonction de quatre paramètres :

- * $\beta_{\text{behaviour}}$ la contribution du "comportement", celle-ci est fonction du rayon de perception du copéode, du rayon des proies et la différence de vitesse de nage entre le copéode et les proies,
- * $\beta_{\text{turbulence}}$ la contribution de la turbulence,
- * N_p la densité des proies dans le milieu,

¹ Le niveau de satiété est proportionnelle à la quantité de nourriture contenue dans l'estomac. Plus l'estomac du copéode est plein, moins il cherche à capturer de nourriture.

² Plus le milieu est turbulent, plus la probabilité de rencontre entre le copéode et les cellules de phytoplancton est grande.

- * F_A une mesure de l'activité de nutrition dépendant de la quantité de nourriture dans l'estomac du copépode et, selon le mode de prédation (en suspension ou en embuscade), la densité des proies dans le milieu.

Ce modèle rend bien compte du processus de capture mis à part, de notre point de vue, la contribution du comportement. En effet, si l'on peut mettre en équation le fait que l'activité de nutrition du copépode est fonction de la densité des proies, du niveau de turbulence du milieu, du mode de prédation et de la quantité de nourriture dans l'estomac, il est, en revanche, délicat de résumer le comportement par une relation qui fait seulement intervenir le rayon de perception, la taille des proies et la vitesse de nage du copépode relativement à celle de ses proies.

2.2 La modélisation multi-agents

Le système étudié est composé d'une masse d'eau (qui représente le milieu des autres agents) dans laquelle sont immergés des "patches"³ de phytoplanctons et des copépodes. Chacun des agents est situé et possède ses propriétés parmi lesquelles son comportement. Les patches de phytoplanctons ont une certaine dimension et sont soumis aux courants et à la turbulence. Le copépode possède différentes caractéristiques (sa masse en azote, le volume de son estomac, sa vitesse de nage...). Son comportement, objet de l'étude pour le biologiste, est défini par un réseau de Pétri pour permettre la description de comportements sophistiqués avec un outil standard et simple. On se place dans une perspective d'agent réactif (cf. Figure 4) : l'état interne de l'agent et les stimuli auxquels il est sujet à un certain instant entraînent (de manière non nécessairement déterministe) un certain comportement et un nouvel état interne.

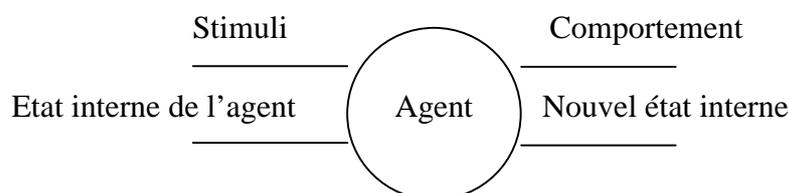


Figure 4. Agent réactif

2.2.1 L'outil de modélisation

Il existe une multitude d'outils de modélisation plus ou moins généralistes (Swarm [13], Manta [12]...). Dans la majorité des cas, ce sont des plates-formes soit adaptées à un ou à une famille de problèmes, soit génériques. A notre connaissance, la généricité entraîne, dans tous les cas, une dépendance vis à vis d'un environnement informatique ou d'un langage de développement mais surtout ces plates-formes sont à base de primitives réutilisables et il faut donc écrire du code ! Toutes ces raisons nous ont poussés à concevoir un nouvel outil de modélisation et de simulation d'agents réactifs, perceptifs et situés.

Nous utilisons un langage objet (Java) dans lequel nous retrouvons tous les concepts du paradigme objet ainsi qu'une arborescence de classes avancées (*thread*, *stream*...). Cette hiérarchisation des classes nous a permis de définir nos classes d'agents (cf. Figure 5). Dans un premier temps, ces classes d'agents ont été accessibles par un langage avec lequel on peut décrire une catégorie d'agents par l'intermédiaire de leur comportement, de leurs propriétés et de leurs moyens de communication. Un second langage rend possible la description de l'environnement. L'ensemble compose une plate-forme où l'on peut décrire les agents, les créer et les activer.

³ Agrégat

A l'heure actuelle, les deux langages sont encapsulés dans une interface graphique écrite en Java (donc portable et accessible sur Internet) et accompagnés d'outils de définition de l'environnement, de représentation des déplacements d'agents situés et de tracé des variations des propriétés des agents.

Les familles d'agents

Notre approche repose sur quatre familles d'agents (cf. Figure 5). Les agents spatiaux sont les briques de base pour la construction de l'environnement en les juxtaposant et en les connectant. Selon le mode de connexion, il est possible de définir des environnements toriques ou bornés. Comme nous le verrons par la suite, l'interconnexion des agents spatiaux entre eux est aussi à la base des canaux de communication entre agents situés et spatiaux.

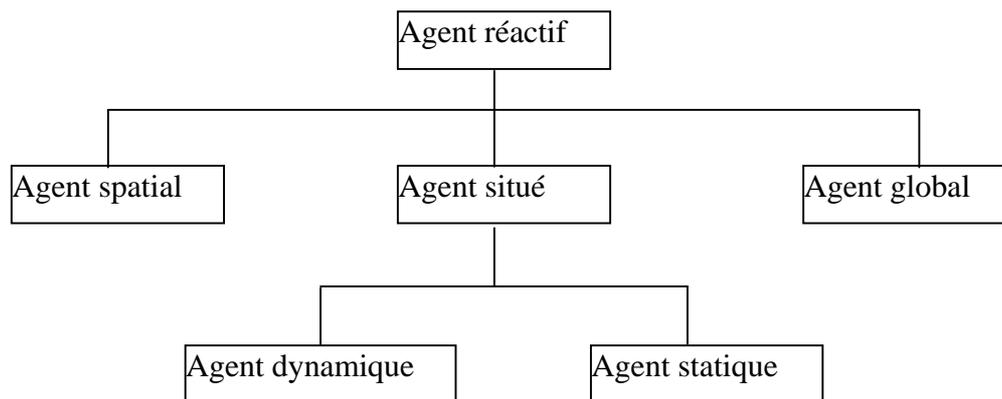


Figure 5. Hiérarchie des familles d'agents dans notre simulateur

Les agents situés constituent la deuxième grande famille. Un agent situé se trouve en relation avec un et un seul agent spatial. Cette relation permet de définir sa position dans l'espace. On distingue deux sous-familles : les agents dynamiques et les agents statiques. La différence fondamentale réside dans le fait que les agents dynamiques ont la faculté de se déplacer par leur propre initiative. En revanche, les agents statiques sont fixes ou peuvent être déplacés par d'autres agents.

Les agents globaux, comme leur nom l'indique, sont des entités communes à l'ensemble des autres agents. Tous les agents peuvent y faire référence.

Les états internes

Malgré leurs différences conceptuelles, tous les agents sont construits autour de l'idée d'agent réactif. Les états internes précisent des valeurs telles que masse en azote, volume de l'estomac, vitesse de nage, fréquence des sauts... , c'est-à-dire toute valeur caractéristique de l'agent susceptible d'intervenir dans son comportement. Ces valeurs sont strictement privées à l'agent et donc, lui seul, en fonction de ce qu'il perçoit et de son comportement, modifie ces valeurs. On peut imaginer, par exemple, qu'après l'absorption d'une cellule de phyto-plancton par un copépode, celui-ci augmente sa masse en azote d'une certaine quantité (une fois assimilée).

La dynamique

La dynamique, directement liée au comportement que l'on cherche à tester, fait appel aux réseaux de Pétri (cf. Figure 6). Celui-ci a pour objectif de modéliser les changements d'état. On retrouve

dans cet outil mathématique les notions d'état (stable) ou de place et de transition. A un état (ou place) est associé un ensemble d'actions (déplacement, mise à jour de l'état interne, envoi de message...) réalisées lors de l'activation de cet état. L'état est activé à chaque réception de jetons. Une transition, quant à elle, marque le passage d'un état stable à un autre. Ce passage s'effectue si le nombre de jetons nécessaires est atteint dans les places amonts, si la condition associée à la transition est valide et après une durée de temporisation déterministe ou stochastique. En d'autres termes, pour qu'il y ait changement d'état, il faut que les conditions de franchissement de la transition soient validées et, après une certaine attente, le nouvel état stable est activé.

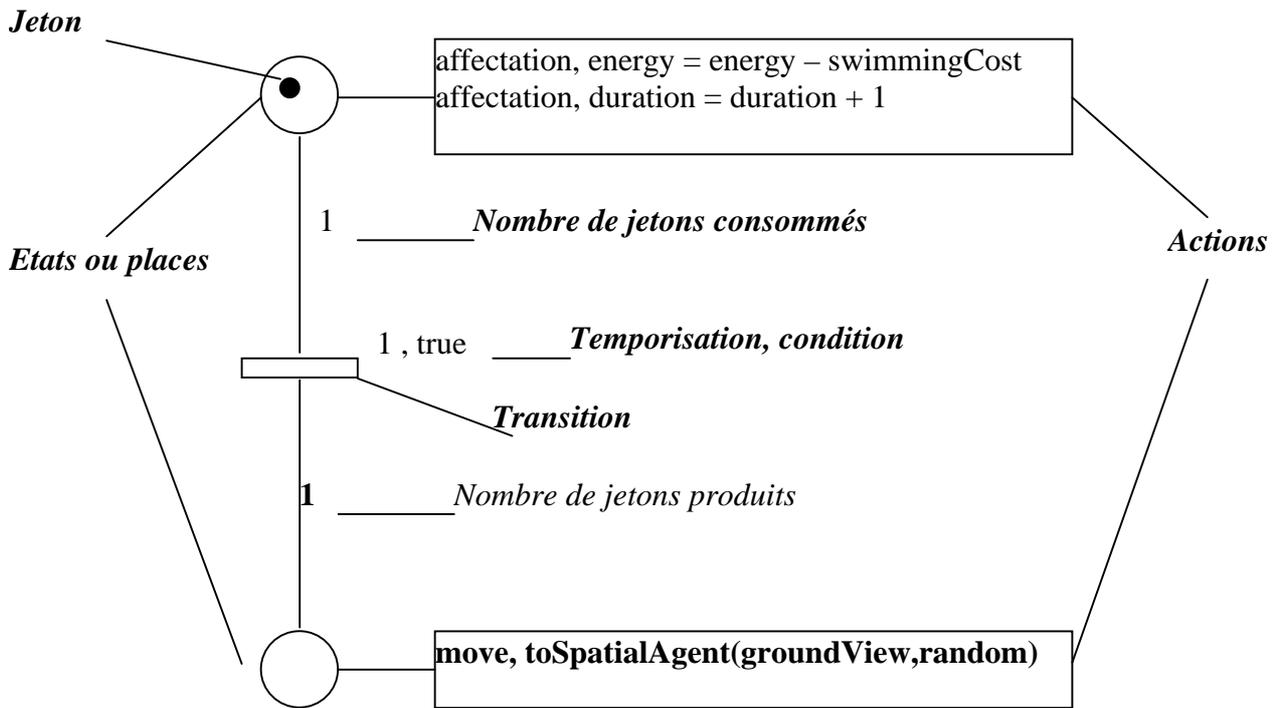


Figure 6. Réseau de Pétri temporisé modifié

L'introduction de durée de temporisation rend les modèles dépendant de la variable temps, si nécessaire. En effet, l'exécution des réseaux de Pétri temporisés construit dynamiquement un échancier.

Nous avons introduit une deuxième technique pour la définition de la dynamique : l'introduction dynamique de classes Java. Ces classes permettent de décrire en Java un comportement. Cette option a l'avantage d'augmenter sensiblement la vitesse d'exécution des simulations.

La perception

Par analogie avec les entités biologiques, tout agent est doué de sens plus ou moins développés. Dans notre modèle, un agent possède plusieurs sens et chacun d'eux est défini par trois paramètres : le type d'agents perçus, le secteur (selon l'orientation de l'agent) et la distance jusqu'à laquelle ces agents sont perçus. Par ce biais, on peut tester différents scénarii selon les perceptions possibles du copépode (par exemple, perception des cellules de phytoplancton situées dans un secteur de 90 degrés et à une distance inférieure à 1mm [2] – cf. Figure 7). De plus, un agent peut ne percevoir qu'une partie des caractéristiques d'un autre agent. On précise dans ce cas les caractéristiques perçues. Un agent peut aussi disposer de plusieurs sens. C'est lors d'actions que l'agent précisera le sens utilisé.

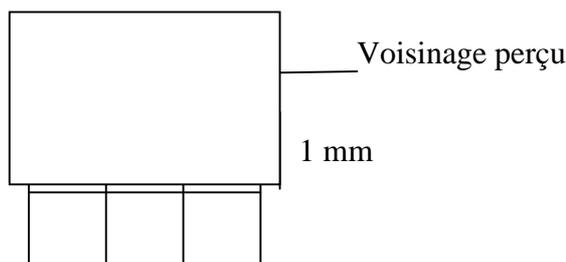


Figure 7. Distance de vision = 1 et secteur = 2

Les sens sont aussi, dans cette approche, à la base du réseau de connaissances des agents. Dynamiquement, l'agent construit son voisinage en fonction de sa position dans l'espace et de ses sens.

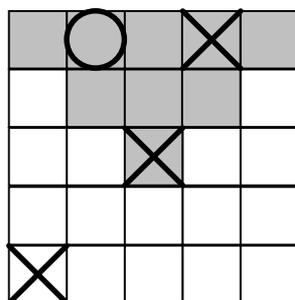


Figure 8. Exemple de réseau de connaissance par voisinage

A titre d'exemple (cf. Figure 8), considérons les agents "croix". Ils perçoivent totalement leurs congénères et les agents spatiaux comme indiqué sur la Figure 8. Les cases grises représentent les agents spatiaux perçus (section de 90° et distance de 2). En revanche, les agents "croix" ne perçoivent pas les agents "rond". Il en résulte que l'agent situé au centre de la grille ne perçoit qu'un seul agent (l'agent "croix" de la première ligne) et l'ensemble des agents spatiaux grisés.

Indirectement, les sens définissent aussi les possibilités de déplacement des agents dynamiques. Si l'environnement se compose de différents types d'agents spatiaux, l'un de ces agents peut ne pas être perçu par un agent dynamique. Prenons l'exemple d'un labyrinthe, il est composé de couloirs et de murs. Les agents qui se déplacent dans ce labyrinthe ne peuvent pas se positionner sur les murs. On interdit donc la perception des murs par les agents mobiles. Lors d'une action de déplacement, l'agent va construire l'ensemble des futures positions possibles en fonction de ses perceptions et choisir parmi elles.

Les messages

La quatrième caractéristique concerne les communications entre agents. Les agents ont la possibilité d'envoyer des messages aux seuls agents qu'ils perçoivent. Ces messages se présentent sous la forme d'un vocabulaire que les agents peuvent émettre (ou non) et entendre (ou non) selon leur type. La réception d'un message par un agent le conduit à effectuer une ou plusieurs actions. Le mécanisme d'envoi de messages est le seul moyen pour un agent de modifier les caractéristiques d'un autre agent (à condition que celui-ci l'accepte). L'envoi d'un message est considéré comme une action au même titre qu'un déplacement. A titre d'exemple, la place n°3 de la Figure 9 contient un exemple d'envoi de message (*send,toSpatialAgentWhereIsIt(Init)*). Ce message est envoyé à l'agent spatial où l'agent expéditeur se situe ; il se nomme *Init* et ne possède pas de paramètres.

Lorsque le message est reçu par l'agent spatial, destinataire du message, celui-ci effectuera les actions associées au message.

2.2.2 Le modèle

Nous sommes partis de l'hypothèse que le copépode adopte deux comportements distincts : une nage orientée à la recherche de nourriture et des sauts aléatoires. Ces comportements ont une influence directe sur le processus d'ingestion des cellules de phytoplancton. Nous allons donc nous intéresser exclusivement à ce processus et laisser de côté la partie digestion qui, néanmoins, n'est pas à négliger pour obtenir un modèle complet.

Généralités

Le système se compose de trois entités : la masse d'eau, les cellules de phytoplancton et les copépodes. Dans un premier temps, nous ne considérons qu'un seul copépode à la fois. La masse d'eau constitue l'environnement (ou le milieu) dans lequel évoluent les autres entités. La taille du copépode (1 mm) sert de longueur de base pour la discrétisation de ce milieu. Le milieu est considéré, pour l'instant, en deux dimensions et découpé en parcelles d'1 mm². Chaque parcelle est prise en charge par un agent spatial. Les cellules de phytoplancton sont très nombreuses (de 10 cellules par litre à 10⁸ cellules par litre soit au maximum 10⁴ cellules par parcelle). Il n'est donc pas raisonnable de modéliser chaque cellule par un agent. La solution retenue consiste à définir, au niveau des agents spatiaux, une propriété "Nombre de cellules". On délègue la gestion de la nourriture aux agents spatiaux c'est-à-dire à l'environnement. Quant au copépode, il est représenté par un agent dynamique dont on va décrire la dynamique dans la partie suivante.

Le pas de temps de la simulation est noté u. t. Celui-ci est fixé par la durée correspondant au temps nécessaire à la plus petite action, c'est-à-dire la manipulation d'une cellule de phytoplancton par le copépode soit 1/20 s.

La dynamique du copépode.

Le réseau de Petri modélisant la dynamique de déplacement du copépode (cf. Figure 9) se divise en quatre parties :

- * dès que l'on atteint la fin d'un cycle de 75 u. t.⁴, le copépode effectue un saut sans considérer ce qu'il l'entoure⁵ - **place n°3**,
- * pendant 20 u. t. (le temps de traverser une parcelle du milieu), le copépode explore l'espace où il se trouve et si de la nourriture s'y trouve, il peut, à chaque unité de temps, capturer une cellule de phytoplancton - **place n°9**,
- * en revanche, s'il n'y a pas de nourriture, il continue à nager pour atteindre la parcelle suivante - **place n°8**,
- * au bout des 20 u. t. nécessaires à la traversée d'une parcelle, le copépode choisit une nouvelle parcelle à explorer et s'y rend - **place n°6**.

Intéressons-nous à la phase active de capture de nourriture. A l'exception des sauts aléatoires (place n°3), le copépode nage et parcourt une parcelle toutes les 20 u. t. Lorsqu'il a atteint ce délai, il change de parcelle (place n°6). Ce changement est fonction de la stratégie que l'on teste. Dans le cas de la Figure 9, la parcelle disposant de plus de nourriture aura plus de chance d'être choisie ce qui se traduit par l'instruction *move,toSpatialAgent(groundView,true,random(currentQuantity))*.

⁴ 75 u. t. est une valeur moyenne issue d'une campagne d'observations.

⁵ pour l'instant, c'est l'hypothèse retenue car on ne sait absolument pas pourquoi il saute.

Cette instruction fait partie de l'ensemble des actions disponibles et permet le déplacement d'un agent dynamique vers un des agents spatiaux perçu par un sens, ici *groundView*, et sélectionné par tirage aléatoire proportionnel à la valeur d'une variable (*currentQuantity*, dans notre cas).

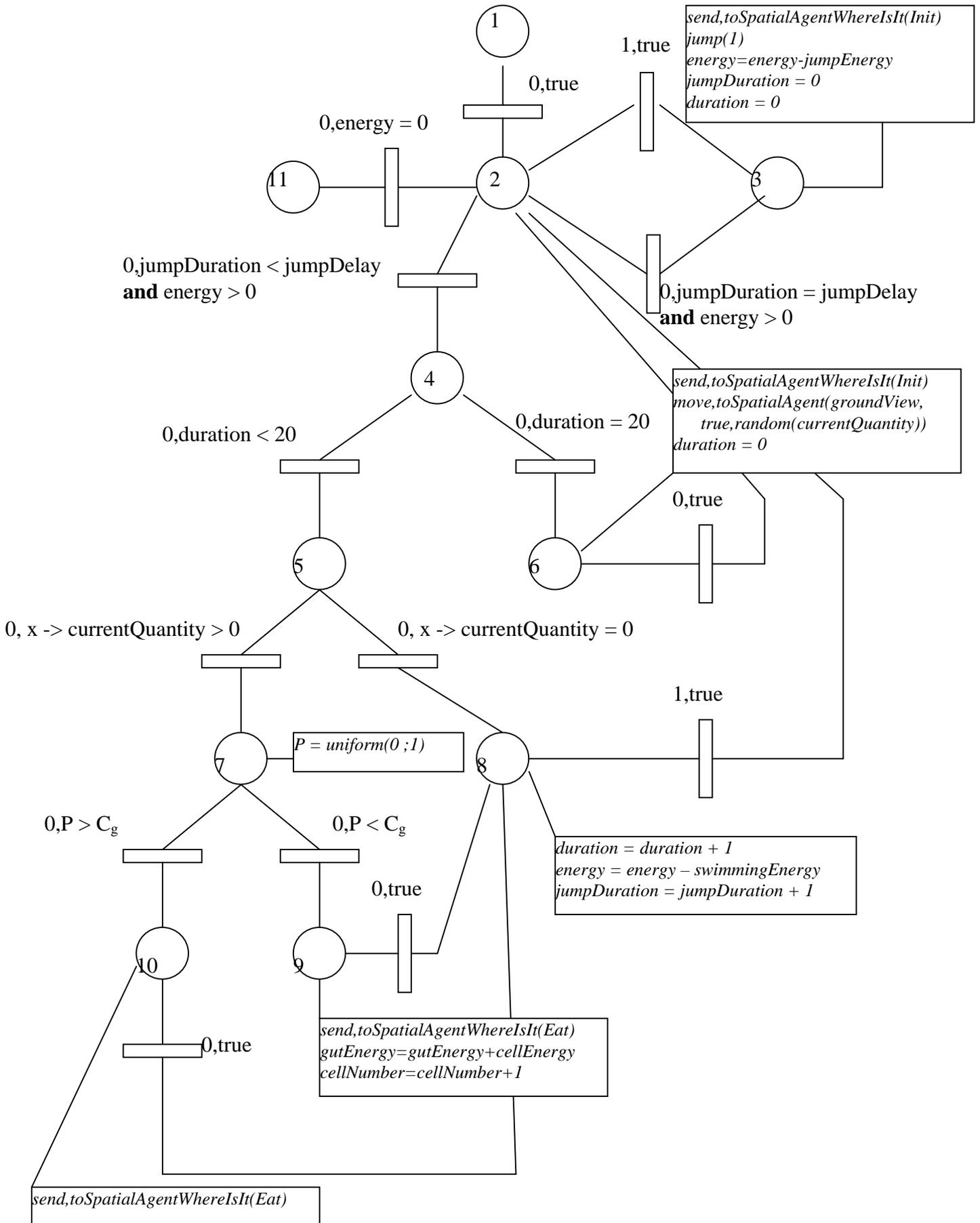


Figure 9. Dynamique de déplacement

Localement, le copépode capture les cellules de phytoplancton. S'il n'a pas encore tout consommé, on vérifie s'il a "envie" de manger (place n°7). En effet, nous avons vu que le copépode diminue la quantité de nourriture qu'il absorbe en fonction de son niveau de satiété, lui-même directement lié, pour l'instant, au nombre de cellules de phytoplancton présentes dans l'estomac. La fonction de satiété est la suivante :

$$C_g = 1 - \left(\frac{V_{\text{proie}} X_1}{\frac{2}{3} V_{\text{estomac}}} \right)^2$$

où $V_{\text{proie}} X_1$ présente le volume des proies non digérées et V_{estomac} le volume de l'estomac du copépode.

Si le tirage aléatoire de la place n°7 est favorable alors il capture la cellule sinon cette cellule disparaît de son champ de vision.

Réflexions et extensions.

Ce premier modèle est une traduction algorithmique du modèle analytique. On y retrouve des valeurs moyennes issues de bilans et des processus aléatoires, tel que celui de la satiété. La même remarque est valable pour la partie digestion. Le seul élément nouveau est la prise en compte du comportement au niveau de la nage. Il faut néanmoins souligné que la construction des agents est nettement plus simple et fait intervenir un nombre réduit de paramètres par rapport au modèle analytique.

Nous nous intéressons maintenant à décomposer les processus "énigmatiques", tels que les sauts aléatoires ou la vitesse de nage constante. Il est évident que derrière ces actions se cachent des règles de décision ou des processus plus complexes. Par exemple, l'observation montre que la vitesse de nage n'est pas constante et que ces variations sont dues à des interactions avec l'environnement. Il reste, aux biologistes, à les définir ou à les imaginer.

D'autre part, si on s'intéresse à la règle de décision de la capture ou non d'une cellule de phytoplancton, on peut se demander si des éléments comme l'état physiologique du copépode ne rendent pas en ligne de compte. Par exemple, une femelle, porteuse d'œufs, a des besoins en nourriture plus important qu'un mâle. Il paraît aussi évident d'un copépode seul ne vit pas de la même façon qu'une colonie de copépodes.

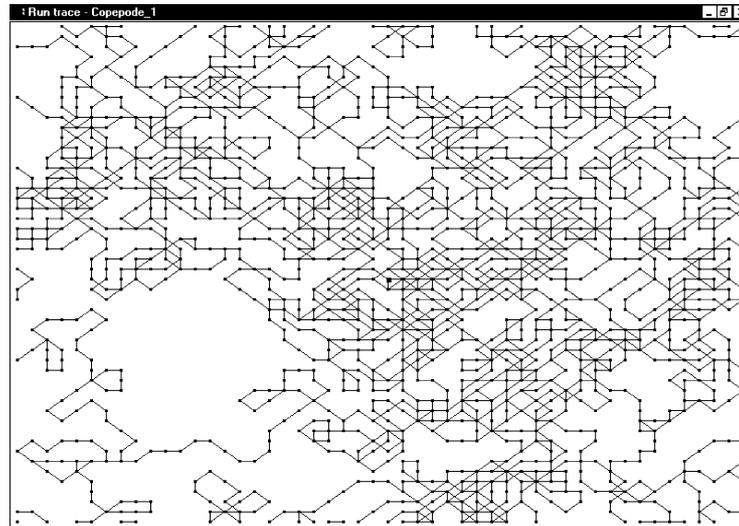
2.3 Résultats expérimentaux

Nous avons défini deux types de copépode en fonction de leur stratégie de nage : aléatoire ou orientée. L'environnement se compose d'une grille 2D de 2500 parcelles carrées (50x50). Chaque parcelle est un agent spatial et est connectée à ces 8 voisines. Les cellules de phytoplancton sont réparties soit par patches (cf. Figure 10 – tâches grises) soit uniformément. Dans les deux cas, la densité globale est identique (1 cellule par parcelle). Pour les patches, différentes densités sont utilisées (représentées par des niveaux de gris différents).

A l'aide des variables définies au niveau des agents *Copéode*, on mesure à chaque pas de la simulation : l'énergie, exprimée en pg^6 d'azote, contenue dans l'estomac (*gutEnergy*), l'énergie utilisable (*energy*), le nombre de cellules de phytoplancton capturées (*cellNumber*) et quatre variables du modèle analytique (X_3 , X_4 , T_g et C_a).

Figure 10.
copéode (nage

Nous disposons
visualisation de
permet de tracer le
par un agent situé
simulation. En
trajectoire du
et la distribution
phytoplancton, on
le cas d'une
cellules de
par patch, la



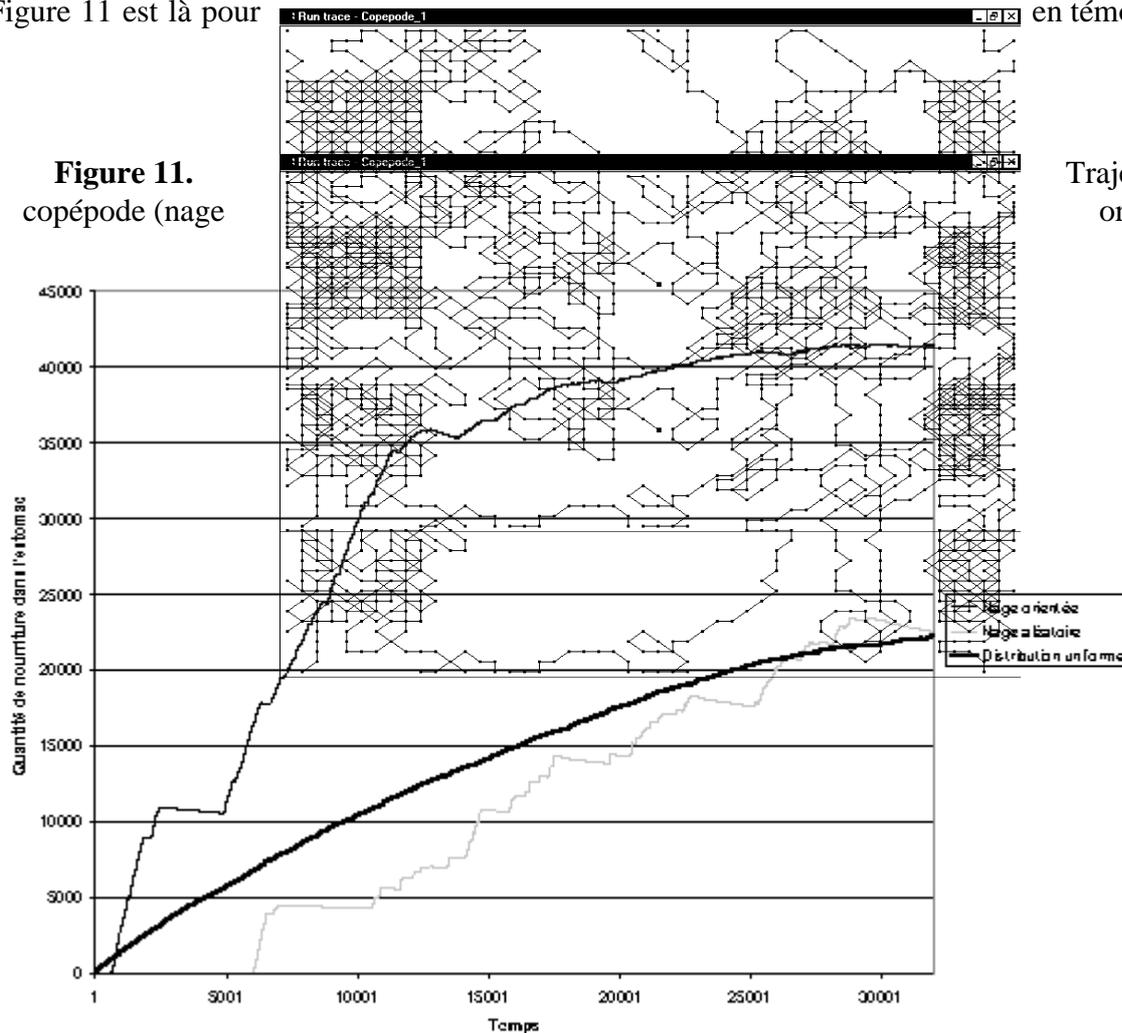
Trajectoire du
aléatoire)

d'un outil de
trajectoire. Il
chemin parcouru
durant la
superposant la
copéode étudié
des cellules de
montre que dans
distribution des
phytoplancton
stratégie de la

nage orientée est plus efficace. La

Figure 11 est là pour

Figure 11.
copéode (nage



en témoigner.

Trajectoire du
orientée)

⁶ picogramme

Figure 12. Quantité de nourriture (en pg d'azote) dans l'estomac du copépode en fonction du temps

Si on superpose les courbes représentant la quantité de nourriture dans l'estomac du copépode en fonction du temps et selon la distribution des cellules de phytoplancton et la stratégie de nage, il apparaît très rapidement que la nage orientée est favorable à l'alimentation du copépode du point de vue énergétique.

Si le copépode est plongé dans un champ homogène, la stratégie de nage n'a aucune influence sur l'alimentation puisqu'il trouvera de la nourriture dans toutes les directions en même quantité. En revanche, dans un champ hétérogène, la stratégie aléatoire conduit le copépode à tomber par hasard sur un patch de cellules de phytoplancton et surtout d'en sortir sans chercher à en profiter. Donc cette stratégie est moins efficace.

En conclusion, on retrouve les principaux résultats énoncés dans [1] pour une configuration de champ uniforme et de nage aléatoire. Il reste à effectuer des comparaisons avec des résultats d'expériences *in vivo* mais qui restent pour l'instant difficile à réaliser. Les seuls points de comparaison possible concerne l'aspect des trajectoires selon la densité de nourriture (cf. [3]).

3 Discussion

Nous avons présenté la démarche et les premiers résultats d'un travail reposant sur une collaboration informatique/biologie qui a pour objectif d'étudier le comportement d'un organisme zoo-planctonique, le copépode. Nos résultats sur un champ de phytoplanctons hétérogène sont les premiers obtenus. Ils montrent clairement la différence par rapport au champ uniforme, qui aura des conséquences importantes sur le comportement, donc sur la dynamique de l'écosystème marin. En effet, si le phytoplancton est distribué de manière hétérogène, le copépode a un comportement qui tend à les concentrer dans des zones où le phytoplancton se trouve. Le copépode étant lui-même la nourriture de certains poissons, ceux-ci auraient à leur tour une répartition différente dans les eaux...

Ce travail réalisé, de nombreuses voies de poursuite sont possibles. Tout d'abord, le passage à un système 3D est envisagé. Ce passage du 2D au 3D ne devrait pas remettre en cause les conclusions de ce travail au niveau biologie. Par contre, pour l'informaticien, le passage à la troisième dimension soulève des problèmes en terme de charge de calcul. Ce problème lié au nombre d'objets et au nombre d'interactions à simuler est plus général que celui du passage au 3D ; c'est un problème que l'on rencontre partout, dès que le système est un peu compliqué. Nous travaillons actuellement sur l'agrégation d'agents pour faire face à ce problème.

Les temps de calcul sont également dus au fait que la plate-forme a été développée en Java pour permettre, notamment, une compatibilité avec Internet. Par ailleurs, le développement d'une application à l'aide d'agents " propres " peut entraîner des surcharges de traitements.

Ce travail conceptuel et cette plate-forme sont actuellement en cours de réutilisation dans le cadre d'une collaboration avec un laboratoire de géologie où l'aspect laboratoire virtuel séduit également beaucoup les chercheurs.

L'utilisation d'un modèle multi-agents nous semble être indispensable pour autoriser l'étude fine d'un comportement, ce qu'un modèle analytique ne nous aurait pas permis.

Au niveau méthodologique de modélisation, nous avons dû étudier le passage d'un modèle reposant sur des bilans à un modèle orienté individu ; nous retrouvons le même type de changement de paradigme dans notre travail avec des géologues. Les variables habituellement manipulées dans les modèles dans ces sciences sont en effet collectives, émergentes de l'activité des individus. Nous pensons que l'approche multi-agents oblige à se concentrer sur les " vraies " questions. Dans le travail décrit ici, le comportement en moyenne d'un copépode n'a aucun sens : le copépode n'a

jamais ce fameux comportement moyen, et ce lissage masque les véritables comportements, altérant finalement l'étude du système en se concentrant sur de faux problèmes et en ignorant les bonnes questions.

4 Références

1. Caparroy P. et Carlotti F., *A model for Acartia tonsa : effect of turbulence and consequences for the related physiological processes*, Journal of Plankton Research, vol. 18, n° 11, pp. 2139-2177, 1996
2. Tiselius P. et Jonsson P. R., *Foraging behaviour of six calanoid copepods : observations and hydrodynamic analysis*, Marine ecology progress series, vol. 66, pp. 23-33, 1990
3. Bundy M. H., Gross T. F., Coughlin D. J. et Strickler J. R., *Quantifying copepod searching efficiency using swimming pattern and perceptive ability*, Bulletin of Marine Science, vol 53, n°1, pp. 15-28, 1993
4. Ferber J., *Les systèmes multi-agents, vers une intelligence collective*, InterEditions, 1995
5. Pavé A., *Modélisation en biologie et en écologie*, ed. Aléas, 559 p., 1994
6. Cambier Ch., *Simdelta --- Un système multi-agents pour simuler la pêche sur le delta central du Niger*, Thèse de doctorat, Université de Paris 6, 1994
7. Cambier Ch., Perrier E., Treuil J-P. et Preux Ph., *Action physique et géométrique, Contribution à une réflexion sur l'utilisation des processus physiques, Application RIVAGE*, poster, 5^{ième} JFIADSMA, avril 1997
8. Marcenac P., *Modélisation et simulation par agents --- Application aux systèmes complexes*, mémoire d'habilitation à diriger des recherches, Université de la Réunion, 1997
9. Marcenac P., *Modélisation de systèmes complexes par agents*, Techniques et Sciences Informatiques, TSI, vol. 16, n°8, pp. 1013-1037, octobre 1997
10. Perrier E., Cambier Ch., *Une approche multi-agents pour simuler les interactions entre des acteurs hétérogènes de l'infiltration et du ruissellement d'eau sur une surface de sol*, Journées CNRS "Tendances nouvelles en modélisation pour l'environnement" du programme environnement, vie et sociétés, janvier 1997
11. Treuil J-P., Perrier E., Cambier Ch., *Directions pour une approche multi-agents de la simulation de processus physiques spatialisés*, 5^{ième} JFIADSMA, avril 1997
12. Drogoul A. et Ferber J., *Multi-Agent Simulation as a Tool for Modeling Societies: Application to Social Differentiation in Ant Colonies*, Actes du Workshop MAAMAW'92, Viterbo, 1992
13. Minar N., Burkhart R., Langton C. et Askenazi M., *The SWARM simulation system : a toolkit for building multi-agent simulations*, 1996