Cybergeo : European Journal of Geography

Systèmes, Modélisation, Géostatistiques

Patrice Langlois, Baptiste Blanpain et Eric Daudé

MAGéo, une plateforme de modélisation et de simulation multi-agent pour les sciences humaines

Avertissement
Le contenu de ce site relève de la législation française sur la propriété intellectuelle et est la propriété exclusive de l'éditeur.
Les œuvres figurant sur ce site peuvent être consultées et reproduites sur un support papier ou numérique sous réserve qu'elles soient strictement réservées à un usage soit personnel, soit scientifique ou pédagogique excluant toute exploitation commerciale. La reproduction devra obligatoirement mentionner l'éditeur, le nom de la revue, l'auteur et la référence du document.
Toute autre reproduction est interdite sauf accord préalable de l'éditeur, en dehors des cas prévus par la législation en vigueur en France.

Revues.org est un portail de revues en sciences humaines et sociales développé par le Cléo, Centre pour l'édition électronique ouverte (CNRS, EHESS, UP, UAPV).

Référence électronique
Patrice Langlois, Baptiste Blanpain et Eric Daudé, « MAGéo, une plateforme de modélisation et de simulation multi-agent pour les sciences humaines », *Cybergeo : European Journal of Geography* [En ligne], Systèmes, Modélisation, Géostatistiques, document 741, mis en ligne le 02 octobre 2015, consulté le 04 octobre 2015. URL : http://cybergeo.revues.org/27236 ; DOI : 10.4000/cybergeo.27236

Éditeur : CNRS-UMR Géographie-cités 8504
http://cybergeo.revues.org
http://www.revues.org

Document accessible en ligne sur :
http://cybergeo.revues.org/27236
Document généré automatiquement le 04 octobre 2015.
© CNRS-UMR Géographie-cités 8504

Patrice Langlois, Baptiste Blanpain et Eric Daudé

MAGéo, une plateforme de modélisation et de simulation multi-agent pour les sciences humaines

1 Introduction

1 La modélisation et la simulation de systèmes complexes (Phan & Amblard, 2007) spatialisés (Benenson & Torrens, 2004) et multi-niveaux (Langlois, 2010) touche de nombreuses thématiques transversales aux sciences humaines et sociales. Les évolutions des systèmes de villes (Pumain *et al.*, 1995) et les étalements urbains (Dubos-Paillard et al., 2003), la diffusion d'innovation (Daudé, 2003) ou d'épidémie (Badariotti *et al.*, 2007), les mobilités urbaines quotidiennes (Banos *et al.*, 2010) ou en situation de catastrophe (Tranouez *et al.*, 2012), les processus de ruissellements de surface (Delahaye *et al.*, 2002) ou les représentations spatiales dans les choix résidentiels (Bonnefoy, 2003) sont quelques exemples significatifs des champs d'exploration possibles. Des plateformes multi-agent sont à la disposition des chercheurs pour décrire, modéliser et simuler ces systèmes complexes. Ces logiciels proposent une grande variété dans les approches de modélisation, et leur utilisation demande plus ou moins de compétences en informatique.

2 Parmi les plateformes existantes[1], on peut distinguer d'un côté celles dans lesquelles la description du modèle se fait via un langage de programmation généraliste comme C++ ou Java, et qui utilisent des bibliothèques de structures et de fonctions adaptées au contexte multi-agent : c'est par exemple le cas de Mason (Luke *et al.*, 2005), de Swarm (Minar *et al.*, 1996) ou de Repast (Collier, 2003). D'autres plateformes proposent un langage de programmation davantage adapté aux modèles multi-agent, et présentent donc un progrès par rapport aux précédentes, mais s'ouvrent encore difficilement à des utilisateurs n'ayant pas une formation d'informaticien : c'est notamment le cas de Netlogo (Wilensky, 1999), de CORMAS (Bousquet *et al.,* 1998) ou de GAMA (Drogoul *et al.*, 2013). En effet, ces plateformes reposent principalement sur la création de modèles par la programmation informatique, ce qui rend les modèles moins explicites, moins adaptés à la communication, et surtout moins accessibles aux utilisateurs issus des sciences sociales, pas toujours enclins à s'initier à un formalisme qui leur semble peu intuitif au premier abord. C'est pourquoi d'autres plateformes ont été développées, dans lesquelles le développement ne se fait plus uniquement par programmation informatique, mais qui utilisent pour l'essentiel une interface graphique pour construire un modèle, ce qui est *a priori* plus convivial et simple d'accès. StarLogo TNG (Klopfer *et al.*, 2009) et Repast Simphony (North *et al.*, 2013) se situent dans cette catégorie. Du point de vue de l'utilisateur, le principal objectif de ces logiciels est de fournir un langage de manipulation des connaissances qui soit à la fois rapidement compréhensible, facilement mobilisable, tout en offrant une grande richesse combinatoire. Cependant certains de ces projets montrent que ce qu'on gagne d'un côté - par exemple, le tout graphique comme StarLogo TNG -, on perd sur les autres - faire un modèle un peu complexe devient très rapidement fastidieux -. Nous avons donc cherché un compromis entre la clarté d'une construction graphique et la richesse combinatoire d'un langage.

3 La plateforme MAGéo fait partie de ce mouvement de création de plateformes conviviales dite *user-friendly*, et propose, dans une démarche totalement intégrée, la construction d'un modèle conceptuel sous forme essentiellement graphique, constitué d'entités - Agents, Organisations, Comportements - liées entre elles par différents types de relations. C'est cette construction qui sera ensuite la structure du modèle de simulation. En effet, une fois le modèle construit et paramétré, l'utilisateur peut instancier et initialiser son modèle. Il peut alors observer son évolution avec les cartes et les graphiques qu'il aura définis dans l'interface de simulation.

4 Dans cette article concernant la plateforme MAGéo, nous ne pouvons entrer d'emblée dans une étude approfondie concernant une application précise ou une focalisation spécifique. Nous

ferons ici une présentation assez généraliste à travers les concepts qui constituent le cadre de pensée que le modélisateur doit bien comprendre pour savoir ce qu'il peut faire avec la plateforme et surtout comment le faire, par les principes qui régissent la réalisation d'un modèle. Il n'est pas question non plus de faire ici une description exhaustive des fonctionnalités et des manipulations de mise en œuvre, ce qui est le rôle de la documentation. Néanmoins, nous voulons que le lecteur ait une idée assez précise de la manière d'interagir pour construire un modèle et de l'économie de moyens que la plateforme offre pour le réaliser. Cette économie réelle ne doit pas faire oublier que la modélisation reste une démarche délicate qui demande une réflexion importante et ne rend pas miraculeusement simple un modèle compliqué. Par contre, cette plateforme permet une construction incrémentale par une succession de cycles conception-construction-simulation-validation qui permet de complexifier progressivement le modèle tout en maîtrisant son fonctionnement.

2 Principaux concepts de MAGéo

2.1 Objectif général

5 Notre objectif est d'offrir une plateforme de modélisation et de simulation qui permette à tout chercheur sans formation préalable en programmation informatique de pouvoir développer un modèle dynamique et spatialisé en un nombre restreint d'étapes. Ceci ne signifie pas que le chercheur ne devra pas au préalable se familiariser avec les concepts qui constituent le cadre de pensée de MAGéo et les entités manipulables par lesquelles ce cadre se concrétise dans la plateforme. Mais cela signifie qu'une fois ces concepts acquis, il pourra utiliser pleinement les potentialités du logiciel pour développer ses modèles avec une liberté intellectuelle importante, tout en respectant ce cadre. A travers les mots choisis dans les dialogues, les structures proposées, la documentation, nous avons cherché à définir un contexte intellectuel qui soit dégagé le plus possible de toute technique informatique. Nous avons voulu aussi définir des entités conceptuelles associées à des formes graphiques explicites permettant de construire un modèle simple à comprendre et dont la combinatoire soit riche de possibilités. Ce système d'entités manipulables est le méta-modèle AOC, Agent – Organisation – Comportement (Langlois *et al.*, 2010, 2014). Ce méta-modèle s'inscrit à la fois dans la théorie des systèmes complexes, l'approche distribuée des systèmes multi-agent spatialisés, mais aussi dans la dynamique des systèmes. MAGéo offre donc un outil transdisciplinaire pouvant être utilisé dans les différentes disciplines des sciences humaines (Varenne, 2011).

2.2 Le principe méthodologique de base : l'intégration du processus de modélisation

6 En général, la construction d'un modèle commence par la réalisation d'un ***modèle conceptuel*** qui permet de définir l'***ontologie*** du modèle, principalement ses entités fondamentales et leurs relations. Cette étape est classiquement représentée par différents schémas, dans le cadre d'une méthodologie de conception, par exemple en UML (Unified Modeling Langage). Ce niveau conceptuel étant validé, on construit le modèle définitif dans un autre cadre technique, en le programmant dans un langage informatique selon les spécifications définies par le modèle conceptuel. On obtient alors un modèle de simulation, sous forme d'application exécutable sur un ordinateur. Bien souvent l'application développée est très éloignée du modèle conceptuel initial, parce qu'ils sont issus de contextes techniques très différents : un schéma UML et un programme en Java par exemple. De plus le modèle final est dans la plupart des cas réalisé par d'autres personnes que les concepteurs initiaux, la logique de départ est alors noyée dans une forêt de détails techniques, de choix de variables, de structures, de paramètres cachés et d'algorithmes. Si bien qu'il est parfois difficile de vérifier l'adéquation entre le modèle conceptuel et le modèle de simulation et qu'il est compliqué de rendre le modèle reproductible, pour pouvoir le faire vérifier, valider ou adapter par d'autres équipes, ce qui est pourtant un critère de scientificité important.

7 Dans MAGéo, les trois phases que sont la conception, le développement et le fonctionnement sont totalement intégrées dans une même application. Ceci permet une simplification

considérable du processus de réalisation d'un modèle, ainsi qu'une grande lisibilité du modèle final, puisque l'on peut descendre progressivement dans les détails du modèle conceptuel directement à partir de sa représentation graphique, contrairement à un modèle entièrement codé dans un langage de programmation. Néanmoins, cette intégration masque deux niveaux de réalisation qu'il importe de comprendre afin d'éviter toute confusion.

8 ***Le niveau conceptuel*** est celui de l'ontologie du modèle, où les entités sont abordées à un niveau abstrait. Par exemple un agent ne sera ici qu'une notion abstraite qui ne décrit pas un objet précis mais définit les propriétés communes à tous les objets de sa « classe ». Le niveau conceptuel définit donc la structuration de l'information des entités d'une classe donnée, mais ne définit pas les valeurs à stocker dans chacune de ces entités.

9 ***Le niveau physique*** est celui de la simulation proprement dite du modèle, de son exécution, de son fonctionnement. Ce modèle physique s'inscrit dans le temps. Pour spécifier qu'un agent est considéré à ce niveau, on parlera donc d'***agent physique***. Pour passer du niveau conceptuel au niveau physique, deux opérations sont nécessaires. Tout d'abord l'***instanciation*** qui crée en mémoire, pour chaque classe d'entité, une collection d'entités individualisées à partir des informations propres à sa classe. Ensuite l'***initialisation*** qui renseigne pour $t = t_0$ (l'instant initial), les différentes variables d'état et la géométrie des entités instanciées. Le système est alors dans son état initial, prêt à évoluer dans le temps. Ainsi un agent physique d'une certaine classe possède un état qui s'inscrit dans l'espace, la matière et le temps de la simulation du modèle.

2.3 Le concept d'agent et de classe d'agent

10 Au niveau conceptuel, la notion d'agent correspond à la notion de ***classe agent***. Une telle classe peut représenter un *agent unique* ou un *agent générique*. Cette classe d'agent produira au niveau physique une seule instance si c'est un agent unique et plusieurs s'il est générique.

11 Tout modèle dans MAGéo commence par un agent unique qui représente la totalité du système modélisé. Un agent générique caractérise une collection d'agents de même structure, mais qui se différencient les uns des autres, notamment par leur nom, par des valeurs distinctes de leurs attributs, par leur contenu interne, ou encore par leur géométrie ou leur extension spatiale.

12 Tout ce qui concerne les liens entre les agents, tel que le voisinage, la localisation ou la composition est du ressort de l'organisation.

2.4 Le concept d'organisation

13 L'organisation est le concept de MAGéo qui permet de spécifier ce qu'il a de plus dans un ***système*** d'agents que dans l'***ensemble*** de ces mêmes agents. Ce qu'il y a de plus, c'est la structure qui permet d'unifier et de faire interagir ses éléments. Si l'agent conceptualise la notion d'***individualité***, l'organisation conceptualise celle de ***collectivité***, au sens où non seulement une organisation contient une collection d'agents, mais décrit en plus un ensemble de liens objectifs entre ces agents qui explicitent comment ils sont organisés entre eux, soit spatialement, soit « socialement ». Ce que nous appelons ici organisation sociale est très général, et recouvre tout type de relation non spatiale qui permet de décrire les liens entre les constituants de l'organisation. Si un groupe d'adhérents à une coopérative entre bien dans la définition courante d'une organisation sociale, nous utiliserons aussi cette dénomination pour décrire des liens non spatiaux entre des entités physiques quelconques.

14 Une organisation est nécessairement encapsulée dans un agent de plus haut niveau, qu'elle organise. Dans MAGéo on appelle ***agent complexe*** (ou ***agent organisé***), un agent contenant au moins une organisation interne, et ***agent simple*** ou ***agent terminal*** un agent non organisé en interne. Une organisation contient une ou plusieurs classes d'agents qui définissent ses ***composants***. Ainsi un agent organisé est lui-même composé d'agents. Par cette propriété récursive selon le schéma Agent – Organisation – Agent, Magéo permet de définir un ***système.*** C'est donc une plateforme systémique dont les modèles se construisent en partant du plus général et en descendant progressivement dans les détails, par une démarche dite « *top-down* ».

15 Nous distinguons dans MAGéo deux catégories d'organisations : les organisations spatiales et les organisations sociales.

2.4.1 Organisations spatiales

16 MAGéo propose 4 types d'organisations spatiales prédéfinies : le maillage régulier en carrés et en hexagones, le maillage irrégulier pour la réalisation de cartes et le réseau.

17 Dans une organisation spatiale, divers éléments de la structure sont paramétrables. C'est le cas de la morphologie de l'espace qui est soit limitée, soit torique, et de la structure topologique des voisinages. Celle-ci peut être induite par la métrique choisie et permet de définir les liens de proximité ou de raccordement entre les entités géométriques, par exemple la frontière entre deux zones d'un maillage irrégulier, ou l'ordre des insertions des tronçons autour d'un nœud de réseau. Dans MAGéo, nous utilisons une structure topologique construite sur le concept de graphe planaire topologique et des cartes combinatoires (Langlois, 1994). Cette structure permet d'utiliser des cartes importées depuis un système d'information géographique (SIG), par exemple un fonds de carte administratif ou le réseau routier d'une ville. L'organisation est alors géoréférencée, elle possède un système de coordonnées que les agents peuvent utiliser.

18 Une organisation spatiale contient au moins une classe agent composants. C'est le cas dans un maillage régulier où la seule classe d'agent générique est un objet surfacique, le carreau. En revanche, dans le cas d'un réseau, l'organisation contiendra deux classes d'agents, des nœuds et des tronçons.

2.4.2 Organisations sociales

19 Comme nous l'avons dit, le terme social prend ici la signification très générale de non spatial, c'est-à-dire qu'une organisation sociale est un ensemble d'agents-composants sur laquelle on peut définir des relations qui définissent sa structure. Les structures sociales dans ce contexte étant très variées, il est difficile de proposer des structures prédéfinies, aussi contrairement aux structures spatiales qui sont prédéfinies, c'est au modélisateur de définir les relations qui structurent les agents d'une organisation sociale.

20 MAGéo propose deux paradigmes pour aborder l'organisation d'un système : le paradigme ***distribué*** et le paradigme ***compartimental***.

2.4.2.1 Le paradigme distribué

21 C'est celui qui considère un système comme composé d'un grand nombre de composants individualisés et en interaction, dont le nombre peut varier au cours de la vie du système. Les organismes vivants constitués d'un grand nombre de cellules en sont un exemple typique (Varela & Maturana, 1994). Dans MAGéo, c'est à travers le concept de ***population*** (formée d'individus) que se traduit le paradigme distribué.

2.4.2.2 Le paradigme compartimental

22 C'est celui qui considère un système comme composé d'entités peu nombreuses – appelées selon le contexte, compartiments, classes, organes, sous-systèmes – et ayant chacune sa fonctionnalité propre. Les compartiments sont en petit nombre et chacun est indispensable à l'intégrité du système. MAGéo propose ce type d'organisation sociale compartimentée.

23 Les compartiments d'une telle organisation sont souvent en interaction[2] de flux, où chacun consomme des flux entrants et produit des flux sortants. Ils sont souvent eux-mêmes organisés en sous-compartiments de niveau inférieur, et ainsi de suite. L'exemple d'un organisme vivant constitué de différents organes en est un exemple typique.

24 Ces deux paradigmes se complètent et ne sont pas antagonistes, de plus il n'y a pas de critère général permettant de choisir l'un ou l'autre. Par exemple le système constitué du découpage administratif des régions françaises (elles-mêmes subdivisées en départements et cantons), peut aussi bien se voir comme distribué que comme compartimental. Malgré l'absence de limite entre ces deux approches, elles sont néanmoins fonctionnellement très différentes. Des outils de modélisation compartimentale comme Stella ou SADT[3] ne parviendront pas par exemple à modéliser un système distribué comme un être vivant composé de milliards de cellules en interaction. Réciproquement un outil distribué comme les plateformes multi-agent n'ont que peu d'intérêt pour modéliser des systèmes très compartimentés, comme un moteur de fusée ou le fonctionnement d'une administration.

2.5 Le temps et les comportements des agents

25 Le concept de temps dans MAGéo est défini indépendamment des évènements qui peuvent se produire dans le modèle en cours de simulation. Le temps de la simulation est conçu comme une suite d'instants consécutifs séparés et ordonnés, codé par des entiers consécutifs à partir de 0 ou de tout autre entier t_0, appelé ***origine des temps***. La durée théorique de l'intervalle de temps entre deux instants consécutifs est constante et constitue un ***pas de temps*** élémentaire (ou ***itération***), au cours duquel le modèle évolue d'un instant t, à l'instant suivant t+1. Le pas de temps peut être relié à une ***unité de temps physique de référence*** (comme la seconde, le jour, l'année). Elle est associée au pas de temps du modèle par une constante entière qui exprime le nombre d'itérations du modèle qui égale la durée de l'unité de temps choisie. On voit donc que le temps de la simulation est fictif, c'est celui qui est affiché dans la zone d'observation et n'a rien à voir le temps réel de fonctionnement qui lui, ne dépend que de la puissance de traitement de l'ordinateur en rapport avec le nombre de calculs à effectuer.

26 Les comportements d'un agent ont pour but de décrire ce qui évolue au cours d'un pas de temps. Un comportement décrit une des dynamiques possibles d'une classe agent, et n'est pas une simple fonction de calcul ou d'action faisant évoluer des valeurs d'attribut et de position de l'agent. Un comportement décrit un mécanisme associé à l'agent basé sur le couple < *pertinence*, *action* >, qui commence par évaluer son environnement et son état interne avant d'agir. Le résultat de cette évaluation permet de définir la ***pertinence*** pour l'agent à effectuer l'***action*** définie dans le comportement. La pertinence n'est pas une simple condition logique qui entraine l'action si elle est vraie, mais elle permet à l'agent de faire un choix parmi plusieurs comportements concurrents à un instant donné. Si plusieurs comportements sont en concurrence, c'est le plus pertinent qui sera sélectionné. La pertinence demande ainsi plus de nuances que le VRAI ou le FAUX, elle peut prendre toute valeur réelle entre 0 et 1 (où 0 signifie *totalement faux* et 1 *totalement vrai,* c'est ce qu'on appelle la logique continue). Des opérateurs de combinaison de comportements élémentaires permettent d'associer plusieurs comportements pour en créer de plus complexes.

27 MAGéo intègre la notion de diagramme de transition, permettant un fonctionnement de type automate cellulaire, où une transition consiste simplement en un changement d'état d'une variable qualitative de l'agent. Une telle variable d'état est soumise à plusieurs transitions qui s'organisent en un graphe pour former un ***diagramme de transition***. Un tel diagramme peut être déterministe ou stochastique, selon la manière de définir les règles de transition. En fait dans MAGéo, un diagramme de transition est intégré dans un comportement complexe constitué des règles de transition du diagramme où chaque règle est formulée comme un comportement élémentaire de la forme $R_i = (\textit{évaluation_de_pertinence}, \textit{action_de_transition})$. L'ensemble des règles R_i sont globalement mises en concurrence entre elles, si bien qu'à chaque instant et pour chaque agent, c'est la règle la plus pertinente du diagramme qui s'exécute. Par exemple dans un modèle de diffusion d'une épidémie, chaque individu possède une variable d'état pouvant prendre les valeurs « sain » (S), « infecté » (I) ou « guéri » (R). La transition d'un état vers un autre dépend de la pertinence qui évalue l'état interne et l'environnement de l'individu tels que le nombre de voisins infectés et un taux de transmission (S → I), règles qui peuvent être stochastiques ou déterministes, comme une durée de guérison en jours par exemple (I → R).

2.6 Relations définies entre les entités

28 MAGéo permet donc de manipuler trois catégories principales d'entités : les agents, les organisations et les comportements. Au fur et à mesure de la construction d'un modèle conceptuel, les entités créées par le modélisateur sont reliées graphiquement les unes aux autres. Ces liens graphiques correspondent à des ***relations de construction*** imposées entre les entités par le méta-modèle de MAGéo. D'autres relations, que l'on pourrait qualifier de ***relations libres,*** doivent être définies « manuellement » par le modélisateur entre différentes classes d'agents : ce sont les relations de localisation, les relations binaires et les relations générales.

2.6.1 Les relations de construction du modèle conceptuel

29 ***L'approfondissement***: lorsque l'on considère un agent comme étant lui-même organisé en interne, un lien graphique est créé de l'agent vers l'organisation, pour représenter la relation suivante :

Approfondissement : Agent → Organisation

30 Une classe agent peut ainsi contenir plusieurs organisations, mais une organisation n'organise qu'une et une seule classe agent. Au niveau physique, une organisation qui approfondit une classe agent générique est instanciée autant de fois qu'il y a d'instances d'agents. Ainsi et comme pour l'agent, au niveau conceptuel on peut parler de classe d'organisation, qui produit au moment de l'instanciation, plusieurs instances d'organisation.

31 La ***composition*** : une organisation est composée d'au moins une classe agent dont les instances sont les composants de l'organisation, mais une classe agent ne compose au plus qu'une seule organisation. Le seul cas où une classe agent ne compose pas une organisation est l'agent-modèle, qui se trouve en haut de la hiérarchie des niveaux de complexité du modèle. La composition est définie par la relation :

Composition : Organisation → Agent

32 L'***héritage*** : cette relation permet de créer des sous-classes agent de la classe initiale, dite ancêtre. Chaque sous-classe hérite des attributs et comportements de la classe ancêtre, mais peut posséder également des attributs et des comportements propres. Une sous-classe ne peut redéfinir un attribut déjà présent dans sa classe ancêtre, mais s'il y a un comportement dans une classe ancêtre qui porte le même nom qu'un comportement dans une sous-classe, celui de la sous-classe masque celui de l'ancêtre. De plus les organisations internes à la classe ancêtre sont également internes à ses sous-classes.

Héritage : Sous-classe Agent → Classe-ancêtre Agent

33 Lorsqu'une classe agent est la classe-ancêtre d'une relation d'héritage, elle devient une classe abstraite, c'est-à-dire qu'elle ne sera pas instanciée. Comme une sous-classe peut elle-même être l'ancêtre d'une classe encore plus fine, seules les sous-classes terminales (qui ne sont ancêtres d'aucune autre) seront instanciées. Le programme d'action n'est présent que dans les classes terminales. Chaque classe terminale contient aussi les fonctionnalités d'instanciation et d'initialisation des instances de sa classe.

34 ***Lien de compétence*** : un dernier lien graphique est automatiquement créé lorsque l'on veut munir une classe agent d'un comportement. L'agent possède alors la compétence définie par ce comportement. Plusieurs classes agent peuvent avoir une même compétence et une même classe agent peut avoir plusieurs compétences.

2.6.2 Les relations libres

35 Un modèle conceptuel MAGéo peut être enrichi par des relations entre classes agents, permettant de localiser un agent sur un autre (c'est la relation de localisation), ou de spécifier des liens entre agents, par exemple la parenté, la hiérarchie, la possession, l'amitié, etc.

36 Le premier type de relation « libre[4] » que nous définissons est la relation de localisation qui est une fonction. Elle est évidemment fondamentale pour permettre la création de modèles spatialisés.

2.6.2.1 La relation de localisation

37 Une ***relation de localisation*** est une fonction entre un ensemble A d'agents appelés ***passagers*** vers un ensemble B d'agents appelés ***hôtes***. Comme toute fonction, un passager x (de A) est relié à un hôte y au plus (dans B), on dit alors que x est ***localisé sur*** y ou que x ***est passager de*** y ou que x est sur y. Eventuellement x n'a pas d'hôte, il n'est donc pas localisé. Une relation de localisation permet de référencer spatialement des agents qui ne sont pas spatialisés au départ, comme les individus d'une population. La relation de localisation est une fonction, puisqu'elle associe à chaque agent-passager un agent-hôte au plus. Au niveau physique cette relation peut être statique ou dynamique. Dans le cas d'agents mobiles, par exemple d'automobiles qui se déplacent sur les tronçons d'un réseau, les véhicules sont les passagers des tronçons (ceux-ci sont alors les hôtes des voitures). Le déplacement des véhicules modifie à chaque pas de temps

la fonction de localisation. On peut préciser des contraintes supplémentaires de cardinalité de la relation, par exemple en fixant le nombre maximum de voitures sur un tronçon.

Localisation : Agent-Passager → Agent-hôte

38 Une relation de localisation donne donc à chaque agent-passager sa position « qualitative », son adresse en quelque sorte sous forme d'un composant hôte, qui est le lieu unique sur lequel il se trouve à un instant donné.

39 Une relation de localisation confère de plus aux passagers une position géométrique dans le cas où son hôte est spatialisé. L'agent qui devient un passager dans une relation de localisation est alors muni d'un repère local qui mémorise sa position et son orientation dans l'espace de son hôte. A chaque action de déplacement du passager, la référence à son hôte est mise à jour, ainsi que sa position géométrique. La relation de localisation peut également être statique, c'est le cas de la localisation des agents de l'organisation « départements » qui sont localisés dans les agents de l'organisation « régions », définies toutes deux comme étant des maillages irréguliers (cartes topologiques zonales). Cette relation peut facilement être initialisée à partir d'un attribut des agents « départements » qui indiquent leur région hôte.

40 La distinction entre ***composant*** et ***passager*** est ainsi fondamentale dans MAGéo. Un composant est un élément constitutif de l'agent composé, il fait partie de son intégrité, de son intérieur. Alors qu'un passager ne l'est nullement, il est de passage, en contact, dessus. Ce lien n'est donc pas définitif si le passager est mobile. Par exemple l'habitant d'une ville n'est pas constitutif de la ville comme pourrait l'être un bâtiment ou une route. L'habitant est un passager mobile dans la ville, le bâtiment en est un composant. Ainsi les concepts de composant et de passager permettent de modéliser des situations très différentes tout en gardant un caractère très général et très intuitif.

41 Les termes de ***composant***, ***composé***, ***passager***, et ***hôte*** ne sont par ailleurs pas des catégories d'agents, ils expriment plutôt un contexte relatif aux différentes ***relations*** entre les entités d'un modèle.

2.6.2.2 Les relations binaires entre classes d'agents

42 Nous avons aussi besoin de relations plus générales que la simple localisation, permettant de créer des liens très variés entre une classe agent source et une classe agent cible. Par exemple pour chaque individu de la classe P, on veut mémoriser son (ou ses) lieu(x) de travail de la classe agent C. On définit alors une relation binaire *f* de P vers C, qui associe à tout individu *i* de P, son (ou ses) lieu(x) de travail *f*(*i*). Cette relation peut ressembler à une localisation, mais ce n'est pas le cas puisqu'un individu n'est pas toujours localisé sur son lieu de travail et peut même en avoir aucun, ou plusieurs.

2.6.2.3 Les relations générales entre classes d'agents

43 Pour être complet, MAGéo permet au modélisateur de définir des relations encore plus générales que les relations binaires entre classes d'agents, on parle alors de ***relation n-aire***. Ce concept est pris ici au sens de l'algèbre relationnelle, c'est-à-dire qu'une relation n-aire définie sur les *n* ensembles A, B, C, … est un ensemble (fini) de ***liens,*** on parle aussi ***d'occurrences*** de la relation. Un lien de cette relation est une suite de *n* termes de la forme (a, b, c, …) où a est un élément de A, b un élément de B, c un élément de C, etc. Les ensembles A, B, C,… sont appelés composants de la relation lorsque ce sont des classes agents, mais ce peut être aussi des attributs complémentaires de type quelconque (entier, réel, texte, etc.).

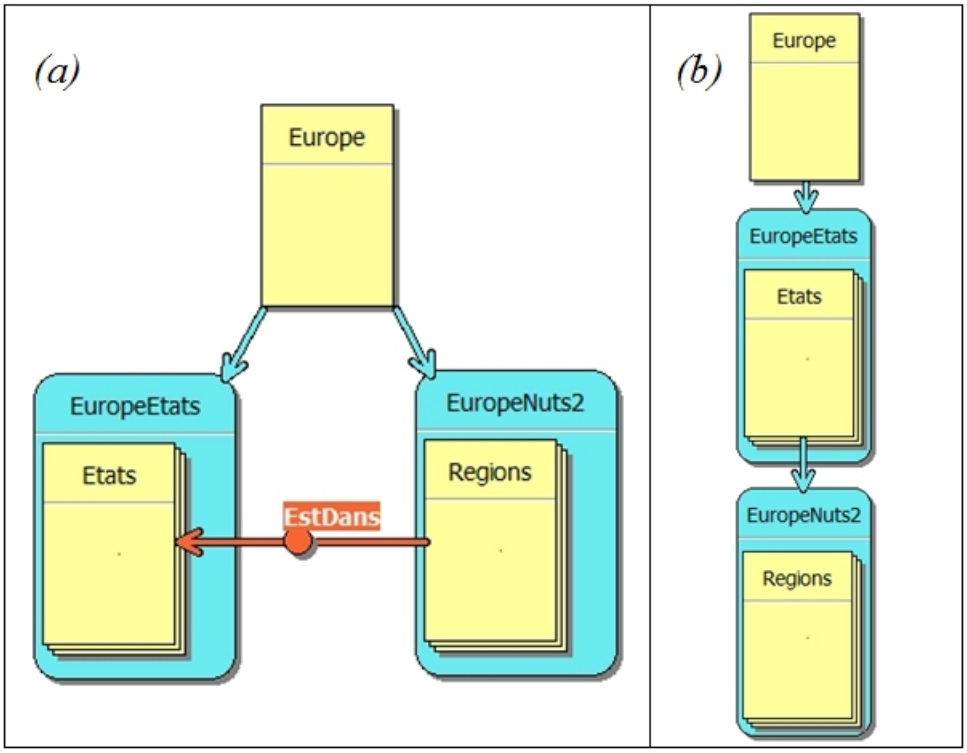
2.7 Conception de modèle complexes

44 La plupart des modèles MAGéo possèdent deux niveaux : le haut niveau qui est celui de l'agent modèle global (agent de niveau 1), est le plus souvent approfondi par au moins une organisation contenant ses propres composants (agents de niveau 2). C'est ce que l'on fait dès que l'on souhaite modéliser un système distribué à base d'agents. Néanmoins on construit aussi des modèles avec un seul agent, en s'arrêtant donc au niveau 1. C'est le cas des modèles basés sur des systèmes dynamiques (équations différentielles). MAGéo offre également la possibilité de définir plus de deux niveaux, mais le modèle devient alors plus compliqué (et plus complexe) et une bonne réflexion conceptuelle doit être menée pour trouver la meilleure structure.

45 Pour définir des modèles multiniveau, il faut bien faire la différence entre la *classe*, qui est une notion abstraite (et relève du niveau conceptuel) et *l'instance* (ou *réalisation*) qui est la concrétisation physique d'une entité d'une certaine classe. En effet, au niveau conceptuel, une organisation est en fait une ***classe d'organisation*** qui peut donner au niveau physique une instance unique ou plusieurs instances selon sa position dans le modèle conceptuel. Par exemple si le modèle est composé de 2 niveaux, au premier niveau, nous n'avons qu'un agent M unique qui représente le modèle dans sa globalité. On approfondit celui-ci en insérant une organisation unique O. Jusqu'ici tout va bien, car chaque classe (d'agent ou d'organisation) donne une seule instance dans le modèle physique, il y a donc une quasi-identité entre les niveaux conceptuel et physique. Mais l'organisation O est composée d'une classe-agent A, qui définit au niveau physique plusieurs instances A_i d'agents formant le second niveau. Si l'on approfondit maintenant la classe-agent A par une nouvelle classe d'organisation P, cela signifie qu'au niveau physique on instancie dans chaque agent A_i une instance d'organisation P_i de la classe P. Cette organisation est donc générique et son paramétrage devra donc différencier ses différentes instances. Si les structures internes de MAGéo qui permettent de traiter les organisations génériques sont d'ores et déjà développées, l'interface ne permet pas encore de les utiliser dans leur plus grande généralité.

46 En ce qui concerne l'utilisation de données géographiques, cette limitation peut être évitée de la manière suivante. Le fait de disposer de données géographiques découpées selon plusieurs échelles ne signifie pas qu'il faille forcément leur associer un modèle multiniveau. Prenons l'exemple d'une hiérarchie de découpage administratif de l'Europe en États et en régions. On pourrait penser qu'un modèle à trois niveaux serait la meilleure structure ontologique adaptée à ce cas. Néanmoins la structuration des données dans un SIG est un peu différente car elle procède par couches (layers, coverages, etc.) et non par emboitements de niveaux. Soit deux couches géographiques qui représentent la carte des États de l'Europe pour la première et la carte des régions pour la deuxième. Il est plus aisé de définir un modèle multicouche formé de deux organisations spatiales correspondant à ces 2 couches et de les relier par une relation de localisation par inclusion spatiale : les régions seront ainsi localisées dans les états. Cette relation permet de définir les liens d'emboitement entre les différentes entités spatiales, comme on pourrait le faire avec un modèle multiniveau, mais en n'utilisant que 2 instances d'organisation au lieu de plusieurs dizaines, comme on peut le voir dans la figure suivante.

Figure 1 : Modèle multicouche (a) et modèle multiniveau (b)



47 Après avoir présenté les principaux concepts utilisés dans MAGéo, la section qui suit détaille les différents éléments pour réaliser un modèle de simulation.

3 Présentation de la plateforme MAGéo

48 La plateforme MAGéo propose par sa structure même un cadre méthodologique de réalisation d'un modèle. Ceci se traduit d'abord par le découpage en deux fenêtres principales : on construit le modèle conceptuel dans la première et l'interface de simulation dans la seconde. Différents outils sont mis à disposition du modélisateur : éditeurs permettent de générer des fonctions graphiquement ou par formulation, éditeur de types énumérés, éditeur de symboles graphiques ; deux inspecteurs d'objets permettent de suivre l'état des entités physiques du modèle (agents, organisations). Par ailleurs un modèle MAGéo peut utiliser des fichiers en entrée (cartes, données tabulaires) et fournir divers fichiers de sortie.

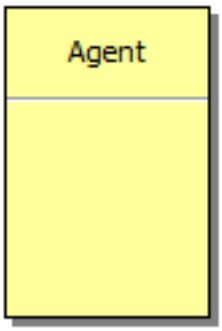
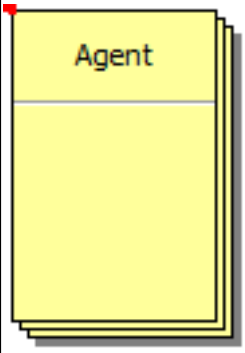
3.1 L'interface de modélisation

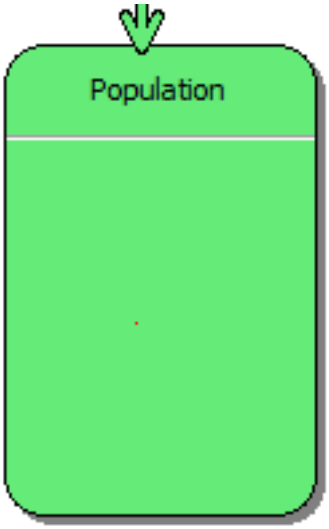
49 La fenêtre de modélisation est celle dans laquelle est construit le modèle conceptuel, sous une forme arborescente composée d'entités et de relations. La méthode de construction d'un modèle consiste à ajouter une ou plusieurs organisations à l'agent modèle (et aux agents en général), organisations qui contiennent elles-mêmes des classes d'agents composants. Puis on ajoute des relations entre classes agent. Vient ensuite l'identification des comportements. A ce niveau on travaille de manière uniquement graphique, sans entrer dans les détails internes, chaque entité conceptuelle étant concrétisée par un symbole graphique particulier et les relations par des flèches et des lignes entre ces entités.

3.1.1 Les entités du modèle conceptuel

50 Les entités d'un modèle conceptuel sont construites selon la symbolique suivante :

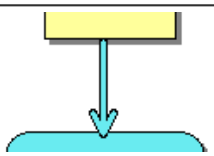
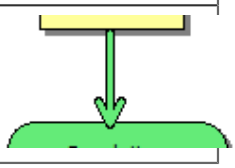
Tableau 1 : Entités du modèle conceptuel

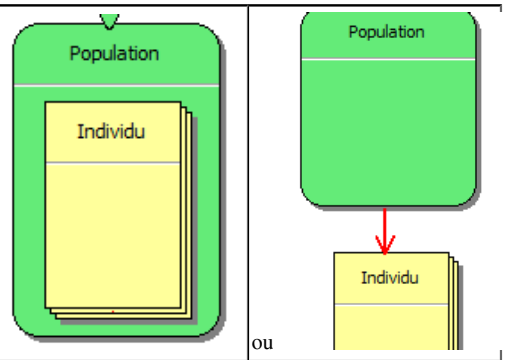
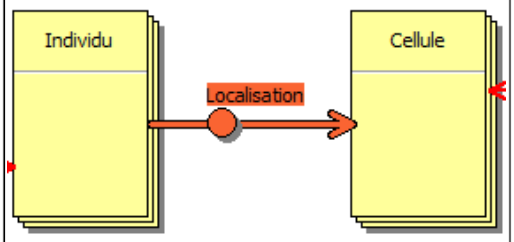
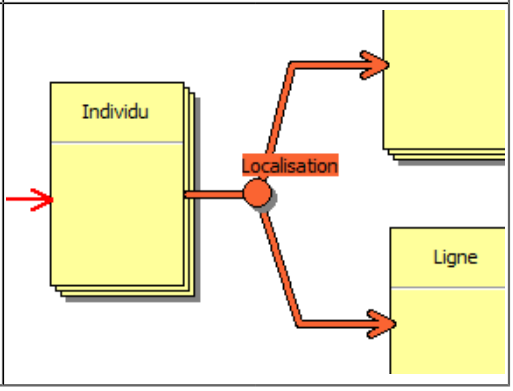
Type d'entité	Symbole graphique
Agent unique	
Agent générique	
Comportement simple	

Comportement de type diagramme de transition	
Organisation sociale (Population)	
Organisation spatiale	

51 Les relations entre les entités graphiques des organisations et des agents sont construites selon la symbolique suivante :

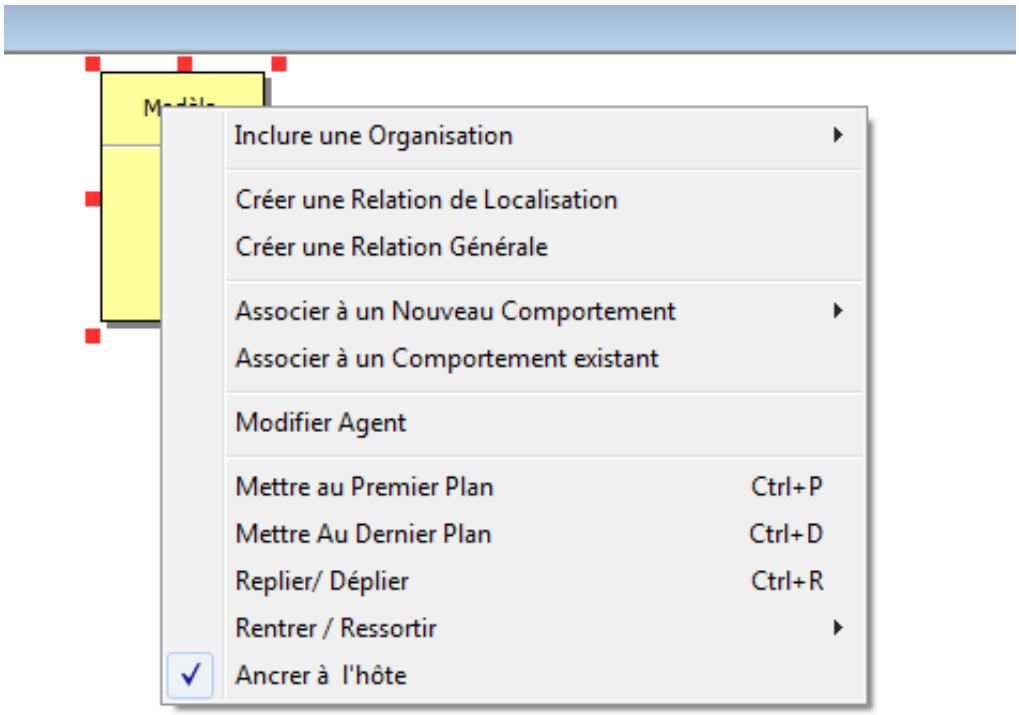
Tableau 2 : Relations du modèle conceptuel

Type de relation	Définition	Symbolique
Approfondissement	Agent → Organisation	 ou 

Composition	Organisation → Agent	
Localisation	Agent → Agent	
Localisation étendue	Agent → (Agent,…, Agent)	

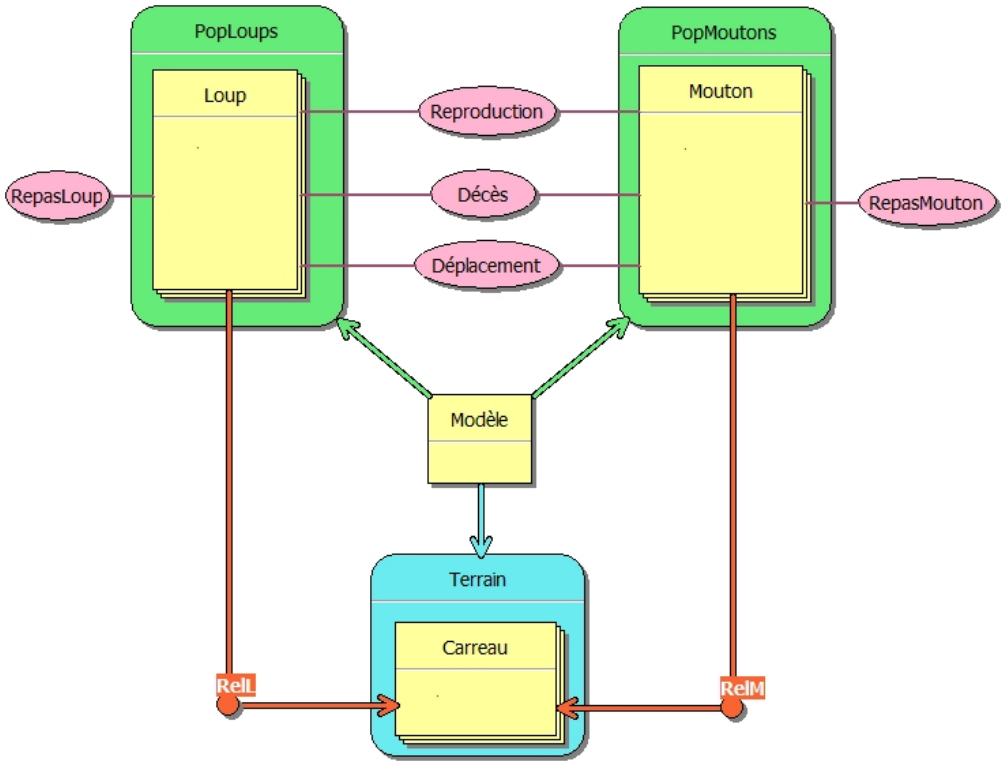
52 La seule entité présente par défaut lors de la création d'un nouveau modèle est l'agent unique appelé « Modèle ». Un clic droit sur cette entité, et sur toute entité en général, permet d'afficher un menu contextuel (Figure 2) qui contient l'ensemble des possibilités pour construire le modèle, c'est-à-dire ajouter des organisations, des relations et des comportements.

Figure 2 : Menu contextuel associé à un type d'entité (ici un agent)



53 En utilisant les choix proposés par ce menu, puis après le positionnement et le dimensionnement des différentes entités créées, le modèle conceptuel se construit progressivement. La Figure 3 illustre un modèle conceptuel complet du modèle Proie-prédateur (Collectif MAPS, 2014).

Figure 3 : Exemple de modèle conceptuel



54 Un simple clic sur une entité de ce modèle permet de la sélectionner, un double-clic permet d'entrer dans l'entité sélectionnée, c'est-à-dire d'afficher son dialogue de paramétrage.

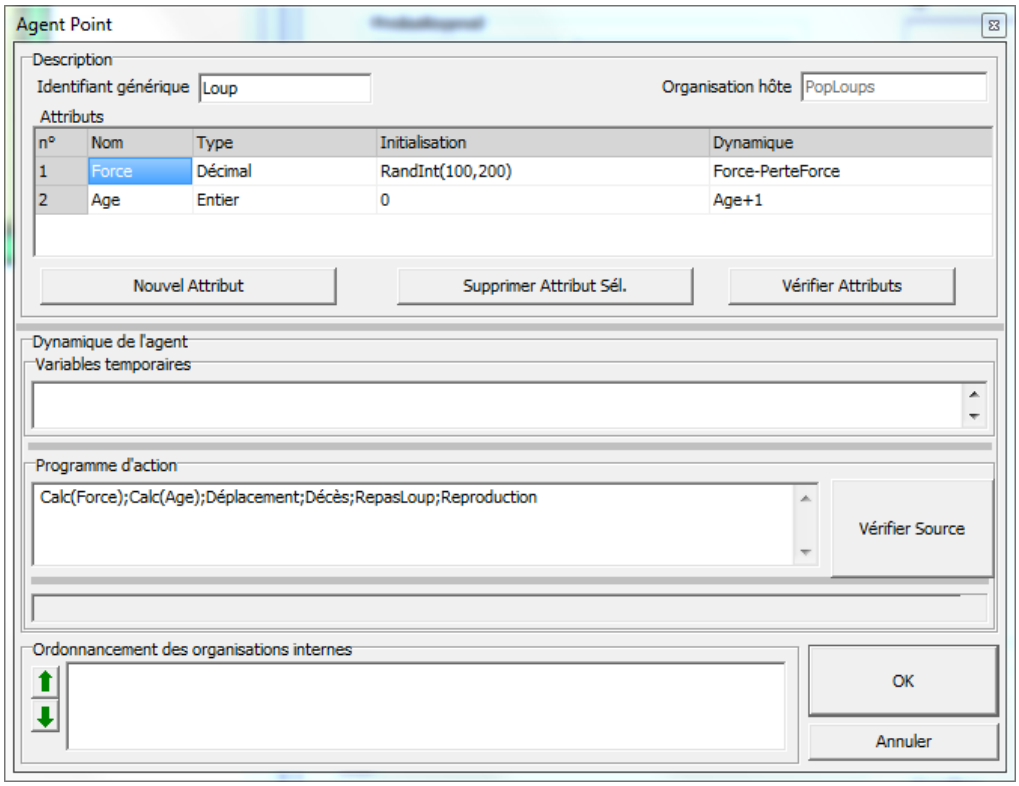
3.1.2 Paramétrage des entités conceptuelles

55 Les organisations et les classes d'agents créées sont ensuite renseignées afin de donner des informations qui seront utilisées lors de l'instanciation des entités physiques.

3.1.2.1 Paramétrage des agents

56 Un agent possède des attributs et un programme d'action. Celui-ci est un script qui indique comment ses différents comportements seront exécutés lors d'un pas de temps. Un agent peut aussi posséder un système interne formé de différentes organisations. Pour gérer son système interne, une liste de ces organisations indique leur ordonnancement lors de l'initialisation et de l'exécution.

Figure 4 : Paramétrage d'un agent



57 L'interface de dialogue d'une classe agent permet de définir le contenu et la dynamique des agents. Elle contient :

- Un identifiant générique de classe, qui sert à définir les identifiants des instances en ajoutant à ce nom un numéro d'ordre ;
- Une liste d'attributs, qui représentent l'état de l'agent ;
- La définition du programme d'action, qui précise l'ordonnancement des comportements de l'agent lors d'un pas de temps, au moyen d'un langage dédié (voir 4.4.1) ;
- L'ordonnancement des organisations internes, qui indique l'ordre dans lequel seront exécutés les comportements des agents internes lors d'un pas de temps.

58 Les attributs d'une classe agent doivent être définis dans le tableau des attributs de cette classe. Cette définition consiste à renseigner les 4 champs suivants :

- Le nom de l'attribut ;
- Le type de cet attribut, qui peut être un nombre entier, un nombre décimal, un nombre logique (nombre décimal entre 0 et 1), une chaîne de caractères, un vecteur, un agent ou un énuméré. Dans MAGéo, une règle fondamentale est qu'un nom d'attribut ne peut être utilisé dans plusieurs classes agents que si le type associé est le même.
- L'expression d'initialisation, écrite en langage MAGéo, qui sera évaluée à l'initialisation du modèle pour fournir la valeur initiale de l'attribut. Cette expression doit donc être du

type indiqué dans la colonne précédente. Par exemple un attribut entier peut prendre une valeur initiale aléatoire entre 0 et 100 par RandInt(0, 100).

- L'expression de la dynamique de cet attribut, également écrite en langage MAGéo, est une expression donnant une nouvelle valeur à l'attribut, et qui sera exécutée à chaque pas de temps si elle est indiquée dans le programme d'action à l'aide de la fonction Calc(NomAttribut). On voit ainsi qu'il y a deux moyens de gérer l'évolution d'un agent, le comportement et la dynamique d'un attribut. Ce dernier est utilisé lorsqu'un attribut a besoin d'être recalculé à chaque itération, sans condition particulière, alors que le comportement est un mécanisme plus sophistiqué qui s'adapte en fonction de l'état du système. Par exemple pour que l'attribut Age soit incrémenté à chaque itération, nul besoin de définir un comportement, on écrit simplement Age+1 dans le champ « *dynamique* » de cet attribut, et on indique Calc(Age) dans le programme d'action.

3.1.2.2 Paramétrage des organisations

59 MAGéo propose actuellement 3 types d'organisations spatiales prédéfinies :

- Les ***maillages réguliers*** en carrés ou en hexagones ;
- Les ***maillages irréguliers*** qui permettent de réaliser des cartes zonales topologiques ;
- Les ***réseaux.***

60 MAGéo propose également 2 types d'organisations sociales :

- Les organisations constituées d'une population générique ;
- Les organisations compartimentées.

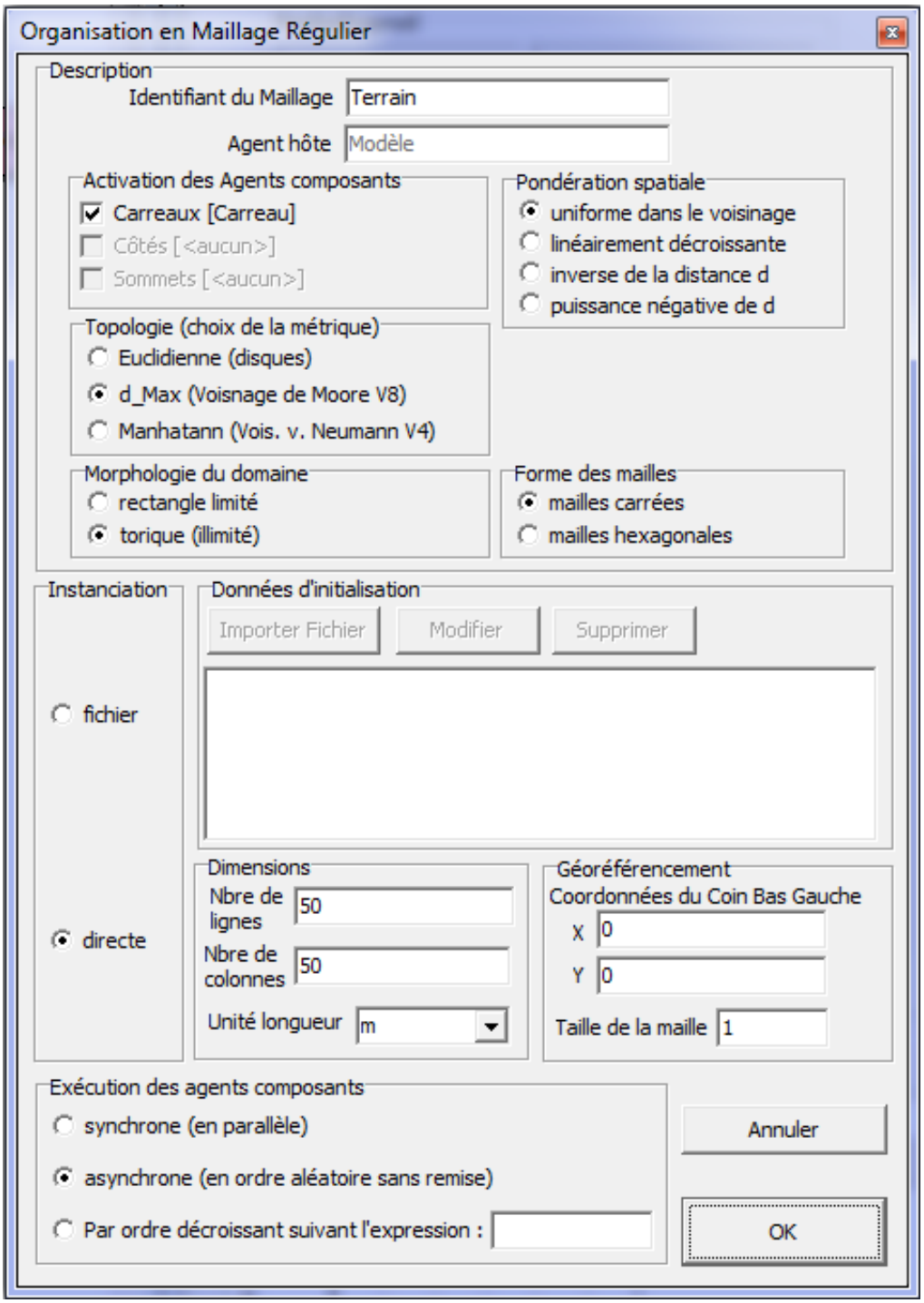
61 Une organisation a pour rôle de définir la structure unifiant une collection d'agents, appelés composants de l'organisation. Elle a également pour rôle de gérer les interactions entre ces composants. Tous les types d'organisation contiennent :

- Un identifiant générique de classe d'organisation, qui sert à définir les identifiants des instances d'organisation, si besoin en ajoutant à ce nom un numéro d'ordre ;
- La méthode d'instanciation de ses agents composants, qui définit le nombre d'agents à créer ;
- La méthode d'initialisation de ses agents composants qui peut-être directe (une expression par exemple) ou réalisée à partir d'un fichier contenant une table attributaire ;
- La méthode d'ordonnancement de ses agents composants. Elle peut être synchrone (les agents s'exécutent en parallèle), asynchrone (ils s'exécutent selon une séquence dont l'ordre est tiré au hasard) ou selon un ordre défini par une expression dépendant de l'état du modèle.

62 Selon les organisations, des paramétrages particuliers sont nécessaires. Pour une organisation spatiale, on doit choisir les relations topologiques qui relient et structurent les voisinages de ses agents composants. Pour une organisation sociale, on doit décrire les différentes relations qui existent entre les individus composants une population. Selon le type d'organisation, un dialogue différent est donc défini.

63 Par exemple, une organisation de type maillage régulier contient différents paramètres comme la forme de la maille (carrée ou hexagonale), la topologie (induite par un type de métrique), la présence de fichier d'initialisation de ses composants ou encore le nombre de lignes et de colonnes (Figure 5).

Figure 5 : Paramétrage d'un maillage régulier



64 Le même genre de dialogue est utilisé pour paramétrer un réseau ou un maillage zonal irrégulier. Dans un maillage zonal on utilise des composants surfaciques, les faces. Dans un réseau on utilise deux classes de composants, des nœuds (composants ponctuels) et des tronçons (composants linéaires) et les données sont chargées depuis un fichier.

65 Une organisation « sociale » telle que définie dans MAGéo est avant tout une organisation non spatiale qui repose sur un ensemble d'agents composants. Ces agents ne sont pas localisés *a priori* mais peuvent le devenir si on définit une relation de localisation de ces agents vers les composants d'une organisation spatiale. A sa création, une organisation sociale ne contient donc pas de structure particulière de l'ensemble de ses composants. C'est au modélisateur de définir les relations qui structureront cette population.

3.1.2.3 Paramétrage des comportements

66 Dans MAGéo, chaque comportement est représenté par une entité graphique ayant la forme d'une ellipse (Tableau 1) qui est reliée aux agents susceptibles de l'utiliser. Le comportement est paramétré dans une boîte de dialogue accessible par un double-clic.

67 Les actions et expressions présentes dans un comportement doivent être écrites en cohérence avec les classes d'agents auxquelles elles sont reliées. Cette cohérence relève principalement de l'utilisation du mot-clé self. Le script d'un comportement fait référence à l'agent en cours d'exécution qui utilise ce comportement, cet agent y est désigné par le mot clé self. Lorsque le comportement fait référence à des attributs de l'agent self ou de tout autre agent, ces attributs doivent évidemment être définis dans la classe de cet agent. Le programme d'action est un script à renseigner dans la classe agent qui précise de quelle manière les différents comportements qu'elle utilise se combinent entre eux au moment où chaque agent de cette classe s'exécute.

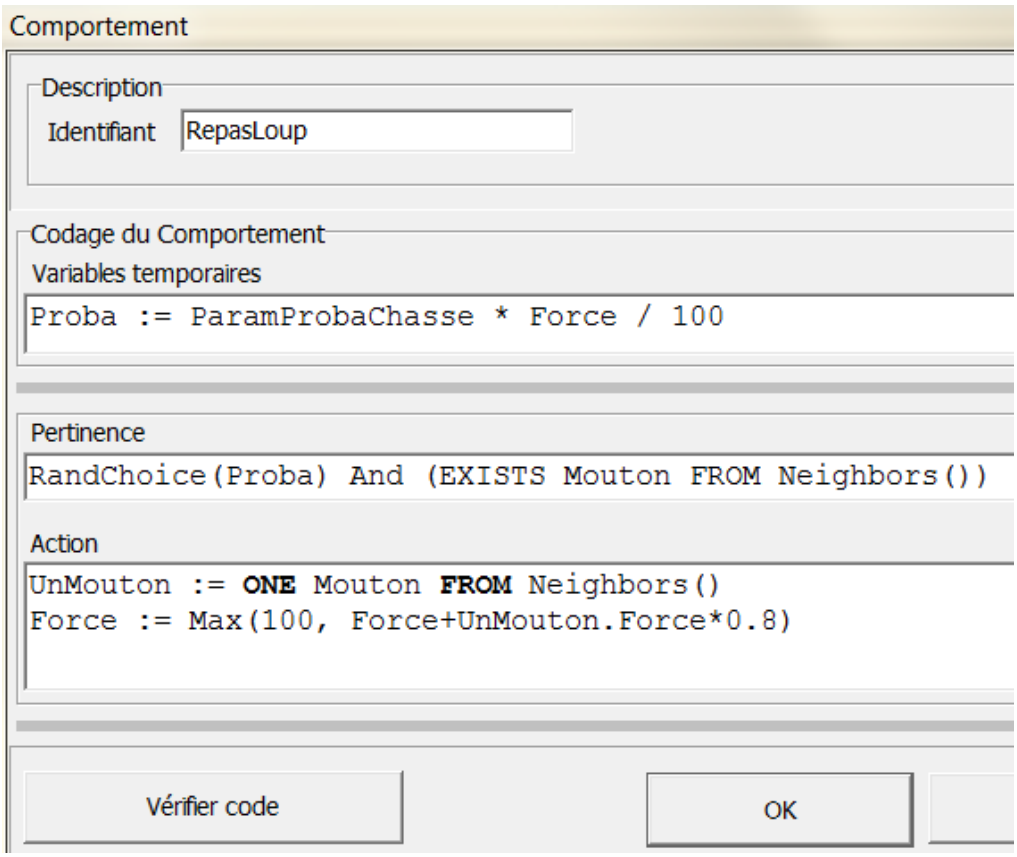
68 Deux types de comportements sont proposés dans MAGéo, les comportements simples et les diagrammes de transition.

3.1.2.3.1 Comportements simples

69 Un comportement simple s'exprime sous la forme d'un couple < *pertinence*, *action* >. La pertinence est une expression dont la valeur est un nombre réel dans l'intervalle [0, 1]. L'action est une séquence d'instructions qui modifie l'état de l'agent, soit ses attributs, sa localisation ou son système interne.

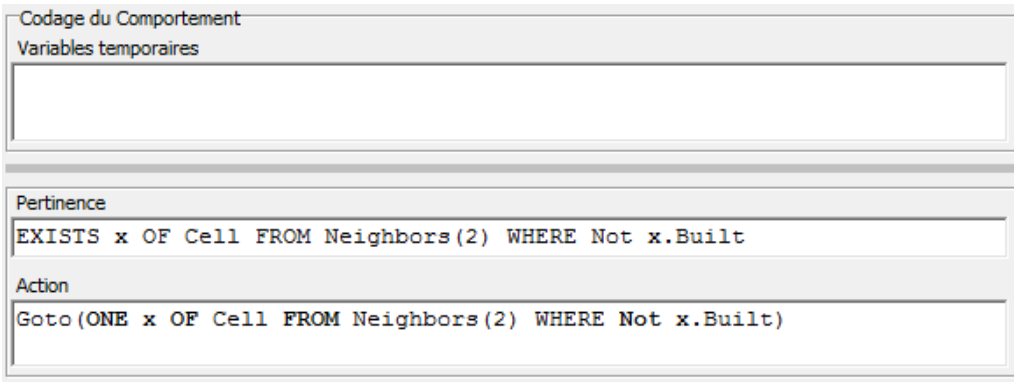
70 Dans le dialogue de paramétrage du comportement, ces deux termes ***Pertinence*** et ***Action*** correspondent chacun à une zone de texte qui permet d'écrire leur script dans le langage MAGéo. Une zone nommée *variables temporaires* permet de faire localement des calculs préparatoires dont les résultats peuvent être utilisés aussi bien dans la zone de pertinence que celle d'action. Dans l'exemple de la Figure 6, le terme « Proba » est une variable locale et temporaire. A un instant donné, si le programme d'action de l'agent décide d'exécuter ce comportement (ici « RepasLoup »), les calculs de la zone des variables temporaires seront exécutés, par contre la partie action ne sera exécutée que si elle est jugée plus pertinente que les autres comportements en concurrence.

Figure 6 : Paramétrage d'un comportement simple



71 L'exemple suivant (Figure 7) montre un comportement nommé *Move*. Celui-ci est pertinent s'il existe au moins une cellule non bâtie dans l'ensemble des cellules du voisinage de rayon 2 de l'agent en cours d'exécution. Son action consiste alors à ce que l'agent change de localisation, en devenant le passager d'une cellule choisie au hasard parmi les cellules non-bâties de son voisinage.

Figure 7 : Comportement Move



3.1.2.3.2 Diagrammes de transitions

72 MAGéo propose également des comportements inspirés des règles de transition du logiciel SpaCelle (Dubos-Paillard *et al.*, 2003). Un comportement modélisé par un diagramme de transition permet de simuler les changements d'états d'un attribut de l'agent, qui doit être ici un attribut de type énuméré, donc qualitatif. Une règle de transition décrit un comportement simple un peu spécial, dont l'action essentielle consiste à faire passer une variable d'un état (E1) vers un autre état (E2). La transition est notée : E1 → E2. La règle contient aussi un mécanisme qui permet de décider du moment et des conditions pour effectuer la transition. Pour cela, une règle de transition contient :

- Une durée de latence, en nombre d'itérations, pendant laquelle la transition est en attente et la règle reste totalement non pertinente. Cette durée peut évidemment être nulle ;

- Dès que la durée de latence est dépassée, l'expression de la pertinence est évaluée ;
- En plus de la transition elle-même, une zone « action » permet d'écrire des actions supplémentaires qui se déclenchent au moment de la transition.

73 Si plusieurs règles ont le même état initial et sont donc en concurrence, seule la plus pertinente sera exécutée. S'il n'y a qu'une seule règle et donc pas de concurrence, elle sera exécutée si sa pertinence est strictement positive. Si plusieurs règles ont des pertinences maximales égales, alors l'une d'entre elles sera choisie au hasard pour être exécutée.

74 Par exemple dans un modèle de diffusion d'une maladie infectieuse, l'agent possède une variable d'état représentée par l'attribut qualitatif « Etat ». Cette variable peut prendre les trois états : « S » lorsqu'il est sain, « I » lorsqu'il est infecté et « R » lorsqu'il est remis et momentanément immunisé après avoir été infecté. Le diagramme permet de définir un graphe des trois transitions (Figure 8) ainsi que les règles de transitions associées (Figure 9) : un individu sain deviendra infecté (R1 : S → I) s'il existe au moins un individu infecté dans son voisinage de rayon 2 et si celui-ci lui transmet l'infection (selon une probabilité de 5%).

Figure 8 : Paramétrage d'un diagramme de transition

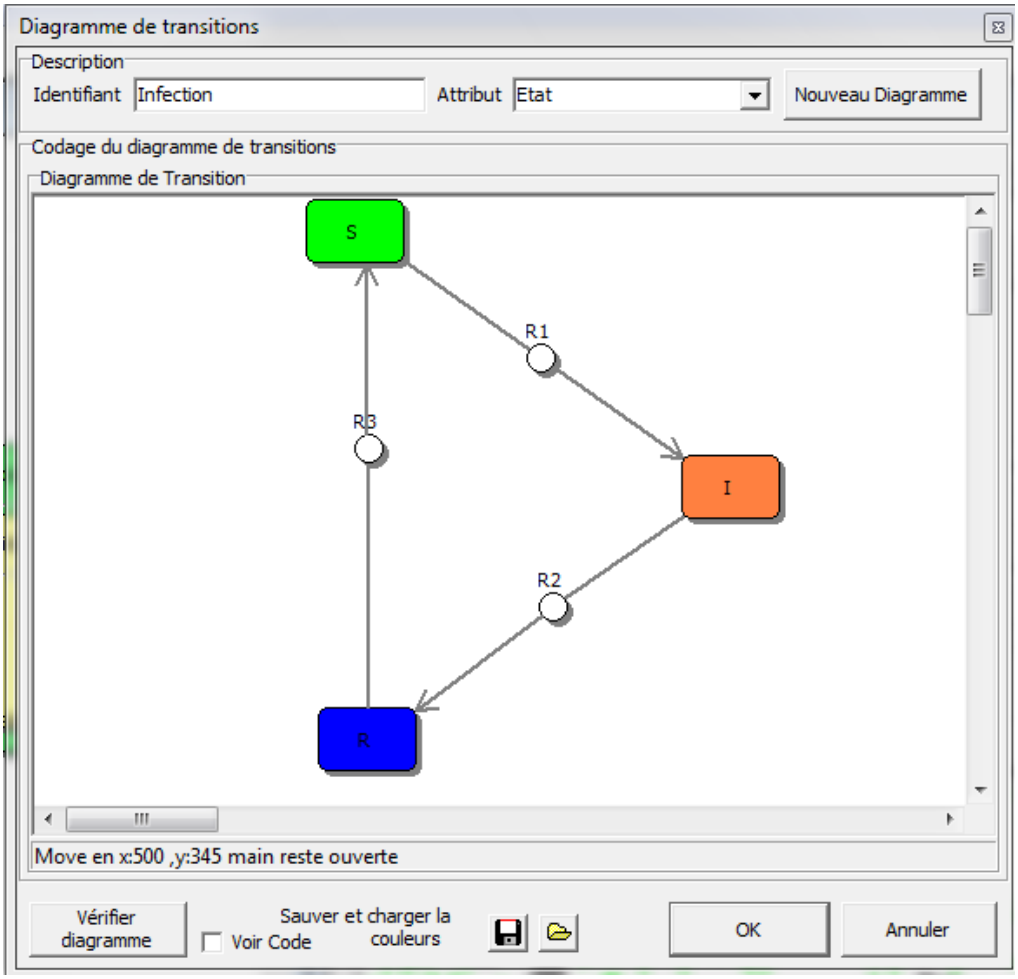
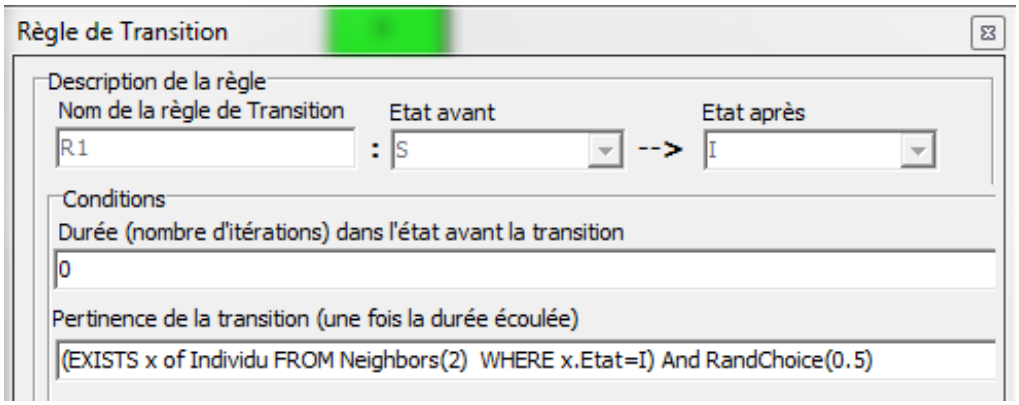


Figure 9 : Définition de la règle de transition R1



3.1.2.4 Paramétrage des relations de localisation

75 Une relation de localisation est définie entre une classe agent qui est la source et une ou plusieurs classes agent qui sont les cibles. Le dialogue associé à une relation de localisation a pour but de préparer son initialisation et de définir des mots-clés qui permettront de la manipuler dans les comportements.

76 L'initialisation d'une relation consiste à définir pour chaque agent passager de la source sur quel agent hôte dans une cible il sera localisé au début de la simulation. Ceci peut être paramétré de quatre façons différentes :

- Localisation aléatoire : l'agent sera localisé sur un agent choisi aléatoirement parmi tous les hôtes disponibles ;
- Par inclusion géométrique : On cherche quel est le composant qui contient le centre de l'agent passager. Cette méthode n'est valable que pour des passagers ponctuels ou de forme convexe, dans le cas contraire il se peut que le centre soit situé à l'extérieur du domaine géométrique du passager. On utilise dans ce cas la méthode suivante ;
- Par clé extérieure : l'agent passager recherche son composant hôte par son identifiant qui doit être mémorisé dans un attribut du passager, nommé clé extérieure ;
- Par expression : l'agent devient passager d'un hôte obtenu par évaluation de l'expression indiquée dans le dialogue. A titre d'exemple, pour localiser chaque agent *Homme* sur un agent *Maison* dont la surface est inférieure à 70 on écrit :

  ONE m OF Maison WHERE m.Surface <= 70

77 On notera que lorsqu'un agent non localisé devient passager d'un composant spatialisé, il se trouve placé au centre de la géométrie de cet agent hôte.

78 Le dialogue d'une relation de localisation permet d'ailleurs de définir trois noms de fonctions utiles à la manipulation de la relation en question :

- L'hôte : cette fonction, par exemple nommée *Loc*, permet de récupérer la localisation d'un agent passager donné, par l'appel *Passager.Loc* ;
- Les passagers : cette fonction, par exemple nommée *Pass*, permet de récupérer la liste des passagers d'un agent hôte donné, par l'appel Hôte.*Pass* ;
- Les copassagers : cette fonction, par exemple nommée *Copass*, permet de récupérer la liste des autres passagers localisés sur le même composant hôte qu'un agent passager donné, par l'appel *Passager.Copass*.

79 Une fois ces noms définis dans le dialogue, les fonctions décrites ci-dessus viennent s'ajouter au langage, et peuvent alors être utilisées dans les comportements.

3.1.2.5 Paramétrage des relations générales

80 Pour définir une relation quelconque, il faut définir :

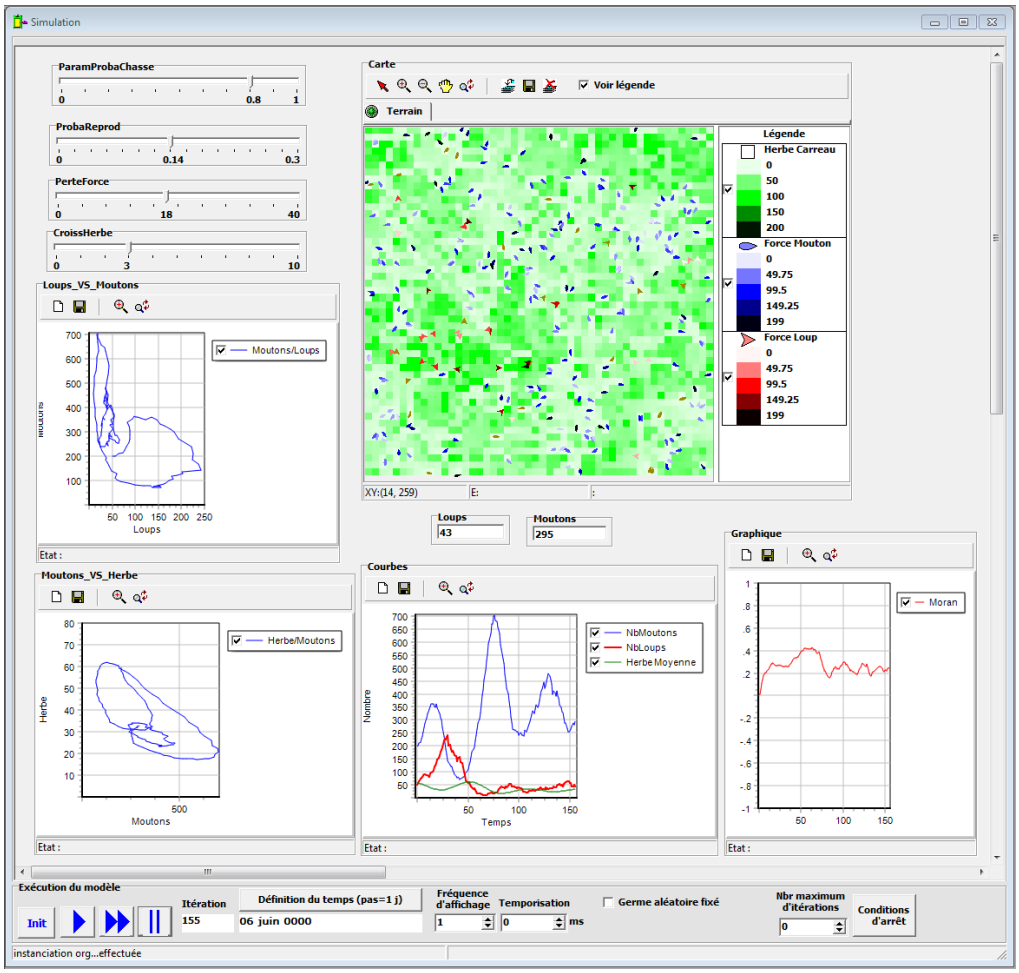
- L'identifiant de la relation ;
- Les classes d'agents qui composent la relation. Celles-ci sont créées graphiquement par les liens qui référencent les classes d'agents participant à la relation. Ces classes d'agents apparaissent comme les premiers attributs de la relation ;

- Les attributs complémentaires qui sont définis dans le dialogue, et sont de type numérique, logiques ou textuels, et apportent des informations complémentaires aux différentes occurrences de la relation ;
- L'instanciation des occurrences de la relation et l'initialisation des attributs, qui peuvent être réalisées par un fichier ou par un script.

3.2 L'interface de simulation

81 La fenêtre de simulation est celle dans laquelle est construite l'interface de simulation, elle est composée de deux parties (Figure 10). La partie haute permet au modélisateur de définir graphiquement les paramètres d'entrée du modèle qu'il souhaite pouvoir régler avant ou en cours de simulation. Dans cette fenêtre le modélisateur dispose également d'objets d'observation de l'état de son modèle dans l'espace (avec des cartes) et dans le temps (avec des courbes), lors de l'initialisation puis au cours de la simulation. Cette fenêtre reçoit donc tous les éléments utiles au paramétrage et à l'observation du modèle physique. La partie basse de cette fenêtre contient des éléments de dialogue qui sont prédéfinis pour l'exécution du modèle. Ils permettent d'instancier et d'initialiser un modèle, d'exécuter le modèle pas à pas ou de le faire tourner et de l'arrêter. Il est également possible de régler différents paramètres généraux comme la temporalité du modèle, la rapidité de simulation, la procédure automatique d'arrêt du modèle ou encore la méthode de génération de nombre aléatoires. Dans ce dernier cas il est possible de fixer le même germe pour toutes les simulations, ou au contraire de fixer un nouveau germe à chaque initialisation du modèle. Pour insérer un objet d'entrée ou de sortie dans la fenêtre de simulation, on utilise comme dans toutes les fenêtres de la plateforme le clic-droit qui fournit un menu contextuel des objets que l'on peut insérer.

Figure 10 : Interface de simulation d'un modèle en fonctionnement (Proies-prédateurs)



3.2.1 Objets d'entrée

82 Avec les curseurs, boutons, champs textuels ou titres, le modélisateur dispose de différents objets d'interface qui s'insèrent dans la fenêtre de simulation. Ceux-ci permettent d'expliciter les paramètres du modèle pour pouvoir régler rapidement et facilement les valeurs avant le début de la simulation. Ces valeurs peuvent rester constantes en cours de simulation, ou éventuellement être modifiées entre deux pas de temps par l'utilisateur.

83 Les curseurs fournis dans MAGéo permettent de définir des paramètres entiers ou décimaux, variant dans une certaine plage de valeurs, et selon un certain pas. Un curseur est utilisable directement par son nom, par exemple dans les scripts d'initialisation ou de comportement. L'utilisateur dispose également de champs de texte servant en entrée comme en sortie, et différents autres objets graphiques d'interface.

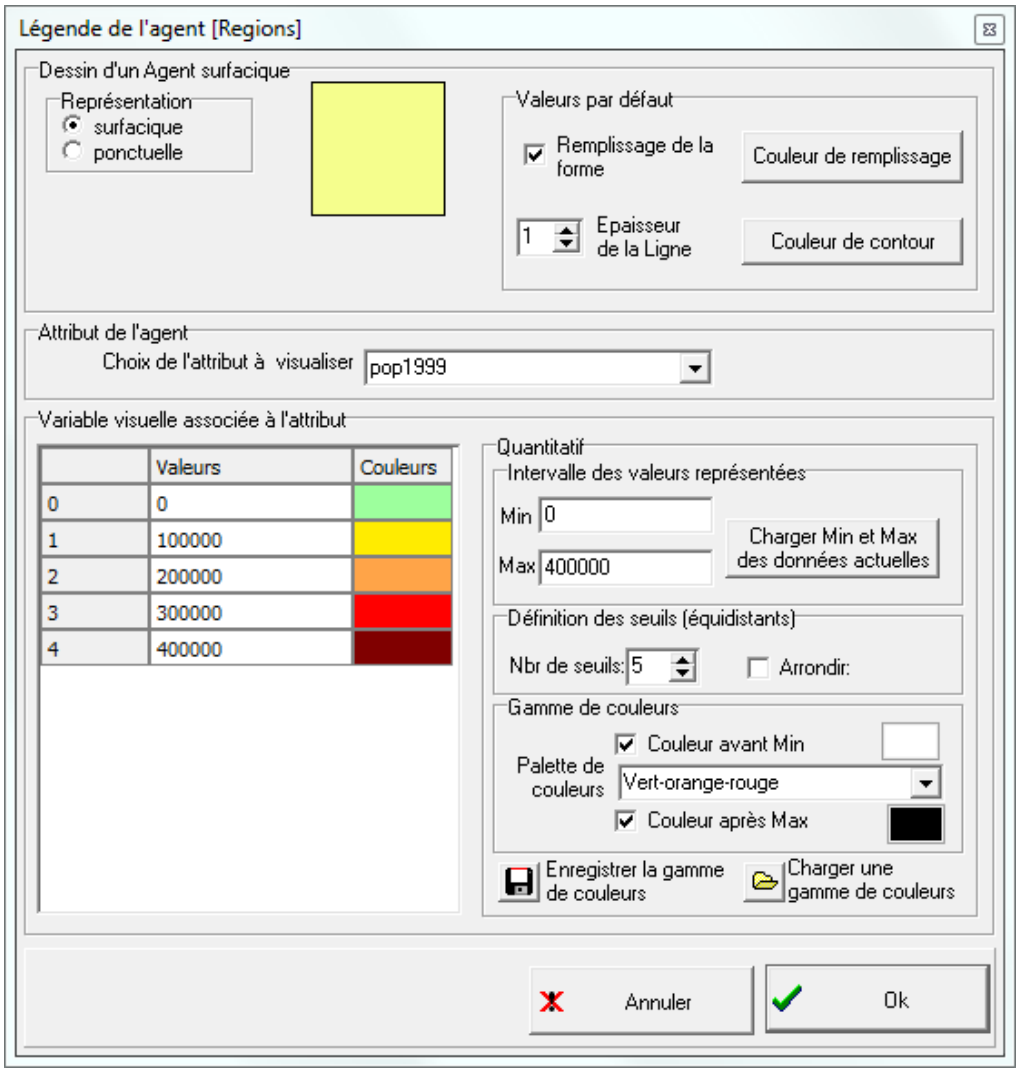
3.2.2 Cartes

84 Une carte MAGéo est un objet d'interface de sortie constituée d'un empilement de couches. Une couche est définie par les valeurs d'un attribut d'un ensemble d'agents (*composants* ou *passagers*) d'une organisation spatiale, elle donne donc une vision synchronique d'un état donné du système. L'ajout d'une couche à une carte implique donc de sélectionner une organisation spatiale, la classe agent qui sera affichée parmi ses composants et passagers, puis l'attribut de cette classe agent qui sera représenté graphiquement dans la couche.

85 Lorsqu'une couche est ajoutée à une carte, un double-clic sur la légende associée donne accès au dialogue de définition de la variable visuelle qui est associée à l'attribut (Figure 11). Les paramètres d'affichage sont :

- Mode de représentation : ponctuel (par symbole) ou surfacique (zonal) ;
- Choix du symbole (carré, rond, voiture etc.) et facteur d'échelle ;
- Paramètres de remplissage et de contour ;
- Choix de l'attribut à visualiser ;
- Variable visuelle associée : nombre et limites de classes, gamme de couleurs.

Figure 11 : Définition de la légende pour l'attribut pop1999 d'agents Régions

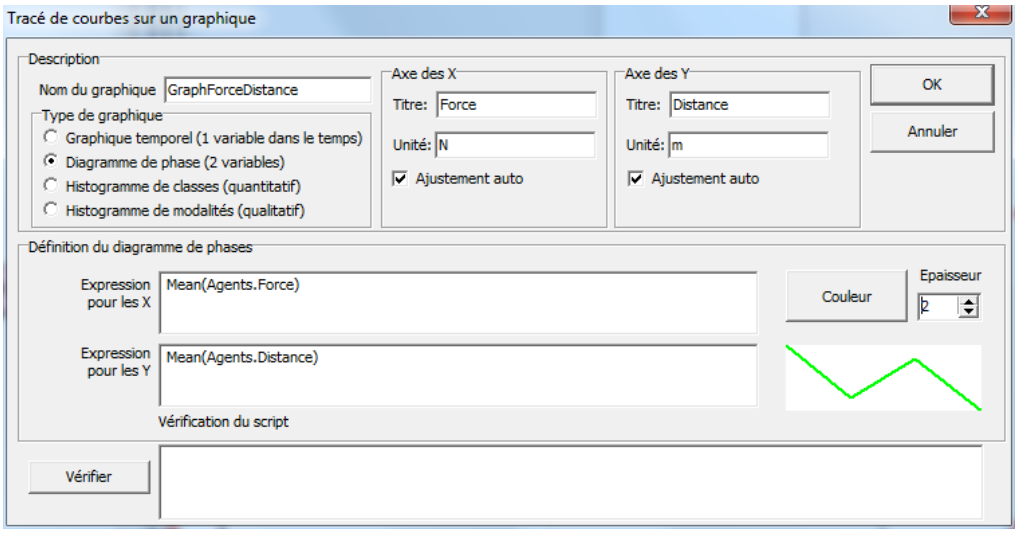


3.2.3 Graphiques

86 Un graphique MAGéo est un objet de sortie qui permet d'observer l'évolution dans le temps d'une ou plusieurs grandeurs associées à un ensemble d'agents, il donne donc une vision diachronique d'un état donné du système. Des graphiques de différentes natures peuvent être créés pour suivre cette évolution :

- Graphiques temporels, qui représentent l'évolution dans le temps d'une quantité tels que la valeur d'un attribut ou le nombre d'agents répondant à un critère ;
- Diagrammes de phase, qui représentent l'évolution dans le temps d'une quantité en fonction d'une autre ;
- Histogrammes de classe, qui sont dynamiques et représentent à chaque instant, soit les fréquences de classes selon la répartition des valeurs d'une expression quantitative, soit les fréquences des différentes modalités d'une variable qualitative (attribut de type énuméré), d'un ensemble d'agents.

Figure 12 : Dialogue de paramétrage d'une courbe (ici un diagramme de phase)



87 Ces différents graphiques se construisent aisément à l'aide de l'interface graphique de MAGéo, réduisant la programmation à la seule rédaction de quelques expressions (une ou deux selon le type de graphique) définissant les quantités considérées (Figure 12).

3.2.4 Le temps

88 MAGéo propose des outils de modélisation du temps qui permettent de définir le repère temporel qui positionnent le temps formel des itérations dans le temps simulé, par une origine et une unité exprimée en nombre d'itérations au cours de l'unité de temps. On choisit également les sous-unités de temps à afficher et le format d'affichage de la date dans l'interface.

89 La définition d'un repère de temps ajoute des possibilités pour la description du comportement des agents, notamment à travers certaines fonctions du langage MAGéo. A titre d'exemples, la fonction booléenne AtTic(i) permet de décomposer l'unité de temps pour effectuer une première action par tous les agents avec la pertinence AtTic(0), puis de passer à une deuxième action effectuée par tous les agents avec la pertinence AtTic(1), etc. La fonction Month() indique le numéro courant du mois de l'année courante (valeur entre 0 pour janvier et 11 pour décembre), tandis que la fonction Months() retourne le nombre total de mois écoulés depuis le début de la simulation. Ce type de fonctions permet de synchroniser les comportements des agents avec le temps défini dans le repère, par exemple en indiquant qu'ils ne sont pertinents qu'à certains moments. Un comportement qui ne se déclenche qu'à chaque mois de février pourrait avoir comme pertinence : Month()=1. Ces fonctions permettent ainsi de définir des temporalités emboîtées lorsque l'on traite un modèle multicouche.

3.3 Des éditeurs pour enrichir les modèles

90 Plusieurs éditeurs sont proposés dans MAGéo afin d'enrichir les modèles. Ces éditeurs sont accessibles via le menu *Outils* de la fenêtre principale.

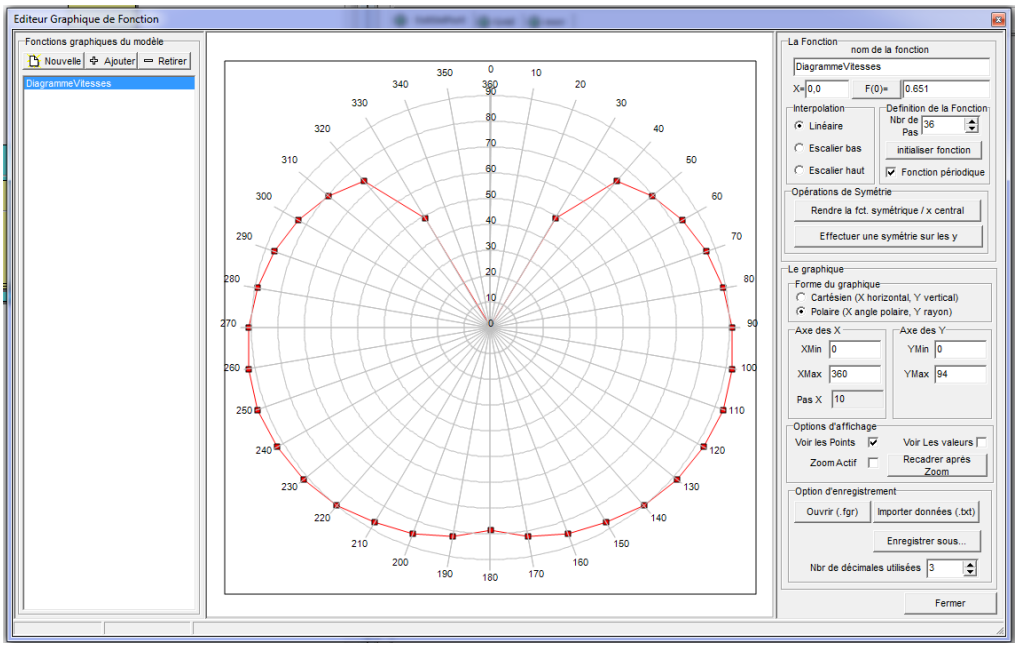
3.3.1 Editeur de fonctions

91 L'éditeur de fonctions permet la création de nouvelles fonctions directement utilisables dans le modèle. Ces fonctions reçoivent et calculent des données de type simple (nombres, texte, valeurs énumérées ou liste). Elles sont écrites par une suite d'instructions selon la syntaxe du langage MAGéo (cf. § 5). Une fonction peut prendre des arguments formels d'entrée qui sont utilisés pour faire appel à la fonction dans différents contextes lors d'une simulation.

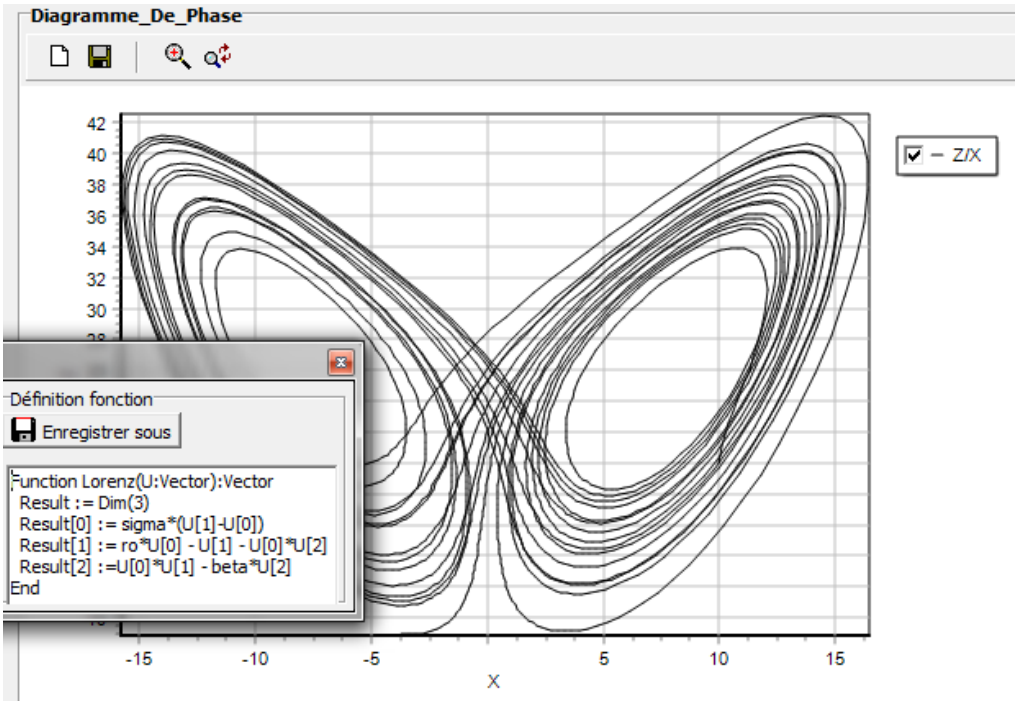
3.3.2 Editeur graphique de fonctions

92 Cet éditeur est utilisé pour créer des fonctions numériques à une variable, de façon purement graphique, c'est-à-dire par positionnement manuel des points de la courbe. Une fois défini l'intervalle de définition de la fonction, l'éditeur propose différents paramètres, notamment un choix entre une visualisation cartésienne classique et une visualisation polaire.

Figure 13 : A - Exemple de fonction construite dans l'éditeur graphique de fonction



B- Exemple de fonction définie dans l'éditeur de fonction



3.3.3 Editeur de types énumérés

93 Outre les types simples usuels (entier, réel, texte, etc.), les attributs d'un agent peuvent prendre leurs valeurs selon un type énuméré, par exemple un type définissant les occupations possibles du sol (Forêt, Prairie, Zone Urbaine). Ces types doivent être créés dans l'éditeur de types énumérés, dans lequel on définit le nom du type et ses différentes valeurs, pour pouvoir être ensuite utilisés dans la définition des attributs d'une classe agent.

3.3.4 Editeur de symboles

94 L'éditeur de symboles permet d'enrichir la bibliothèque de symboles disponibles pour la visualisation des simulations dans les cartes géographiques. La création du symbole est réalisée par un dessin vectoriel.

3.4 Deux inspecteurs pour explorer les modèles

3.4.1 L'inspecteur général du modèle

95 L'exploration d'un modèle peut être réalisée en cours de simulation à l'aide de l'inspecteur du modèle instancié, accessible par le menu *Outils* de la fenêtre principale. Il propose une visualisation arborescente et récursive d'un modèle. Cet inspecteur nous renseigne sur l'état des agents (valeur des attributs, liste des hôtes et passagers) et des organisations (liste des agents composants).

3.4.2 L'inspecteur d'agent

Un second inspecteur est activable à partir d'une carte. Pour cela il faut activer l'outil de sélection des composants ().

Un double-clic sur un composant le sélectionne et provoque l'ouverture de l'inspecteur d'agent qui affiche les valeurs de ses attributs ainsi que la liste de ses passagers. Un double clic sur un des passagers permet d'ouvrir un autre inspecteur sur cet agent. On peut également observer l'évolution de l'état de plusieurs agents en cours de simulation. On peut enfin observer l'évolution d'agents non localisés à partir de l'inspecteur général du modèle, où par un double-clic sur l'agent choisi, on ouvre l'inspecteur de cet agent.

3.5 Les fichiers

3.5.1 L'interopérabilité des modèles MAGéo

96 Un modèle MAGéo est sauvegardé sous forme de fichier XML, dans un fichier portant le nom du modèle, avec l'extension « *.mgo* ». En effet un modèle étant essentiellement structuré de manière arborescente, le langage XML convient tout à fait pour enregistrer une telle structure. De plus les fichiers XML de MAGéo sont conformes à un XML SCHEMA nommé *UAOC_SchemaXML.xsd* ce qui permet de valider un fichier en dehors de MAGéo, d'en comprendre l'organisation dans un éditeur XML et éventuellement d'en modifier le contenu. La présence d'un tel schéma offre une ouverture du logiciel en permettant éventuellement l'utilisation de ses modèles dans d'autres applications.

3.5.2 Organisation des données

97 Pour l'initialisation d'organisations complexes qui utilisent des tables attributaires, des cartes ou des réseaux, mais aussi pour initialiser des valeurs d'attributs des agents, d'autres données peuvent être nécessaires. Dans ce cas, elles sont enregistrées dans un dossier créé automatiquement lors de l'importation des données et porte le nom du modèle avec l'extension « *.dgo* ». Ainsi, le modèle est enregistré sous *modèle.mgo* et les données dans le dossier *modèle.dgo*. Ces données sont stockées sous forme de fichiers de divers formats qui sont référencés dans le fichier principal « *modèle.mgo* ». Ainsi nous aurons :

- Table attributaire en format ascii tabulé ;
- Fichiers grid ascii (format ArcView) ;
- Fichiers cartographiques : Les maillages topologiques (utilisés pour les maillages irréguliers et les réseaux), sont enregistrés selon un format propriétaire en binaire qui comprend :
    - La géométrie, ou fichier des points (extensions « .g2D » ou « .g3D »)
    - Les arcs, qui contient la structure du graphe topologique, (extension « .topz »)
    - Les faces, pour les maillages irréguliers (extension « .faces »)

98 L'importation de fichiers cartographiques issus de SIG est possible dans MAGéo à l'aide d'une application externe, « *ImportShapeFile.exe* », qui se lance lorsqu'on demande à MAGéo d'importer une carte et permet d'importer des ShapeFiles et des fichiers au format *generate* (ESRI, ArcInfo).

99 D'autres types de fichiers de données existent, par exemple pour les réseaux :

- Table des distances entre nœuds, pour les réseaux (extension « .dist »)
- Table des pancartes, contenant les chemins optimaux entre nœuds (extension « .pan »)

100 Enfin des fichiers de données générales utilisables dans les modèles sont présents :

- Fichier des symboles : ce fichier contient la description des symboles en format texte, il doit être présent dans le dossier contenant l'exécutable de MAGéo ;
- Fichiers de légendes : à tout moment on peut enregistrer la légende associée à une représentation cartographique pour pouvoir la réutiliser (Extension « .lgc »).

4 Comment MAGéo simule-t-il un modèle ?

101 Lorsqu'un modèle conceptuel est apte à être exécuté, le dialogue de la partie basse de la fenêtre de simulation permet de gérer toutes les phases de la simulation. Le bouton ***init*** exécute la phase d'instanciation, puis celle d'initialisation du modèle. Le système est alors dans son état initial, prêt à la simulation, soit pour exécuter un pas de temps, soit pour le faire « tourner », c'est-à-dire enchaîner l'exécution des pas de temps successifs, jusqu'à faire une pause demandée par l'utilisateur, ou jusqu'à ce que la méthode d'arrêt soit activée.

4.1 Instanciation (agents, organisations, comportements, relations)

102 La phase d'instanciation parcourt le modèle conceptuel de manière récursive depuis l'agent-modèle et instancie chaque classe d'agent pour réaliser les instances (en mémoire de l'ordinateur) d'agents individualisés, selon leur structure attributaire, leurs comportements et leur système interne. Elle instancie ensuite les relations entre classes, qui produisent les instances de ces relations entre les instances d'agents. Le système interne d'un agent est composé de références à des classes d'organisation. Ces classes sont aussi instanciées pour produire des instances d'organisations dans les instances d'agents.

4.2 Initialisation (attributs)

103 La seconde phase, une fois que les entités physiques (les instances) sont créées, consiste à définir les valeurs des différents attributs des agents et des relations, qui constitue l'état initial du système $E(t_0)$. Ces valeurs initiales peuvent être soit calculées par différents scripts soit chargées depuis différents fichiers.

4.3 Exécution d'un modèle

4.3.1 Fonctionnement général

104 A partir de l'instant initial ($t = t_0$) comme à tout autre instant ($t > t_0$), lorsqu'un modèle est exécuté, le temps du modèle passe de la valeur t à l'instant suivant $t+dt$ selon le pas de temps constant dt. Le modèle évolue alors en calculant son nouvel état $E(t+dt)$ à partir de son état $E(t)$ à l'instant t. Pour calculer son nouvel état, il doit passer en revue toutes les entités physiques actives i à tous les niveaux du modèle, dans un certain ordre, de manière à calculer leur nouvel état $e_i(t+dt)$. Cette modification se fait à partir de tous les comportements, des règles contenues dans les diagrammes de transition, des champs calculés dans les attributs, des programmes d'action des agents, de l'ordre d'exécution des organisations internes aux agents, ainsi que toutes les expressions qui permettent l'affichage et les sorties du modèle. Le modèle suit pour cela un ordonnancement prédéfini dans lequel les entités doivent être exécutées.

105 Ainsi, à chaque pas de temps et dans une logique montante, allant des entités les plus fines (terminales) aux entités les plus complexes (composées d'entités plus fines), le modèle va se transformer récursivement, niveau par niveau, pour passer de l'état $E(t)$ à $E(t+dt)$.

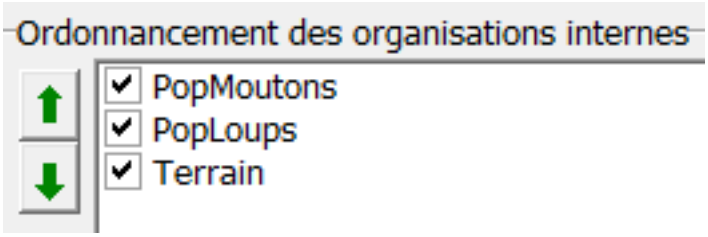
106 Plus précisément, au cours d'un pas de temps, l'agent modèle (mais c'est la même chose pour tout agent) commence par exécuter son niveau inférieur, c'est-à-dire chacune de ses organisations internes, puis il exécute son propre niveau, c'est-à-dire son *programme d'action* qui définit la manière dont ses comportements doivent se mettre ou non en action.

107 L'ordonnancement de MAGéo est donc réalisé de façon à la fois décentralisée et hiérarchisée, chaque entité étant responsable du déroulement des opérations liées à son contenu. Nous privilégions ici les rapports verticaux (d'emboîtement) entre les entités avant les rapports horizontaux (de voisinage). En d'autres termes, un agent attend que ses composants en profondeur soient calculés avant de transformer ses propres attributs par ses comportements, puis on passe à l'agent suivant du même niveau qui effectue également une descente avant de se transformer lui-même.

4.3.2 Ordonnancement des organisations internes à un agent

108 Dans le cas où un agent possède plusieurs organisations internes, il incombe à cet agent de décider de l'ordre dans lequel elles s'exécuteront. Ceci peut être paramétré dans le dialogue de l'agent, où l'on peut modifier à l'aide de flèches l'ordre d'initialisation et d'exécution des organisations internes. Dans l'exemple de la *Figure 14*, l'agent global du modèle Proie-prédateur exécute dans l'ordre suivant ses 3 organisations internes : la population de *moutons*, la population de *loups,* puis l'organisation spatiale *terrain*. La modification de l'ordonnancement se fait directement sur l'interface par la sélection d'une organisation et son repositionnement par les flèches.

Figure 14 : Exemple d'ordonnancement des organisations internes à un agent

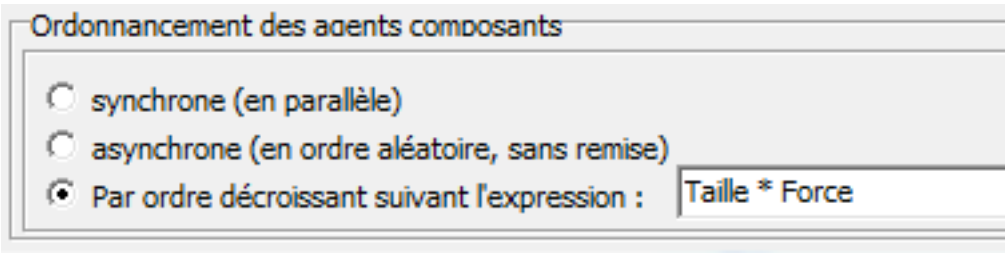


4.3.3 Ordonnancement des agents composant une organisation

109 Lors de l'exécution d'une organisation, celle-ci exécute chacun de ses composants selon le choix indiqué dans *Ordonnancement des agents composants* (voir Figure 15). Le dialogue de paramétrage d'une organisation permet de choisir la méthode d'ordonnancement de ses composants. Trois types d'ordonnancement sont possibles :

- ***Synchrone*** : les agents exécutent leurs comportements en parallèle les uns des autres. Cependant, le parallélisme parfait étant impossible sur un ordinateur séquentiel, le mode synchrone simule le parallélisme de la manière suivante : pour ne pas être influencé par un agent voisin qui aurait changé d'état avant lui, chaque agent (et pour chacun de ses comportements) calcule d'abord la pertinence à agir à partir de l'état de tous les agents à l'instant *t*. Une fois toutes ces pertinences calculées, les agents agissent tous en séquence. Cette méthode permet à la fois d'approcher le paradigme du parallélisme, tout en évitant des conflits entre agents. On parle dans ce cas de pseudo-parallélisme ;
- ***Asynchrone*** : les agents réalisent leur programme d'action les uns après les autres, dans un ordre aléatoire différent à chaque pas de temps, de façon à ne pas générer de biais systématique ;
- ***Par ordre de priorité*** : les agents réalisent leur programme d'action les uns à la suite des autres, par ordre décroissant d'une priorité donnée par une expression. Dans l'exemple ci-dessous, cette expression est le produit de deux attributs des agents concernés, de sorte que les premiers agents à réaliser leur programme d'action seront ceux pour lesquels ce produit est le plus grand.

Figure 15 : Ordonnancement des agents composants d'une organisation



4.4 Comment s'exécute un comportement ?

110 Un comportement élémentaire est défini comme un couple < ***pertinence***, ***action*** >. Si la pertinence est positive, l'action est effectuée si aucun comportement concurrent n'a calculé une plus grande pertinence. Si plusieurs comportements sont en concurrence et de même pertinence, un tirage aléatoire uniforme est réalisé pour choisir celui qui sera effectué.

4.4.1 Ordonnancement des comportements par le programme d'action d'un agent

111 Le programme d'action d'un agent est un script de la classe d'agent qui permet de combiner plusieurs de ses comportements pour définir la dynamique finale des agents de la classe au cours d'un pas de temps. Pour cela un ensemble d'opérateurs d'ordonnancement des comportements est défini.

112 Si A et B sont deux comportements, chaque opération ci-dessous combine A et B pour former un nouveau comportement, qui est exécuté au cours d'un même pas de temps :

- l'opération « A **;** B » est ***la mise en séquence*** des comportements A et B. Dans ce cas, A est effectué, puis B est effectué à la suite et dans le même pas de temps.
- l'opération « A **|** B » ou « A **|** B**|** …**|** C » est ***la mise en concurrence*** de deux ou plusieurs comportements. Dans ce cas, c'est le comportement le plus pertinent d'entre eux qui seul sera effectué. Si plusieurs ont une pertinence maximale égale, un tirage au sort est effectué entre eux pour en exécuter un seul.
- l'opération « A **?** B » correspond à un tirage aléatoire entre les deux comportements.
- l'opération « A **>** B » est une implication : si A est pertinent, alors A puis B sont effectués en séquence, sinon aucun des deux n'est exécuté.

113 Enfin on peut utiliser des parenthèses pour mieux structurer une combinaison de plusieurs comportements. Par exemple, dans un modèle de fourmis qui ramassent des grains épars et tentent de les rassembler en tas, on pourra écrire le programme d'action :

(ramasse **|** dépose) **;** bouge

114 Le comportement *ramasse* est pertinent si la fourmi n'est pas déjà chargée et s'il y a un grain isolé à prendre où elle se trouve, le comportement *dépose* est pertinent si elle est chargée et s'il y a d'autres grains à proximité, enfin le comportement *bouge* est tout le temps pertinent. Selon l'ordonnancement de ces 3 actions élémentaires, la fourmi met donc en concurrence les actions ramasse et dépose, puis enchaine avec l'action bouge.

4.4.2 Ordonnancement des règles de transition dans un diagramme de transition

115 Comme on l'a vu, un diagramme de transition est un graphe dont les sommets sont les états d'une variable d'état de l'agent et les arcs sont les transitions possibles entre ces états. A chacun des n arcs du diagramme de transition est associée une règle de transition R_i qui est un comportement élémentaire particulier. Le programme d'action implicite qui ordonnance les règles d'un diagramme équivaut à un comportement résultant R défini par :

$$R = R_1 \mid R_2 \mid \ldots \mid R_i \mid \ldots \mid R_n$$

5 Le langage MAGéo

116 MAGéo propose un langage permettant d'écrire les différents scripts présents dans les comportements et dans tout autre objet graphique. La plupart du temps, un script se résume à une expression ou à quelques instructions (comme dans la partie *action* d'un comportement ou dans une fonction). Néanmoins le langage MAGéo permet aussi d'écrire des scripts plus algorithmiques puisqu'il possède des formes syntaxiques de contrôle des langages algorithmiques, comme les tests ou les boucles[5]. Ces fonctionnalités algorithmiques sont inutiles dans la grande majorité des modèles développés sous MAGéo, ce qui répond aux objectifs initiaux de ne pas recourir à la logique algorithmique, mais permet néanmoins aux modélisateurs plus experts de les utiliser si besoin. Ce langage a été conçu pour écrire des scripts courts, de une à quelques lignes. Une expression ressemble quelquefois à une requête SQL, mais elle est totalement adaptée à la modélisation par agents de l'environnement MAGéo.

5.1 Où a-t-on besoin du langage ?

5.1.1 Scripts d'initialisation et de dynamique d'attribut et autres petits scripts

117 Le langage MAGéo est pour l'essentiel utilisé dans la définition des comportements et des fonctions, et secondairement dans les circonstances suivantes :

- Pour initialiser les attributs des agents ;
- Pour initialiser les relations (de localisation, les relations binaires ou n-aires) ;
- Pour définir l'évolution d'un attribut, dans la rubrique « dynamique » de l'attribut ;
- Pour définir les règles de transition d'un diagramme de transition ;
- Pour définir le programme d'action d'une classe agent ;
- Dans un graphique temporel, pour calculer la valeur à mettre en ordonnée à chaque pas de temps ;
- Dans un diagramme de phase, pour calculer les valeurs de x et de y à chaque pas de temps ;
- Dans un histogramme de classes, pour désigner la quantité évaluée pour chaque individu de la population, en vue de la répartir dans des classes ;
- Dans l'éditeur de fonctions ;
- Dans une organisation, pour définir sa taille.

118 Mais revenons aux comportements, où le langage prend toute son importance.

5.1.2 Scripts de comportement simple

119 Nous avons vu que le dialogue d'un comportement simple comprend trois zones de texte où se répartit le script de celui-ci : la zone des variables temporaires, la zone de pertinence et la zone d'action.

5.1.2.1 Script de la zone des variables temporaires

120 Dans la zone des variables temporaires, on écrit des affectations - une par ligne - de la forme :

```
VarTemporaire := Expression
```

121 Le langage de MAGéo n'étant pas déclaratif, la variable VarTemporaire est créée automatiquement et son type est déterminé par le type de l'expression. On peut aussi définir des variables temporaires dans la partie action si celles-ci ne sont pas utilisées dans la pertinence.

5.1.2.2 Script de la zone de pertinence

122 Dans la zone de pertinence, on doit écrire une expression en logique continue. Par exemple « Loc.Herbe > 0 » est une expression purement booléenne (avec « True = 1 » et « False = 0 »). On pourrait écrire aussi l'expression sous forme de rapport : « Loc.Herbe / MaxHerbe », qui est ici une expression en logique continue à condition que 0 # Loc.Herbe # MaxHerbe. La pertinence produit donc un résultat logique et ne modifie pas l'état du système. Lors de la simulation, les variables temporaires ainsi que la pertinence sont instanciées dans chaque agent et mémorisent des informations propres à celui-ci.

5.1.2.3 Script de la zone d'action

123 Le code de la zone d'action est en revanche destiné à modifier l'état du système, à travers des affectations d'attributs, des appels de fonctions (par exemple de mobilité, de destruction ou de création de nouveaux agents). Chaque action est écrite sur une ligne. Par exemple :

```
Force := Force / 2
VarClone := Self.Clone()
VarClone.Age := 0
```

124 La variable Self est une variable prédéfinie indiquant l'instance d'agent qui exécute le comportement, on parlera d'agent courant. Force est ici un attribut de l'agent courant, et VarClone une variable locale de type Agent. L'appel de fonction Self.Clone() s'applique à l'agent Self, elle permet la création d'un nouvel agent identique. La notation VarClone.Age fait référence à l'attribut Age du nouvel agent VarClone.

5.2 Le langage d'expression

125 Une expression possède un type : entier (integer), réel (decimal) dont le séparateur décimal est le point, logique (logic), liste (list), vecteur (vector) ou texte (text).

126 Une expression numérique est construite par combinaison de variables définies dans le modèle (attributs, paramètres…), à l'aide des opérations arithmétiques habituelles avec des parenthèses ou des appels de fonctions, selon les règles usuelles des expressions algébriques (par exemple : Force+3*(age–17), ou bien : k*sin((x-x0)*pi/180).

127 Une expression peut être de type logique (booléenne ou continue), on la construit là aussi à l'aide des opérateurs classiques (And, Or, <, IN, …). Elle peut aussi être de type liste, ensemble ou vecteur. Une liste est une suite homogène de données (des nombres, des références d'agents, des valeurs de types énumérés comme les jours de la semaine), un ensemble est une liste sans doublons et un vecteur est une liste de nombres d'une certaine longueur (sa dimension).

128 MAGéo propose de nombreux outils pour leur construction, par exemple des opérations ensemblistes (Union, intersection, inclusion…), ainsi que différentes syntaxes de sélection pour affecter une liste L comme :

```
L := ALL x FROM List WHERE Condition
L := ClasseAgent.Attribut
```

129 Une fonctionnalité importante du langage, qui simplifie le traitement des vecteurs, est la généralisation des opérations arithmétiques aux listes de nombres. De manière générale, une opération sur un ou plusieurs vecteurs équivaut au vecteur des opérations sur ses termes. Par exemple, la somme de deux vecteurs (de même dimension) est le vecteur des sommes terme-à-terme des éléments des deux vecteurs. Et une règle d'inflation considère un nombre x comme le vecteur [x, x, …, x], s'il est combiné avec un vecteur V = [y1, y2,…, yn]. Par exemple :

$$x + V = [x, x, \ldots, x] + V = [x+y1, x+y2, \ldots, x+yn]$$
$$\log(V) = [\log(y1), \log(y2), \ldots, \log(yn)]$$

5.3 Syntaxe des fonctions dépendant du contexte

130 Dans une organisation de type réseau, un mobile ne se déplace pas de la même manière que dans une organisation de type carroyage ou dans un espace continu. De ce fait certaines fonctions d'action, comme les déplacements, dépendent du contexte dans lequel l'agent se situe au moment d'agir.

5.3.1 Fonction indépendantes du contexte, en contexte d'agent, en contexte d'organisation

131 Lorsqu'une fonction est indépendante du contexte, il suffit de l'appeler par son nom avec éventuellement certains paramètres. Par exemple les fonctions mathématiques comme sin(x) sont indépendantes du contexte, on les écrit ainsi, sans autre précision. Par contre il existe des fonctions qui dépendent de l'agent, d'autres qui dépendent d'une organisation. On peut même rencontrer la nécessité de donner l'agent et l'organisation dans lequel il se situe.

132 Lorsque l'on doit indiquer que la fonction Fonc(x) opère sur l'agent Agt, on place la référence de l'agent en préfixe de la manière suivante : Agt.Fonc(x)

133 Lorsque l'on doit indiquer que la fonction Fonc(x) opère dans l'organisation Org, on place la référence de l'organisation en préfixe de la manière suivante : Org:Fonc(x)

134 Lorsque l'on doit indiquer que la fonction Fonc(x) opère à la fois sur l'agent Agt et dans l'organisation Org, on place les deux références en préfixe de la façon suivante : Org:Agt.Fonc(x)

135 Néanmoins la plupart du temps, lorsqu'il n'y a pas d'ambigüité sur le contexte, le compilateur le trouve tout seul sans qu'il soit nécessaire de le lui indiquer. Ainsi, l'expression Réseau:MoveToward(MonTravail,30) équivaut par exemple à Réseau:Self.MoveToward(MonTravail,30), et signifie que l'agent-passager courant Self se déplace par pas de 30m sur l'organisation appelée Réseau, en direction de l'agent MonTravail.

5.3.2 Fonctions définies par l'utilisateur

136 Les fonctions prédéfinies dans MAGéo ne peuvent pas couvrir toutes les thématiques de la modélisation, aussi l'utilisateur peut-il lui-même définir les fonctions spécifiques à son modèle, sous forme de graphique ou par un script.

5.3.2.1 Fonctions définies par une courbe

137 La création d'une fonction sous forme de graphique a été décrite précédemment. Une telle fonction F(x) peut être utilisée, dans le corps d'un comportement, en précisant la valeur du paramètre d'entrée, par exemple, F(Self.force) ou F(0.2317).

5.3.2.2 Fonctions définies par un script

138 L'éditeur de fonctions également décrit précédemment permet d'écrire le script d'une fonction en deux parties : la partie facultative des variables temporaires, puis le corps de la fonction écrite entre l'en-tête de fonction : « FUNCTION Nom(ListeParamètres):type » et le mot de fin « END ». Le résultat de la fonction doit être affecté à la variable prédéfinie Result.

6 Bilan et perspectives

139 Depuis ses débuts en 2009, MAGéo a connu de nombreuses évolutions qui ont pu être testées notamment lors de sessions d'enseignement en Master de Géographie et en programme intensif Erasmus[6]. Ces sessions ont d'une part confirmé l'intérêt d'une approche de modélisation individu-centrée et spatialisée chez les jeunes chercheurs et d'autre part ont confirmé l'attrait que pouvait susciter une plateforme de modélisation aussi intuitive que MAGéo auprès de chercheurs non enclins à la programmation informatique. Cet attrait s'est également révélé lors de la présentation de MAGéo au salon « Innovatives SHS » de la valorisation en sciences humaines et sociales organisé par le CNRS et l'INSHS en 2013 et en juin 2015 ou lors des ateliers de formation du collectif MAPS (Collectif MAPS, 2014).

140 MAGéo V3.5 est maintenant dans un état de développement qui lui permet de modéliser des systèmes complexes spatialisés dans diverses thématiques, par exemple sur la circulation automobile en situation de catastrophe, sur la dynamique des comportements de vote ou dans le cadre de la diffusion d'épidémie. Ces différents domaines de recherche permettent de mesurer les champs d'applications de MAGéo et de faire évoluer les outils utiles aussi bien à la modélisation qu'à la simulation.

141 Plusieurs évolutions sont en cours de réalisation ou envisagées à plus ou moins court terme :

- L'amélioration de l'interface pour les modèles multiniveau, en particulier pour gérer les instances d'organisation dans un contexte distribué ;
- L'introduction de la notion d'interaction, utile pour la simulation d'objets mobiles soumis à des contraintes, des objets qui s'entrechoquent, etc. Une interaction est vue comme un « comportement » particulier qui met en action réciproque et simultanée plusieurs agents ;
- MAGeo permet actuellement de simuler très simplement les systèmes d'équations différentielles. Aussi, l'introduction d'outils de gestion de systèmes dynamiques dont les diagrammes dynamiques (Forrester, 1984) qui permettent de définir graphiquement la dynamique de variables quantitatives et simplifient la conception de systèmes dynamiques basés sur des équations différentielles est envisagée ;
- L'introduction de composants d'observation sous forme de scène en 3D.

Bibliographie

Banos A., Boffet-Mas A., Charbonnel S., Lang C., Marilleau N., Thevenin T., 2010, « Simuler la mobilité urbaine quotidienne : le projet MIRO », in Banos A., Thevenin T. (dir.), *Mobilités urbaines et risques des transports : approches géographiques*, Hermès, 51-88.

Benenson I., Torrens P., 2004, *Geosimulation: Automata-based modeling of urban phenomena*, England, John Wiley & Sons Ltd.

Bonnefoy J.-L., 2003, « From households to urban structures: space representations as engine of dynamics in multi-agent simulations », *Cybergeo* : *http://cybergeo.revues.org,* No.234, 31 janvier 2003.

Bousquet F., Bakam I., Proton H. Le Page C., 1998, « Cormas: common-pool resources and multi-agent Systems », *Lecture Notes in Artificial Intelligence,* No.1416, 826-838.

Badariotti D., Banos A., Laperrière V., 2007, « Vers une approche individu-centrée pour modéliser et simuler l'expression spatiale d'une maladie transmissible : la peste à Madagascar », *Cybergeo* : *http://cybergeo.revues.org,* No.393, 16 juillet 2007.

Bretagnolle A., Daudé É., Pumain D., 2006, « From theory to modelling: urban systems as complex systems », *Cybergeo* : *http://cybergeo.revues.org,* No.335, 08 mars 2006.

Collectif MAPS, 2014, *Recueil de fiches pédagogiques du réseau MAPS : Modélisation multi-agents appliquée aux phénomènes spatialisés 2009-2014*, RNSC-CNRS.

Collier N., 2003, « Repast: An extensible framework for agent simulation », *The University of Chicago's Social Science Research*, No.36.

Daudé É., 2003, « Apports de la simulation multi-agents à l'étude des processus de diffusion », *Cybergeo* : *http://cybergeo.revues.org,* No.255, 20 février 2003.

Daudé É., 2005, « Systèmes multi-agents pour la simulation en géographie : vers une Géographie Artificielle », in Y. Guermont (dir.), *Modélisation en Géographie : déterminismes et complexités*, Hermès, 353-380.

Delahaye D., Guermond Y., Langlois P., 2002, « Spatial interaction in the run-off process », *Cybergeo* : *http://cybergeo.revues.org*, No.213, 10 mai 2002.

Drogoul A., Amouroux E., Caillou P., Gaudou B., Grignard A., Marilleau N., Taillandier P., Vavaseur M., Vo D.-A., Zucker J.-D., 2013, « GAMA: A Spatially Explicit, Multi-level, Agent-Based Modeling and Simulation Platform », *Practical Applications of Agents and Multi-Agent Systems*, Lecture Notes in Computer Science, Spain, Springer Berlin Heidelberg, 271-274.

Dubos-Paillard E., Guermond Y., Langlois P., 2003, « Analyse de l'évolution urbaine par automate cellulaire : le modèle SpaCelle », *L'espace géographique*, No.4, 357-378.

Forrester J.W., 1984, *Principes des Systèmes*, Presses Universitaires de Lyon.

Klopfer E., Scheintaub H., Huang W, Wendel D., 2009, « StarLogo TNG: Making Agent Based Modeling Accessible and Appealing to Novices In Artificial Life Models in Software ».

Langlois P., 1994, « Formalisation des concepts topologiques en géomatique », *Revue internationale de géomatique*, n° 4, 181-205.

Langlois P., 2010, *Simulation des systèmes complexes en géographie, fondements théoriques et applications*, Paris, Hermès-Lavoisier.

Langlois P., Daudé É., Blanpain B., Sapin E., 2010, « AOC, une ontologie formelle pour la modélisation de systèmes complexes en géographie », *SAGEO'2010, Conférence internationale de Géomatique et Analyse Spatiale*, Toulouse.

Luke S., Cioffi-Revilla C., Panait L., Sullivan K., Balan G., 2005, « MASON : A multiagent simulation environment », *Simulation*, No.81(7), 517-527.

Lytinen S.L., Railsback S.F., 2012, « The evolution of agent-based simulation platforms: A review of NetLogo 5.0 and ReLogo », *Proceedings of the Fourth International Symposium on Agent-Based Modeling and Simulation*, Vienna, Austria.

Marietto M., David N., Sichman J., Coelho H., 2003, « Requirements analysis of agent-based simulation platforms: state of the art and new prospects », *Multi-Agent-Based Simulation II*, Lecture Notes in Computer Science, vol. 2581, 125–141.

Minar N., Burkart R., Langton C., Askenazi M., 1996, « *The swarm simulation system: A toolkit for building multi-agent simulations* », Working paper 96-06-042, Santa Fe Institute, USA.

North M.J., Collier N., Ozik J., Tatara E., Macal C., Bragen M., Sydelko P. 2013, « Complex Adaptive Systems Modeling with Repast Simphony », *Complex Adaptive Systems Modeling*, Heidelberg, Springer.

Phan D., Amblard F., (dir.) 2007, *Agent-based modelling and simulation in the social and human sciences*, Oxford, Bardwell Press.

Pumain D., Sanders L., Mathian H., Guerin-Pace F., Bura S., 1995, « SIMPOP, A Multi-Agents Model for the Urban Transition », in Sikos T., Bassa L., Fischer M. (eds), *Recent Developments in Spatial Information, Modelling and Processing*, Budapest, Studies in Geography in Hungary

Railsback S.F., Lytinen S.L, Jackson S.K., 2006, « Agent-based Simulation Platforms: Review and Development Recommendations », *Simulation*, No.82(9), 609–623.

Tranouez P., Daudé É., Langlois P., (2012), « A multiagent urban traffic simulation, », *Journal of Nonlinear Systems and Applications*, vol. 3, No.2, 98-106

Varela F., Maturana H., *1994*, *L'arbre de la connaissance*, Addison-Wesley.

Varenne F., 2011, *Modéliser le social*, Paris, PsychoSup.

Wilensky U., 1999, *NetLogo,* http://ccl.northwestern.edu/netlogo, Center for Connected Learning and Computer-Based Modeling, Northwestern Institute on Complex Systems, Northwestern University, Evanston, IL.

Notes

1 Pour une revue et une comparaison des plateformes, voir http://www.openabm.org/page/modeling-platforms

2 La notion d'interaction utilisée dans cet article est celle définie dans (Langlois 2010, chap. 6, *l'interaction spatiale*) qui propose que l'interaction entre des entités spatiales est le résultat d'interactions élémentaires entre couples d'entités. Une interaction élémentaire entre deux entités est un changement d'état simultané de ces entités (c'est-à-dire au cours d'un même pas de temps), qui est provoqué par l'action d'un médiateur. Une interaction de flux entre des entités spatiales est le résultat des interactions élémentaires médiatisée par ce flux qui provoque le passage simultané d'une certaine quantité (de matière, d'énergie ou d'information) d'une entité à l'autre, modifiant ainsi l'état du stock dans chaque entité au cours du temps.

3 Voir méthode SADT par exemple : http://en.wikipedia.org/wiki/Structured_Analysis_and_Design_Technique

4 Libre au sens où elle n'est pas imposée par le méta-modèle de MAGéo, mais laissée à la liberté du modélisateur.

5 Par exemple les tests : *IF condition THEN BlocAction1 [ELSE BlocAction2] ENDIF*, les boucles : *LOOP n BlocAction ENDLOOP, ou encore : FOR x IN liste DO BlocAction ENDFOR)*

6 http://www.unil.ch/mastergeographie/home/menuinst/ateliers-master--stages/intensive-program---mgm.html

---

Pour citer cet article

Référence électronique

Patrice Langlois, Baptiste Blanpain et Eric Daudé, « MAGéo, une plateforme de modélisation et de simulation multi-agent pour les sciences humaines », *Cybergeo : European Journal of Geography* [En ligne], Systèmes, Modélisation, Géostatistiques, document 741, mis en ligne le 02 octobre 2015, consulté le 04 octobre 2015. URL : http://cybergeo.revues.org/27236 ; DOI : 10.4000/cybergeo.27236

À propos des auteurs

Patrice Langlois
CNRS, UMR IDEES 6266, CNRS-Université de Rouen, Professeur des Universités
patrice.langlois@univ-rouen.fr
Baptiste Blanpain
CNRS, UMR IDEES 6266, CNRS-Université de Rouen, Ingénieur de recherche
baptiste.blanpain@univ-rouen.fr
Eric Daudé
CNRS, UMIFRE CSH 20, CNRS-MAE Delhi, Inde, Chargé de recherche
eric.daude@univ-rouen.fr

Droits d'auteur

© CNRS-UMR Géographie-cités 8504

Résumés

Cet article présente MAGéo (Modélisation Agent Géographique), une plateforme de simulation multi-agent dédiée aux problématiques spatialisées et multiniveau, suffisamment visuelle et intuitive pour être accessible aux utilisateurs sans expérience en programmation informatique. La plateforme MAGéo intègre en un seul outil toutes les phases du cycle de vie d'un modèle, de la conception à la réalisation et jusqu'à la simulation dans l'espace et le temps permettant la validation. Ce cycle est réitérable dans une spirale de complexification progressive. Chaque modèle est conçu sous forme graphique à travers son modèle conceptuel (ou ontologie) et le modélisateur construit une interface graphique de pilotage et d'observation propre à chaque modèle. Ces constructions se font sous forme graphique, à l'aide de boîtes

et de relations à organiser et à paramétrer. L'ontologie du modèle repose sur le méta-modèle AOC qui intègre les entités Agent, Organisation, Comportement et qui permet de décrire des modèles complexes et multiniveau. Le modèle dynamique est exécuté dans l'interface de simulation à l'aide de différents objets graphiques de paramétrage du modèle, tels que curseurs, boutons, boites de saisie, ainsi que des cartes et des graphiques pour ses sorties. L'ensemble de ces éléments font de MAGéo un logiciel de géosimulation idéal aussi bien pour l'apprentissage des systèmes complexes spatialisés et de leur modélisation, dans le cadre d'activités d'enseignement et de séminaires de formation par exemple, que pour les activités de recherche en SHS. MAGéo permet d'imaginer une gamme très importante de modèles dynamiques et répond donc aux besoins des modèles les plus exigeants. Enfin la rapidité et la transparence dans laquelle sont construits des modèles font également de MAGéo un outil de construction de prototypes de modèles de complexité croissante très utile dans les projets pluridisciplinaires.

This paper presents MAGeo (Modeling Agent in Geography), a user-friendly agent-based simulation tool dedicated to spatialized and multilevel researches. Model development with MAGeo does not require any particular programming skills. The conceptual model is realized through graphical objects, such as boxes and links to organize them. Model's ontology is based on the following three entities: Agent - Organization - Behavior that allow the building of multilevel model. The modeler has access to the simulation window in order to develop its model's dynamic. Simulation model is built using cursors and other graphical tools to parametrize the model, as well as maps and graphs to explore model's behavior. MAGeo is readily accessible to social scientists which aim to develop spatialized and complex models but without programming skill. Because learning MAGeo and developing model is not time-consuming, MAGeo is also well designed for teaching complex system. Finally its graphical approach produces perfect systemic and multilevel graphics for communication in multidisciplinary teams.

Entrées d'index

Mots-clés : plateforme de simulation, modélisation individu centrée et spatialisée, systèmes complexes, simulation multi-agent, ontologie
Keywords : simulation platform, Agent Based Modeling (ABM), complex systems, Multi-Agent Simulation, ontology