

**REPUBLIQUE DU SENEGAL**



**Un Peuple – Un But – Une Foi**

-----

**MINISTRE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE**

**Université Gaston Berger de Saint-Louis**



-----

**U.F.R DE SCIENCES APPLIQUEES ET DE TECHNOLOGIE**

**SECTION D'INFORMATIQUE**

-----

Mémoire recherche

***Sujet : Un modèle informatique multi-thématique des conditions d'invasion d'une espèce de rongeur; le cas de la gerbille nigériane (Gerbillus nigeriae) au nord du Sénégal***

*de*

**Moussa SALL**

Sous l'encadrement de :

**M. Jean Le Fur**, Chercheur, Institut de Recherche pour le Développement  
**M. Jean Marie Dembele**, Enseignant-chercheur, Section d'informatique UFR SAT

Année académique : 2015/2016

N° d'ordre :

## Remerciements

Alhamdoulilahi Rabil halamine !

J'adresse mes sincères remerciements à:

Mes Parents qui n'ont ménagé aucun effort pour me mettre dans les bonnes conditions de réussite,

mon maître de stage, M. Jean Le Fur pour ses longues séances d'explications et de travail à deux pour ma bonne et rapide compréhension, pour son amour pour le travail accompli et surtout pour sa patience.

mon encadreur à l'université Gaston Berger de Saint-Louis, M. Jean Marie Dembele pour sa disponibilité, sa participation incommensurable à la bonne tenue de mon mémoire et ses remarques pertinentes,

M. Pape Adama Mboup, pour son soutien continu et la bonne ambiance durant tout le stage,

M. Laurent Granjon, pour les moments agréables de partage de connaissance et pour m'avoir permis de participer à la conférence des étudiants de Baillarguet 2017,

A M. Christophe Diagne alias My Lord pour son soutien continu et la bonne ambiance au labo,

à tous les doctorants, stagiaires et techniciens du laboratoire Biopass,

à l'ensemble du corps enseignant et administratif de l'UGB,

ainsi qu'à toutes les personnes qui ont participé de près ou de loin à la réalisation de ce travail.

**Mots clés:** Modèle à base d'agents, architecture PDE, comportement invasif, *Gerbillus nigeriae* (gerbille nigériane), chouette effraie.

## Résumé :

Les rongeurs constituent l'un des ordres de mammifères les plus répandus sur terre. Ils peuvent être vecteurs de nombreux pathogènes zoonotiques et causent des dégâts importants aux cultures et denrées stockées. Dans le cadre du projet CERISE<sup>1</sup>, nous nous sommes intéressés à l'invasion du Sénégal par la gerbille nigériane. Cette espèce de rongeur, arrivée au Nord du Sénégal très probablement au milieu des années 1990, est un déprédateur important des cultures de mil et de sorgho qui étend sa distribution vers le sud du Sénégal. Des travaux méthodologiques ont été coordonnés sur cette question à plusieurs échelles : ensemble de la bande sahélienne, Afrique de l'Ouest et Sénégal. Notre travail se propose d'utiliser la modélisation à base d'agents pour l'étude intégrée des processus effectifs de colonisation de la gerbille nigériane à l'échelle de l'individu et de son habitat. Après avoir mis en place l'environnement durant notre master professionnel, nous avons documenté puis tenté de représenter de façon intégrée, les processus biologiques (reproduction), et trophiques (prédation, alimentation) ainsi que les processus de décision des agents qui intègrent ces composantes associés à cette invasion actuelle. L'architecture basée sur le paradigme Perception-Délibération-Exécution (PDE) a été utilisée pour implémenter les processus de décision des agents. Les résultats obtenus concernent (i) le comportement, les traits de vie spécifiques et (ii) les divers types d'interactions des gerbilles avec leurs prédateurs, notamment la chouette effraie (*Tyto alba*), ou leurs aliments (végétation). Des composants du simulateur seront présentés et discutés. Ce modèle permet d'apporter des éléments de compréhension, à cette échelle, sur les processus qui sont susceptibles de favoriser ou non l'invasion de la gerbille au Sénégal. Il s'intègre aux autres approches du projet réalisé à d'autres échelles.

---

<sup>1</sup> CERISE : Scénarios d'invasion de rongeurs au Sahel : Impact des changements globaux sur l'expansion de la gerbille nigériane et la souris domestique au Sénégal

**Key-words:** agent-based model, PDE architecture, invasive behavior, *Gerbillus nigeriae* (nigerian gerbil), *Tyto alba* (barn owl).

### **Abstract:**

Rodents distributed worldwide can be vectors of various zoonotic pathogens and are considered as major pests for materials and crops. We focused on the Senegal invasion by the Nigerian gerbil, depredator of millet and sorghum crops. As a part of the CERISE project, we focused on the Nigerian gerbil (*Gerbillus nigeriae*) currently invading North Senegal since the 1990s. This rodent species, known as important pest for millet and sorghum crops, is currently expanding its distribution area southwards in Senegal. Methodological works have been conducted in this question at different scales: the sahelian eco-region, West Africa and Senegal. In this work, we used agent-based modelling for an integrative study of effective processes to the individual scale and its habitation. After works on the environment during our professional master, we documented then attempted to represent in an integrated way, the biologic (reproduction), trophic (predation, food) as well as decisional processes of the agents integrating these components. Architecture based in Perception-Deliberation-Execution (PDE) paradigm is used to implement decision agents process. Results concern (i) the behavior, life traits and, (ii) the various interaction types between gerbils and their predators, especially the barn owl (*Tyto alba*), or their foods (vegetation). Aspects of model's simulations will be presented and discussed. This model provides elements of understanding, at this scale, of the processes that are likely to favor or not the gerbil invasion of Senegal.

## Table des Matières

Remerciements .....	1
Résumé : .....	2
Table des Matières .....	4
Abréviations .....	6
Liste des Figures.....	7
INTRODUCTION.....	8
CHAPITRE 1: Présentation générale et état de l’art.....	10
I- Présentation Générale .....	10
I.1- Problématique.....	10
I.2- Contexte .....	12
I.3- Objectifs .....	15
II- Etat de l’art .....	15
II.1- Etat des lieux .....	15
II.2- Modèle informatique multi-thématique .....	24
II.3- Les systèmes multi-agents .....	25
CHAPITRE 2: Modélisation .....	32
I- Protocole de la gerbille .....	32
II- Formalisation des agents du système.....	34
II.1- Gerbille .....	35
II.2- Chouette effraie.....	39
III- Mise à jour de l’environnement de simulation.....	43
III.1- Chronogramme de simulation.....	43
III.2- Gestion des steps .....	44
CHAPITRE 3 : Résultats et Discussion .....	46
I- Présentation d’un premier résultat .....	46

II- Discussion.....	47
CONCLUSION .....	48
REFERENCES.....	50
ANNEXE : codes source.....	56
C_ProtocolGerbil :.....	56
C_RodentGerbil :.....	63
C_BarnOwl : .....	67

## Abréviations

**ADN** : Acide DésoxyriboNucléique

**BDI** : Belief-Desire-Intention

**CERISE** : sCEnario of Rodent Invasion in the SahEl

**CBGP** : Centre de Biologie pour la Gestion des Populations

**FIFO** : First In First Out

**IGN** : Institut National de l'information Géographique et Forestière

**IRD** : Institut de Recherche pour le Développement

**M2ITIC** : Master 2 d'Ingénierie Informatique et Technologies de l'Information et de la Communication

**SQD** : Système Quasi Décomposable

**PDE** : Perception-Délibération-Exécution

**UGB** : Université Gaston Berger

**UICN** : Union Internationale pour la Conservation de la Nature

**UML** : Unified Modeling Language

**SMA** : Systèmes Multi-Agents

**TRMM3B42** : Tropical Rainfall Measuring Mission 3B42 version

**CCL** : Center for Connected Learning

**CEAMITIC** : Centre d'Excellence Africain en Mathématiques Informatique et TIC

## Liste des Figures

Figure 1 : Gerbille nigériane ( <i>Gerbillus nigeriae</i> ) photo. ....	11
Figure 2 : Chouette effraie, <i>Tyto alba</i> : (Photo Jean-Marc Duplantier / IRD) .....	12
Figure 3 : Quelques cas d'utilisation intégrés dans SimMasto. ....	14
Figure 4 : Deux exemples de simulations d'études de cas dans SimMasto. ....	16
Figure 5 : Diagramme de classe basé sur UML des agents du modèle, .....	18
Figure 6 : Diagramme de classe basé sur UML de la partie génétique du modèle .....	19
Figure 7 : Diagramme de classe basé sur UML des types d'organisation du sol, .....	20
Figure 8 : Schéma de conversion de coordonnées géodésiques en coordonnées discrètes de la grille ; .....	22
Figure 9 : Exemples de zones d'intérêt sur la petite emprise (source: Modèle de l'approche petite emprise JLF 12-11-2015) .....	22
Figure 10 : Simulation de la pluviométrie de l'année 1999 (Sall, 2016).....	23
Figure 11 : Capture d'écran de la simulation de l'évolution de la pluviométrie en fonction de la quantité de pluie .....	24
Figure 12 : Architecture Croyance-Désire-Intension d'un agent (adapté de Taillandier <i>et al</i> , 2012).....	29
Figure 13 : Architecture perception-délibération-exécution d'un agent (adapté de Ferber, 1995).....	30
Figure 14 : Diagramme de séquence du processus d'exécution du protocole gerbille .....	34
Figure 15 : les cas de figure possible entre le disque de perception de la gerbille et celui de la couverture végétale, .....	36
Figure 16 : pseudo code de la délibération de la gerbille .....	37
Figure 17 : diagramme de séquence du processus de décision de la gerbille .....	38
Figure 18 : pseudo code de la délibération de la chouette.....	40
Figure 19 : diagramme de séquence des processus de décision de la chouette.....	41
Figure 20 : Processus de décision d'un agent de gerbille, .....	42
Figure 21 : capture d'écran de la simulation d'agents gerbilles et chouettes évoluant dans leur environnement .....	46



## INTRODUCTION

Fortement liées aux effets de la mondialisation et ceux des changements climatiques, les invasions biologiques constituent l'une des sources importantes de dommages environnementaux et économiques (Mooney, 1999, Lowe *et al*, 2000). Différentes études ont été soutenues dans le monde afin de comprendre et d'appréhender ce phénomène. Ces études fournissent aux chercheurs des connaissances sur tous les niveaux de la vie, gènes, cellules, organismes, populations et communautés, et jusqu'au niveau de l'écosystème. C'est dans ce cadre que le projet CERISE, centré sur l'étude de l'impact des changements globaux sur l'expansion de la gerbille nigériane et de la souris domestique au Sénégal, a vu le jour. Il a été proposé dans ce projet, la mise en place d'outil de modélisations différentes (modèles bayésiens, modèles multi-agents) en vue de produire des scénarios d'expansion d'espèces invasives.

Le travail que présente ce rapport s'articule autour du modèle multi-agents sur les conditions d'invasion de la gerbille nigériane au nord du Sénégal proposé dans cadre du projet CERISE. Après un travail préliminaire sur la modélisation des processus environnementaux (pluies) et édaphiques (couverture du sol) du modèle, durant notre stage de master professionnel (Sall, 2016), nous avons poursuivi avec la modélisation des processus biologiques (reproduction), trophiques (prédation, alimentation) ainsi que particulièrement les processus de décision de la gerbille nigériane et de son prédateur la chouette effraie sur une petite échelle spatiale (grande échelle en géographie) caractéristique de la zone sahélienne du Sénégal.

La prise en considération de ces caractéristiques dans un modèle multi-agents nécessite une appréhension de l'aspect multidimensionnel des éléments et de leurs interactions (Meek *et al*, 2001) et, conséquemment, de mettre en œuvre une collaboration interdisciplinaire (Wheeler, 2007). C'est dans ce cas de figure que nous proposons un modèle multi-agents multi-thématique basé sur le paradigme Perception-Délibération-Exécution (PDE), développé avec le langage Java et intégré dans un modèle général des connaissances sur la coévolution bioécologique des petits rongeurs et de leur parasites (projet SimMasto, Le Fur *et al*, 2017).

Le chapitre 1 de ce rapport propose une présentation générale dans laquelle nous définissons la problématique, le contexte et l'état de l'art. Le chapitre 2, centré sur la modélisation informatique sera consacré à la présentation du protocole lié à l'étude de cas considérée, la formalisation des agents ainsi que les aspects de mise à jour de l'environnement. Le troisième

et dernier chapitre présentera et discutera avant de conclure sur les perspectives qu'offre ce travail.

# CHAPITRE 1: Présentation générale et état de l'art

Cette première partie du document aborde en premier la présentation du contexte de ce travail, la problématique posée et les objectifs fixés. Ensuite, nous passerons à la description de l'existant avant l'adjonction de notre travail et à la revue générale des connaissances sur le thème abordé.

## I- Présentation Générale

### I.1- Problématique

Les effets de la mondialisation combinés à ceux liés aux changements climatiques ont entraîné une multiplication des cas d'espèces invasives à l'échelle de la planète (Granjon, 2013). Avec la sécheresse de 1972 survenue au Sénégal, une progression de l'aridification vers le sud sous l'action d'une pluviométrie déficiente et d'une déforestation intense a été notée (Bille, 1974). Entraînant ainsi l'apparition de nouvelles espèces invasives et la rareté ou disparition des espèces natives (Duplantier *et al*, 1991).

#### I.1.1- Invasion biologique

D'après l'UICN<sup>2</sup>, l'invasion biologique constitue la deuxième cause d'extinction d'espèces dans le monde, avec des estimations de 30% des oiseaux, 15% des plantes, 11% des amphibiens et 8% des mammifères inscrits dans la liste des catégories d'espèces menacées. Documenté par de multiples recherches et soutenu par les grands programmes nationaux et internationaux, le problème des invasions biologiques constitue une préoccupation récurrente des sciences de l'environnement (Lowe *et al*, 2000). Dans le passé lointain, les océans et les montagnes représentaient des barrières naturelles infranchissables aux déplacements. Mais, avec l'augmentation considérable des voies de communication et l'intensification des trafics routier, ferroviaire, aérien et maritime, la fréquence d'introduction des espèces allogènes dans le monde s'accroît inexorablement, entraînant ainsi des changements importants de l'écosystème d'accueil. Différentes études ont été menées afin de comprendre et appréhender le phénomène d'invasion biologique. C'est dans ce cadre, qu'un projet d'étude des critères d'invasion de la gerbille nigériane au nord du Sénégal a été mis en place.

---

<sup>2</sup> Union internationale pour la conservation de la nature

### I.1.2- Gerbille nigériane

Inféodée aux sols sableux des zones arides et semi-arides, la gerbille nigériane est présente sur toute la zone sahélienne depuis le Tchad jusqu'à la côte Mauritanienne (Granjon et Duplantier, 2009). Elle a été identifiée pour la première fois au Sénégal vers la fin des années 90 (Bâ *et al*, 2006) et est devenue en peu de temps (moins d'une décennie) l'espèce de rongeur dominante dans la région nord du Sénégal (Thiam, 2007), Diagne *et al*, 2016 données non publiées). Elle est réputée potentiellement pullulante au Niger (Nomao et Gautun, 2001) et au Burkina-Faso (Sicard, 1987) avec des densités moyennes de 0 à 150 individus par hectare. La gerbille nigériane est petite de taille 25 à 26g (Thomas et Hinton, 1920) environ et présente une activité exclusivement nocturne. Déprédateur de cultures sèches et insectivore pendant les périodes de disette, elle cause beaucoup de dégâts sur les cultures de mil et de sorgho. Des études d'analyses de pelotes de réjection de la chouette effraie ont montré qu'elle constitue la proie principale de ce rapace sur toute l'étendue du nord du Sénégal.



Figure 1 : Gerbille nigériane (*Gerbillus nigeriae*) photo...

### I.1.3- Chouette effraie

La chouette effraie est l'un des oiseaux terrestres dont l'aire de répartition est la plus étendue. On la trouve sur tous les continents, mis à part l'Antarctique (Vallée, 1999). Avec une charge alaire de  $0.25\text{g/cm}^2$  et une ouïe très développée, elle patrouille face au vent pour ralentir davantage son vol et à faible altitude (1 à 5m) au-dessus de sa surface de chasse. Elle peut patrouiller sur une surface de rayon 2,5 km (Taberlet, 1983) en moyenne par jour et présente une activité principalement nocturne, ajustée à celle de ses proies. La chouette consomme 70 à 105g de proie par jour (Mikkola et Willis, 1983) environ trois (3) à quatre (4) rongeurs en moyenne par jour. Elle régurgite les parties indigestes de ses proies (poils, os, carapaces, ...)

sous formes de boulettes appelées pelotes. L'analyse des pelotes de chouette effraie collectées par Thiam *et al* (2008), entre 1989 et 2015, dans 96 localités situées dans la partie nord du Sénégal, a montré que 68% du menu de la chouette est constitué par des rongeurs. Et sur ce pourcentage, la gerbille nigériane absente durant les années 90, prend le plus grand pourcentage à partir des années 2000.



Figure 2 : Chouette effraie, *Tyto alba* : (Photo Jean-Marc Duplantier / IRD)

## I.2- Contexte

### I.2.1- Cadre de l'étude

La compréhension et le contrôle des écosystèmes a toujours été un enjeu phare des chercheurs sur toute l'étendue de la planète. Cependant, la complexité des systèmes naturels surtout ceux marqués par le vivant (systèmes biologiques, sociaux, écologiques...) constitue un défi pour le génie humain. Ce dernier est, en effet, limité dans ses capacités à appréhender un très grand nombre d'informations simultanées, et à obtenir un résultat maîtrisable de mécanismes co-agissants et interagissant à toutes les échelles (Coquillard *et al*, 1997). Avec l'essor des technologies et des concepts informatiques, d'importants modèles ont été mis en œuvre en vue de formaliser de façon explicite des entités et leurs comportements. C'est dans ce sillage que les acteurs du projet CERISE (sCENARIO of Rodent Invasion in the SahEl) ont proposé de mettre en place des outils de modélisation des processus d'invasion de rongeurs dans le Sahel.

### I.2.2- Projet CERISE

CERISE est un projet sur les scénarios d'invasion de rongeurs au Sahel. Il est orienté sur deux axes d'études que sont (i) les impacts de la progression récente de la souris domestique d'Ouest en Est, suivant le réseau des agglomérations, les infrastructures et les transports en développement, et (ii) la progression récente de la gerbille au Sénégal, du Nord vers le Sud avec l'aridification du climat et/ou l'anthropisation du milieu. Pour apporter une touche

innovante au projet, différents outils de modélisation ont été mis en œuvre en vue de produire des scénarios d'expansion d'espèces invasives. Ainsi, nous avons noté :

Pour la souris domestique :

- ✓ un modèle bayésien de niche écologique sur le Sénégal (Leyli Borner *et coll.*),
- ✓ un modèle multi-agents du transport au Sénégal (Jean Le Fur *et coll.*)

Pour la gerbille nigériane:

- ✓ un modèle bayésien de niche écologique à grande emprise sur le Sahel (Leyli Borner *et coll.*),
- ✓ un modèle multi-agents populationnel sur la moyenne emprise sur l'Afrique de l'Ouest (Françoise Durafour *et coll.*),
- ✓ et le modèle multi-agents populationnel individu centré sur une petite échelle (Moussa Sall *et coll.*).

La plate-forme SimMasto a été choisie pour l'implémentation du modèle multi-agents du transport au Sénégal sur la souris domestique et le modèle individu centré des populations de gerbilles sur la petite emprise.

### I.2.3- Le projet SimMasto

SimMasto est un projet expérimental de plate-forme de simulation développé par Jean Le Fur et son équipe et qui vise à être un centre de connaissances dynamiques sur la coévolution bioécologique des petits rongeurs et de leurs parasites (Le Fur *et al.*, 2017). Il est hébergé par le laboratoire CBGP<sup>3</sup> et est développé sous Repast Symphony. L'objectif de SimMasto est de participer à une meilleure compréhension de l'état, du rôle et du destin de ces nuisibles en coévolution. Il vise à élaborer un modèle permettant une représentation intégrée des connaissances sur ces nuisibles couvrant de nombreux niveaux fonctionnels, de l'ADN aux écosystèmes, ainsi que de multiples échelles d'observation. SimMasto est conçu avec un souci de robustesse et de flexibilité ce qui lui procure la possibilité de réunir plusieurs cas de simulation comme (Figure 1) : les processus d'hybridation, de capture marquage recapture dans l'étude de la dynamique d'une population de rongeurs, de diffusion et de propagation multi-échelles du rat-noir, ..., et pour ce travail, la mise en place d'un modèle informatique multi-thématique des conditions d'invasion d'une espèce de rongeur ; le cas de la gerbille

---

<sup>3</sup> Centre de Biologie pour la Gestion des Populations

nigériane (*Gerbillus nigeriae*) au nord du Sénégal. La validité de SimMasto réside dans son aptitude à intégrer toute problématique du domaine, quelle que soit sa dimension spatiale, temporelle ou fonctionnelle et quelle que soit son actualité, autrement dit qu'elle corresponde à des problématiques définies il y a dix ans, de nos jours ou dans dix ans.

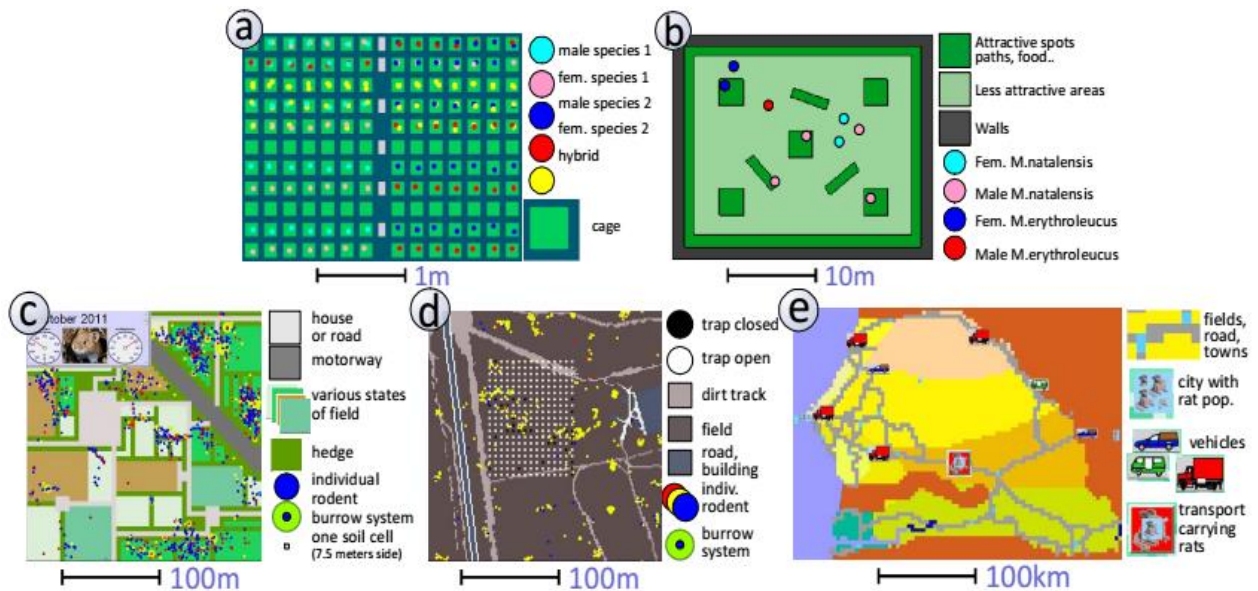


Figure 3 : Quelques cas d'utilisation intégrés dans SimMasto. (a) expériences d'hybridation en cage et (b) enclos sur des espèces jumelles, (c) population de campagnols évoluant dans un paysage agricole, (d) expérience de capture-marquage-recapture dans une réserve africaine, (e) épidémiologie et transport du rat noir par des véhicules commerciaux au Sénégal (Afrique de l'ouest). (source : Mboup, 2015)

Afin de représenter indifféremment tout type de connaissance un schéma modulaire a été conçu comportant (i) les objets des mondes étudiés, du gène au paysage, (ii) un système minimaliste d'intégration des connaissances fondé sur la notion générique d'événements applicables indistinctement aux différents types de savoirs, (iii) un système épiphyte<sup>4</sup> permettant d'observer les systèmes sans interférer sur leur fonctionnement ainsi que (iv) un ensemble de protocoles permettant de distinguer chaque étude de cas. Avec un schéma basé sur une architecture trois tiers (data, business, présentation), chaque étude de cas modélisée apporte de nouveaux objets, un (des) environnement(s) et processus réclamés par le contexte particulier de chacune des études à représenter et bénéficie des briques déjà existantes qui peuvent être réutilisées et améliorées (Le Fur *et al*, 2017).

<sup>4</sup> Un système épiphyte permet d'observer les systèmes sans interférer sur leur fonctionnement (dans SimMasto, le système épiphyte se charge de recenser et de restituer, au besoin et de façon multimodale, les données produites par une simulation)

### I.3- Objectifs

L'enjeu principal de ce travail est d'apporter des éléments pour la compréhension de l'invasion de la gerbille par la modélisation multi-agents. En effet, la modélisation simplifie et reproduit le comportement du réel le mieux possible en fonction des objectifs fixés pour le cadre de l'étude. Ainsi, l'enjeu fixé doit se matérialiser par un travail de simulation du comportement individuel de la gerbille et de son prédateur sur une petite échelle spatiale (grande échelle en géographie) qui est la zone sahélienne du Sénégal. Cette approche permet de décrire à un niveau microscopique des phénomènes intervenant localement entre plusieurs individus pour observer le résultat émergent (à un niveau macroscopique) de ces interactions (Grimm, 2012).

## II- Etat de l'art

### II.1- Etat des lieux

Fondé sur une approche incrémentale (Le Fur *et al*, 2017), la plate-forme de simulation SimMasto a pour finalité d'élaborer un modèle permettant d'intégrer les connaissances sur les rongeurs. Ces connaissances couvrent de nombreux niveaux fonctionnels allant de l'ADN aux écosystèmes, ainsi que de multiples échelles d'observation. Avec ce modèle d'architecture, des études de cas contrastées sont successivement représentées au sein de la plate-forme. Chaque étude enrichit ainsi la plate-forme avec de nouveaux types d'objets et bénéficie des développements antérieurs. La contrainte de robustesse qui fonde la validité de l'approche nécessite que le modèle général soit compatible simultanément avec toutes les études de cas représentées. Le passage d'une étude à l'autre est accompagné par un retour sur la structure générale obtenue. En effet, une révision du code à travers trois types d'opérations est effectuée. (i) Une factorisation de méthodes partagées est réalisée en première phase avec l'exploitation du système des classes abstraites et des interfaces ainsi que le polymorphisme et l'héritage offerts par l'approche objet. (ii) Avec l'émergence de structures communes, un processus de réagencement est effectué sur les paquetages. Ce qui se traduit par un ré-usinage des noms de méthodes pour la clarification de la logique en construction, le contrôle des unités de temps et d'espace qui doivent être partagées. Enfin, des ajustements sont menés pour assurer la rétrocompatibilité du code.

Dans ce qui suit, nous présentons une brève description de la plate-forme (avec l'aide de diagrammes de classe) avant l'intégration de notre travail suivie d'un résumé de ce qui a été réalisé en préliminaire lors de mon master professionnalisant.



### II.1.1- Présentation de l'existant

A mon arrivée, sept études de cas ont été implémentées sur SimMasto en collaboration avec les chercheurs de terrain titulaires de la réalisation et des résultats de l'étude concrète modélisée. Ces études de cas modélisées sont :

- ✓ la dynamique des campagnols dans un paysage agricole avec la considération des notions de paysages, d'itinéraire technique, de rotation de culture, d'enfouissement et de comportement social reproducteur (Figure 4),
- ✓ l'expérimentation d'espèces jumelles de rongeurs africains en cage et en enclos. Ce qui a permis l'implémentation du niveau chromosomique, du gène jusqu'aux processus de méiose et de fécondation en passant par la réification des chromatides, des chromosomes et des génomes,
- ✓ l'expérience de capture-marquage-recapture dans l'étude de la dynamique d'une population de rongeurs dans la réserve de Bandia (Sénégal) avec la formalisation du dispositif de piégeage et sa mise en œuvre,
- ✓ trois études sur le transport de rongeurs commensaux au Sénégal avec la prise en compte de plusieurs échelles spatiales, réseaux routiers, véhicules de transports, transporteurs, villes, marchés, zones économiques, etc. Mais aussi la mise au point d'une fonctionnalité permettant de réaliser les simulations à toutes les échelles temporaires de la seconde à l'année,
- ✓ la dispersion de souris au sein de pièces d'habitations villageoises avec l'introduction de rythmes d'activité circadiens.

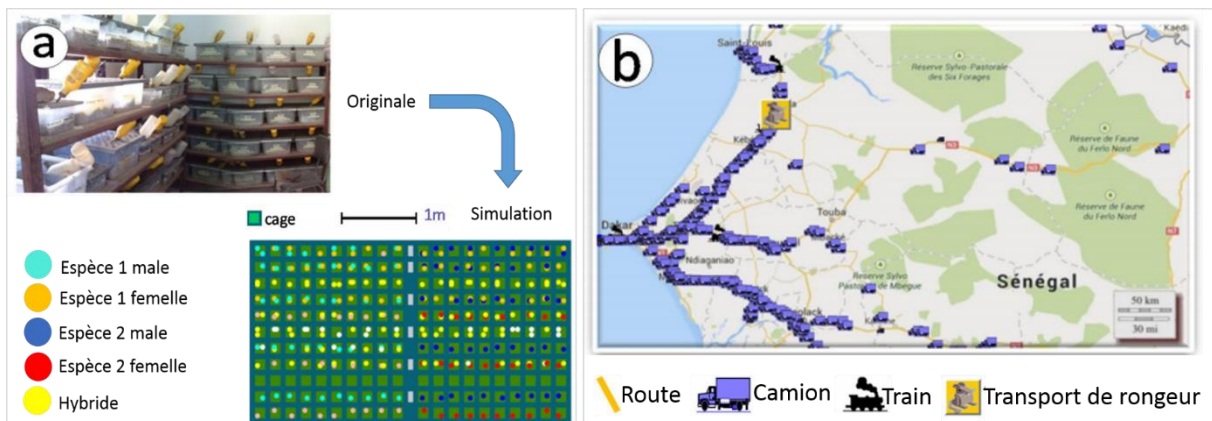


Figure 4 : Deux exemples de simulations d'études de cas dans SimMasto.

**a)** Visualisation de la simulation sur l'expérimentation d'espèces jumelles de rongeurs en cage. Le challenge ici est d'étudier les limites de fertilisation liées aux différences chromosomiques entre deux espèces.

**b)** Simulation de la colonisation du Sénégal par le rat noir (source : Le Fur *et al*, 2017).

La compilation de ces études de cas a favorisé l'émergence d'un modèle à trois domaines de diversités. Ces domaines de diversités sont :

- ✓ les entités concrètes simulées,
- ✓ l'association de génome avec les organismes vivants,
- ✓ les différents types de matières (substrat) dans lesquels les objets et les agents peuvent être localisés.

Les entités concrètes simulées déterminent l'arbre principal du modèle et a pour objectif de décrire les agents qui peuvent intervenir dans celui-ci et qui sont appelés à être des systèmes quasi décomposables (SQD). Ces agents sont identifiés dès leur existence (c'est-à-dire nommés) et vivent selon une durée qui est formalisée par une méthode de vieillissement. Ils portent un génome présentant des valeurs transcrites en propriétés ou paramètres leur permettant de mener à bien leurs actions. Ces génomes leur procurent la possibilité de se reproduire avec d'autres organismes de la même espèce et de transmettre une partie de leur ADN à leurs descendants. Toute fonction identifiée au sein d'une classe de ce domaine a été adaptée systématiquement à une justification biologique : les animaux peuvent se reproduire, les amniotes peuvent porter des zygotes, les rongeurs fouisseurs creusent des terriers, et ainsi de suite.

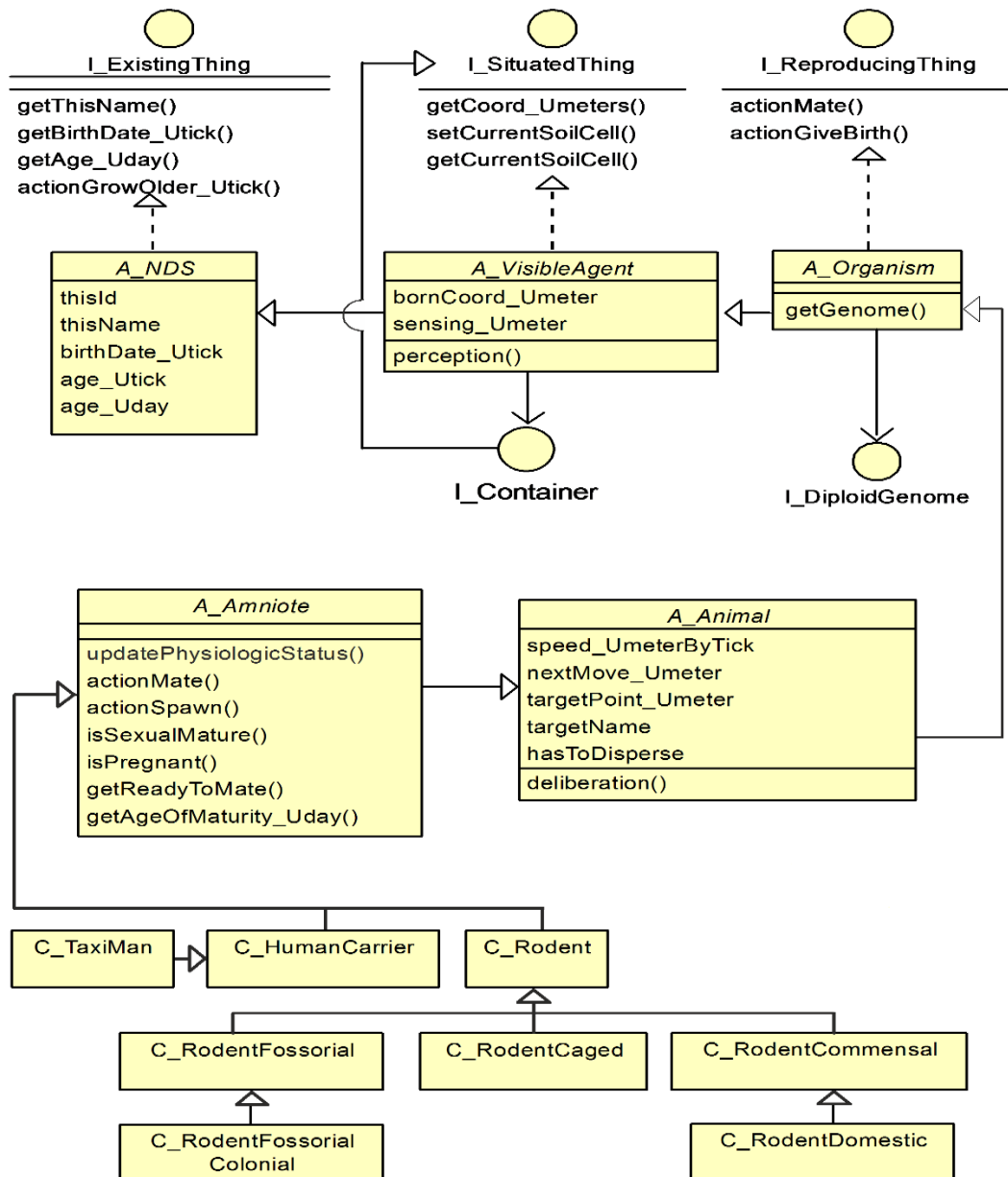


Figure 5 : Diagramme de classe basé sur UML des agents du modèle, obtenu avec les études de cas et conçu selon les classifications naturelles. Pour des raisons de clarté, seules les méthodes, les propriétés ou les relations pertinentes sont présentés. Légende : A\_ : classe abstraite, I\_ : interface, C\_ : classe, \_Uxxx : unité de la méthode ou du propriété, NDS : système quasi décomposable (Simon, 1962), (source : Le Fur *et al.*, 2017).

L'association de génome avec les organismes vivants fait intervenir dans la plate-forme tous les aspects liés à la génétique des organismes vivants. Il comprend les éléments mécaniques (gènes, allèles, etc.) requis pour les opérations génomiques tels que la méiose, la ségrégation, la fertilisation, la mutation et la recombinaison. La nature de la composition des gènes, chromosomes, paires de chromosomes et génome est adaptée du travail de Shaw et Wagner (2008) sur les génomes acridiens. Ainsi, diverses paires de chromosomes sont agrégées sur chaque génome et elles peuvent être recombinées et héritées en partie, pendant la reproduction. L'exploitation de l'analogie entre l'approche biologique (phylogénie) et la

programmation objets (héritage) sur la plate-forme SimMasto a permis de mettre en place les caractéristiques génétiques des agents comme la vitesse de déplacement, la taille de la portée, l'âge de la maturité sexuelle, etc.

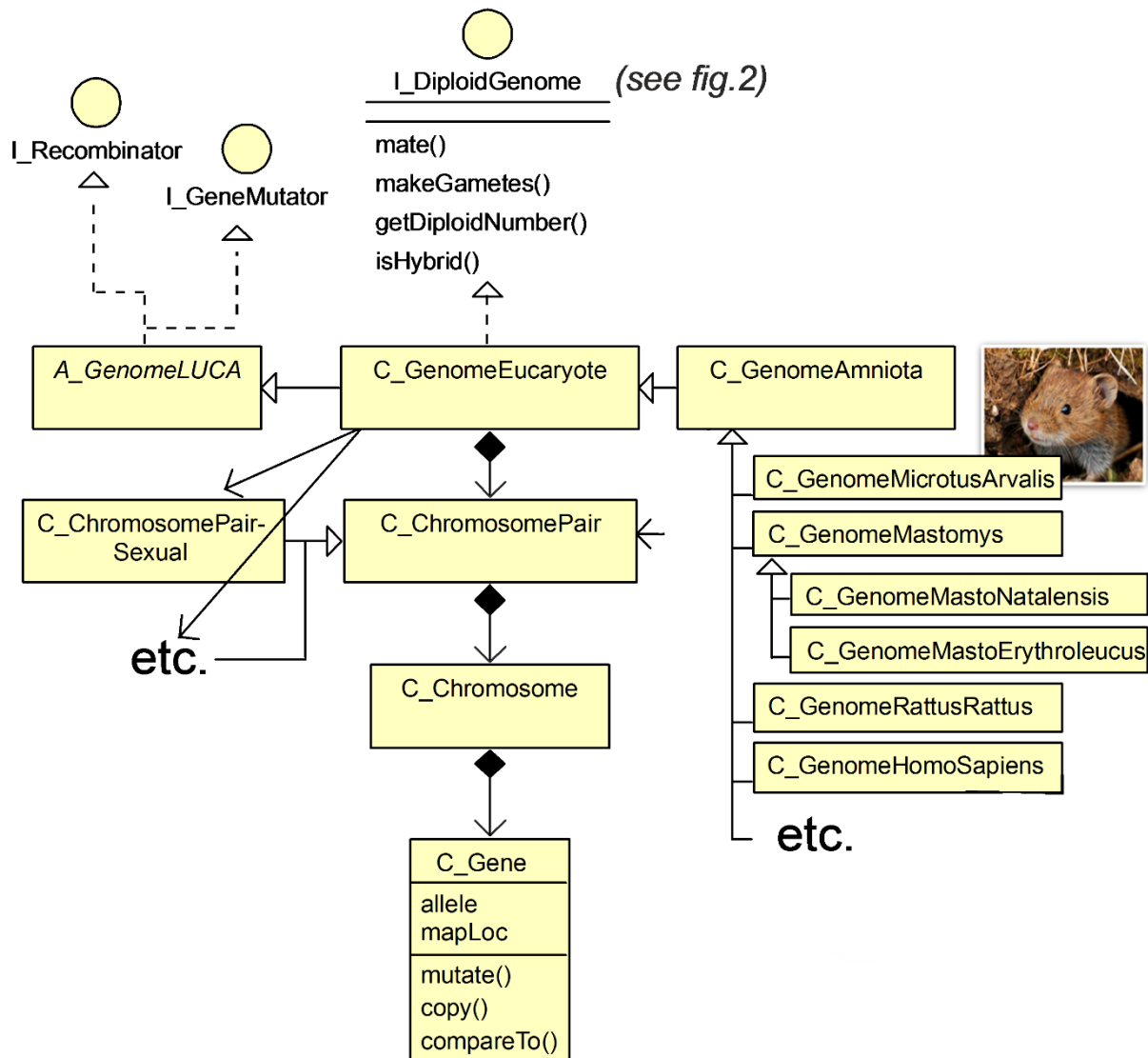


Figure 6 : Diagramme de classe basé sur UML de la partie génétique du modèle avant agrégation de mon travail, tel que obtenu à partir des études de cas. Gauche: structure génétique permettant la transmission du patrimoine génétique d'un agent simulé à sa «progéniture» après accouplement. La structure de l'arbre sur la partie du bas ressemble à une base de connaissances pour l'instanciation, dans le schéma général, de tout type d'espèce (rongeur, prédateur, homme, etc.) et ses caractéristiques (espérance de vie, âge de la maturité sexuelle, etc.). (source : Le Fur *et al*, 2017).

Dans SimMasto, les différents types de matières (substrat) désignent l'espace d'évolution des agents. En effet, avec la mise en place du système de 'conteneur', chaque agent est situé dans le modèle sur un point précis de l'espace. Les conteneurs (cellules, terriers, nids, véhicules, etc.) sont définis comme un système récursif, car pouvant eux-mêmes contenir des conteneurs. Cet aspect de récursivité permet de déterminer la liste des agents dans un conteneur et/ou récursivement dans les conteneurs qu'il contient. Les conteneurs sont des

agents visibles et, comme NDS, ils ont un âge et une capacité à vieillir, ou, en d'autres termes, à changer au fil du temps selon les spécifications.

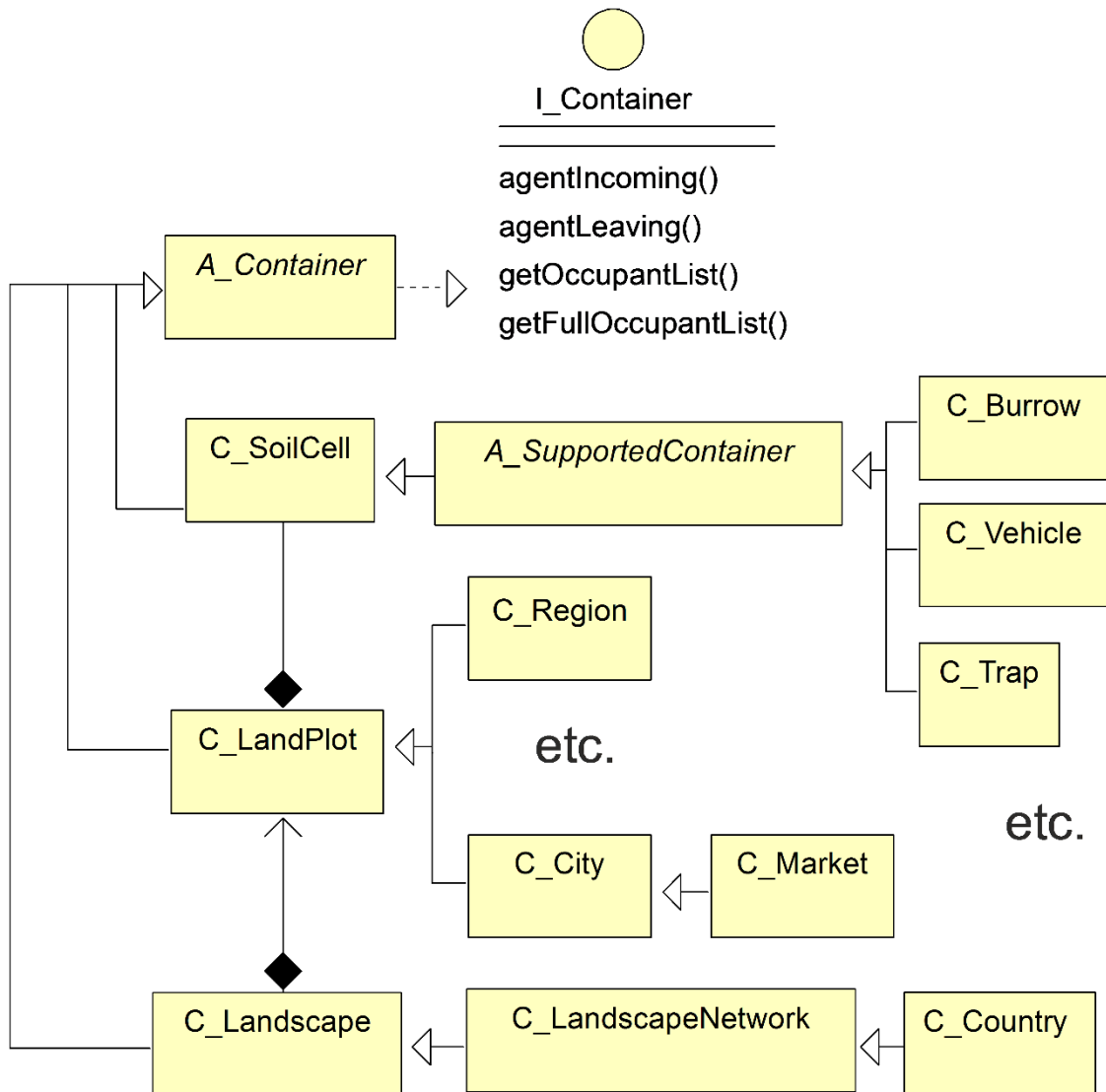


Figure 7 : Diagramme de classe basé sur UML des types d'organisation du sol, déterminé à partir des études de cas agrégées. Chaque instance de cette hiérarchie partage les fonctionnalités des conteneurs. On distingue trois niveaux de composition : paysage, terrain, cellule. Chaque niveau de la composition est lui-même la racine d'une hiérarchie qui peut être étendue (à droite) pour décrire diverses fonctionnalités des mondes simulés. (Source : Le Fur *et al*, 2017).

L'approche utilisée, fondée sur l'articulation incrémentale d'études de cas contrastés dans une même structure formelle, a fait que la plate-forme SimMasto était déjà assez robuste (après 7 études de cas) pour représenter les gerbilles (déplacement, reproduction,...) et leur terrier. Cependant, elle présentait des limites pour la représentation des phénomènes liés à notre sujet. Ces limites concernaient :

1. la simulation de l'évolution de la pluviométrie,
2. l'extraction de zone de questionnement (zooms) dans le modèle initial,

3. la prise en compte des types de végétations dans le modèle,
4. la mise en relation de l'évolution de la végétation par rapport à la quantité de pluies disponible,
5. la prédation,
6. la prise en compte des phénomènes naturels déclencheurs de la saison de reproduction,
7. la gestion de l'activité des agents suivant leur rythme circadien
8. l'utilisation des coordonnées géodésiques dans le modèle, par exemple : positionner les agents suivant les points de piégeage.

Cet ainsi qu'un travail de mise en place de l'environnement sahélien avec la prise en compte de la pluviométrie, des types de végétation et des coordonnées géographiques a été effectué durant mon master professionnel. Le travail réalisé durant cette période est présenté dans la partie II.1.2- .

### II.1.2- Espace sahélien

La modélisation de l'évolution de la population de gerbille sur le nord du Sénégal vise à lier de façon fonctionnelle et dynamique les populations de rongeurs, la couverture du sol et les données pluviométriques aux données obtenues *in situ* par piégeage ou analyse de pelotes de chouettes. Durant mon précédent master, j'ai travaillé sur la formalisation et la simulation du milieu sahélien dynamique propice à l'invasion et à la survie de la gerbille nigériane. Ce travail a été marqué par la mise en place d'un convertisseur de coordonnées géodésiques, d'un outil permettant d'extraire à la demande une zone de questionnement fixée par les biologistes, la simulation de l'évolution de la pluviométrie et la mise en relation de la pluviométrie avec la croissance de la végétation.

#### a. Conversion de coordonnées géodésiques

La prise en compte des données fournies par les différents thématiciens du projet CERISE sur SimMasto nécessite de convertir les coordonnées géographiques en coordonnées discrètes de la grille. En effet, utilisées couramment dans le monde de la recherche, les coordonnées géographiques permettent notamment de repérer les points de capture de rongeurs et de ramassage de pelotes de chouette. Ces données de capture et de ramassage sont exploitées par la plate-forme SimMasto pour comprendre le scénario d'invasion de la gerbille au Sénégal. Deux méthodes différentes ont été utilisées pour effectuer la conversion et vérifier la validité des algorithmes. La méthode de la transformation des coordonnées géodésiques de Clarke

(IGN/SGN, 2013) a été utilisée pour la conversion de coordonnées géodésiques en coordonnées discrètes de la grille et la méthode de la projection orthographique pour la conversion inverse. (voir M2ITIC-UGB 2015/2016, (Sall, 2016))

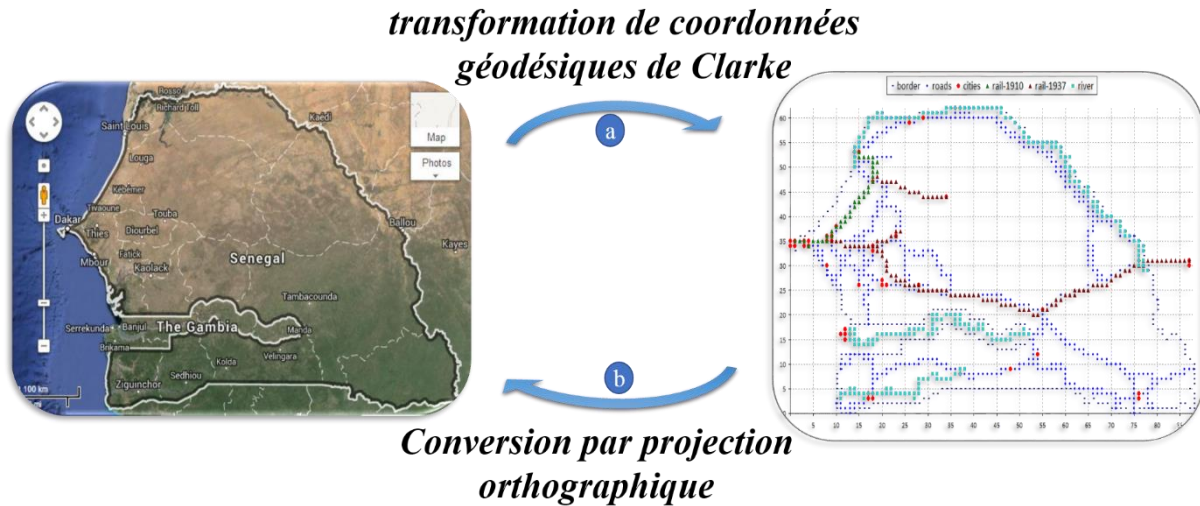


Figure 8 : Schéma de conversion de coordonnées géodésiques en coordonnées discrètes de la grille ;

a) conversion des coordonnées géographiques en coordonnées discrètes de la grille en utilisant la transformation géodésique de coordonnées de Clarke (IGN/SGN, 2013). b) conversion de coordonnées discrètes de la grille en coordonnées géographiques avec la méthode de conversion par projection orthographique (Snyder, 1993)

#### b. Extraction de zoom

La démarche de compréhension et de contrôle d'une invasion biologique est toujours accompagnée par des questionnements auxquels les chercheurs tentent de répondre. Dans le cadre de notre travail durant le master professionnel, les acteurs du projet nous ont proposé de mettre en place un outil permettant d'extraire des zones de questionnement à la demande. Ainsi, nous avons élaboré un algorithme qui permet d'extraire un zoom à partir de fichier texte, mais aussi de fichier bitmap sachant qu'une origine et un nombre de cellules sont fixés au préalable.

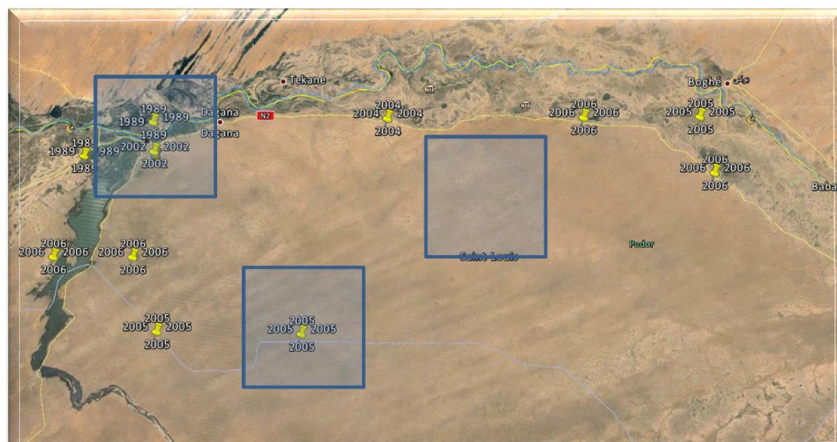


Figure 9 : Exemples de zones d'intérêt sur la petite emprise (source: Modèle de l'approche petite emprise JLF 12-11-2015)

### c. Evolution de la pluviométrie

La quantité de biomasse de la couche herbacée observée est souvent proportionnelle à la quantité d'eau présente dans le milieu. En effet, la pluviométrie constitue l'un des facteurs déterminants de la croissance de la végétation. Ainsi, il devient nécessaire de la prendre en compte dans tout modèle de simulation de la végétation. Pour les besoins de notre modèle, nous avons utilisé des données satellites du produit TRMM3B42 (Huffman *et al*, 2007) obtenues sur la presque totalité de l'Afrique de l'Ouest et représentant des cumuls par mois stockés sous forme de carte. Pour exploiter ces données nous avons procédé (i) au ré-échantillonnage de l'image en classes de valeur et en résolution, suivi par (ii) la reclassification de couleur et enfin (iii) la digitalisation de l'image en matrice numérique et la sauvegarde du résultat.

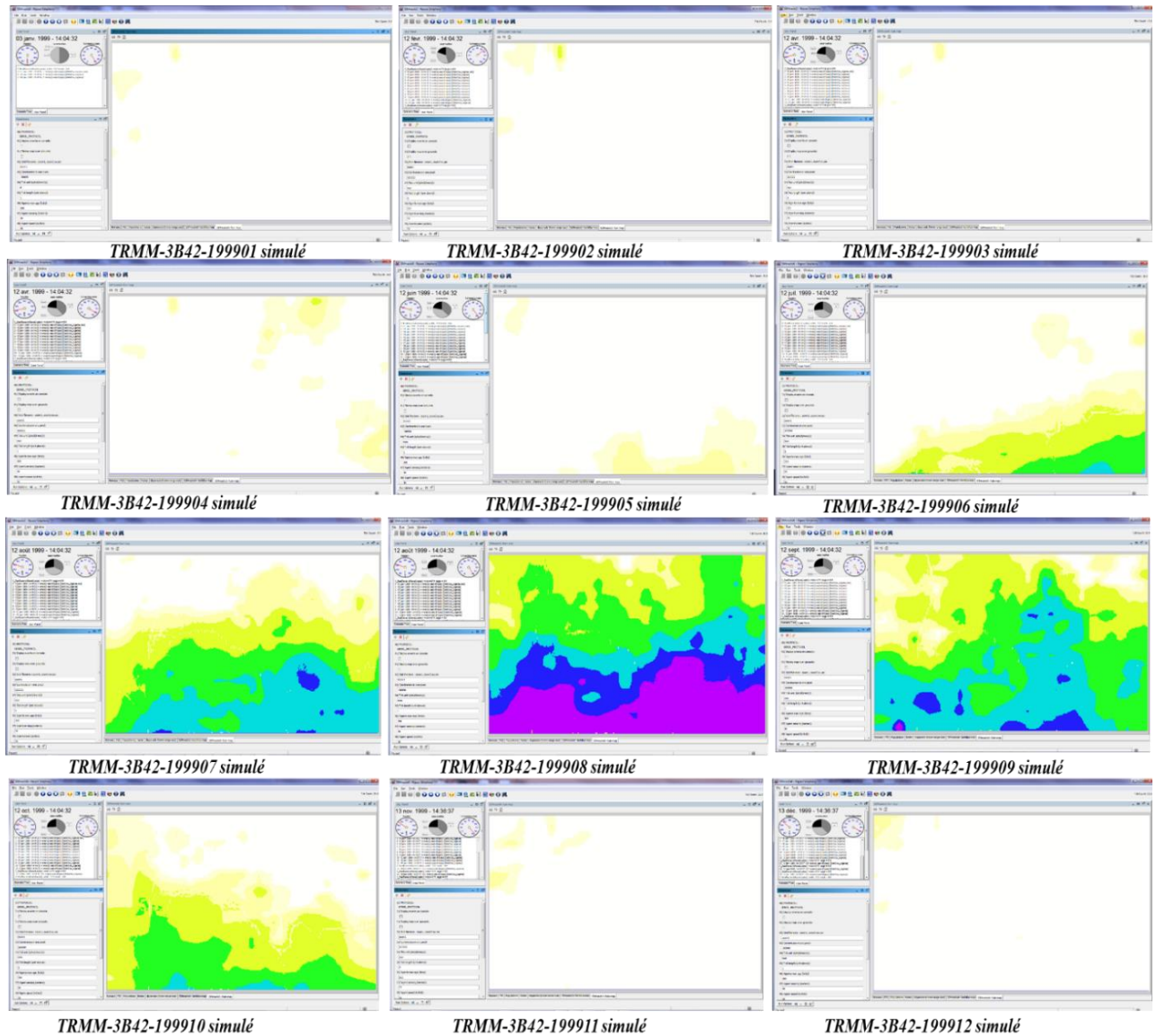


Figure 10 : Simulation de la pluviométrie de l'année 1999 (Sall, 2016)



#### d. Croissance de la végétation

La végétation dans les milieux sahélo-soudaniens se présente sous une forme variée avec la présence de certaines espèces d'herbes, d'arbustes, d'arbres et de cultures. Elle constitue une source de nourriture indispensable pour la survie des granivores comme la gerbille. La croissance de cette végétation est conditionnée par plusieurs facteurs écologiques dont la pluviométrie. Cette dernière connaît de fortes fluctuations suivant la saison et constitue ainsi le facteur le plus déterminant de la croissance de la végétation (Camberlin *et al*, 2007). Ainsi, pour obtenir la variation de cette dernière en fonction de la pluviométrie, nous avons fait appel à un algorithme basé sur la fonction de croissance logistique de Verhulst (Verhulst, 1845).

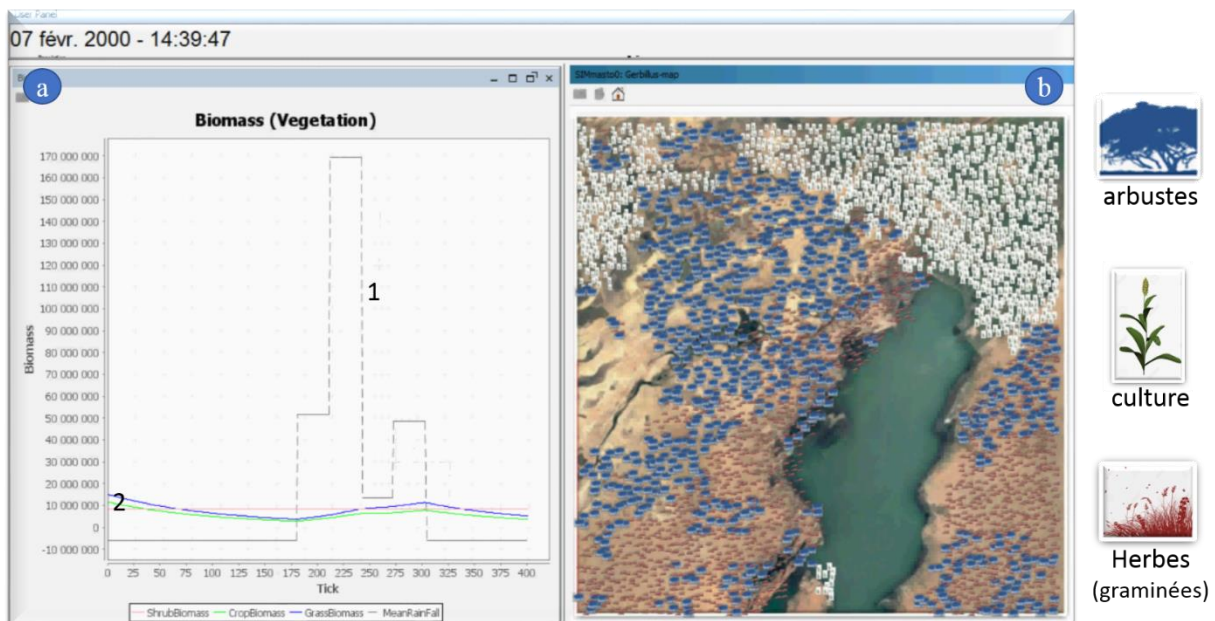


Figure 11 : Capture d'écran de la simulation de l'évolution de la pluviométrie en fonction de la quantité de pluie présente dans la cellule dans un Zoom sur le lac de Guiers au Nord du Sénégal (extrait de la communication au colloque d'étudiant de Baillarguet de Montpellier).

a) Diagramme d'évolution de la biomasse en fonction de la quantité de pluies d'axes Tick( pas de temps) et Biomass(Biomasse de la végétation). 1) représente l'histogramme d'évolution de la pluviométrie en fonction du temps, 2) représente les variations de biomasse des végétaux herbes, cultures et arbustes en fonction de la quantité de pluies.

## II.2- Modèle informatique multi-thématique

L'appréhension et la compréhension des systèmes naturels et leurs changements ont longtemps fait l'objet de travaux scientifiques. En effet, les systèmes naturels sont caractérisés par des unités constitutives interagissant entre elles (Huraux, 2015) pour préserver l'intégrité, la structure et le comportement du système plus vaste auxquelles elles appartiennent. Les difficultés d'interprétation et de compréhension de ces caractéristiques nécessitent inévitablement le retour vers l'idée de complexité. Un système complexe est un ensemble

constitué d'un grand nombre d'entités en interaction qui empêchent l'observateur de prévoir sa rétroaction, son comportement ou évolution par le calcul. La complexité d'un système est définie par la difficulté, voire l'impossibilité, de réduire le nombre de ses paramètres ou de ses variables caractéristiques, sans perdre ses propriétés fonctionnelles essentielles globales (Pavard et Dugdale, 2006). Ainsi, pour pouvoir traiter ces phénomènes, il est nécessaire d'appréhender l'aspect multidimensionnel des éléments et de leurs interactions (Meek *et al*, 2001) et, conséquemment, de mettre en œuvre une collaboration interdisciplinaire (Wheeler, 2007). Chaque brique de connaissance élaborée par les disciplines constitue à la fois un élément légitime (McMurtry et Henry, 2009) et une composante de l'ensemble des processus ayant conduit au fait constaté. Ainsi, l'intégration formelle des connaissances disciplinaires pourrait permettre de mieux identifier et donner du sens aux systèmes complexes étudiés (Newell, 2001). C'est dans ce contexte que différents modèles informatiques intégrant des connaissances distribuées et assez mal partagées ont été mis en œuvre. Parmi ces modèles informatiques nous pouvons citer la modélisation par les systèmes multi-agents.

### II.3- Les systèmes multi-agents

La modélisation par les systèmes multi-agents est un domaine assez connu et largement utilisé pour étudier les systèmes complexes (Mboup, 2017) dans les domaines de l'intelligence artificielle, des systèmes distribués, de la robotique ou de la vie artificielle (Marcenac, 2017). Les systèmes multi-agents ont vocation à être des modèles explicatifs (Bousquet *et al*, 2002) basés sur des entités en interaction entre elles, avec et à travers un environnement (Ferber, 1997). Dans ce formalisme, la dynamique globale du système émerge des interactions des agents (Grimm *et al*, 2005, Parunak *et al*, 1998). La recherche dans les systèmes multi-agents (SMA) vise à comprendre comment une société d'entités autonomes appelées agents peuvent s'organiser de façon à résoudre des problèmes, réaliser des tâches et d'une manière générale produire des phénomènes globaux, qu'aucun agent ne peut réaliser avec la même efficacité individuellement (Bousquet *et al*, 2002). Ainsi, au fur des années et suivant les auteurs, différentes définitions ont été attribuées à la notion de système multi-agents. Parmi ces définitions, nous allons retenir celle de Ferber (1995).

#### II.3.1- Définition

On appelle système multi-agents (ou SMA), un système composé des éléments suivants :

- a. Un environnement E, c'est-à-dire un espace disposant généralement d'une métrique;

- b. Un ensemble d'objets  $O$ . Ces objets sont situés, c'est -à-dire que, pour tout objet, il est possible, à un moment donné, d'associer une position dans  $E$ . Ces objets sont passifs, c'est-à-dire qu'ils peuvent être perçus, créés, détruits et modifiés par les agents;
- c. Un ensemble  $A$  d'agents, qui sont des objets particuliers ( $A \subseteq O$ ), lesquels représentent les entités actives du système;
- d. Un ensemble de relations  $R$  qui unissent des objets (et donc des agents) entre eux ;
- e. Un ensemble d'opérations permettant aux agents de  $A$  de percevoir, produire, consommer, transformer et manipuler des objets de  $O$ ;
- f. Des opérateurs chargés de représenter l'application de ces opérations et la réaction du monde à cette tentative de modification, que l'on appellera les lois de l'univers.

### II.3.2- Agents

En se basant sur la définition précédente, nous pouvons dire que l'agent est l'entité autonome active du système multi-agents, qui selon ses comportements, effectue des actions et des interactions avec son environnement (Parunak *et al*, 1998). Ainsi, Ferber (1995) propose une définition minimale et commune du terme agent dans laquelle l'agent est une entité physique ou virtuelle :

- a. qui est capable d'agir dans un environnement,
- b. qui peut communiquer directement avec d'autres agents,
- c. qui est mue par un ensemble de tendances (sous la forme d'objectifs individuels ou d'une fonction de satisfaction, voire de survie, qu'elle cherche à optimiser),
- d. qui possède des ressources propres,
- e. qui est capable de percevoir (mais de manière limitée) son environnement,
- f. qui ne dispose que d'une représentation partielle de cet environnement (et éventuellement aucune),
- g. qui possède des compétences et offre des services,
- h. qui peut éventuellement se reproduire,
- i. dont le comportement tend à satisfaire ses objectifs, en tenant compte des ressources et des compétences dont elle dispose, et en fonction de sa perception, de ses représentations et des communications qu'elle reçoit.

La conception de la dynamique d'agents complexes capables d'agir de manière crédible, en particulier lorsque leur comportement est dirigé par de nombreux besoins et désirs contradictoires (Caillou *et al*, 2017), suscite certaines questions dont : comment peuvent-ils

coopérer? Quels sont les modes de communication nécessaires pour qu'ils se répartissent des tâches et coordonnent leurs actions? Quelle architecture peut-on leur donner pour qu'ils satisfassent leurs buts? (Ferber, 1995). Les tentatives de réponse à ces questions ont favorisé la mise en place de paradigmes pour formaliser l'architecture interne de ces agents complexes. Parmi ces paradigmes, nous allons vous présenter le paradigme de l'architecture BDI (Belief, Desire, Intention) et PDE (Perception, délibération, Exécution).

### II.3.3- Architecture BDI

L'approche BDI (croyance, désir, intension) a été proposée pour la première fois dans le domaine de l'Intelligence Artificielle (Bratman, 1987) et a été conçue dans le but de représenter la façon dont les agents peuvent faire un raisonnement complexe. Son succès dans le monde de la recherche est basé sur sa simplicité en réduisant le Framework d'explication pour le comportement humain complexe (Dennett, 1989). Une autre force du modèle BDI est l'utilisation constante de notions psychologiques populaires (Pokahr *et al*, 2005) qui correspondent de près à la façon dont les humains communiquent (Norling, 2004). Elle a été adoptée avec succès dans des domaines aussi variés que la simulation de tactiques militaires, les systèmes de contrôle des processus d'usine, la gestion des processus métier, le système de diagnostic de panne dans la navette spatiale (Georgeff et Rao, 1996) ou encore le diagnostic dans les réseaux de télécommunications. L'architecture BDI présente une sémantique logique abstraite, qui a été reprise et développée largement dans la communauté des chercheurs (Rao et Georgeff, 1998, Schild, 1998). En parallèle, plusieurs architectures prenant en compte les agents BDI ont été développées dans les systèmes multi-agents. Certaines de ces architectures BDI sont incluses dans des frameworks permettant de les utiliser directement dans différentes applications (Caillou *et al*, 2017). Trois processus principaux sont définis dans l'architecture BDI: la perception dans laquelle l'agent acquiert des informations de l'environnement, l'interprète central qui aide l'agent à délibérer en fonction de ses désirs puis à sélectionner les actions disponibles et l'exécution de l'intention qui représente les réactions des agents (Caillou *et al*, 2017). Selon Rao (1991), les croyances, les désirs et les intentions sont définis comme des attitudes mentales représentées par des états possibles du monde. Pour un agent BDI :

- ✓ La croyance représente les connaissances internes que l'agent a sur le monde.
- ✓ Les désirs sont l'ensemble des objectifs que l'agent souhaiterait accomplir. Comme la base de croyance, la base de désirs est mise à jour pendant la simulation. Les désirs

peuvent être liés par des liens hiérarchiques (sous / super-désirs) lorsqu'ils sont créés en tant qu'objectif intermédiaire.

- ✓ L'intention décrit ce que l'agent a choisi de faire. Elle détermine le plan sélectionné et peut être mise en attente, par exemple lorsqu'un désir secondaire est nécessaire d'être atteint. Ainsi, une file (FIFO) est utilisée pour gérer la liste d'intentions de l'agent, la dernière est l'intention actuelle, et la seule qui n'est pas en attente.

Martha Pollack dans (Georgeff *et al*, 1998), subdivise le travail sur les architectures BDI en trois catégories:

- a. Modèles généraux pour le raisonnement pratique, en fonction des concepts BDI.
- b. Modèles informatiques basés sur l'architecture des machines à dérivation intelligente (IRMA) (Bratman *et al*, 1988), présentant une correspondance étroite avec la philosophie de Bratman.
- c. Le modèle de calcul utilisé dans la famille de systèmes PRS (Georgeff *et al*, 1987, Ingrand *et al*, 1992).

Certains auteurs ont montré l'intérêt d'utiliser le paradigme BDI dans le contexte de la simulation (e.g., Adam *et al*, 2011, Shendarkar *et al*, 2008), mais il est encore rarement utilisé (Le *et al*, 2013). Une raison en est que la plupart des architectures d'agents basées sur le paradigme BDI (e.g., JAM (Huber, 1999), JASON (Bordini *et al*, 2007)) sont complexes à comprendre et à utiliser par des scientifiques non informatiques.

En fait, la conception d'agents capables de réaliser un raisonnement complexe et d'interagir avec leur environnement biophysique ainsi qu'avec d'autres agents est une tâche difficile, en particulier pour les chercheurs qui n'ont pas de compétences en programmation (Taillandier *et al*, 2012).

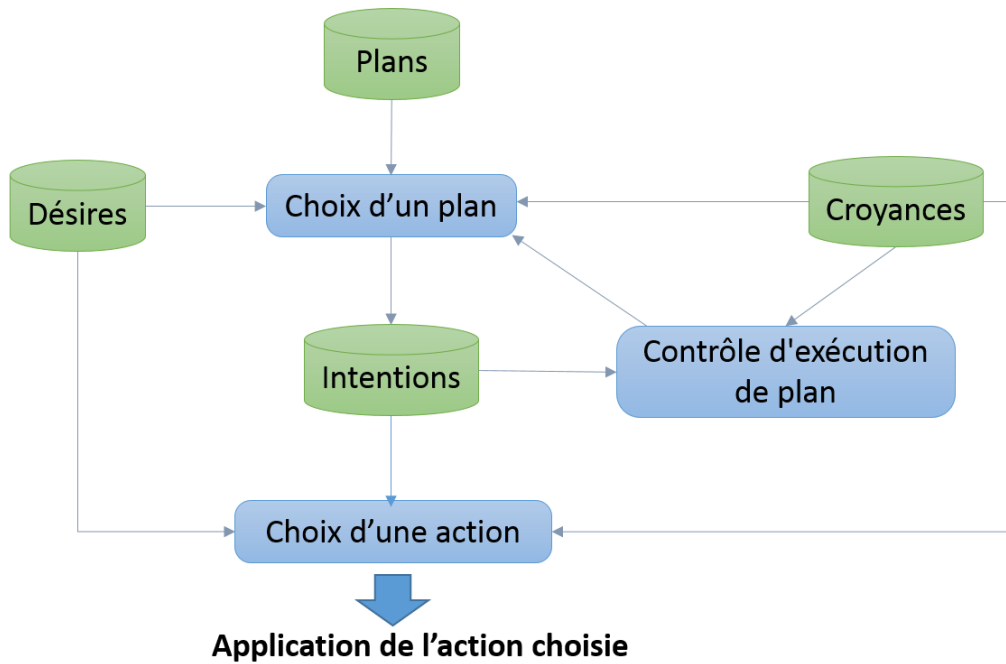


Figure 12 : Architecture Croyance-Désire-Intension d'un agent (adapté de Taillandier *et al*, 2012)

#### II.3.4- Architecture PDE

Cette architecture dite PDE « perception, délibération, décision, exécution » est fortement inspirée du travail des pionniers Genesereth et Nilsson (1987), qui ont proposé une représentation algébrique de leur système et des comportements des agents (Ferber, 1995) (Pérez *et al*, 2005), l'architecture PDE est une théorie qui permet de décrire rigoureusement les interactions complexes entre les agents situés (Ferber et Müller, 1996). Elle offre un plus grand niveau d'abstraction réduisant la complexité du problème (Ferber, 1995) et permet de couvrir la dynamique des systèmes multi-agents constitués d'agents logiciels mobiles adaptés aux systèmes évolutifs (Pérez *et al*, 2005). L'architecture PDE a été adoptée dans diverses domaines comme : le contrôle des systèmes biologiques (Pérez *et al*, 2008), la robotique (Martinez et Perez, 2011), les neurorégulateurs biologiques (Fernández *et al*, 2006), etc. Elle confère une grande adaptabilité et une bonne approximation des systèmes biologiques (García et Maciá, 2000).

Les agents de l'architecture PDE sont des entités capables de prendre en compte ce qui les entoure, ce qui se traduit pour les agents purement situés par la capacité de percevoir l'environnement et d'agir en exécutant des actions qui tendent à modifier l'état du système (soit en agissant directement sur l'environnement, soit en émettant des messages à l'adresse d'autres agents), après avoir délibéré de ce qu'il convenait de faire. Ce concept sur les agents est basé sur une structure de perception, délibération et exécution (Pérez, 2001, Ferber, 1995).

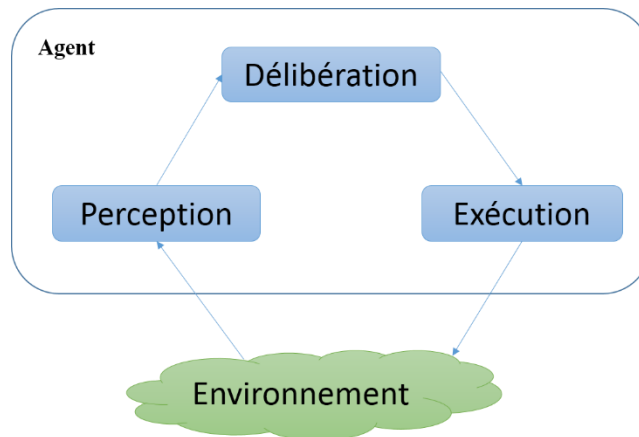


Figure 13 : Architecture perception-délibération-exécution d'un agent (adapté de Ferber, 1995)

Avec les agents PDE, deux types d'interactions sont notés : les interactions entre un agent et son environnement et les interactions entre les agents à travers l'environnement. Ce qui permet par conséquent, le traitement des agents tropiques (réactifs) et des agents hystérétiques (agents avec mémoire) en tant que cas particuliers du système multi-agents.

Pour un agent PDE, la perception représente la qualité de pouvoir classer et distinguer différents états non seulement en ce qui concerne les caractéristiques les plus importantes de l'environnement, mais aussi en ce qui concerne les actions qui en découlent (Pérez *et al*, 2005).

La délibération représente l'ensemble des fonctions qui rendent possible la sélection des tâches (Pérez *et al*, 2005). Elle se déroule entre la perception et l'action et permet de différencier les agents tropiques, sans mémoire, des agents hystérétiques, qui ont des états internes et qui peuvent prendre en compte les situations passées (Michel, 2006). La délibération, sauf à considérer des agents extrêmement simples, c'est là que s'élaborent les objectifs, les prises de décisions, les facultés de mémorisation et, en ce qui concerne les agents cognitifs, les représentations du monde et les concepts utilisés pour décider de l'action à entreprendre.

Enfin, l'action pour un agent PDE représente l'intention d'agir sur le système.

L'intérêt de cette architecture provient de l'impossibilité de planifier à l'avance l'ensemble des actions des agents. Cette approche fournit un cadre capable de fournir une capacité expressive raisonnable à s'attaquer au développement de systèmes complexes distribués, en tenant compte d'un large éventail de contingences, d'un comportement émergent et de la possibilité d'une modification de la structure à mesure que de nouvelles avancées s'opèrent (Ferber, 1995). Ce qui fait que, l'interaction entre les différents agents peut donner des

comportements émergents qui ne sont pas définis dans le comportement initial programmé des agents individuels (Fernández *et al*, 2004).

### II.3.5- Les plateformes de simulations multi-agents

Avec l'avancée de la technologie, différents outils (plates-formes, toolkits, frameworks) de modélisation et de simulation à base d'agents ont été développés (Allan, 2009). La plupart d'entre eux sont restés au stade de développement et n'ont jamais été connus par les utilisateurs. Seul un petit nombre d'entre eux sont adoptés par les modélisateurs et sont devenus populaires (Vo, 2012). Parmi ces outils nous pouvons citer : GAMA, NetLogo et Repast Simphony.

**GAMA** ("Gis et Agent-based Modelling Architecture") est considérée comme la plate-forme la plus avancée dans la modélisation multi-agents et multi-niveau et permettant d'étudier les phénomènes géographiques (Drogoul *et al*, 2013). Elle permet de construire des simulations spatiales explicites basées sur des agents (Grignard *et al*, 2013).

**NetLogo** (Wilensky et Evanston, 1999) est une plate-forme de modélisation à base d'agents développée au CCL ("Center for Connected Learning and Computer-Based Modeling") et qui offre des concepts adaptés au développement de modèles à base d'agents spécialisés. Son langage ReLogo, basé sur le langage Logo (Papert, 1980), cache toute la complexité d'un langage d'usage général comme Java, permettant ainsi aux non modélisateurs non informaticiens d'écrire facilement leur modèle.

**Repast** ("The Recursive Porous Agent Simulation Toolkit") **Simphony** est un système multi plates-formes interactif de modélisation à base d'agents basé sur Java. Il est développé au Laboratoire National d'Argonne à Chicago (Collier *et al*, 2003, North *et al*, 2013) et incorporé dans l'environnement de développement Eclipse. Il est open-source et est réputé très flexible pour le développement de modèles d'interaction d'agents exécutables sur des ordinateurs personnels et sur des machines en clusters de petite taille. Il a été utilisé comme base de départ pour le développement de la plateforme SimMasto.

Ces Plateformes offrent des environnements de développement de modèles basés sur des langages de programmation objet (Python, Java, C ++, etc.), les langages de domaine spécifique ( ReLogo utilisé dans NetLogo (Wilensky *et al*, 1999), GAML de GAMA (Drogoul *et al*, 2013), etc. ) ou les langages à base de graphiques (Groovy (North *et al*, 2013), etc.).



## CHAPITRE 2: Modélisation

La modélisation, apparue il y a longtemps (Coquillard *et al*, 1997), est aujourd'hui largement utilisée dans les approches scientifiques pour décrire les systèmes complexes. Avec l'essor des nouvelles technologies et des concepts informatiques, une véritable révolution s'est opérée dans ces disciplines traditionnelles comme la modélisation de systèmes biologiques longtemps cantonnée dans une approche analytique ou stochastique (Coquillard *et al*, 1997). C'est dans ce cadre d'idée qu'il a été proposé dans le projet CERISE, l'utilisation de la modélisation par les systèmes multi-agents pour la mise à disposition d'outil d'aide à la compréhension de l'invasion de la gerbille au Sénégal Ainsi, dans une approche de mise en place du modèle, nous avons procédé à :

- ✓ l'agrégation du protocole gerbille dans la plate-forme SimMasto,
- ✓ la formalisation des agents du système avec l'aide des données fournies par les biologistes et la littérature,
- ✓ la mise en œuvre des processus de décision des agents basés sur l'architecture PDE décrite dans le chapitre 1, paragraphe II.3.4- page II.3.4- 29.
- ✓ la mise à jour de l'environnement avec les données d'entrées du modèle.

### I- Protocole de la gerbille

La classe protocole dans la plateforme SimMasto, représente le point d'entrée pour une étude de cas formalisée. En effet, elle permet de réaliser le monde correspondant à chaque étude considérée et à la question particulière posée par le chercheur (choix des échelles, des objets et des agents, indicateurs de sortie sélectionnés, des sources de données d'entrée). Elle constitue la classe centrale du simulateur. A l'aide du protocole, chaque étude de cas agrégée dans le modèle est caractérisée, distinguée et mise en œuvre de manière autonome. Chaque protocole étend la classe abstraite A\_Protocol et est composé par :

- ✓ un gestionnaire de données d'entrée,
- ✓ un gestionnaire d'environnement,
- ✓ un ou plusieurs inspecteurs permettant de collecter les données produites par le simulateur au cours de la simulation et de les représenter comme données dans la partie visuelle du simulateur (la partie observateur),
- ✓ d'un calendrier permettant de convertir la durée des pas de temps de simulation en unités temporelles, par exemple une durée de simulation peut valoir 1 heure ou 2 jours

etc. ce calendrier s'incrémente au cours de la simulation et permet de connaître l'âge des agents, la saison de reproduction etc.

Le rôle principal du protocole est de distribuer les tâches aux autres composants du modèle, de créer les agents, les référencer dans les inspecteurs concernés et de demander au gestionnaire d'espace (*C\_Landscape*) de les positionner sur l'environnement. A l'aide d'un scheduler intégré dans Repast (*@Scheduler()*), le Protocole gère à chaque pas de temps (*tick*) son environnement et active à chaque agent pour qu'il réalise son propre *step* (ses actions pour ce *tick*) (Mboup, 2017). La durée du pas de temps dépend du niveau de détail choisi lors d'une étude de cas, par exemple : pour notre modèle, la simulation de la zone sahélienne (Sénégal du nord) avec un pas de temps de 1 jour ou la simulation du zoom sur une portion du lac de Guiers au Nord du Sénégal avec un pas de temps de 1 heure, etc.

Les différents points du processus d'exécution du protocole sont figurés ci-après à l'aide d'un diagramme de séquence Figure 14.

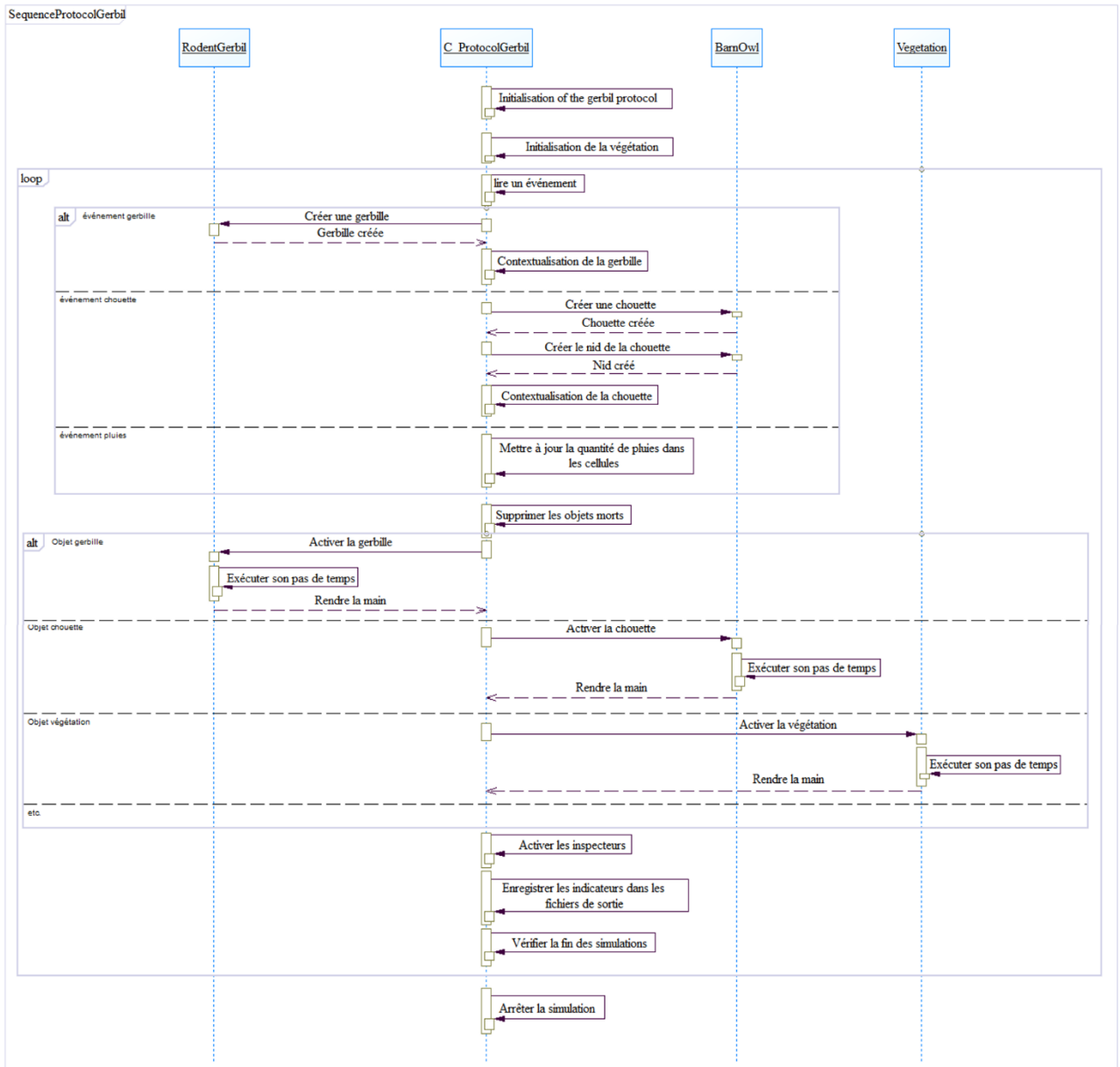


Figure 14 : Diagramme de séquence du processus d'exécution du protocole gerbille  
 alt : alternative ; opt : optionnel ; loop : boucle

## II- Formalisation des agents du système

Formaliser les agents revient à représenter de manière simple et générale leur comportement individuel à savoir leurs processus biologiques (croissance, reproduction, gestation, etc.) et trophiques (alimentation, prédation). L'utilisation des systèmes multi-agents (SMA) cognitifs nous a permis de mettre l'accent sur les capacités des agents en s'intéressant à leurs propriétés individuelles. Ainsi, grâce aux données fournies par les chercheurs de terrain et la littérature, nous avons procédé à la mise en place des traits de vie des agents ainsi que des processus de

décision de ces derniers. Ces processus de mise en œuvre sont décrits dans les points qui suivent.

## II.1- Gerbille

Le comportement de la gerbille dans son milieu naturel est caractérisé par une suite d'actions quotidiennes. Elle se nourrit de graines et de jeunes pousses de cultures et d'herbes mais aussi d'insectes pendant les périodes de disette. Cette action lui permet de renouveler son stock d'énergie nécessaire à sa survie. Elle élabore également des actions de fuite (se terrer, se cacher sous un arbuste, etc.) lorsqu'un prédateur est aperçu, de construction de terrier lorsqu'elle en a besoin (nid pour mettre bas, cachette, etc.) et qu'elle n'en possède pas, d'exploration de nouvelles zones à la recherche de milieu favorable ou lorsqu'elle est un jeune mâle. Des études ont montré que le début de la saison des pluies est étroitement lié au début de la saison de reproduction des gerbilles nigérianes. En effet, en dehors de la disponibilité des ressources, l'apparition de jeunes pousses de plantes annuelles semble être un déclencheur de la reproduction (Nomao et Gautun, 2001). Ces plantes procurent aux gerbilles non seulement l'eau nécessaire à la lactation, mais aussi une substance (6-methoxybenzoalnone) dont l'effet stimulateur sur la reproduction a été montré (Hima, 2010). La présence de cette substance dans la végétation a été formalisée dans le modèle avec la prise en compte des quantités de pluies présentes dans la cellule. Ainsi, une gerbille adulte qui se nourrit d'une végétation contenant cette substance se met en rut. Et cette période de rut est maintenue jusqu'à ce qu'elle se nourrisse d'une végétation ne contenant pas la substance (ce processus fait encore l'objet de discussion avec les biologistes). Ainsi, la formalisation d'un agent de gerbille dans le modèle a été effectuée comme suit

- ✓ création de l'agent gerbille avec un génome dédié *C\_GenomeGerbillusNigeriae*,
- ✓ contextualisation de la gerbille sur le substrat suivant sa position (coordonnée de piégeage ou sur un cercle de rayon 2500m lorsqu'elle a été trouvée par analyse sur une pelote de chouette),
- ✓ prise en compte du rythme circadien (gerbille active pendant la nuit et au nid à l'aube)
- ✓ exécution de ses processus de décision

### II.1.1- Processus de décision de la gerbille

Les processus de décision de la gerbille sont effectués suivant le modèle PDE et sont tels que décrit ci-après.

### a. Perception

Avec une fonction de perception, la gerbille sillonne ses alentours et enregistre dans une liste les objets (sauf ceux de la végétation) dont leur position appartient au cercle d'origine la position de la gerbille sur la grille et de rayon égal au rayon de perception de la gerbille défini selon le pas de temps choisi (ex : 30 mètres si le pas de temps est 1 jour ou 30/24 mètres si le pas de temps est 1 heure). Pour la perception des objets de végétation voir **Erreur ! Source u renvoi introuvable.**, la gerbille cherche sur l'ensemble des objets de végétation présents dans la cellule ceux dont le disque de couverture végétale présente une intersection non vide avec celui du disque de perception de la gerbille. A la fin, les objets de végétation vérifiant la condition précédente sont ajoutés sur la liste des objets perçus.

Ce processus fait encore l'objet de discussion avec les biologistes

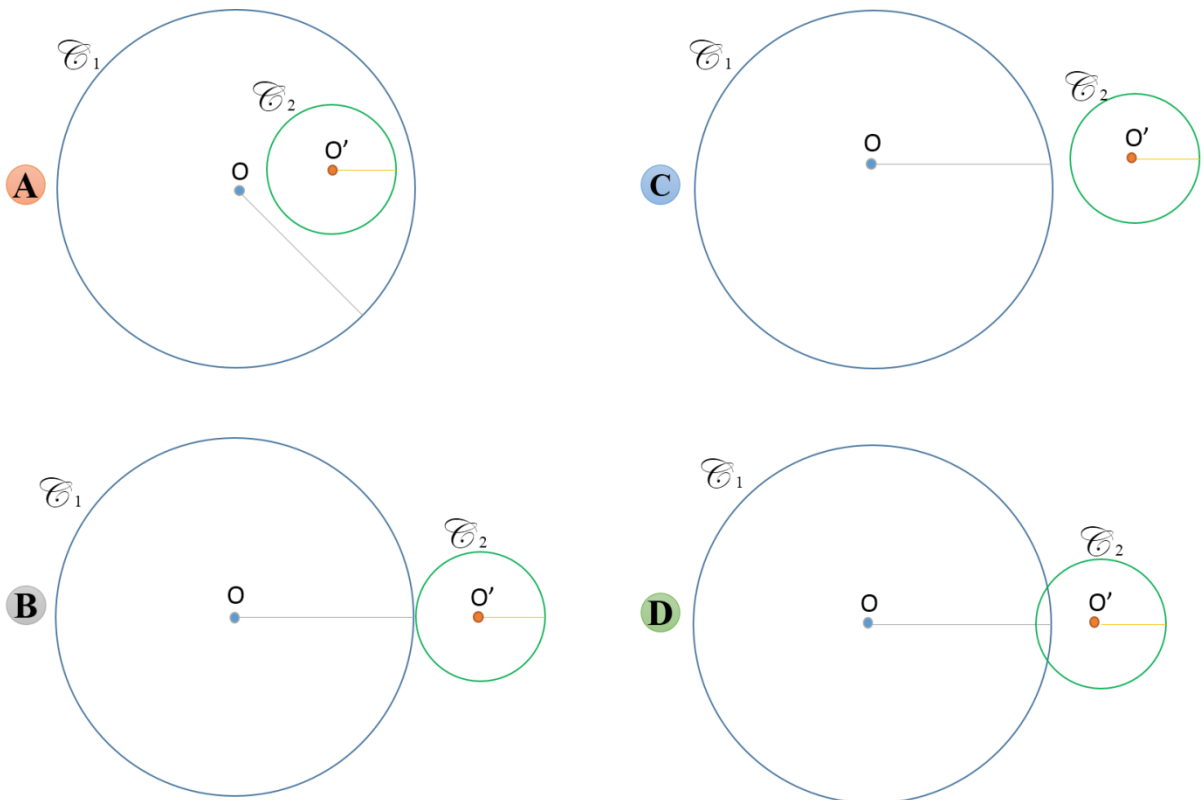


Figure 15 : les cas de figure possible entre le disque de perception de la gerbille et celui de la couverture végétale,  $\mathcal{C}_1$  : disque de couverture de la végétation,  $\mathcal{C}_2$  : disque de perception de la gerbille,  $O$  : position de la végétation sur la grille,  $O'$  : position de la gerbille sur la grille. Dans **A**, **B** et **D** la gerbille perçoit la végétation et dans **C** la gerbille ne perçoit pas la végétation

### b. Délibération

Etant un agent autonome, la gerbille évalue sa situation en se basant sur la liste des objets perçus et ses désirs (physiologique) **Erreur ! Source du renvoi introuvable.** Après avoir éterné la liste des objets perçus, la gerbille procède à une prise de décision suivant un ordre

de priorité. Cet ordre de priorité est surtout défini selon le besoin présent. Un pseudo code est ainsi utilisé pour décrire la délibération d'un agent gerbille sur notre modèle.

```

Si l'aube
|
| Si terrier n'est pas perçu
| | CreuserUnTerrier ()
| | retournerTerrier ()
Sinon
|
| Si chouette perçue
| | Si terrier ou arbuste dans la liste perçue
| | | Retourner terrier ou arbuste
| | Sinon
| | | Détaler ()
Sinon
|
| Si période de reproduction
| | Chercher un partenaire
| | Si partenaire trouver
| | | Retourner le partenaire
Sinon
|
| Si besoin de manger
| | Chercher un objet de culture ou d'herbe
| | Si végétation trouvée
| | | Retourner la végétation

```

Figure 16 : pseudo code de la délibération de la gerbille

### c. Exécution ou interaction

L'exécution ou l'interaction pour la gerbille consiste à faire l'action correspondant à la décision prise lors de la délibération. Chaque processus de délibération dans notre modèle renvoie un objet avec lequel la gerbille doit interagir. Ainsi, avant d'interagir avec l'objet renvoyé, la gerbille se déplace à la position de ce dernier, si elle n'y est pas déjà, puis interagit avec lui (terrier : se terrer, végétation : s'alimenter, partenaire : s'accoupler, etc.).

Les processus de décision de la gerbille ont été décrits à l'aide du diagramme de séquence de la Figure 17.

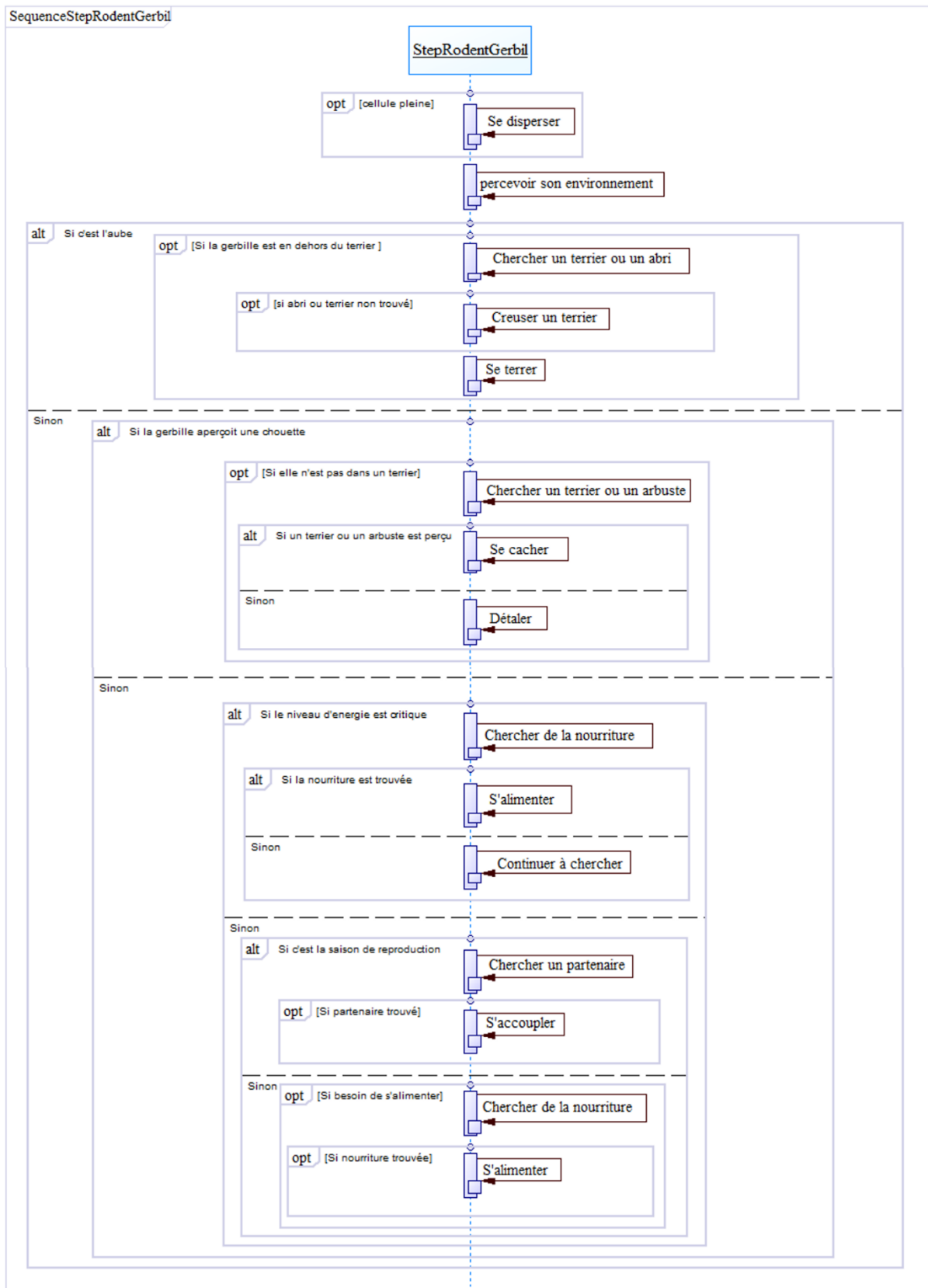


Figure 17 : diagramme de séquence du processus de décision de la gerbille  
 alt : alternative ; opt : optionnel

## II.2- Chouette effraie

La chouette effraie, comme la gerbille nigériane, est dotée d'un ensemble de comportements étudié et formalisé dans notre modèle. Principalement nocturne, elle élabore des suites d'actions suivant ses besoins physiologiques. En effet, pendant la nuit, elle sillonne son environnement à la recherche de proie (principalement des rongeurs) et retourne dans son nid une fois rassasiée ou à la première lueur de soleil. La chouette effraie consomme en moyenne 70 à 105g de nourriture par jour (Mikkola et Willis, 1983), ce qui coïncide au poids moyen de trois (3) gerbilles. Dans la prise en compte de la relation proie-prédateur entre la chouette et la gerbille dans notre modèle, nous sommes amenés à formaliser certaines caractéristiques biologiques et trophiques de la chouette. Ainsi, nous avons procédé à :

- ✓ création de l'agent chouette avec un génome dédié *C\_TytoAlba*,
- ✓ création du nid de la chouette aux coordonnées de ramassage de pelote,
- ✓ attribution du nid à la chouette
- ✓ contextualisation du nid sur le modèle puis de la chouette sur le nid,
- ✓ prise en compte du cycle circadien de la chouette (chouette active pendant la nuit et au nid à l'aube)
- ✓ exécution de ses processus de décision

### II.2.1- Processus de décision de la chouette

Comme tout agent du système, la chouette effectue des processus de décision suivant ses désirs présents en se basant sur l'architecture PDE adoptée dans le modèle.

#### a. Perception

Suivant un rayon de perception défini selon le pas de temps choisi, la chouette sillonne son milieu à la recherche de gerbille. Elle procède ainsi à l'enregistrement dans une liste d'agents, des gerbilles se trouvant dans la zone couverte par son disque de perception et n'étant pas à l'intérieur d'un terrier.

#### b. Délibération

Une fois la perception effectuée, la chouette sélectionne l'(les) action(s) à effectuer. Deux processus de décision de la chouette sont actuellement pris en compte dans notre modèle. En effet, dans le souci d'assurer une base solide de la relation proie-prédateur dans notre modèle, nous avons choisi de mettre en œuvre le processus de chasse et le besoin de retourner dans son habitat une fois rassasiée ou au lever du soleil. Le pseudo code ci-dessous décrit le processus de délibération de la chouette mis en œuvre dans notre modèle.



```

Si l'aube
|
| Si position ≠ nid
| | retournerAuNid ()
Sinon
|
| Si besoin de chasser
| | Si gerbille dans la liste des objets perçus
| | | Retourner gerbille
| | Sinon
| | | Continuer de chercher
Sinon
|
| Si position ≠ nid
| | retournerAuNid()

```

Figure 18 : pseudo code de la délibération de la chouette

### c. Exécution ou interaction

Après avoir formulé son désir pendant la délibération, une entité du modèle est retournée. Ainsi, la chouette procède à la mise en œuvre de la tâche choisie en effectuant l'action correspondante. Comme pour la gerbille, la chouette se déplace à la position de l'entité retournée par la délibération avant d'interagir avec elle. Les processus de décision de la chouette sont décrits dans le diagramme de séquence de la Figure 19.

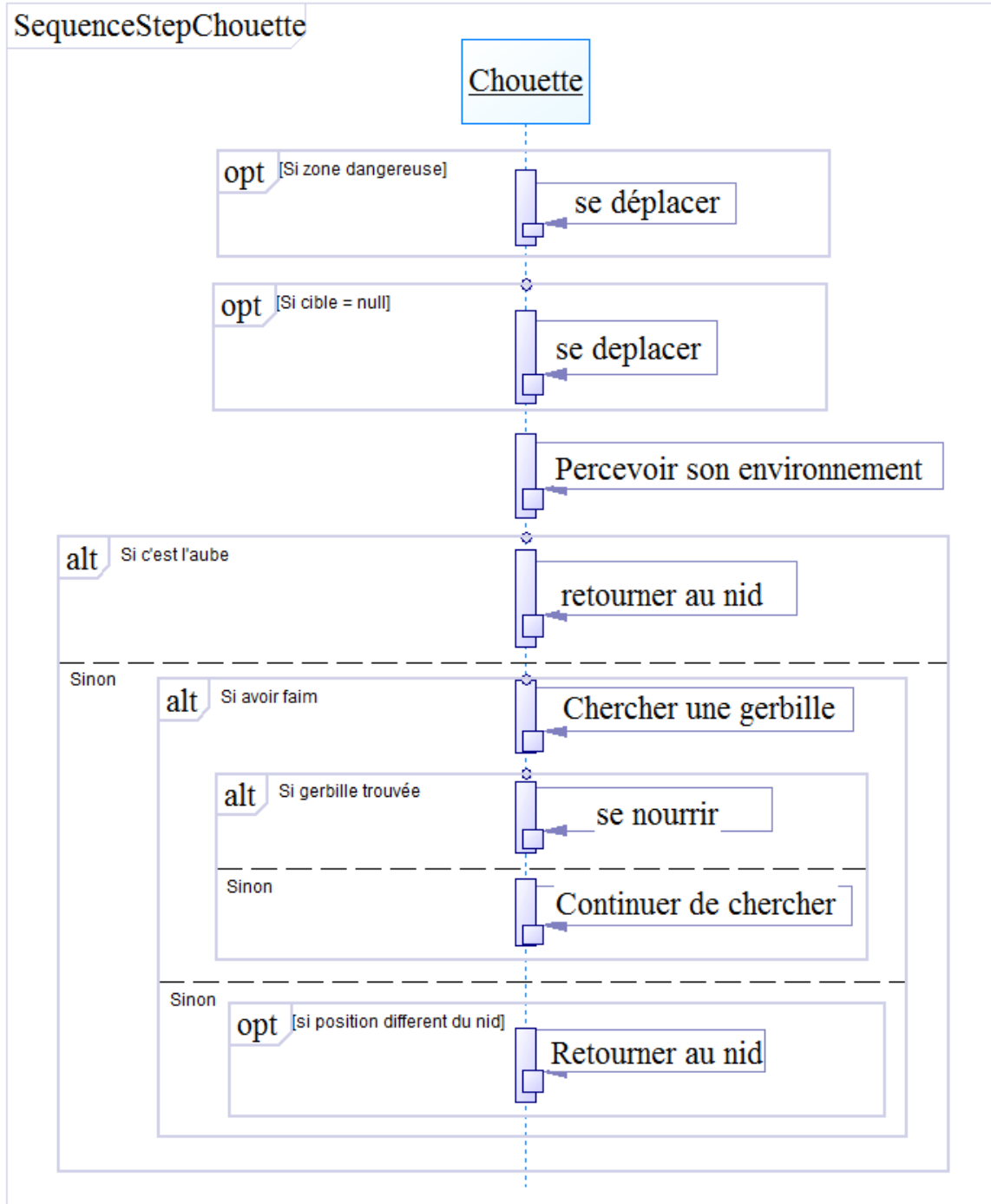


Figure 19 : diagramme de séquence des processus de décision de la chouette  
 alt : alternative ; opt : optionnel

**NB** : Chaque action exécutée dans notre modèle entraîne une perte et/ou un gain d'énergie.

**Processus de décision dans notre modèle : exemple d'un agent de gerbille**

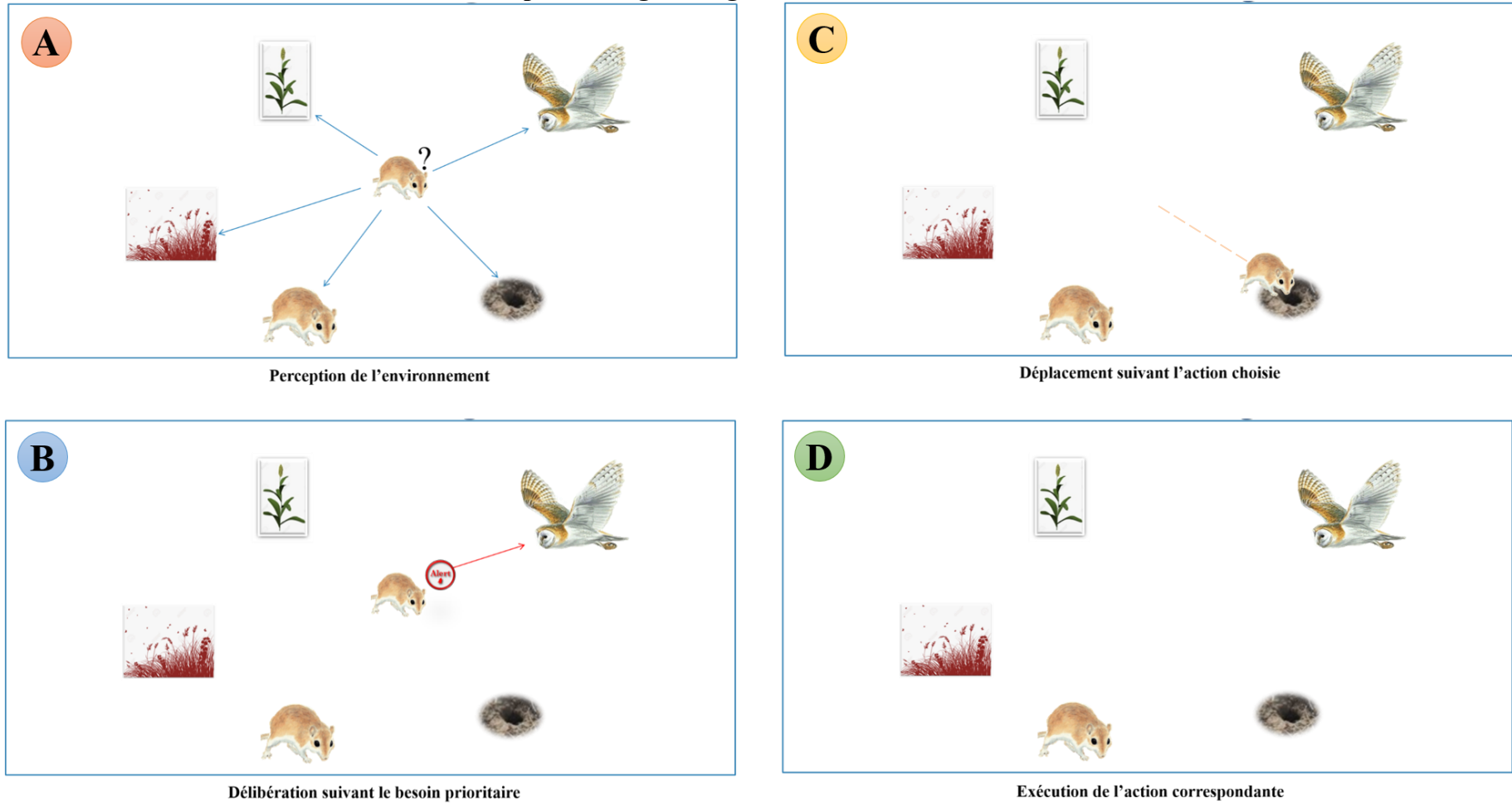


Figure 20 : Processus de décision d'un agent de gerbille, **A)** perception de son milieu, **B)** délibération suivant le besoin pressant, **C)** et **D)** exécution de l'action choisie (déplacement vers le terrier ensuite se terrer)

### III- Mise à jour de l'environnement de simulation

Dans le modèle d'étude de la gerbille, l'évolution de l'environnement est assurée d'une part par l'intégration dans le modèle des données fournies par les chercheurs de terrain contribuant au projet (géographe, biologiste, écologiste, etc.) et d'autre part, par la somme des influences produites par les agents semi autonomes (gerbilles, chouettes) à chaque pas de simulation.

#### III.1- Chronogramme de simulation

Le chronogramme est un système minimaliste mis au point par J. Le Fur et collaborateur pour traiter indistinctement des événements de type géographique, biologique, écologique, climatologique, au sein d'une historique, etc.(Mboup, 2015). Il sert à intégrer les données fournies par les thématiciens et à mettre à jour l'environnement. Le chronogramme est défini *in situ* par un fichier csv accessible aux non-informaticiens. Il s'agit d'un tableau dans lequel chaque ligne ou bloc de ligne représente un événement complet. Il est construit avec un format unique et est structuré suivant un ordre chronologique.

Ce format lui permet de répondre et par ordre aux trois questions suivantes :

- ✓ Quand : à quelle date l'événement a été enregistré ?
- ✓ Où : à quel endroit ?
- ✓ Quoi : de quel type d'événement s'agit-il ?

Optionnellement, le chronogramme permet d'ajouter un complément d'information de type comment et/ou combien. Les différentes questions auxquelles doit satisfaire le chronogramme sont ainsi présentées dans le Tableau 1.

**Tableau 1** : Extrait du chronogramme 20151029\_GerbilEventsOfficiel.2a.jlf.csv utilisé dans le modèle gerbille source : (Sall, 2016)

DATE EVENT	X	Y	EVENT	VALUE1	VALUE2	COMMENT
01/01/1999		0	0 rain			2015.10.29 M.Sall & J.Le Fur
17/07/2011	-14.82825	15.81816667	Gerbillus_nij F		489846	20151005-CapturesGnigeriaePE.jlf.ms
17/07/2011	-14.82825	15.81816667	Gerbillus_nij M		489848	20151005-CapturesGnigeriaePE.jlf.ms
01/08/2011		0	0 rain			2015.10.29 M.Sall & J.Le Fur
01/09/2011		0	0 rain			2015.10.29 M.Sall & J.Le Fur
25/09/2011	-14.412	16.495	owl	Gerbillus_nij		2 Dodel/ G. henleyi / nigeriae remplacé par Gerbillus_nigeriae, JLF-MS, 29.10.2015
25/09/2011	-14.412	16.495	owl	Gerbillus_nij	13	Dodel, JLF-MS, 29.10.2015

On note ainsi, la date de l'événement (DATE\_EVENT) qui répond à la question du Quand, les coordonnées (X et Y) de l'événement de piégeage de gerbilles ou de ramassage de pelotes de chouette pour la question du lieu (Où), la colonne EVENT décrivant le type d'événement

noté (piégeage, pluies ou ramassage de pelote), la colonne VALUE1 qui renseigne le sexe pour les captures ou le type de rongeurs pour les pelotes, la colonne VALUE2 pour le numéro d'identification des gerbilles capturées ou le nombre de gerbilles trouvé par analyse sur les pelotes ramassées, enfin la colonne COMMENT qui représente les commentaires qui peuvent être notés sur l'évènement.

La date 01/01/1999 est utilisée dans notre modèle comme la date de démarrage de la simulation. Elle coïncide aussi dans le tableau, à l'arrivée du premier évènement de pluie. Les colonnes représentant des évènements de gerbille dans le chronogramme (ex : la capture d'une gerbille (*Gerbillus\_Nigeriae*) femelle à la date 17/07/2011 et aux coordonnées (14,83°W, 15,82°N)) correspondent dans notre modèle à la création d'un agent de gerbille à la position fournie par les biologistes du projet. En outre, les évènements de pelote (*owl*) sont utilisés pour la création d'une part d'une chouette et son nid aux coordonnées fournies et d'autre part à la création des agents gerbilles correspondant à ceux trouvés par analyse sur les pelotes sur le cercle de chasse de la chouette de rayon 2,5 km<sup>2</sup> et de centre la position définie par les coordonnées du point de ramassage.

Pour gérer les évènements pendant la simulation, deux classes de SimMasto sont sollicitées. *C\_Chronogram* et *C\_Event* : sont utilisées par le modèle pour stocker, parcourir et traiter les évènements du chronogramme. La classe *C\_Chronogram* se charge de contenir les évènements (par ordre chronologique) permettant la création et la mise à jour de l'environnement. Elle copie les lignes du chronogramme et construit ensuite des objets évènements avec la classe *C\_Event* et les stocke dans un collectionneur d'objets. Ce procédé générique permet aux autres études de cas de l'utiliser (mis en œuvre dans l'étude de cas sur la diffusion du rat noir au Sénégal et réutilisé sur notre modèle) moyennant une adaptation du traitement des évènements particuliers.

### III.2- Gestion des steps

La simulation est considérée comme un modèle plongé dans le temps (Coquillard *et al*, 1997). Ce dernier est représenté dans les simulations en temps discret (que nous réalisons) par un référentiel monodimensionnel défini par une origine, une unité de temps, un pas de temps élémentaire et un ordonnanceur élémentaire (scheduler). Le step dans SimMasto est une fonction exécutée à chaque pas de temps et elle permet au protocole de réaliser ses actions et de déclencher celles des autres entités du système. En effet, à chaque pas de temps, le protocole compare la date de la simulation à la date courante du chronogramme. Si les dates

correspondent alors il procède à la lecture des lignes du chronogramme ayant la même date. Chaque lecture de ligne du chronogramme correspond soit à la mise à jour de la quantité de pluie présente dans la cellule, soit à l'entrée d'un nouvel agent gerbille dans le modèle ou l'ajout d'une chouette, son nid et le nombre de gerbilles trouvées par analyse dans les pelotes récoltées. Après avoir terminé la lecture des données, le protocole met à jour son environnement en procédant à la suppression des objets morts. En outre, avant d'enregistrer les indicateurs, le protocole active de façon successive à chaque agent du système. Ces derniers exécutent leur pas de temps et retournent la main au protocole qui procède à la vérification de la fin de simulation.

## CHAPITRE 3 : Résultats et Discussion

Dans ce chapitre, nous allons d'abord vous présenter le premier résultat obtenu avec la prise en compte des entités dans le modèle, ensuite nous procéderons à la discussion du travail effectué.

### I- Présentation d'un premier résultat

Les résultats dans une simulation proviennent d'un procédé d'exécution successive de plusieurs fonctionnalités du modèle par exemple, la mise en place de l'environnement, la gestion de l'arrivée d'un agent, etc. La simulation suit une chronologie partant de la lecture des données à la visualisation des résultats en passant par l'exécution des procédés de traitement. Les principaux résultats obtenus durant ce travail tournent autour de la mise en place des processus biologiques et trophiques de la gerbille et de son prédateur la chouette effraie. Un aperçu des simulations obtenu est présenté sur la Figure 21. Sur cette figure apparaissent quelques populations de gerbilles et de chouettes évoluant dans leur environnement dynamique caractérisé par une végétation diversifiée (herbe, arbuste et culture) et dont la dynamique est assurée par une pluviométrie qui évolue dans le temps.

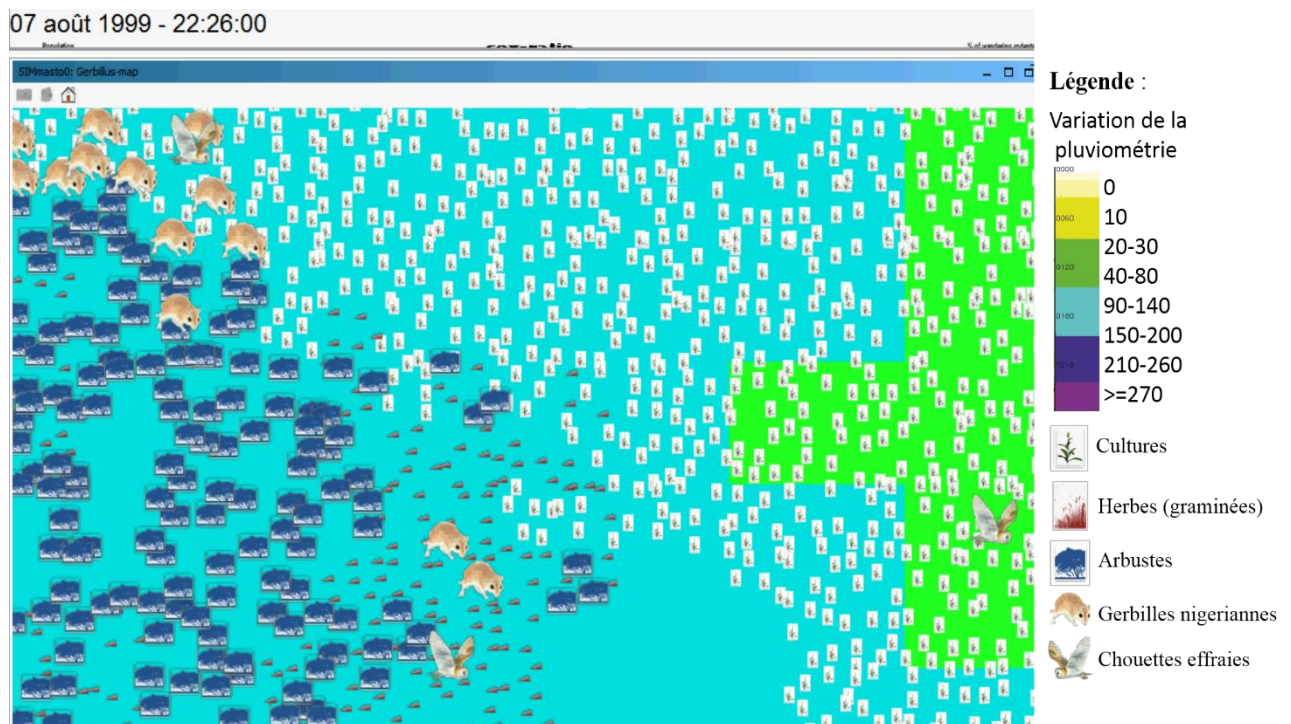


Figure 21 : capture d'écran de la simulation d'agents gerbilles et chouettes évoluant dans leur environnement

## II- Discussion

La complexité du modèle obtenu réside dans sa dimension multi-thématique. En effet, les connaissances de la gerbille et de son prédateur sont obtenues à plusieurs échelles d'observation et couvrent plusieurs niveaux fonctionnels, de l'ADN aux écosystèmes. Pour mettre au point un modèle répondant aux besoins formulés dans l'introduction, nous avons choisi d'utiliser un système multi-agents basé sur l'architecture PDE. Cette architecture nous a permis de mettre en œuvre les processus biologiques et trophiques des agents cognitifs représentés. Ainsi, ces derniers se révèlent capables de faire un raisonnement complexe qui aboutit sur une interaction entre eux ou entre agents et l'environnement.

Cependant, ce choix de paradigme PDE est une des nombreuses possibilités qui pourraient être proposées. Il aurait été possible d'adopter les formalismes tels que, « Capture / Sortie » (Vo, 2012), « Emergence / Création (de groupe) » (Servat *et al*, 1998), « Agrégation / Désagrégation » (Navarro *et al*, 2011, Soyez *et al*, 2012), « Agent / Groupe / Rôle » (Ferber *et al*, 2004) ou « Croyance / Désir / Intention » (Caillou *et al*, 2017). Ce dernier paradigme très proche de l'architecture PDE a fait l'objet de plusieurs applications dans des domaines très variés. Parmi ces domaines d'applications nous pouvons citer : l'aéronautique, le comportement humain, les jeux vidéo, etc. Nonobstant le nombre important de publication, le paradigme BDI présente des freins comme la lourdeur de calcul avec la consommation de beaucoup de temps pour la simulation d'un agent (Adam *et al*, 2011) et donc pas adaptée aux simulations avec des milliers d'agents (Caillou *et al*, 2017), la difficulté d'appréhension pour les scientifiques non informaticiens (Taillandier *et al*, 2012), etc....

Étant donné, la nécessité d'une perpétuelle évolution de notre modèle, il pourrait être judicieux de concilier les paradigmes PDE et BDI. En effet, l'incorporation sur l'architecture PDE de l'aspect intention du paradigme BDI serait un plus dans la gestion du comportement des agents cognitifs. Par exemple, un agent gerbille qui prend l'intention d'accomplir une action dans un pas de temps antérieur verrait son intention gardée dans une pile si elle n'est pas satisfaite. Cette approche visant à étendre le paradigme PDE devrait produire une architecture qui offre un meilleur support pour la modélisation des processus biologiques et trophiques des agents étudiés, en maintenant la force de l'architecture, mais aussi, en répondant à certaines de ses faiblesses.



## CONCLUSION

L'objet de ce travail était de simuler les comportements de la gerbille et de son prédateur la chouette effraie dans un environnement physique dynamique. Ainsi, l'approche utilisée et le paradigme adopté nous ont permis de développer un modèle traitant les interactions complexes à la fois entre les agents et leur environnement et entre les agents à travers l'environnement. Les deux travaux combinés ont apporté à la plateforme :

- ✓ une végétation diversifiée et dynamique,
- ✓ une pluviométrie évolutive,
- ✓ la possibilité d'extraire des zones de questionnement fixées par les biologistes,
- ✓ l'utilisation des coordonnées géodésiques,
- ✓ les aspects énergétiques de l'activité des agents,
- ✓ la prise en compte de stimulateur de saison de reproduction
- ✓ les bases d'une relation proie-prédateur,
- ✓ etc.

En dehors de l'étude de cas sur la gerbille, les fonctionnalités développées restent cohérentes avec toutes les études de cas implémentés grâce au modèle général SimMasto et peuvent être réutilisées par celles à venir. Ces aspects permettent au modèle gerbille d'être en phase avec les exigences du modèle général dont le soubassement est fondé sur le souci de robustesse et de flexibilité.

Le travail mené durant ce stage nous a permis de nous accoutumer avec la littérature mais également de consolider nos connaissances sur les systèmes multi-agents. Ce travail a fait également l'objet d'une communication au colloque étudiant : « Le printemps de Baillarguet » à Montpellier en mai 2017. Cette présentation nous a permis de recevoir un point de vue extérieur sur le travail en cours mais aussi de nous familiariser avec un public nouveau.

Bien entendu, de multiples améliorations peuvent être apportées au modèle.

- ✓ La calibration doit être poursuivie et ses résultats confirmés avec des procédures de validation robustes du type analyse de sensibilité multi-paramètres, qui sont adéquates pour ce type de modèle.
- ✓ La relation proie-prédateur doit aussi connaître une suite avec la prise en compte d'autres comportements de la chouette comme la reproduction.

- ✓ Nos agents, bien que cognitifs, ne font pas d'apprentissage avec des possibilités de mémorisation.
- ✓ L'utilisation des agents hystérétiques apporterait plus de réalité au modèle avec mémorisation d'un chemin, d'un milieu dangereux.
- ✓ etc.

NB : un article intitulé « *Multi-Thematic Computer Model of a Rodent Species Invasion; Case of the Nigerian Gerbil (Gerbillus nigeriae) in Northern Senegal* » est prévu pour une valorisation du travail effectué.

Nous avons enfin pu bénéficier d'un financement de thèse du Centre d'Excellence Africain en Mathématiques Informatique et TIC (CEAMITIC) sur le sujet « *Modélisation informatique des conditions de propagation et de transmission de zoonoses induites par les rongeurs nouvellement invasifs en milieu rural au Sénégal* ».

## REFERENCES

- Adam, C., Gaudou, B., Hickmott, S. et Scerri, D., 2011. Agents BDI et simulations sociales. *Revue d'Intelligence Artificielle (RIA)-Num. Spec. Simul. Multi-Agent* 25: 11 - 42.
- Allan, R. J., 2009. Survey of agent based modelling and simulation tools. No. DL-TR-2010-007.
- Bâ, K., Thiam, M., Dobigny, G., Granjon, L., Mane, Y., Volobouev, V., Duplantier, J. M., 2006. Hypothesis on the origin of the invasion of Senegal by *Gerbillus nigeriae* based on chromosomal data. *mammalia* 70: 303–305.
- Martinez, B. J. V. and Perez, F. M., 2011. Robotic control based on the human nervous system. *International Journal of Artificial Intelligence & Applications* 2: 107.
- Bille, J., 1974. Recherches écologiques sur une savane sahelienne du Ferlo septentrional, Sénégal : 1972, année sèche au sahel. *Station d'Ecologie ORSTOM, B.F. '20 Richard Toll, Sénégal*.
- Bordini, R. H., Hübner, J. F. and Wooldridge, M., 2007. Programming multi-agent systems in AgentSpeak using Jason. *John Wiley & Sons*. Vol. 8.
- Bousquet, F., Page, L. C. et Müller, J. P., 2002. Modélisation et simulation multi-agent. *deuxiemes assises du GDRI3*.
- Bratman, M., 1987. Intention, plans, and practical reason. *Harvard University Press, Cambridge, Mass*.
- Bratman, M. E., Israel, D. J. and Pollack, M. E., 1988. Plans and resource-bounded practical reasoning. *Computational intelligence* 4: 349–355.
- Caillou, P., Gaudou, B., Grignard, A., Truong, C. Q. and Taillandier, P., 2017. A Simple-to-use BDI architecture for Agent-based Modeling and Simulation. *Advances in Social Simulation 2015*, 15–28. Springer.
- Camberlin, P., Martiny, N., Philippon, N. and Richard, Y., 2007. Determinants of the interannual relationships between remote sensed photosynthetic activity and rainfall in tropical Africa. *Remote Sensing of Environment* 106: 199 - 216.
- Collier, N., Howe, T. and North, M., 2003. Onward and upward: The transition to Repast 2.0. *Proceedings of the First Annual North American Association for Computational Social and Organizational Science Conference*, 5.
- Coquillard, P., Hill, DR. and Sime-Ngando, T., 1997. Modélisation et simulation d'écosystèmes: des modèles déterministes aux simulations à événements discrets. *Masson Paris*.
- Dennett, D. C., 1989. The intentional stance. *MIT press*.

- Drogoul, A., Amouroux, E., Caillou, P., Gaudou, B., Grignard, A. and Marilleau, N., 2013. Gama: A spatially explicit, multi-level, agent-based modeling and simulation platform. *International Conference on Practical Applications of Agents and Multi-Agent Systems*, 271–274. Springer.
- Duplantier, J. M., Granjon, L. et Bâ, K., 1991. Découverte de trois espèces de rongeurs nouvelles pour le Sénégal: un indicateur supplémentaire de la désertification dans le nord du pays. *Mammalia* 55: 313–315.
- Ferber, J., 1995. Les Systèmes Multi Agents: vers une intelligence collective.
- Ferber, J., 1997. Les systèmes multi-agents: un aperçu général. *Techniques et sciences informatiques* 16.
- Ferber, J., Gutknecht, O. and Michel, F., 2004. From agents to organizations: an organizational view of multi-agent systems. *Lecture notes in computer science*: 214–230.
- Ferber, J. and Müller, J. P., 1996. Influences and reaction: a model of situated multiagent systems. *Proceedings of Second International Conference on Multi-Agent Systems (ICMAS-96)*, 72–79.
- Fernández, D. R., Chamizo, J. M. G., Pérez, F. M. and Payá, A. S., 2004. Modeling the distributed control of the lower urinary tract using a multiagent system. *International Conference on Modeling Decisions for Artificial Intelligence*, 104–114. Springer.
- Fernández, D. R., Chamizo, J. G., Pérez, F. M. and Payá, A. S., 2006. Robust modelling of biological neuroregulators. *Engineering in Medicine and Biology Society, 2005. IEEE-EMBS 2005. 27th Annual International Conference of the*, 2981–2984. IEEE.
- García, J. M. and Maciá, F., 2000. A Mobile Agent-based Model for a Node Regeneration System. *International Conference on Knowledge Based Computer Systems (KBCS 2000). Mumbai (India)*, 82–93.
- Genesereth, M. R. and Nilsson, N. J., 1987. Logical foundations of artificial. *Intelligence. Morgan Kaufmann* 58.
- Georgeff, M. P., Lansky, A. L. and Schoppers, M. J., 1987. Reasoning and planning in dynamic domains: An experiment with a mobile robot. *DTIC Document*.
- Georgeff, M., Pell, B., Pollack, M., Tambe, M. and Wooldridge, M., 1998. The belief-desire-intention model of agency. *International Workshop on Agent Theories, Architectures, and Languages*, 1–10. Springer.
- Georgeff, M. P. and Rao, A. S., 1996. A profile of the Australian artificial intelligence institute. *IEEE Expert: Intelligent Systems and Their Applications* 11: 89–92.
- Granjon, L., 2013. Modelling and scenarios of biodiversity.
- Granjon, L. et Duplantier, J. M., 2009. Les rongeurs de l’Afrique sahélo-soudanienne.

- Grignard, A., Taillandier, P., Gaudou, B., Vo, D. A., Huynh, N. Q. and Drogoul, A., 2013. GAMA 1.6: Advancing the art of complex agent-based modeling and simulation. *International Conference on Principles and Practice of Multi-Agent Systems*, 117–131. Springer.
- Grimm, V. and Railsback, S. F., 2005. Individual-based modeling and ecology. *BioOne*.
- Grimm, V. and Railsback, S. F., 2012. Pattern-oriented modelling: a ‘multi-scope’ for predictive systems ecology. *Phil. Trans. R. Soc. B* 367: 298–310.
- Hima, K., 2010. Biologie évolutive de *Gerbillus nigeriae* (Rodentia, Gerbillinae), principal ravageur des cultures céréalières au Niger : aspects chromosomique, morphologique et populationnel.
- Huber, M. J., 1999. JAM: A BDI-theoretic mobile agent architecture. *Proceedings of the third annual conference on Autonomous Agents*, 236–243. ACM.
- Huffman, G. J., Bolvin, D. T., Nelkin, E. J., Wolff, D. B., Adler, R. F., Gu, G., Hong, Y., Bowman, K. P. and Stocker, E. F., 2007. The TRMM multisatellite precipitation analysis (TMPA): Quasi-global, multiyear, combined-sensor precipitation estimates at fine scales. *Journal of Hydrometeorology* 8: 38–55.
- Huraux, T., 2015. Simulation multi-agent d’un système complexe: combiner des domaines d’expertise par une approche multi-niveau-Le cas de la consommation électrique résidentielle.
- IGN/SGN, 2013. Transformation entre systèmes géodésiques.
- Ingrand, F. F., Georgeff, M. P. and Rao, A. S., 1992. An architecture for real-time reasoning and system control. *IEEE expert* 7: 34–44.
- Le Fur, J., Mboup, P. A. and Sall, M., 2017. A simulation model for integrating multidisciplinary knowledge in natural sciences : Heuristic and application to wild rodent studies.
- Le, V. M., Gaudou, B., Taillandier, P. et Vo, D. A., 2013. A new BDI architecture to formalize cognitive agent behaviors into simulations. *Advanced Methods and Technologies for Agent and Multi-Agent Systems* 252: 395.
- Lowe, S., Browne, M., Boudjelas, S., De Poorter, M., 2000. 100 of the world’s worst invasive alien species: a selection from the global invasive species database.
- Pérez, F. M., García, C. J. M., Payá, A. S. and Fernández, D. F., 2005. Multiple mobile agent system framework suitable for pervasive computing.
- Marcenac, P., 2017. Modélisation et simulation par agents. Application aux systèmes complexes.
- Mboup, P. A., 2015. Des Connaissances à la Simulation Multi-agents: Modélisation orientée événements de la Colonisation du Rat noir au Sénégal par les Transports Humains sur un Siècle.

- Mboup, P. A., 2017. Modélisation orientée connaissances et événements de mondes à plusieurs échelles historiques et géographiques pour la simulation multi-agents de la diffusion du rat noir au Sénégal.
- McMurtry, A. and Henry, S., 2009. Knowers and phenomena: Two different approaches to interdisciplinarity and interprofessionalism. *Issues in Interdisciplinary Studies*.
- Meek, J., Wentworth, J. and Sebberson, D., 2001. The practice of interdisciplinarity: Complex conditions and the potential of interdisciplinary theory. *Issues in Interdisciplinary Studies*.
- Michel, F., 2006. Le modèle IRM4S: le principe Influence/Réaction pour la simulation de systèmes multi-agents. *JFSMA*, 9–22.
- Mikkola, H. and Willis, I., 1983. Owls of Europe. *Buteo Books Vermillion, SD USA*.
- Mooney, H. A., 1999. Species without frontiers. *Nature Publishing Group*.
- Navarro, L., Flacher, F. and Corruble, V., 2011. Dynamic level of detail for large scale agent-based urban simulations. *The 10th International Conference on Autonomous Agents and Multiagent Systems-Volume 2*, 701–708. International Foundation for Autonomous Agents and Multiagent Systems.
- Newell, W. H., 2001. A theory of interdisciplinary studies. *Issues in Integrative Studies*. No 19, p. 2.
- Nomao, A. et Gautun, J. C., 2001. Dynamique de l'abondance et reproduction de la population de *Gerbillus nigeriae* (Rodentia, Gerbillinae) dans la ferme de Kolo (Niger).
- Norling, E., 2004. Folk psychology for human modelling: Extending the BDI paradigm. *Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems-Volume 1*, 202–209. IEEE Computer Society.
- North, M. J., Collier, N. T., Ozik, J., Tatara, E. R., Macal, C. M., Bragen, M. and Sydelko, P., 2013. Complex adaptive systems modeling with Repast Symphony. *Complex adaptive systems modeling 1*: 3.
- Papert, S., 1980. Mindstorms: Children, computers, and powerful ideas. *Basic Books, Inc.*
- Parunak, H. V. D., Savit, R. and Riolo, R. L., 1998. Agent-based modeling vs. equation-based modeling: A case study and users' guide. *International Workshop on Multi-Agent Systems and Agent-Based Simulation*, 10–25. Springer.
- Pavard, B. and Dugdale, J., 2006. The contribution of complexity theory to the study of socio-technical cooperative systems. *Unifying themes in complex systems*, 39–48. Springer.
- Pérez, F. M., 2001. Modelos de administración de redes heterogéneas de computadores. Sistema de regeneración de nodos de red.
- Pérez, F. M., Chamizo, J. M. G., Payá, A. S. and Fernández, D. R., 2008. A robust model of the neuronal regulator of the lower urinary tract based on artificial neural networks. *Neurocomputing* 71: 743–754.

- Pokahr, A., Braubach, L. and Lamersdorf, W., 2005. Jadex: A BDI reasoning engine. *Multi-agent programming*: 149–174.
- Rao, A. S. and Georgeff, M. P., 1991. Modeling Rational Agents within a BDI-Architecture. *KR 91*: 473–484.
- Rao, A. S. and Georgeff, M. P., 1998. Decision procedures for BDI logics. *Journal of logic and computation* 8: 293–343.
- Sall, M., 2016. Conception, implémentation et simulation de la dynamique de l'environnement sahélien lié à l'invasion de la gerbille au Sénégal. *Mémoire de master professionnel M2ITIC* : 51.
- Schild, K., 1998. On the relationship between BDI logics and standard logics of concurrency. *International Workshop on Agent Theories, Architectures, and Languages*, 47–61. Springer.
- Servat, D., Perrier, E., Treuil, J. P. and Drogoul, A., 1998. When agents emerge from agents: Introducing multi-scale viewpoints in multi-agent simulations. *Multi-Agent Systems and Agent-Based Simulation*, 55–61. Springer.
- Shaw, K. L. and Wagner, K., 2008. Cricketsim: a Genetic and Evolutionary Computer Simulation. *Journal of Artificial Societies and Social Simulation* 11: 3.
- Shendarkar, A., Vasudevan, K., Lee, S and Son, Y. J., 2008. Crowd simulation for emergency response using BDI agents based on immersive virtual reality. *Simulation Modelling Practice and Theory* 16: 1415–1429.
- Sicard, B., 1987. Mécanismes écologiques et physiologiques de régulation des variations régulières et irrégulières de l'abondance des Rongeurs du Sahel (Burkina Faso). *Thèse d'Etat, USTL Montpellier*, 308p.
- Simon, H. A., 1962. The architecture of complexity. *Proceedings of the American philosophical society* 106: 467–482.
- Snyder, J. P., 1993. Flattening the earth: two thousand years of map projections. *The University of Chicago Press, Chicago London*.
- Soyez, J. B., Morvan, G., Dupont, D. and Merzouki, R., 2012. A methodology to engineer and validate dynamic multi-level multi-agent based simulations. *International Workshop on Multi-Agent Systems and Agent-Based Simulation*, 130–142. Springer.
- Taberlet, P., 1983. Evaluation du rayon d'action moyen de la chouette effaie, *Tyto alba* (Scopoli, 1769), à partir de ses pelotes de réjection.
- Taillandier, P., Therond, O. and Gaudou, B., 2012. A new BDI agent architecture based on the belief theory. Application to the modelling of cropping plan decision-making. *International environmental modelling and software society (iEMSs)*.
- Thiam, M., 2007. Les changements climatiques et l'invasion des gerbilles (Rongeurs, Muridés) au Sénégal : importance et causes du phénomène, compétition avec les espèces résidentes. *Thèse de Doctorat. Université Cheikh Anta Diop de Dakar*.

Thiam, M., Bâ, K. and Duplantier, J. M., 2008. Impacts of climatic changes on small mammal communities in the Sahel (West Africa) as evidenced by owl pellet analysis. *African Zoology* 43: 135–143.

Thomas, O. and Hinton, M., 1920. *Novitates Zoologicae*. London.

Vallée, J. L., 1999. La chouette effraie. *Delachaux et Niestlé*: 208.

Verhulst, P. F., 1845. Recherches mathématiques sur la loi d'accroissement de la population. *Nouveaux Mémoires de l'Académie Royale des Sciences et Belles-Lettres de Bruxelles* 18: 14–54.

Vo, D. A., 2012. An operational architecture to handle multiple levels of representation in agent-based models.

Wheeler, T. J., 2007. Analysis, modeling, emergence & integration in complex systems: a modeling and integration framework & system biology. *Complexity* 13: 60–75.

Wilensky, U. and Evanston, I., 1999. NetLogo : Center for connected learning and computer-based modeling. *Northwestern University, Evanston, IL*: 49–52.



## ANNEXE : codes source

### C\_ProtocolGerbil :

```
package simmast0.protocol;
import java.util.ArrayList;
import java.util.Calendar;
import java.util.List;
import java.util.TimeZone;
import java.util.TreeSet;
import com.vividsolutions.jts.geom.Coordinate;
import data.C_Chronogram;
import data.C_Event;
import data.C_Parameters;
import data.C_ReadRaster;
import data.constants.I_ConstantGerbil;
import data.constants.I_ConstantNumeric;
import data.converters.C_ConvertGeographicCoordinates;
import presentation.display.C_CustomPanelSet;
import presentation.epiphyte.C_InspectorVegetation;
import repast.simphony.context.Context;
import simmast0.C_ContextCreator;
import simmast0.C_LandscapeSahelianSavanna;
import thing.C_BarnOwl;
import thing.C_RodentGerbil;
import thing.C_Vegetation;
import thing.I_SituatedThing;
import thing.dna.C_GenomeAcacia;
import thing.dna.C_GenomeAnimalia;
import thing.dna.C_GenomeBalanites;
import thing.dna.C_GenomeFabacea;
import thing.dna.C_GenomeGerbillusNigeriae;
import thing.dna.C_GenomePoacea;
import thing.dna.C_GenomeTytoAlba;
import thing.dna.C_XsomePairSexual;
import thing.ground.C_SoilCellSavanna;
```

```
/** Initialize the simulation and manage inputs coming from the csv events file
 * @author J.Le Fur, 10.2014, rev. M.Sall 12.2015 */
```

```

public class C_ProtocolGerbil extends C_ProtocolFossorial implements I_ConstantGerbil, I_ConstantNumeric {
    //
    // FIELDS
    //
    private C_ConvertGeographicCoordinates geographicCoordinateConverter = null;
    protected C_InspectorVegetation vegetationInspector;
    //
    // CONSTRUCTOR
    //
    public C_ProtocolGerbil(Context<Object> ctxt) {
        super(ctxt);
        // Choose value of cell size
        cellSize.set(0, CELL_SIZE);
        vegetationInspector = new C_InspectorVegetation(this.landscape.getGrid());
        inspectorList.add(vegetationInspector);
        for (int i = 0; i < this.landscape.dimension_Ucell.width; i++) {
            for (int j = 0; j < this.landscape.dimension_Ucell.height; j++) {
                TreeSet<I_SituatedThing> agentList = this.landscape.getGrid()[i][j].getOccupantList();
                for (I_SituatedThing agent : agentList) {
                    if (agent instanceof C_Vegetation) vegetationInspector.addVegetationToList((C_Vegetation) agent);
                }
            }
        }
        C_CustomPanelSet.addVegetationInspector(vegetationInspector);
        // Create and build the dataFromChrono from the csv file
        chronogram = new C_Chronogram(I_ConstantGerbil.CHRONO_FILENAME);
    }
    //
    // METHODS
    //
    @Override
    /** Initialize the protocol with the raster origin */
    public void initProtocol() {
        this.geographicCoordinateConverter = new C_ConvertGeographicCoordinates(new Coordinate(
            I_ConstantGerbil.rasterLongitudeWest_LatitudeSouth_Udegree.get(0), I_ConstantGerbil.rasterLongitudeWest_LatitudeSouth_Udegree.get(1)));
        this.initFixedParameters();
        super.initProtocol();
    }
    @Override
    /** In gerbil protocol, the landscape raster values (i.e., affinity) contain the landcover values */
    protected void initLandscape(Context<Object> context) {

```

```

this.setLandscape(new C_LandscapeSahelianSavanna(context, C_Parameters.RASTER_URL, VALUE_LAYER_NAME, CONTINUOUS_SPACE_NAME));
// Comment the following lines to undisplay soil cells, JLF 10.2015, 11.2015
for (int i = 0; i < this.landscape.dimension_Ucell.width; i++) {
    for (int j = 0; j < this.landscape.dimension_Ucell.height; j++) {
        context.add(this.landscape.getGrid()[i][j]);
        this.landscape.moveToLocation(this.landscape.getGrid()[i][j], this.landscape.getGrid()[i][j].getCoordinate_Ucs());
        this.contextualizeVegetationInSavannaCell((C_SoilCellSavanna) this.landscape.getGrid()[i][j]);
    }
}
}
@Override
public void initCalendar() {
    protocolCalendar.set(1999, Calendar.JANUARY, 1);
}
@Override
/** Default rodents are of the Gerbillus nigeriae species */
public C_RodentGerbil createRodent() {
    return new C_RodentGerbil(new C_GenomeGerbillusNigeriae());
}
public C_BarnOwl createBarnOwl() {
    return new C_BarnOwl(new C_GenomeTytoAlba());
}
/** Add vegetation in soil cell at the requested position from the value of land cover (i.e., cell affinity) */
public void contextualizeVegetationInSavannaCell(C_SoilCellSavanna currentSoilCellSavanna) {
    String[] vegetationInLandcover = LANDCOVER_TO_VEGETATION.get(currentSoilCellSavanna.getAffinity());
    List<Coordinate> coordinateList = new ArrayList<Coordinate>();
    if (vegetationInLandcover != null) {
        Coordinate oneCoordinate = null;
        // Randomly add vegetation in soil cell, check that vegetation are not too close
        for (int i = 0; i < 3; i++) {
            for (int j = 0; j < 2; j++) {
                oneCoordinate = getVegetationCoordinate(DISTANCE_THRESHOLD, coordinateList, currentSoilCellSavanna);
                C_Vegetation oneVegetation = createVegetation(vegetationInLandcover[j]);
                if (oneVegetation != null) {
                    contextualizeNewAgentInCell(oneVegetation, currentSoilCellSavanna);
                    this.landscape.moveToLocation(oneVegetation, oneCoordinate);
                    coordinateList.add(oneCoordinate);
                }
            }
        }
    }
}
}
}

```

```

}
/** Compute one coordinate, compare its position with the other in the list and if it's correct return the position */
public Coordinate getVegetationCoordinate(double distanceThresHold, List<Coordinate> coordinateList, C_SoilCellSavanna currentCell) {
    Coordinate oneCoordinate = null;
    while (oneCoordinate == null) {
        oneCoordinate = new Coordinate(currentCell.getLineNo() + C_ContextCreator.randomGeneratorForInitialisation.nextDouble(), currentCell
        .getColNo() + C_ContextCreator.randomGeneratorForInitialisation.nextDouble());
        if (coordinateList.size() != 0) {
            int i = coordinateList.size();
            while ((oneCoordinate != null) && (i > 0)) {
                double vegetationDistance = .0;
                Coordinate secondCoord = coordinateList.get(i - 1);
                vegetationDistance = Math.sqrt(((oneCoordinate.x - secondCoord.x) * (oneCoordinate.x - secondCoord.x) + (oneCoordinate.y
                - secondCoord.y) * (oneCoordinate.y - secondCoord.y)));
                if (vegetationDistance < distanceThresHold) oneCoordinate = null;
                i--;
            }
        }
        return oneCoordinate;
    }
}
/** Create vegetation with genome given in args */
public C_Vegetation createVegetation(String vegetationInLandcover) {
    C_Vegetation oneVegetation = null;
    switch (vegetationInLandcover) {
        case SHRUBS :
            oneVegetation = new C_Vegetation(new C_GenomeBalanites());
            break;
        case CROPS :
            oneVegetation = new C_Vegetation(new C_GenomeFabacea());
            break;
        case GRASSES :
            oneVegetation = new C_Vegetation(new C_GenomePoacea());
            break;
        case TREES :
            oneVegetation = new C_Vegetation(new C_GenomeAcacia());
            break;
    }
    return oneVegetation;
}
@Override

```

```

public boolean isSimulationEnd() {
    // TODO MS 02-2017 if this function necessary to be declare
    if ((chronogram != null) && (chronogram.isEndOfChrono)) {
        A_Protocol.event(protocolCalendar.stringShortDate() + ": A_Protocol.isSimulationEnd(), chronogram exhausted; halting simulation",
            isNotError);
        return true;
    }
    return false;
}
/** Author MSall 10.2015<br>
 * @see A_Protocol#manageOneEvent */
public void manageOneEvent(C_Event event) {
    Coordinate coordinateCell_Ucs = null;
    if (event.whereX_Ucell == null) { // then: 1) suppose that y is also null, 2) double are values in decimal degrees
        coordinateCell_Ucs = this.geographicCoordinateConverter.convertCoordinate_Ucs(event.whereX_Udouble, event.whereY_Udouble);
        event.whereX_Ucell = (int) coordinateCell_Ucs.x;
        event.whereY_Ucell = (int) coordinateCell_Ucs.y;
    }
    if (coordinateCell_Ucs == null) coordinateCell_Ucs = new Coordinate(event.whereX_Ucell, event.whereY_Ucell);
    switch (event.type) {
        case RAIN :// file name example: 199901-PE-Rain.txt or 199901-TPE-Rain.txt
            String url; {
                Calendar calendar = Calendar.getInstance(TimeZone.getDefault());
                calendar.setTime(event.when_Ucalendar);
                // Month of simulation begin in 0 why we need to add 1 in the month value and put 0 before the month value between 0 and 8
                if (calendar.get(Calendar.MONTH) < 9)
                    url = RASTER_PATH + rainUrl_suffixRainFile.get(0) + calendar.get(Calendar.YEAR) + "0" + (calendar.get(Calendar.MONTH) + 1)
                        + rainUrl_suffixRainFile.get(1);
                else
                    url = RASTER_PATH + rainUrl_suffixRainFile.get(0) + calendar.get(Calendar.YEAR) + (calendar.get(Calendar.MONTH) + 1)
                        + rainUrl_suffixRainFile.get(1);
            }
            int[][] matriceLue = C_ReadRaster.txtRasterLoader(url);
            int imax = this.landscape.getDimension_Ucell().width;
            int jmax = this.landscape.getDimension_Ucell().height;
            // Change rain value of the cell with the value in the corresponding rain file
            for (int i = 0; i < imax; i++) {
                for (int j = 0; j < jmax; j++) {
                    int value = matriceLue[i][j];
                    ((C_SoilCellSavanna) this.landscape.getGrid()[i][j]).setRainLevel(value);
                    ((C_LandscapeSahelianSavanna) this.landscape).setRainGridValueLayer(value, i, j);
                }
            }
        }
    }
}

```

```

    }
}
    break;
case OWL_EVENT : {
    // Verify that the thing location is within the domain
    if ((coordinateCell_Ucs.x < width_heightRaster_Ukilometer.get(0)) && (coordinateCell_Ucs.y < width_heightRaster_Ukilometer.get(1))) {
        C_SoilCellSavanna eventSC = (C_SoilCellSavanna) this.landscape.getGrid()[event.whereX_Ucell][event.whereY_Ucell];
        C_BarnOwl one_barnOwl = this.createBarnOwl();
        contextualizeNewAgentInCell(one_barnOwl, eventSC);
        this.landscape.moveToLocation(one_barnOwl, coordinateCell_Ucs);
        one_barnOwl.actionMakeNest();
    }
}
    break;
case GERBIL_EVENT : {
    if ((coordinateCell_Ucs.x < width_heightRaster_Ukilometer.get(0)) && (coordinateCell_Ucs.y < width_heightRaster_Ukilometer.get(1))) {
        C_SoilCellSavanna eventSC = (C_SoilCellSavanna) this.landscape.getGrid()[event.whereX_Ucell][event.whereY_Ucell];
        C_RodentGerbil oneRodent = this.createRodent();
        this.addSexToRodent(oneRodent, event.value1);
        contextualizeNewAgentInCell(oneRodent, eventSC);
        this.landscape.moveToLocation(oneRodent, coordinateCell_Ucs);
        if (event.value2 != null) oneRodent.setThisName(oneRodent.getThisName() + NAMES_SEPARATOR + "n°" + event.value2);
        // Randomly define age of Gerbil
        int ageOfRodent = (int) (C_ContextCreator.randomGeneratorForInitialisation.nextDouble() * ((C_GenomeAnimalia) oneRodent
        .getGenome()).getMaxAge_Uday());
        oneRodent.setAge_Uday(ageOfRodent);
    }
}
    break;
}
super.manageOneEvent(event);
}
// TODO MS 2016-09 randomly add Gerbil in soil cell
public void randomlyAddGerbil(Coordinate coordinate_Ucs, String gender) {
    if ((coordinate_Ucs.x < width_heightRaster_Ukilometer.get(0)) && (coordinate_Ucs.y < width_heightRaster_Ukilometer.get(1))) {
        C_SoilCellSavanna soilCell = (C_SoilCellSavanna) this.landscape.getGrid()[Coordinate.X][Coordinate.Y];
        C_RodentGerbil oneRodent = this.createRodent();
        this.addSexToRodent(oneRodent, gender);
        contextualizeNewAgentInCell(oneRodent, soilCell);
        oneRodent.setThisName(GERBIL_EVENT + "-UnknowName");
        int ageOfRodent = (int) (C_ContextCreator.randomGeneratorForInitialisation.nextDouble() * ((C_GenomeAnimalia) oneRodent.getGenome())

```

```

        .getMaxAge_Uday());
        oneRodent.setAge_Uday(ageOfRodent);
    }
}
public void addSexToRodent(C_RodentGerbil one_rodent, String sex) {
    if ("MF".contains(sex)) { // determine its gender
        String rodentName = one_rodent.getThisName();
        one_rodent.setThisName(rodentName.substring(0, 1) + sex + rodentName.substring(2));
        int sexeGene = SEX_GENE_Y;
        boolean isMale = true;
        if (sex.equalsIgnoreCase("F")) {
            sexeGene = SEX_GENE_X;
            isMale = false;
        }
        ((C_GenomeGerbillusNigeriae) one_rodent.getGenome()).setGonosome(new C_XsomePairSexual(sexeGene));
        one_rodent.setMale(isMale);
    }
    // else A_Protocol.event("Protocol Gerbil, gerbil event - gender unknown: " + sex, isError);
}

@Override
public String toString() {
    return "protocolGerbil";
}

@Override
public void readUserParameters() {
    super.readUserParameters();
    String[] raster_Parameters = RASTER_PARAMETERS.get(C_Parameters.RASTER_URL.toLowerCase());
    if (raster_Parameters != null) {
        C_Parameters.RASTER_URL = (raster_Parameters[0]);
        rainUrl_suffixRainFile.set(0, raster_Parameters[1]);
        rainUrl_suffixRainFile.set(1, raster_Parameters[2]);
        width_heightRaster_Ukilometer.set(0, Integer.parseInt(raster_Parameters[3]));
        width_heightRaster_Ukilometer.set(1, Integer.parseInt(raster_Parameters[4]));
        rasterLongitudeWest_LatitudeSouth_Udegree.set(0, Double.parseDouble(raster_Parameters[5]));
        rasterLongitudeWest_LatitudeSouth_Udegree.set(1, Double.parseDouble(raster_Parameters[6]));
    }
}
}
}

```

## C\_RodentGerbil :

```
package thing;
import java.util.TreeSet;
import data.constants.I_ConstantGerbil;
import simmasto0.protocol.A_Protocol;
import thing.dna.C_GenomeFabacea;
import thing.dna.C_GenomePoacea;
import thing.dna.I_DiploidGenome;
import thing.ground.C_BurrowSystem;
import thing.ground.C_SoilCell;

/** Case study on Gerbillus nigeriae: gerbils choose food, partners or shelter depending on their energetic state and their perceived environment.
 * @author Moussa Sall, 2015, rev. 2016, JLF 04.2017 */
public class C_RodentGerbil extends C_RodentFossorial implements I_ConstantGerbil {
    //
    // FIELD
    //
    protected boolean methoxybenzoxazolinone; // 6-methoxybenzoquinone is a molecule produced by young shoots that may trigger rodents'reproduction
    //
    // CONSTRUCTOR
    //
    public C_RodentGerbil(I_DiploidGenome genome) {
        super(genome);
    }
    //
    // METHODS
    //
    /** Add vegetation in the whole cell to perceived list / M.Sall 10.2015
     * @see A_VisibleAgent#perception
     * @return TreeSet<I_situated_thing> Perceived things including vegetation objects */
    @Override
    protected TreeSet<I_SituatedThing> perception() {
        TreeSet<I_SituatedThing> perceivedThings = super.perception();
        TreeSet<I_SituatedThing> cellOccupants = this.currentSoilCell.getOccupantList();
        for (I_SituatedThing agent : cellOccupants)
            if (agent instanceof C_Vegetation) perceivedThings.add(agent);
        return perceivedThings;
    }
    /** If dawn, rest in burrow, else if owl: hide, else if reproduction, else if hungry, else uses A_Animal deliberation (super)<br>
     * Author MS 2016, rev. JLF&MS 05.2017.
```



```

* @return list of candidates selected */
@Override
protected TreeSet<I_SituatedThing> deliberation(TreeSet<I_SituatedThing> perceivedThings) {
    // Search for shelter if night ends
    if (A_Protocol.protocolCalendar.isDawn()) { // Put gerbil in existing burrow system or dig burrow and get inside
        if (!(this.currentSoilCell instanceof C_BurrowSystem)) {
            TreeSet<I_SituatedThing> shelter = this.chooseShelter(perceivedThings);
            if (shelter.isEmpty()) {
                C_BurrowSystem newBurrow = (C_BurrowSystem) C_BurrowSystem.createAndPutBurrowInContainer(this.currentSoilCell);
                myLandscape.moveToLocation(newBurrow, myLandscape.getThingCoord_Ucs(this)); // Place burrow at the rodent position
                shelter.add(newBurrow);
            }
            return shelter;
        }
    }
    else {
        for (I_SituatedThing onePerceivedThing : perceivedThings)
            if (onePerceivedThing instanceof C_BarnOwl) return this.chooseShelter(perceivedThings);
        // Methoxybenzoquinone
        if (this.methoxybenzoxazolinone && this.isSexualMature()) return this.choosePartner(perceivedThings);
        if (this.isHungry()) return this.chooseFood(perceivedThings);
    }
    return super.deliberation(perceivedThings);
}
/** Use list perceived things and choose the candidate shelter which can be one burrow, a vegetation list or nothing */
public TreeSet<I_SituatedThing> chooseShelter(TreeSet<I_SituatedThing> perceivedThings) {
    TreeSet<I_SituatedThing> shelters = new TreeSet<I_SituatedThing>();
    for (I_SituatedThing onePerceivedThing : perceivedThings) {
        if (onePerceivedThing instanceof C_BurrowSystem) { // Choose burrow and stop
            shelters.clear();
            shelters.add(onePerceivedThing);
            break;
        }
        else if ((onePerceivedThing instanceof C_Vegetation) && ((C_Vegetation) onePerceivedThing).getVegetationType().equals(SHRUBS)
            && canInteractWith((C_Vegetation) onePerceivedThing)) {
            shelters.add(onePerceivedThing);
        }
    }
    return shelters;
}
/** Use list perceived things and choose grass or crop */

```

```

public TreeSet<I_SituatedThing> chooseFood(TreeSet<I_SituatedThing> perceivedThings) {
    TreeSet<I_SituatedThing> vegetationSelected = new TreeSet<I_SituatedThing>();
    for (I_SituatedThing onePerceivedThing : perceivedThings) {
        if ((onePerceivedThing instanceof C_Vegetation) && (((C_Vegetation) onePerceivedThing).getGenome() instanceof C_GenomePoacea)
            || (((C_Vegetation) onePerceivedThing).getGenome() instanceof C_GenomeFabacea)) && canInteractWith(
                (C_Vegetation) onePerceivedThing)) {
            vegetationSelected.add(onePerceivedThing);
            break;
        }
    }
    return vegetationSelected;
}
/** Use list perceived thing to choose the potential partner to interact and return the list */
protected TreeSet<I_SituatedThing> choosePartner(TreeSet<I_SituatedThing> perceivedThings) {
    TreeSet<I_SituatedThing> rodentCandidate = new TreeSet<I_SituatedThing>();
    for (I_SituatedThing onePerceivedThing : perceivedThings) {
        if (onePerceivedThing instanceof C_RodentGerbil) { // add potential partner in the interact list
            rodentCandidate.add(onePerceivedThing);
        }
    }
    return rodentCandidate;
}
/** Check distance between this and the vegetation and return true if he can interact with her */
public boolean canInteractWith(C_Vegetation one_vegetation) {
    double distanceGerbilToVegetation_Umeter = this.getCurrentSoilCell().getCoordinate_Umeters().distance(one_vegetation.getCoordinate_Umeters());
    if ((vegetationCoverageRadius_Umeter + this.sensing_UmeterByTick) > distanceGerbilToVegetation_Umeter) {
        if (distanceGerbilToVegetation_Umeter > vegetationCoverageRadius_Umeter) { // Move on the radius of vegetation cover
            double distanceToBrowse_Umeter = distanceGerbilToVegetation_Umeter - vegetationCoverageRadius_Umeter;
            nextMove_Umeter.x = (Math.abs(distanceGerbilToVegetation_Umeter - distanceToBrowse_Umeter) / distanceGerbilToVegetation_Umeter) *
                one_vegetation.getCoordinate_Umeters().x + Math.abs(distanceToBrowse_Umeter / distanceGerbilToVegetation_Umeter) * this
                .getCoordinate_Umeters().x;
            nextMove_Umeter.y = (distanceToBrowse_Umeter / (distanceGerbilToVegetation_Umeter - distanceToBrowse_Umeter)) *
                Math.abs(one_vegetation.getCoordinate_Umeters().y - this.getCoordinate_Umeters().y) + this.getCoordinate_Umeters().y;
        }
        return true;
    }
    return false;
}
/** Use list candidate shelter and interact with the nearest */
public void actionHide(I_SituatedThing oneThingPerceived) {
    // Put Gerbil in the burrow
}

```

```

// TODO MS 2017.01 de jlf revoir avec le terrier dans le target
if (oneThingPerceived != null) {
    if (oneThingPerceived instanceof C_BurrowSystem) this.actionEnterBurrow((C_BurrowSystem) oneThingPerceived);
    else this.actionEnterBurrow(this.actionDig());
}
else this.setHasToDisperse(true); // When rodent can't hide
}
/** Consume vegetation biomass and return gain energy */
protected void actionEat(I_SituatedThing perceivedThings) {
    if (perceivedThings != null) {
        C_Vegetation oneVegetation = (C_Vegetation) perceivedThings;
        oneVegetation.setBiomass_Ugram(oneVegetation.getBiomass_Ugram() - dailyConsumptionOfGerbil_Ugram);
        energy_Ukcal += alimentation_Ukcal;
        this.methoxybenzoxazolinone = oneVegetation.getMethoxybenzoxazolinone();
    }
}
/** Generate a new animal, necessary to use spawn
 * @see A_Amniote#spawn */
@Override
public A_Animal giveBirth(I_DiploidGenome genome) {
    return new C_RodentGerbil(genome);
}
@Override
public void step_Utick() {
    if (this.currentSoilCell.isFull()) this.setHasToDisperse(true);
    // If night, rodent comes out of the burrow
    if (A_Protocol.protocolCalendar.isNightTime() && (this.getCurrentSoilCell() instanceof C_BurrowSystem)) actionRandomExitOfBurrow();
    I_SituatedThing destination = selectDestination(deliberation(perception()));
    if ((destination instanceof C_BurrowSystem) || ((destination instanceof C_Vegetation) && ((C_Vegetation) destination).getVegetationType().equals(SHRUBS))) {
        this.actionHide(destination);
    }
    else if (destination instanceof C_RodentGerbil) this.actionInteract((C_Rodent) destination);
    else if ((destination instanceof C_Vegetation) && !((C_Vegetation) destination).getVegetationType().equals(SHRUBS))
        this.actionEat(destination);
    // JLF 10.2016 TODO MS rodent can it move if it's trapped on board?
    if (((C_SoilCell) this.getCurrentSoilCell()).getAffinityLandPlot() != null) {
        if (preMature && ((C_SoilCell) this.getCurrentSoilCell()).getAffinityLandPlot().isFull()) {
            this.setHasToDisperse(true);
            this.setNewRandomMove();
            this.actionMoveToDestination();
        }
    }
}

```

```

        if (destination instanceof C_RodentGerbil) this.actionInteract((C_Rodent) destination);
    else if ((destination instanceof C_Vegetation) && !((C_Vegetation) destination).getVegetationType().equals(SHRUBS)) this.actionEat(destination);
    }
    // JLF 10.2016 TODO MS rodent can it move if it's trapped on board?
    if (((C_SoilCell) this.getCurrentSoilCell()).getAffinityLandPlot() != null) {
        if (preMature && ((C_SoilCell) this.getCurrentSoilCell()).getAffinityLandPlot().isFull()) {
            this.setHasToDisperse(true);
            //this.getRandomDisplacement();
            this.actionMoveToDestination();
        }
    }
    if ((this.getCurrentSoilCell() instanceof C_BurrowSystem) && this.hasToDisperse) actionRandomExitOfBurrow();
} else {
}
}
}
}

```

## C\_BarnOwl :

```

package thing;
import java.util.TreeSet;
import data.constants.I_ConstantGerbil;
import simmast0.C_ContextCreator;
import simmast0.protocol.A_Protocol;
import thing.dna.I_DiploidGenome;
import thing.ground.C_BurrowSystem;
import thing.ground.C_Nest;
import thing.ground.I_Container;

public class C_BarnOwl extends A_Amniote implements I_ConstantGerbil {
    //
    // FIELD
    //
    protected C_Nest myNest;
    //
    // CONSTRUCTOR
    //
    public C_BarnOwl(I_DiploidGenome genome) {
        super(genome);
    }
}

```

```

//
// METHODS
//
@Override
protected TreeSet<I_SituatedThing> perception() {
    TreeSet<I_SituatedThing> perceptList = super.perception();
    TreeSet<I_SituatedThing> barnOwlPerceptList = new TreeSet<I_SituatedThing>();
    if (!perceptList.isEmpty()) {
        for (I_SituatedThing oneThing : perceptList) {
            if (((oneThing instanceof C_Rodent) && !(oneThing.getCurrentSoilCell() instanceof C_BurrowSystem)) || (oneThing instanceof C_Nest))
                barnOwlPerceptList.add(oneThing);
        }
    }
    return barnOwlPerceptList;
}
/** Get list of perceived things, create list of rodent gerbil and call interact function
 * @return empty list of soilcell
 * @param perceivedThings Authors MS 10.2016 */
@Override
public TreeSet<I_SituatedThing> deliberation(TreeSet<I_SituatedThing> perceivedThings) {
    if (A_Protocol.protocolCalendar.isDawn()) {
        if (this.currentSoilCell != myNest) {
            TreeSet<I_SituatedThing> newPerceivingList = new TreeSet<I_SituatedThing>();
            newPerceivingList.add(myNest);
            return newPerceivingList;
        }
    }
    else {
        if (isHungry()) return this.choosePrey(perceivedThings);
    }
    return super.deliberation(perceivedThings);
}
/** Choose the prey if barn owl need to eat and return rodent list perceived */
public TreeSet<I_SituatedThing> choosePrey(TreeSet<I_SituatedThing> perceivingRodent) {
    TreeSet<I_SituatedThing> rodentList = new TreeSet<I_SituatedThing>();
    for (I_SituatedThing oneThing : perceivingRodent) {
        if (oneThing instanceof C_Rodent) rodentList.add(oneThing);
    }
    return rodentList;
}
@Override

```

```

public void step_Utick() { // TODO MS de JLF 2017.04 je ne comprends pas
    // TODO MS 03.2017 change the target if barn owl need to eat
    if (A_Protocol.protocolCalendar.isNightTime() && (this.target == myNest) && isHungry()) {
        this.setTarget(null);
    }
    super.step_Utick();
}
@Override
protected boolean actionInteract(A_Animal prey) {
    if (this.isHungry()) {
        this.actionMoveToDestination();
        prey.setDead(true);
        this.energy_Ukcal += prey.energy_Ukcal;
        return true;
    }
    else return false;
}
/** Create the origin nest of barn owl and contextualize it */
public void actionMakeNest() {
    I_Container curentSoilCell = this.getCurrentSoilCell();
    myNest = new C_Nest(curentSoilCell.getAffinity(), currentSoilCell.getColNo(), curentSoilCell.getLineNo());
    C_ContextCreator.protocol.contextualizeNewAgentInCell(myNest, curentSoilCell);
    C_ContextCreator.protocol.contextualizeOldAgentInCell(this, myNest);
    this.energy_Ukcal--;
}
@Override
public A_Animal giveBirth(I_DiploidGenome genome) {
    return new C_BarnOwl(genome);
}
}

```