

La modélisation individus-centrée en biologie des populations

C. Bertelle

Maîtrise de Biologie
Université du Havre

0-0

C. Bertelle

Plan

1. L'étude des systèmes complexes
2. Le concept objet et les modèles individu-centrés
3. Quelques simulations en biologie et en écologie
4. Mise en œuvre élémentaire avec une plate-forme de développement (Madkit)

1. L'étude des systèmes complexes

La biologie des populations et des écosystèmes sont des exemples de systèmes composés d'éléments nombreux, hétérogènes et en interaction : c'est le cadre d'études auquel s'intéresse la systémique.

Références bibliographiques introductives

- D. Durand "*La systémique*", Que sais-je ? PUF.
- J. de Rosnay "*Le macrocosme*", Edition du seuil.
- S. Frontier "*Les écosystèmes*", Que sais-je ? PUF.

2- Modélisation individus-centrée

1.1 Systèmes et systémique

Un système - au sens de la systémique - est un ensemble constitutif et des propriétés le structurant en tant que tel.

- Ensemble constitutif d'entités en interactions mutuelles et en interaction avec un milieu extérieur ou environnement ;
- Propriétés caractéristiques conférant la structure de système :
 - dépendance interactive des éléments/entités du système, indissociable de leur dynamique (la modification d'une interaction ou d'un élément se répercute sur l'ensemble)

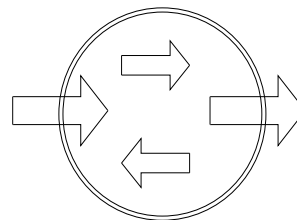
3- Modélisation individus-centrée

- existence ou émergence d’une organisation globale constitutive du système, identifiable et possédant une autonomie globale tout en étant en relation/dépendance avec son environnement. L’organisation émergente a des propriétés nouvelles par rapport aux entités dont elle est issue : *“le tout est plus que la somme des parties”*
- rétroaction de l’organisation globale sur ses parties constitutives : *“le tout est moins que la somme des parties”* (E. Morin)

4- Modélisation individus-centrée

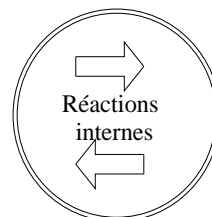
1.2 Systèmes fermés et systèmes ouverts

Système ouvert : interagit avec environnement, transfère énergie/matière, permet formation d’organisations émergentes structurantes.



Système ouvert

Système fermé : coupé du monde extérieur, sans émergence dynamique.



Système fermé

5- Modélisation individus-centrée

1.3 Systèmes complexes

- Ajoute à la notion générale de systèmes le caractère hétérogène des constituants de nombreux systèmes naturels (biologiques, économiques, ...)
- Dissociation entre “système compliqué” et “système complexe” (d’après J.L. Lemoigne) :
 - système compliqué : on peut le simplifier pour découvrir son intelligibilité ;
 - système complexe : en le simplifiant, on détruit son intelligibilité que l’on ne peut découvrir qu’en le modélisant dans sa totalité.

6- Modélisation individus-centrée

- Principe d’organisation en vue d’une fonction d’objectifs (survie, adaptation, reproduction):
 - comportement téléologique, c’est à dire guidé par ses buts.
 - Le concept d’instrumentation d’un tel comportement est le “feed-back” informationel (issu de la notion de contrôle en cybernétique)

7- Modélisation individus-centrée

- Organisation hiérarchique de systèmes et sous-systèmes assurant sa stabilité.
- Stratégies de développement adaptatives (réorganisations structurales face à des fluctuations) ou paradoxales (coexistence d'éléments contradictoires, complémentaires, concurrents ou antagonistes).

1.4 Evolution des systèmes complexes

En général succession de périodes d'auto-organisation et de périodes de stress.

- auto-organisation : période sans perturbation majeure. Le système a tendance à se complexifier pour un perfectionnement fonctionnel et une amélioration de son pouvoir d'adaptation.
- stress : période de perturbation où le système se simplifie et se déstructure partiellement (ou totalement).

1.5 Les écosystèmes - approche systémique

- Ecologie (Haeckel, 1866) : science de l'habitat. D'abord études disjointes des espèces puis nécessité de prendre en compte les autres espèces et le milieu environnant avec lesquels chaque espèce interagit.
- Ecosystème (Tansley, 1935) : système d'interaction entre les populations de différentes espèces vivant dans un site et entre ces populations et le milieu physique.

10- Modélisation individus-centrée

Caractéristiques des écosystèmes

- Structuration hiérarchique sur une grande variété d'échelles (d'une souche d'arbre en décomposition à l'organisation écologique planétaire) ;
- Flux d'énergie les traversants (rayonnement solaire, énergie de mise en mouvement de matière telle que l'eau ou l'air) ;
- Interactions multiples et non réductibles ;

11- Modélisation individus-centrée

Plusieurs étapes d'évolution des écosystèmes

- état juvénile : croissances importantes mais instables ;
- état de maturité : stabilisation par complexification des espèces permettant l'adaptabilité et la mise en place de structures de survie ;
- état de vieillissement : prédominance de certaines espèces aux dépens des autres et contribuant à la simplification globale ;

12- Modélisation individus-centrée

- + des perturbations générant des stress et remettant en cause l'organisation en provoquant la disparition d'espèces et favorisant le développement rapide d'autres espèces ... rajeunissement.

13- Modélisation individus-centrée

2. Le concept objet et les modèles individu-centrés

2.1 Le concept d'objet

Programmation objet basée sur la construction de composants informatiques, appelés *classes* d'objets contenant :

- des *attributs* caractéristiques ,
- des *méthodes* de traitement spécifiques.

Instanciation d'une classe = un objet informatique dont la nature correspond à la classe. On parle d'*instancier* la classe lorsque l'on crée l'objet.

Exemple de programme basé sur une approche objet

On décrit ci-dessous, dans un pseudo-langage, différentes classes permettant de simuler un troupeau de cervidés :

```
//Description d'une classe
classe Cervide {
  // attributs :
  coordonneeX, coordonneeY,
  age, poids, ...
  // methodes :
  seDeplace( ... ),
  mange( ... ), ...
}
```



```
// Implementation methodes de la classe Cervide
Cervide::seDeplace(direction, distance, ...) {
    mise a jour de coordonneeX et coordonneeY
}
...
// Description d'une classe troupeau
classe troupeau {
    groupe de cervide
    ...
}
```

```
Programme simul_cervides {
    creation d'une instance itroupeau de troupeau
    pour une semaine
        pour chaque animal de itroupeau
            faire executer seDeplace
            faire executer mange
            ...
        fin pour
    observer ou afficher itroupeau
    fin pour
}
```

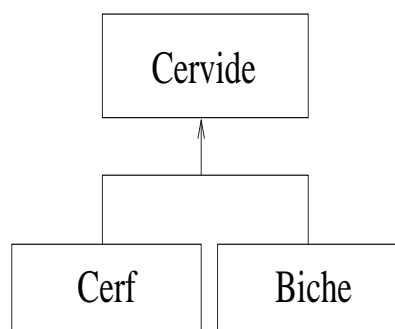
2.2 Concept d'héritage en programmation objet

- L'héritage est une notion essentielle de la programmation objet qui consiste à créer une classe à partir d'une autre en lui transférant ses caractéristiques (attributs + méthodes).
- On peut alors compléter ou spécialiser la classe dérivée ou encore redéfinir certaines des caractéristiques de la classe mère.
- Similaire aux classifications taxonomiques.

18- Modélisation individus-centrée

Exemple d'héritage

```
classe Biche
  hérite de Cervide {
  // attributs
  // spécifiques :
  nombre de gestations,
  ...
  // méthodes
  // spécifiques :
  mettre bas,
  allaiter,
  ... }
```



19- Modélisation individus-centrée

2.3 Agents

On peut définir des agents comme des entités informatiques fonctionnellement plus riches que des objets.

- Ils existent dans un environnement : contexte informatique avec lequel ils interagissent, c'est tout ce qu'ils perçoivent individuellement ;
- Ils peuvent agir sur leur environnement, c'est à dire le modifier ;

- Ils interagissent entre eux et s'organisent ainsi en "société", nommées systèmes multiagent. Ils sont adaptés, par nature, à la modélisation des systèmes complexes, au sens de la systémique, et donc à la modélisation des écosystèmes.
- Issus de l'Intelligence Artificielle Distribuées, ils peuvent intégrer une représentation minimale de comportements cognitifs.

2.4 Approche constructiviste et mondes virtuels

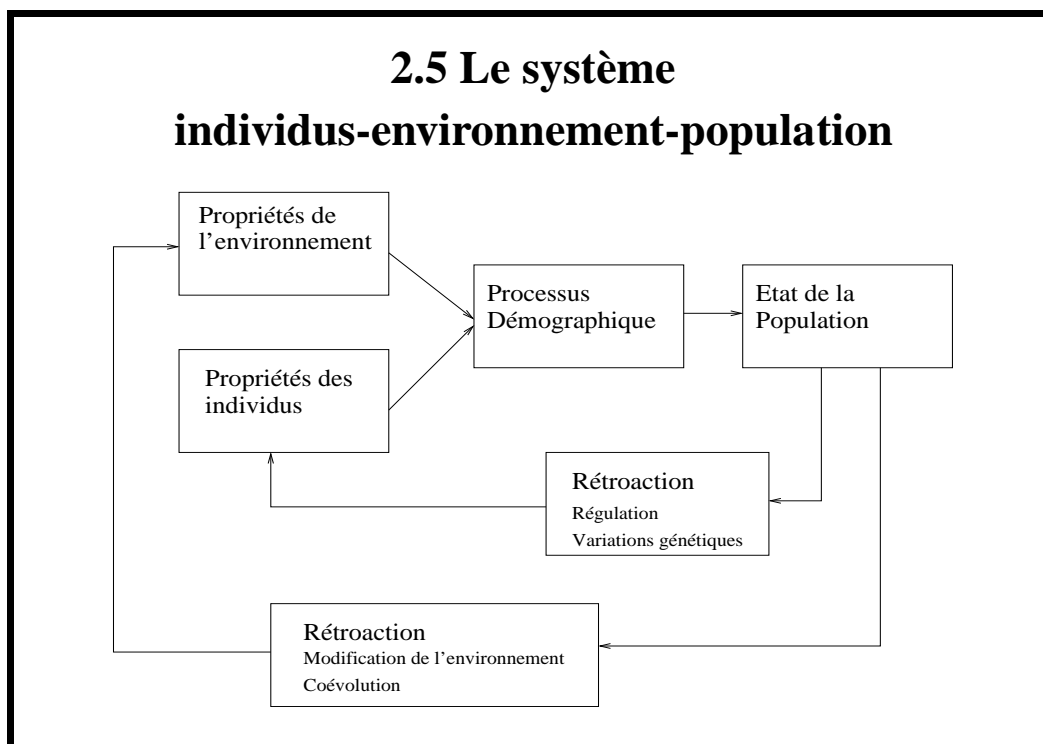
Méthodes d'analyse des phénomènes observés dans la nature :

- Approche phénoménologique : modèles exprimés à l'aide d'un formalisme mathématique exprimant - réduisant - le phénomène global ;
- Approche constructiviste : représentation des composants et de leurs interactions conduisant à des simulations d'où émerge un comportement global.

22- Modélisation individus-centrée

L'ordinateur devient ainsi un outil d'investigation (macrocosme au sens de J. de Rosnay) où paramètres, temps et espace peuvent obéir à de nouvelles règles imposées par l'expérimentateur-modélisateur, maître de la table de contrôle d'un *laboratoire virtuel* dans lequel évoluent des mondes *in silico*.

23- Modélisation individus-centrée



24- Modélisation individus-centrée

Limitation de la représentation d'une population par des variables d'état telles qu'un effectif global :

- Unicité de chaque individu du fait de son patrimoine génétique et de l'influence de son environnement qui conditionnent sa physiologie et son comportement ;
- Localisation des individus affectés essentiellement par leur environnement proche.

La prise en compte des différences entre individus ainsi que de leur répartition spatiale dans une population peut influencer l'évolution globale de cette population.

25- Modélisation individus-centrée

2.6 La simulation individus-centrée

- On représente chaque individu par une entité informatique ;
- On représente un environnement dans lequel évoluent les individus dans l'espace et le temps ;
- On modélise le comportement de chaque individu en réaction à leur environnement ;

3. Quelques simulations en biologie et en écologie

On décrit brièvement un certain nombre de réalisations et d'implémentations basées sur des modèles individus-centrés ou des systèmes multiagents et sur des problématiques de nature écologique.

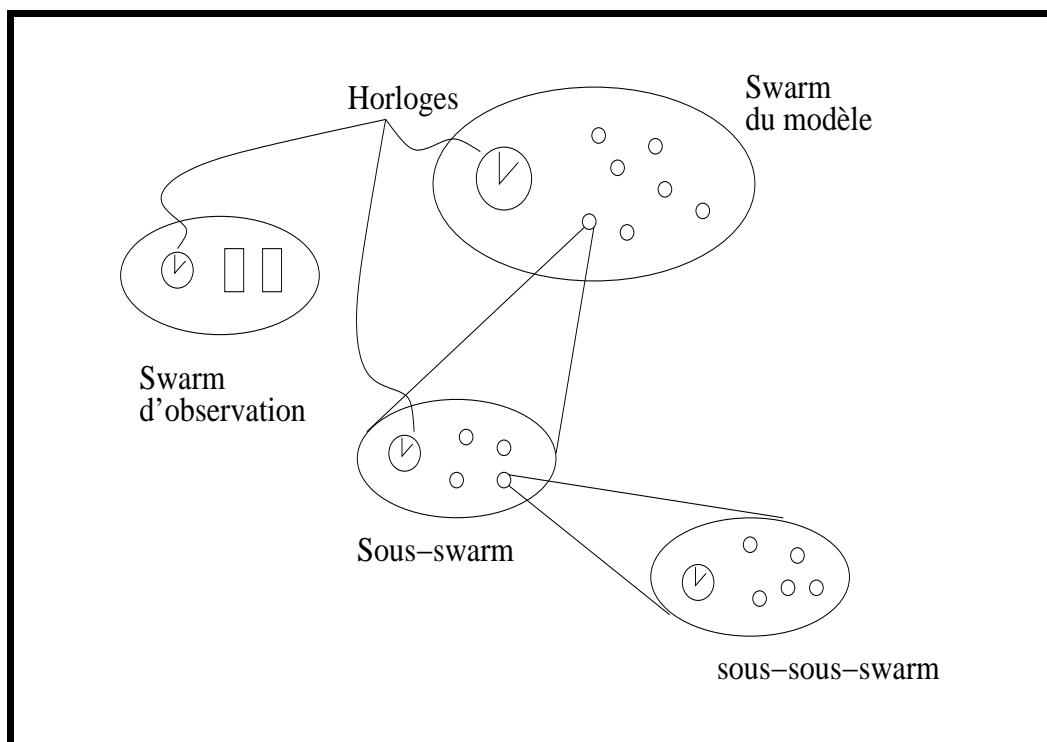
3.1 Swarm/Gecko - www.swarm.org

- Swarm est une plate-forme générique, constituée d'une bibliothèque logicielle (en Objective C, avec une interface Java) qui permet de développer des simulations à base d'agents. Sa conception originale est due à C. Langton.
- Objectif : définir une plate-forme de développement permettant de faire des "expérimentations informatiques" décentralisées à évènement discrets de systèmes complexes.

- Démarche :
 1. Création du modèle : environnement artificiel spatial et temporel où évoluent des entités qui ont des états internes, des règles de comportement et leur propre perception de l'environnement.
 2. Création d'agents d'observation qui enregistrent et analysent les données en provenance de la perception des agents du modèle (système de sondes).

3. Fonctionnement de l'ensemble (monde virtuel + observateurs) dans une simulation discrétisée en temps avec des horloges synchronisées. Les règles de concurrences et dépendances temporelles doivent être explicitement codées.

- Possibilités de hiérarchies et de structures récursives.



De nombreuses applications ont été développées avec Swarm, notamment à caractère biologiques ou écologiques.

Gecko est une de ces applications qui permet de simuler des dynamiques d'écosystèmes.

- Les agents sont représentés par des sphères, ils peuvent se déplacer, grossir ou diminuer de volume.
- Des comportements compétitifs sont mis en place pour le partage de ressources nutritives. Des processus de reproduction asexuée sont également implémentés.
- Des systèmes énergétiques individuels sont modélisés et prennent en compte les assimilations, transformations de

nourriture, ainsi que les coûts de métabolisme et de croissance.

- Des classes spécifiques pour des végétaux, herbivores et carnivores sont définies.
- On peut ainsi étudier des systèmes trophiques complexes constitués de nombreuses espèces d'individus interagissant entre eux de différentes manières.

3.2 SimDelta

C. Cambier *Simdelta : un système multi-agents pour simuler la pêche sur le Delta Central du Niger*”, thèse de l’université Paris 6, 1994.

Laboratoire virtuel pour l’étude de la pêche :

- But : Synthétiser les connaissances de spécialistes en halieutique, écologie, anthropologie, ... sur l’étude du système de pêche du delta du Niger ;
- Simulation de la dynamique de population des poissons en prenant en compte des facteurs biologiques, topologiques variés, le comportement et l’interaction des pêcheurs ;

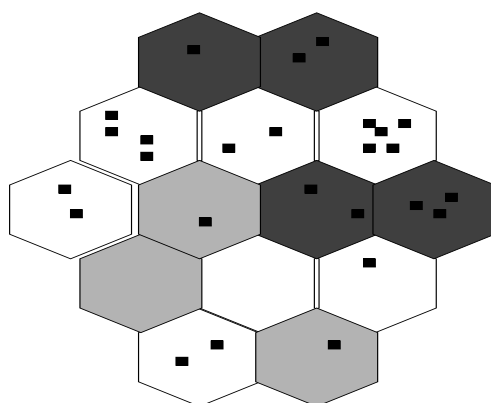
- Trois types d’agents :
 - biotopes représentant des portions d’environnement. Ils sont interconnectés de manière dynamique (modifications lors des crues, par exemple) et possèdent des ressources nutritives variables pour les poissons.
 - poissons avec comportement réactifs. Ils peuvent avoir des comportements collectifs, comme des bancs qui sont eux-mêmes réifiés en agents capables d’adaptation. On intègre des connaissances précises des biologistes sur la reproduction, la croissance, la mortalité et la migration de ces populations.

- pêcheurs avec comportement cognitif. Ils contiennent une base de connaissance comprenant croyances, mémoire et règles de stratégies.
- Plusieurs simulations ont permis de faire des prévisions, dans le cas d'augmentation de l'effort de pêche, sur les modifications induites au niveau de la composition et de la taille des poissons.

3.3 Sealab

C. Lepage "*Biologie des populations et simulations individus centrés*", thèse de l'université Paris 6, 1996 (peut se télécharger).

- Etude de l'influence d'un milieu hétérogène et fluctuant, s'exprimant par l'intermédiaire de comportements reproductifs sur l'abondance d'une population de poissons.
- L'environnement considéré, une portion de milieu marin, est modélisé par un pavage d'hexagones, chacun représentant un fragment d'habitat possédant un indice de conditions hydro-climatiques.



- Les individus-poissons se déplacent et se reproduisent dans cet environnement en fonction des conditions d’habitat. Deux stratégies de comportement sont implémentées :

38- Modélisation individus-centrée

- comportement “opportuniste” où l’individu se déplace vers les habitats présentant les meilleures conditions ;
- comportement “obstiné” où l’individu recherche toujours des conditions de vie analogues à celles de son début de vie.
- On expérimente l’évolution des populations avec des modifications brutales de l’environnement : seules des populations développant les 2 comportements précédents peuvent survivre.

39- Modélisation individus-centrée

3.4 LIL - Université du Littoral - Equipe Ph. Preux

Coopération entre les laboratoires d'informatique et de biologie marine de Lille qui vise à définir des laboratoires virtuels par des SMA pour des études à caractère pluridisciplinaire comme, par exemple, le déplacement de zoo-plancton. Le comportement des agents-individus se fait grâce à des réseaux de Pétri : E. Ramat, P. Preux, Y. Lagadeuc et L. Seuront "*Modélisation et simulation multi-agents en biologie marine - étude du comportement du copépode*", smaget 98. A télécharger sur le site du LIL.

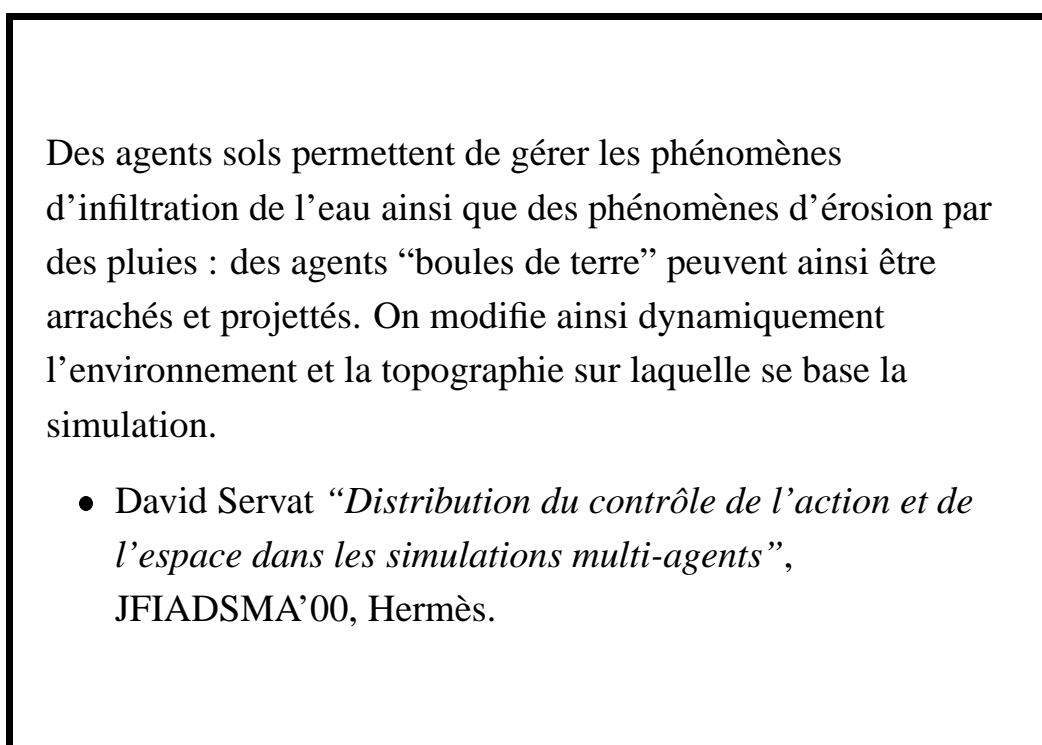
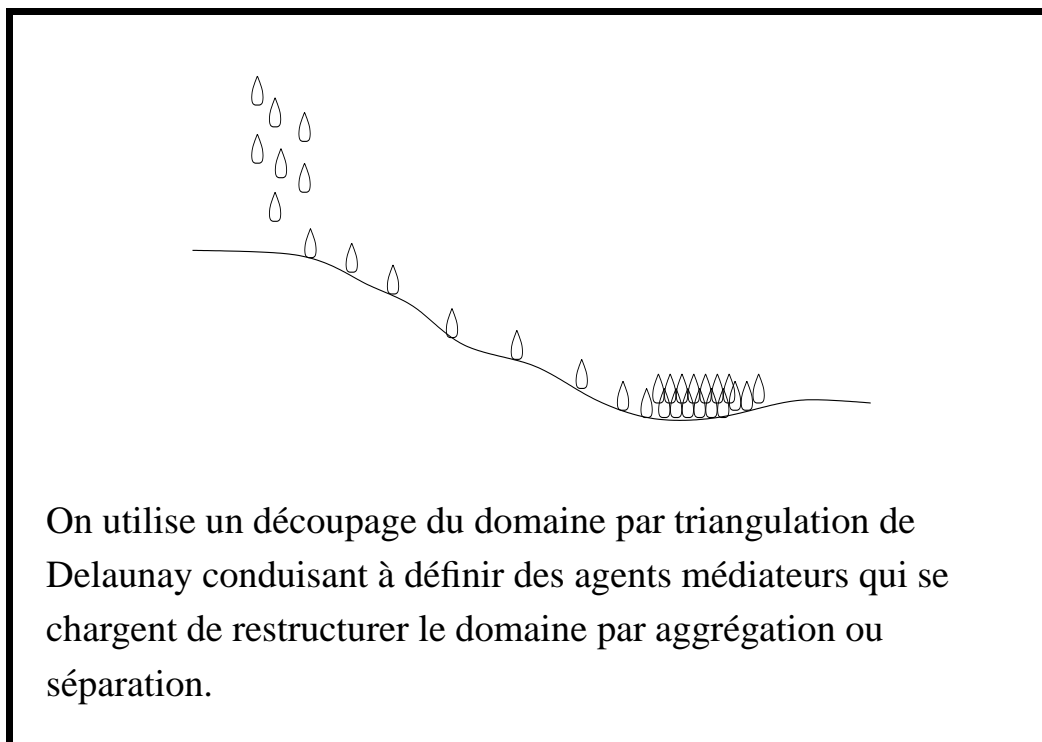
40- Modélisation individus-centrée

3.5 Rivage - thèse D. Servat (27/11/2000)

Modèle décrivant les processus physiques de ruissellement et d'infiltration d'eau pouvant conduire éventuellement à la formation de mares, de ravines ou chemins d'eau.

On simule le comportement d'agents "boules d'eau" qui sont créés par des pluies. Ils se déplacent par pesanteur sur un terrain dont on connaît la topographie, puis s'aggrègent éventuellement.

41- Modélisation individus-centrée



3.6 Manta

Plateforme qui simule des organisations sociales de fourmis :

A. Drogoul “*De la simulation multi-agents à la résolution collective de problèmes*”, thèse de l’université Paris 6, 1993. A télécharger sur sa page personnelle du LIP6 (www.lip6.fr) groupe OASIS.

44- Modélisation individus-centrée

Manta (2)

- Modélisation comportementale de fourmis *Ectatomma ruidum* Roger pour l’étude de l’émergence de structures sociales au sein d’une colonie ;
- Application de l’EthoModeling Framework.



45- Modélisation individus-centrée

Manta (3)

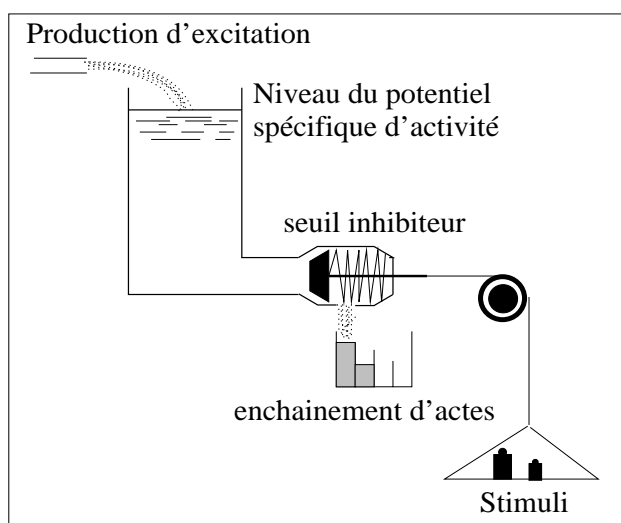
EMF

- Proposé par A. Drogoul (1991-2000) ;
- Utilise le modèle d'activité instinctive de Lorenz
Konrad Lorenz
Les fondements de l'éthologie
1984, Champs Flammarion

46- Modélisation individus-centrée

Manta (4)

Modèle d'activité instinctive



47- Modélisation individus-centrée

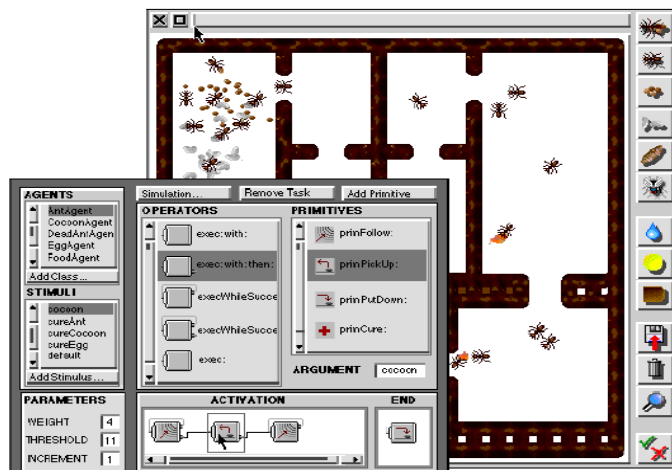
L'activité instinctive est :

- stéréotypée, la marge de variation chez des individus d'une même espèce est très faible ;
- déclenchée par un stimulus externe spécifique par l'intermédiaire du **Mécanisme Inné de Déclenchement** ;
- le stimulus est renforcé par le **Potentiel d'Action Spécifique** qui correspond à la *motivation interne* de l'animal ;
- la réponse à cela se traduit par un enchaînement d'actes élémentaires.

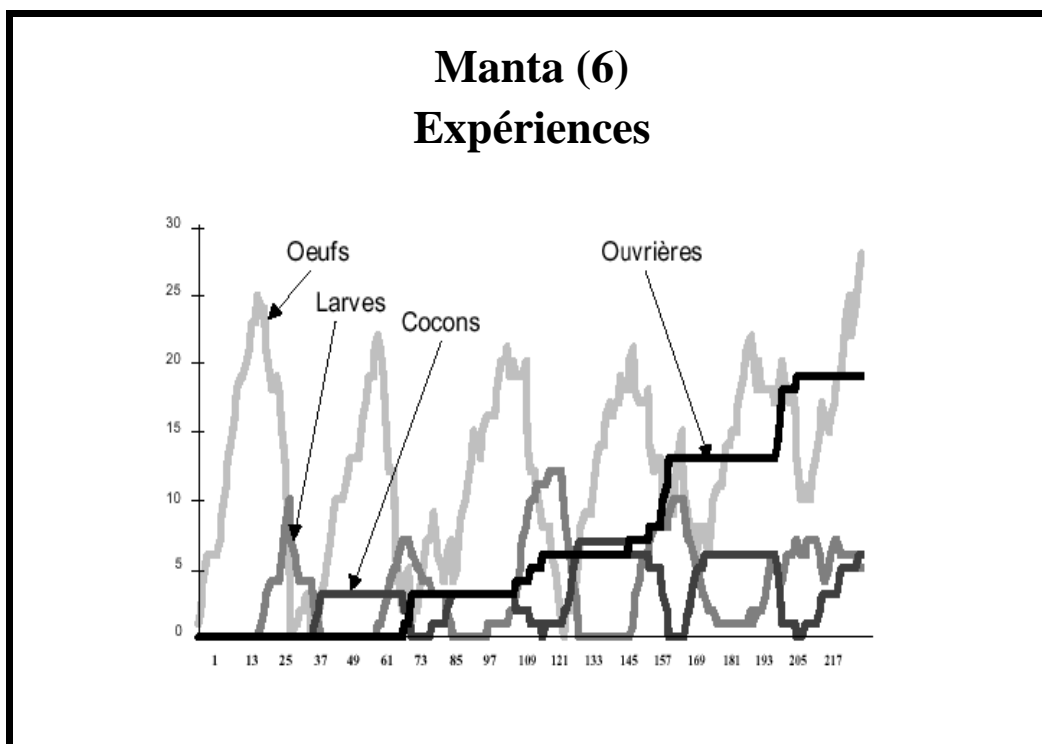
48- Modélisation individus-centrée

Manta (5) Laboratoire virtuel

Outil de programmation dédié aux éthologues



49- Modélisation individus-centrée



50- Modélisation individus-centrée

- Dynamique démographique, organisation sociale évolutive (cas d'études : plusieurs reines, restriction de nourriture, ...)
- Division du travail : stigmergie/feed-back positif, constitution de castes comportementales ;
- Hiérarchie sociale (reine/dominantes/ouvrières).

51- Modélisation individus-centrée

Manta (7)

Conclusion (A. Drogoul)

- Capacité démontrée à
 - Simuler des phénomènes collectifs émergeant de l'interaction entre individus ;
 - Proposer des modèles de comportements, des hypothèses vérifiables dans la réalité ;
 - Interroger la biologie sur ses interprétations.
- La simulation multi-agent est ici un outil essentiel pour la construction de modèles explicatifs de phénomènes complexes.

52- Modélisation individus-centrée

4. Présentation d'une plate-forme de développement : MadKit

- MadKit : Multi-Agent Development Kit écrite dans le langage JAVA et créée par O. Gutknecht & J. Ferber.
- Une plate-forme qui se déclare générique avec des spécifications légères.

53- Modélisation individus-centrée

4.1 MadKit : modèle conceptuel

- Agent : entité d'un ou plusieurs groupes.
- Groupe : ensemble d'agents.
 - un agent ne peut communiquer qu'avec des agents appartenant au même groupe que lui ;
 - un agent peut appartenir à plusieurs groupes ;
 - un groupe peut être fondé n'importe quand par n'importe quel moment et un agent peut demander à appartenir à n'importe quel groupe ;
- Rôle : fonction que remplit un agent au sein d'un groupe. Un agent peut s'occuper de plusieurs rôles.

4.2 Un agent MadKit

- méthode `activate` : création de l'agent en tant que tel avec déclaration de son appartenance à un ou des groupes et en définissant son ou ses rôles ... avec possibilité d'évolution dynamique.
- méthode `live` : décrit le comportement de l'agent au cours de la simulation ;
- méthode `end` : supprime l'agent de la simulation.

4.3 La plate-forme MadKit

- Lancement des agents dans une console, ou par l'interface graphique : Gbox.
- Gestion automatique des groupes et de ses membres ;
- Organisation dynamique pouvant permettre à certains agents de supprimer ou de générer des groupes ou d'autres agents.

4.4 Un exemple élémentaire

La classe PingPong :

```
import madkit.kernel.*;
import java.io.*;
import java.util.*;

public class PingPong extends Agent
{ AgentAddress other = null;

  public void setOther(AgentAddress add)
  { other = add;
  }

  public AgentAddress getOther()
```

```
{ return other;
}

protected void activate()
{ println("PingPong agent Activated");
  println("Looking for a ping-pong group...");
  if (isGroup("ping-pong"))
  {   println ("Yeah ! I join");
      joinGroup("ping-pong");
  }
  else
  {   println ("Nope ! I create one");
      foundGroup("ping-pong");
  }
}
```

```
    requestRole("ping-pong", "player");
}

protected void live()
{ println("Looking for a sport partner...");
  do
  { pause(800);
    AgentAddress[] v =
      getAgentsWithRole("ping-pong", "player");
    for (int i=0; i < v.length; i++)
    { AgentAddress agent = v[i];
      if (!agent.equals(getAddress()))
        other = agent;
    }
  }
}
```

```
    }
    while (other == null);

    println("Other is :"+other);
    sendMessage(other, new Message());

    for (int i = 5; i > 0; i--)
    { waitNextMessage();
      println("GEE ! My turn..." + new Date());
      pause(1000);
      sendMessage(other, new Message());
    }
  }
}
```

```
public void end()
{ println ("PingPong agent Ended");
}
}
```

Script Scheme :

```
(launch-agent 'PingPong 'pong)
(launch-agent 'PingPong 'ping)
```

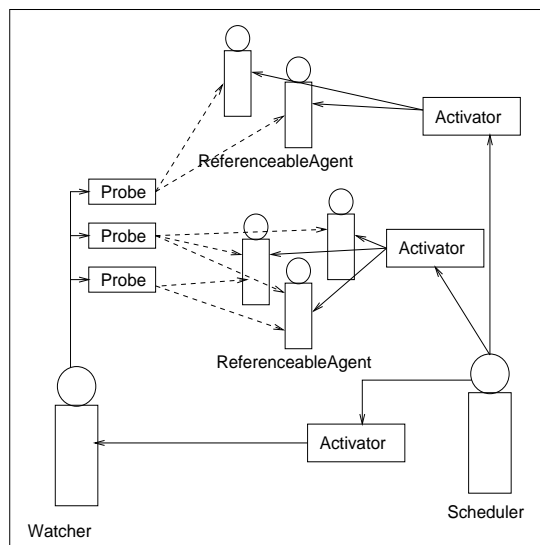
Commande de lancement :

```
export CLASSPATH=.:$CLASSPATH$: ...
/madkit/madkit-1.4-PREDEV/madkit.jar
java madkit.platform.console.Booter
--config config.scm
```

4.5 Gestion de SMA de grande taille avec le “moteur synchrone”

Objectif : permettre la construction de SMA avec un grand nombre d’agents (des centaines), mais aussi la mise en place d’une synchronisation dans le fonctionnement des agents du SMA.

4.5.1 Architecture générale du moteur synchrone



4.5.2 Agent threadé ou non threadé

- Les threads sont des “*processus légers*” qui peuvent s’exécuter de manière concurrente, tout en partageant un même espace mémoire.
- Le nombre de threads gérés par les systèmes, notamment les machines virtuelles comme celle de Java, est en général limité.
- Les agents peuvent être associés chacun à un thread ou alors ils peuvent s’exécuter dans un même processus et ceci de manière séquentielle.

Les classes d’agents de base :

- `AbstractAgent` classe d’agents de base, implémentant l’organisation Agent/Groupe/Rôle, et la gestion des messages.
- `Agent` classe dérivée de la précédente où chaque agent s’exécute dans son propre thread.

Les références d'agents :

- Par défaut, on accède à la référence d'un agent Madkit grâce à la fonction `getAddress` qui renvoie un résultat de la classe `AgentAddress` : format d'adressage propre à Madkit.
- Un agent d'une classe dérivant de `AbstractAgent` et implémentant l'interface (vide) `ReferenceableAgent` peut être référencé directement par son adresse physique. Cet adressage physique est nécessaire lorsqu'on veut gérer l'agent par des ordonnanceurs ou des observateurs.

4.5.3 Les agents ordonnanceurs et les agents activateurs

Un agent ordonnanceur (`Scheduler`) gère des agents activateurs. Il appartient à la classe `Agent` et il est donc threadé.

Un agent activateur (`Activator`) va permettre de définir les actions devant être ordonnancées. Il est lié à un groupe et à un rôle. Il possède une méthode `update()` qui permet de connaître, à un moment donné, tous les agents jouant un certain rôle dans un groupe.

4.5.4 Les agents observateurs et les agents sondes

Un agent observateur (`Watcher`) va gérer un ensemble de sondes (`probe`) qui vont extraire des informations des agents appartenant à un groupe et jouant un rôle donné : par exemple, des données membres publiques. Les agents sondes possèdent une méthode `update()` comme les agents activateurs.

4.5.5 Des messages non bloquants

Les transmissions de messages se font entre les agents qui gèrent chacun leur propre boîte aux lettres. On utilise les différentes méthodes suivantes :

- `sendMessage (AgentAdress, message)` : envoi vers un destinataire précis ;
- `broadcastMessage (group, role, message)` : envoi diffusé à l'ensemble des agents d'un groupe ;
- `isMessageBoxEmpty()` prend la valeur `false` si un message non lu est dans la boîte aux lettres ;

- `nextMessage ()` renvoie le premier message de la boîte aux lettres en le retirant ;
- `getSender ()` renvoie l'AgentAdress de l'expéditeur du message ;
- `waitNextMessage ()` : réception bloquante des messages (agent en attente jusqu'à réception d'un message dans sa boîte aux lettres) mais **ne peut être utilisé que par des agents threadés du type Agent et pas par des agents du type AbstractAgent non threadés** .

Références bibliographiques complémentaires

- P. Coquillard et D.R.C. Hill "*Modélisation et simulation d'écosystèmes*", Masson.
- J. Ferber "*Les systèmes multi-agents*", InterEdition.
- F. Blasco coord. "*Tendances nouvelles en modélisation pour l'environnement*", Elsevier.
- D.L. DeAngelis et L.J. Gross Ed. "*Individual-based models and approaches in Ecology*", Chapman & Hall.