

# Des données géographiques à la simulation à base d'agents : application de la plate-forme GAMA

## Introduction

Ces dernières années ont été marquées par l'essor des approches scientifiques liées à la complexité. Ces approches offrent de nombreuses perspectives tant pour la recherche fondamentale que pour la prospective territoriale. Elles fournissent en effet des concepts (émergence, non-linéarité) et des méthodes (modélisation, simulation) permettant d'étudier un territoire à différentes échelles et sous différents angles.

Parmi ces approches, une a été particulièrement adoptée par les géographes : la modélisation à base d'agents (Crooks et Castle, 2012). Cette approche consiste à étudier un système en modélisant les entités qui le composent sous la forme d'agents dont les interactions permettent l'émergence d'une dynamique globale. Un agent est une entité (informatique) autonome ayant ses propres caractéristiques et comportements. Le concept d'agent, qui provient de l'intelligence artificielle, est versatile : un agent peut représenter différents types et niveaux d'entités (par exemple un être humain, une route, une maison, un groupe social ou une ville). L'utilisation de la modélisation à base d'agents pour l'étude des phénomènes spatialisés s'est largement répandue depuis une dizaine d'années. On peut trouver dans la littérature de nombreux exemples d'utilisation d'une telle approche : la diffusion des épidémies dans les zones urbaines (Eubank, 2004), ou les évolutions et les trajectoires urbaines (White et Engelen, 1993 ; Sanders et al., 1997 ; Barros, 2003 ; Daudé, 2004 ; Batty, 2005 ; Banos et al., 2005).

La construction d'un modèle à base d'agents requiert un travail important de programmation informatique car elle nécessite de décrire sous la forme d'algorithmes l'ensemble des agents et leurs comportements. Afin de faciliter le travail des modélisateurs, différentes plates-formes ont été développées (Railsback, 2006). Ces plates-formes peuvent être classées en trois catégories (non-exclusives) en fonction du type de langage utilisé pour définir les modèles :

- La première catégorie comprend les plates-formes qui permettent de définir les modèles à l'aide d'un langage de programmation générique (Java, C++, python...). Ces plates-formes sont en général dédiées aux informaticiens et sont souvent les plus adaptés au développement de modèles complexes comprenant de nombreux agents et processus. Dans cette catégorie, on trouve entre autres les plates-formes SWARM (Minar et al, 1996) et Repast Symphony (North et al., 2013).
- La seconde catégorie comprend les plates-formes qui permettent de définir des modèles à l'aide d'un langage de modélisation dédié. Ces plates-formes sont généralement plus simples à utiliser que celles de la première catégorie et sont donc destinées à une plus large gamme d'utilisateurs. Elles requièrent néanmoins des connaissances en algorithmique. La plate-forme Netlogo (Tisue et Wilensky, 2004) fait partie de cette catégorie.
- La dernière catégorie comprend les plates-formes qui permettent de définir des modèles à l'aide d'un langage de modélisation graphique. Ces plates-formes ne requièrent en général pas ou peu de connaissances en algorithmique mais sont assez limitées en termes de richesse des modèles qu'elles permettent de construire. On trouve dans cette catégorie les plates-formes StarLogo TNG (Resnick, 1996), MAGéo (Langlois et al, 2013) ou AgentSheets (Repenning, 2011).

La plate-forme GAMA (*Gis & Agent-based Modelling Architecture*) se situe dans la deuxième catégorie. À l'image de Netlogo, elle offre un langage complet de modélisation - le GAML (*Gama Modeling Language*) - permettant aux modélisateurs de construire simplement et

rapidement des modèles. Néanmoins, contrairement à Netlogo qui se limite à la construction de modèles simples, GAMA permet la construction de modèles très complexes, aussi riches que ceux construits par un informaticien à partir d'outils tels que Repast Symphony. En particulier, GAMA propose des outils très avancés pour ce qui concerne la gestion de l'espace. Cet article se concentre sur cet aspect de GAMA et plus particulièrement sur les dernières avancées de la plate-forme dans ce domaine. La première partie est dédiée à une présentation générale de la plate-forme GAMA, ce qui permettra ensuite de détailler la façon dont GAMA prend en charge l'espace, puis de présenter les opérateurs spatiaux inclus dans GAMA. La dernière partie se concentre sur la gestion de la 3D.

## Présentation générale de la plate-forme GAMA

GAMA est une plate-forme libre de modélisation et de simulation à base d'agents développée depuis 2007 (GAMA site internet, 2013). GAMA permet de construire des simulations à temps discrets. Cela signifie que le temps est segmenté en un ensemble de pas de simulation ayant tous la même durée : à chaque pas de simulation, tous les agents sont activés et peuvent agir (modification de leur état, de l'état d'autres agents ou de l'état du monde).

La version 1.6 de la plate-forme est disponible depuis octobre 2013. Cette version est déjà utilisée dans le cadre de différents projets de recherche touchant à des problématiques aussi variées que l'épidémiologie, l'évolution de l'occupation des sols, les risques naturels ou technologiques, la gestion de l'eau avec le projet MAELIA (Taillandier et al., 2012 ; Gaudou et al., 2013), la mobilité urbaine avec le projet MIRO 2 (Banos et al., 2012 ; Banos et al., 2013), ou encore la reconstitution d'événements géo-historiques...

La plate-forme peut être librement téléchargée sur le site internet de la plate-forme. Ce site propose un guide pour l'installation de la plate-forme (GAMA installation, 2013), une documentation complète (GAMA documentation, 2013), ainsi que plusieurs tutoriels présentant différents aspects de la plate-forme (GAMA tutoriels, 2013). Une vidéo de démonstration de GAMA est également disponible (GAMA IJCAI démonstration, 2013).

Comme mentionné en introduction, GAMA fournit aux modélisateurs un langage complet de modélisation : le GAML. En outre, GAMA offre un environnement de développement intégré (IDE) permettant de faciliter l'écriture de modèles. Cet IDE, qui intègre un éditeur de code GAML, fournit tous les outils modernes d'aide à la programmation : coloration du texte, auto-compilation des modèles, détection automatique des erreurs, auto-complétion, correction automatique d'erreurs... Le langage GAML a été conçu pour être simple à utiliser et à assimiler, même pour des modélisateurs ayant un faible niveau de programmation. Il se caractérise, à l'image du langage de Netlogo, par une grammaire très simple et par une grande bibliothèque d'opérateurs (près de 300). Le GAML est un langage de programmation orienté agent. Définir un modèle en GAML consiste à définir trois types de bloc : le bloc global, les espèces et les expérimentations.

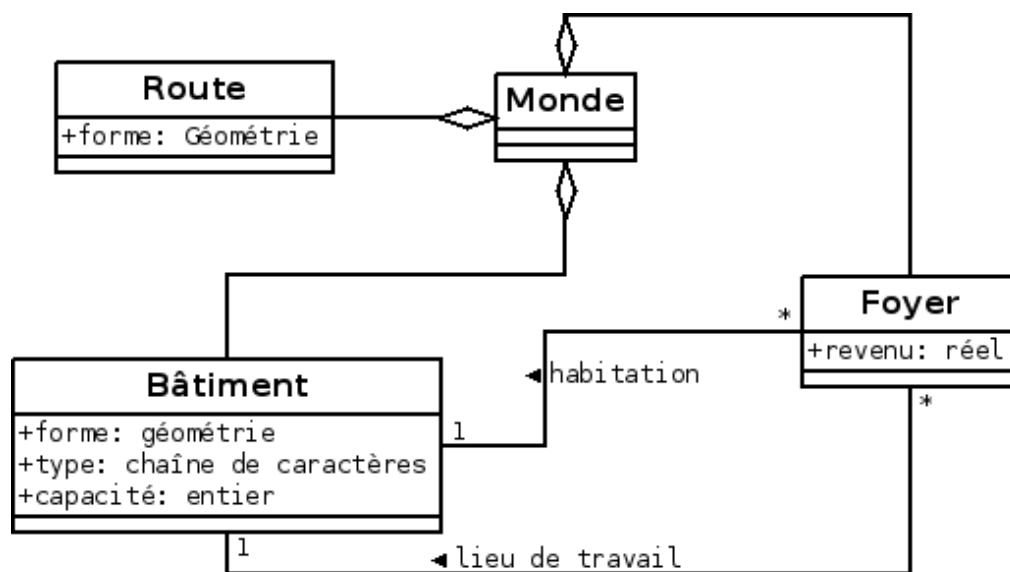
- Le bloc global représente le modèle du monde et en décrit les caractéristiques et les dynamiques (qui peuvent concerner tous les agents). Le lancement d'une simulation correspond à la création du monde qui peut à son tour créer d'autres agents.
- Une espèce – qui correspond en programmation orientée objet à une classe – est un archétype d'agent qui définit pour un type d'agents quels sont ses caractéristiques, ses comportements et ses représentations visuelles.
- Les caractéristiques d'une espèce d'agents sont définies au travers de variables qui définissent l'état interne des agents. Les variables en GAML peuvent être statiques ou dynamiques (recalculées à chaque pas de simulation ou lorsque nécessaire) et sont typées (entier, réel, point, liste, dictionnaire, géométrie, graphe...).

Le comportement d'une espèce d'agents est défini au travers d'actions et de réflexes. Une action représente une capacité que l'agent possède et à laquelle il pourra faire appel. Il s'agit d'un bloc d'instructions qui est exécuté lorsque l'action est appelée. Une action correspond à une méthode en programmation orientée objet. Un réflexe est un bloc d'instructions qui est exécuté à chaque pas de simulation lorsque sa condition d'exécution est vérifiée.

La représentation visuelle d'un agent est définie au travers d'aspects. Un aspect correspond à une façon de visualiser un agent. A noter qu'un agent peut disposer de plusieurs aspects (il est possible de représenter l'agent de différentes façons) et que ces derniers peuvent être (ou non) totalement décorrélés de la géométrie interne de l'agent. GAMA offre dans ce cadre beaucoup de liberté pour l'affichage des agents, permettant d'afficher simultanément tous les types de géométries, des images ou du texte.

Le dernier type de bloc à définir dans un modèle sont les expérimentations qui fournissent des cadres d'exécution d'une simulation. Elles permettent en particulier de définir les entrées (paramètres) et sorties (fenêtre d'affichage, moniteurs) qui seront accessibles aux utilisateurs. Afin d'illustrer la création d'un modèle avec GAMA ainsi que les différentes fonctionnalités de la plate-forme que nous souhaitons mettre en avant, nous proposons de présenter la construction d'un modèle simple sur la ségrégation urbaine. Ce modèle très simple s'inspire fortement du modèle de Schelling (Schelling, 1971). La question posée est celle de l'émergence d'un phénomène de ségrégation basé dans notre cas sur le niveau de revenu des foyers. Nous partons d'un ensemble de foyers habitant dans une ville : chaque foyer est caractérisé par un revenu, un lieu d'habitation et un lieu de travail. Si le revenu moyen des foyers voisins est trop différent du sien, un foyer ne sera pas satisfait et choisira de déménager. Le choix d'une nouvelle habitation se fera aléatoirement parmi les habitations libres – chaque bâtiment a une capacité en termes de nombre de foyers pouvant habiter dedans directement dépendante de sa surface – qui ne sont pas trop éloignées du lieu de travail en prenant en considération la distance par la route. Ce modèle fait intervenir trois types (ou espèces) d'agents : les foyers, les bâtiments et les routes. La figure 1 présente sous la forme d'un diagramme UML ces trois espèces d'agents.

**Figure 1 : Diagramme UML représentant les trois espèces du modèle.**



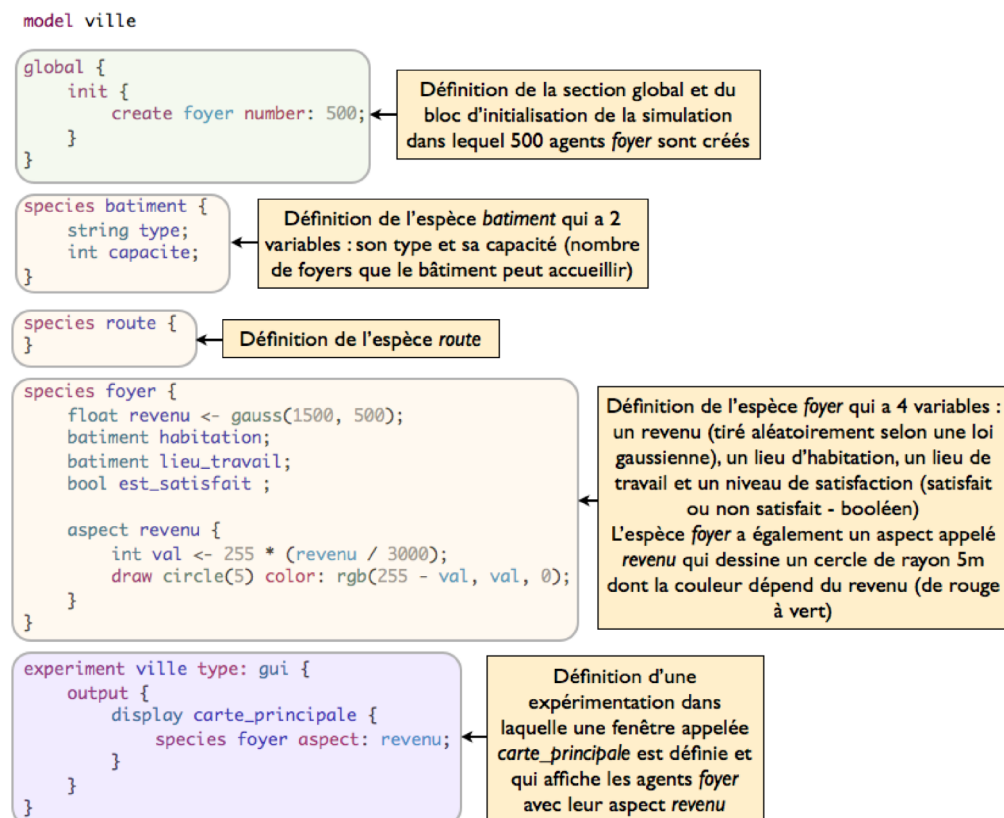
La figure 2 est la traduction de ce diagramme en GAML. Nous avons défini trois espèces : foyer, bâtiment et route.

- L'espèce foyer est caractérisée par 3 variables : le revenu (de type réel), l'habitation (de type bâtiment) et le lieu de travail (de type bâtiment). Nous avons défini en plus un aspect appelé revenu permettant de représenter un agent foyer par un cercle de rayon 5m et dont la couleur est directement dépendante de son revenu : la couleur va varier du rouge (en cas de revenu de 0) au vert (en cas de revenu supérieur ou égal à 3000).
- L'espèce bâtiment est caractérisée par 2 variables en plus de sa géométrie : le type (de type chaîne de caractères et qui est soit « Residential », soit « Industrial ») et la capacité (de type entier). À noter qu'aucune variable géométrie n'a été définie car dans GAMA tout agent est pourvu par défaut d'une géométrie. Nous reviendrons sur cet aspect dans la partie suivante.

- Enfin l'espèce route n'a aucune variable.

Nous avons défini en plus le bloc global qui permet, à l'initialisation d'une simulation, de créer 500 agents foyer (ces derniers seront situés aléatoirement dans l'environnement qui est par défaut dans GAMA un carré de 100m de coté). Enfin, nous avons défini une expérimentation proposant comme sortie une fenêtre de visualisation affichant tous les agents de l'espèce foyer avec leur aspect revenu.

**Figure 2 : Définition de la structure du modèle : espèces bâtiment, route et foyer ; initialisation globale ; expérimentation avec fenêtre de visualisation.**

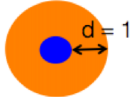

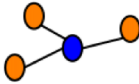


## Gestion de l'espace avec GAMA

Si la plupart des plates-formes permettent de gérer différents types d'environnement (continu, grilles, graphes), elles sont souvent limitées dans leur gestion de multiples environnements ayant des topologies différentes. Il est en effet nécessaire pour certains modèles de pouvoir définir un environnement continu (pour intégrer des données vectorielles, (bâtiments, routes), différents environnement rasters (MNT, grille d'occupation des sols) et différents graphes (graphe social, graphe des routes).

GAMA permet de définir autant d'environnements que nécessaire. L'ensemble des environnements est synchronisé par l'utilisation d'un environnement continu de référence. Chaque environnement est muni d'une topologie qui fournit des services aux agents composant la topologie, en particulier concernant le calcul de voisinage et de distance. Il existe dans GAMA trois types de topologies : l'espace continu, la grille et le graphe (Tab. 1). GAMA permet également de gérer l'aspect toroïdal des environnements.

**Tableau 1 : Les différentes topologies dans GAMA (en orange, le voisinage à une distance de 1 pour chacune de ces topologies).**

Topologie	Continue	Grille	Graphe
Voisinage (distance 1)		 Moore                  Von Neumann	

Une autre spécificité de GAMA est que tout agent est situé dans l’environnement de référence et contient une géométrie – accessible au travers de sa variable *shape*. Basée sur une représentation vectorielle, cette géométrie peut être simple (un point, une polygone, un polygone) ou complexe (composée de plusieurs sous-géométries). La géométrie d’un agent peut être modifiée à tout moment durant la simulation. Elle peut être définie par l’utilisateur (par une liste de points par exemple) ou directement chargée depuis un fichier *shapefile* ou *raster* (voir partie suivante). Par défaut, la géométrie d’un agent est un point. À noter que de manière à faciliter la manipulation des agents dans l’espace, les agents disposent également d’une variable de localisation (appelée « *location* ») de type point représentant le centroïde de la géométrie de l’agent. Modifier la géométrie d’un agent modifie automatiquement la localisation de l’agent. De la même manière, modifier la localisation d’un agent permet d’effectuer automatiquement une translation sur sa géométrie. Afin de faciliter la manipulation des géométries, GAMA permet de définir des variables de type géométrie. GAMA fournit de nombreux opérateurs pour effectuer des opérations sur ces géométries.

Concernant les grilles, chaque cellule de la grille (cellule rectangulaire ou cellule hexagonale) est un agent. L’ensemble de ces agents forme une topologie de type grille. Comme tout agent, un agent cellule a une géométrie (polygone), peut avoir des variables, des comportements (réflexes et actions) et des aspects.

GAMA fournit également de nombreux outils pour définir et travailler sur les graphes. Les sommets et les arrêtes d’un graphe seront soit des agents, soit des géométries. Afin de faciliter la manipulation des graphes, GAMA permet de définir des variables de type graphe. Nous reviendrons sur la création et les opérateurs de graphe dans la troisième partie.

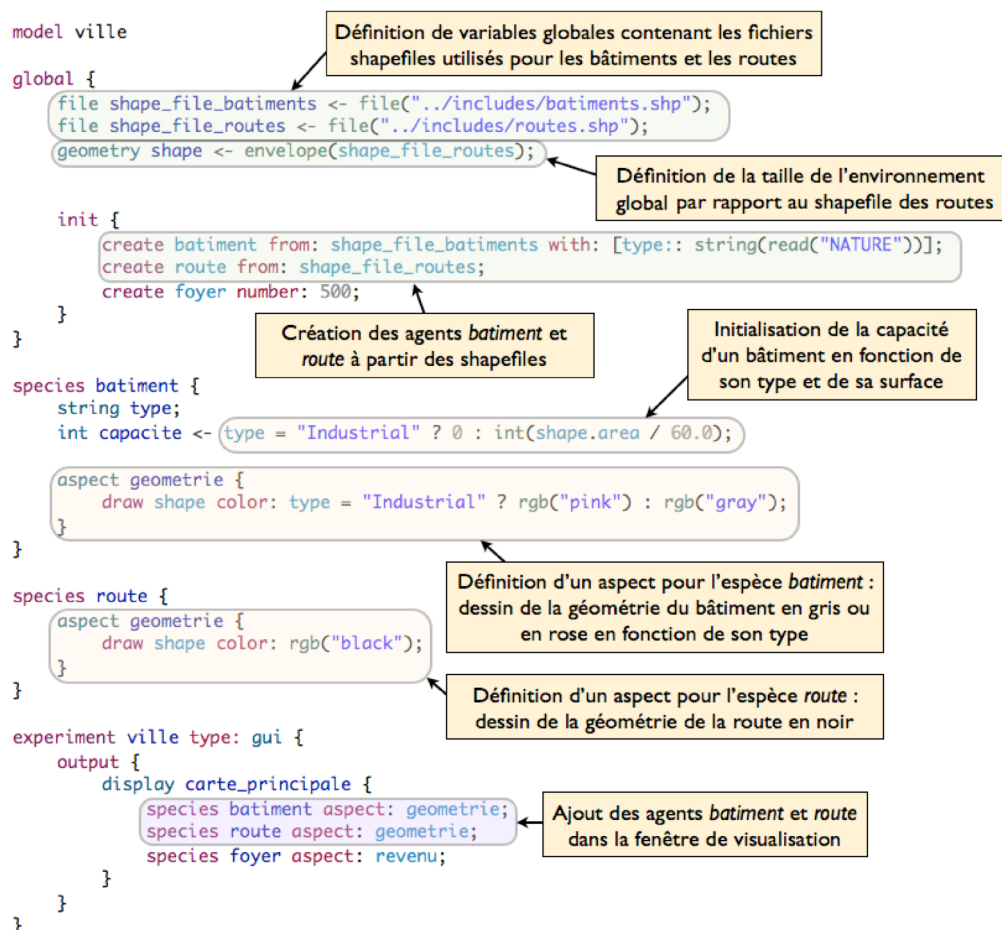
Aujourd’hui, de plus en plus de modèles se basent sur des données géographiques issues d’un SIG pour améliorer leur réalisme. Si la plupart des plates-formes de simulation permettent de lire et d’écrire des données géographiques, la liberté offerte par ces dernières pour manipuler ces données est plus inégale. GAMA offre de nombreux outils pour intégrer des données géographiques. En particulier, GAMA permet, au travers d’une simple instruction GAML, d’utiliser un fichier *shapefile* pour créer des agents d’une espèce donnée : chaque objet contenu dans les données géographiques sera utilisé pour instancier un agent (la géométrie de l’agent sera celle lue dans le *shapefile*), GAMA se chargeant de gérer la projection spatiale des données et si nécessaire de lire les valeurs des attributs dans le *shapefile*.

De la même manière, GAMA permet d’utiliser un fichier *raster* (*asc* ou *image*) pour créer une grille aux dimensions voulues. La valeur contenue dans le fichier *raster* sera automatiquement utilisée pour initialiser la valeur de la variable *grid\_value* de chaque agent cellule. GAMA se charge également de construire la géométrie (ici un rectangle) correspondante au fichier pour chaque agent cellule.

GAMA permet également, à l'aide d'une simple instruction GAML, de sauvegarder les géométries d'une liste d'agents sous la forme d'un shapefile, avec la possibilité de sauvegarder en plus dans le shapefile la valeur des variables souhaitées des agents.

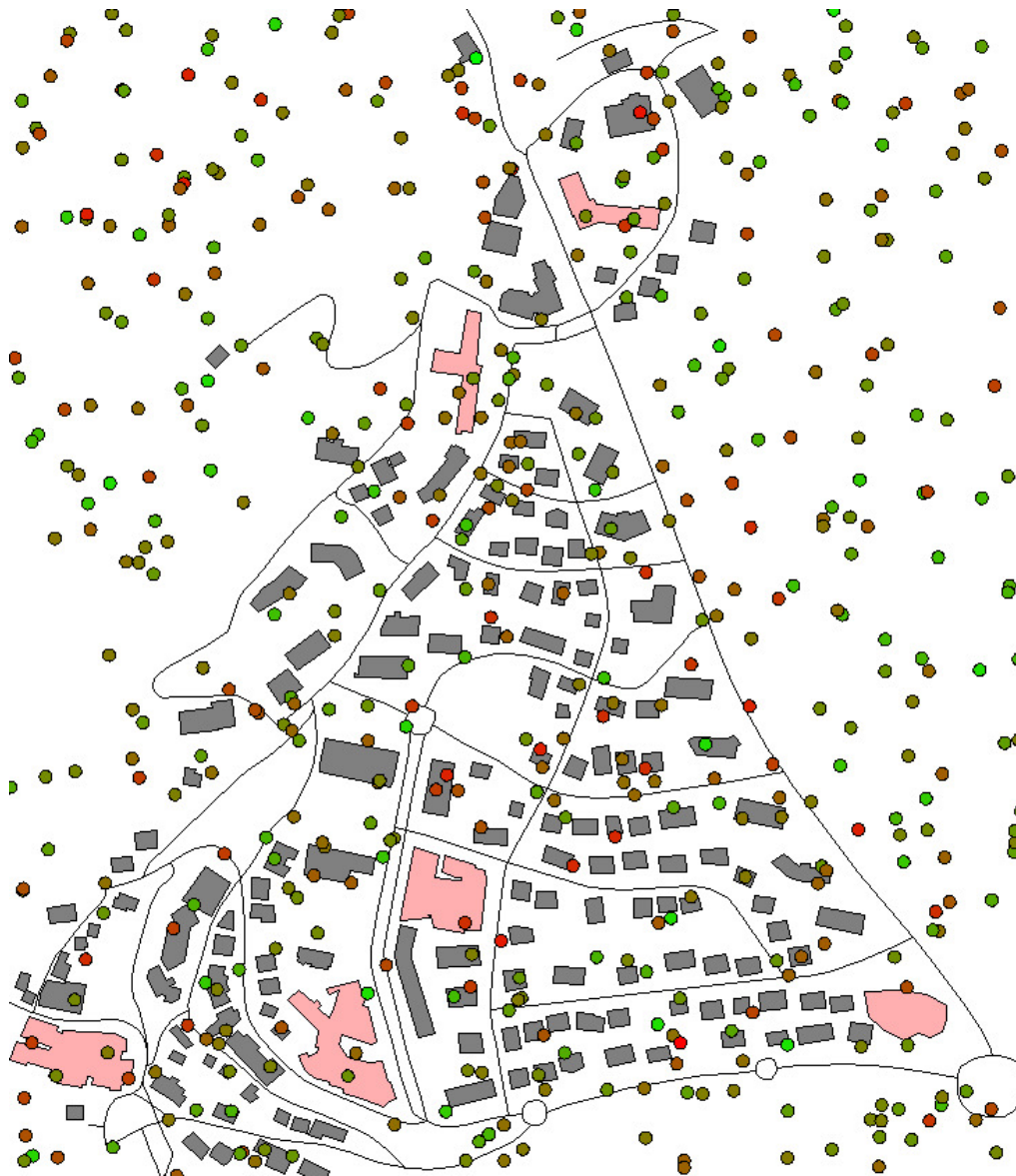
La figure 3 présente un exemple de lecture de shapefile pour notre modèle de ségrégation. Dans cet exemple qui repart du modèle présenté en figure 1, deux fichiers shapefiles sont définis : l'un pour les routes et l'autre pour les bâtiments. A partir de ces fichiers, des agents route et bâtiment sont créés. Chaque objet des shapefiles devient un agent : un agent route pour le shapefile des routes, et un agent bâtiment pour le shapefile des bâtiments. Pour les agents bâtiment, nous utilisons de plus l'attribut « NATURE » contenu dans le shapefile pour initialiser l'attribut type des agents. Nous avons défini pour chacune de ces deux espèces un aspect qui permet de visualiser la géométrie des agents (cette géométrie est accessible au travers de leur variable shape). Enfin nous avons ajouté ces deux espèces d'agents sur la fenêtre de visualisation appelée « carte\_principale ». A noter que la définition d'une fenêtre d'affichage est similaire à celle des SIG : l'affichage se déroule par couche auxquelles il est possible d'ajouter de la transparence. Dans notre exemple, la couche foyer sera dessinée par-dessus la couche route qui sera elle-même dessinée par-dessus la couche bâtiment.

**Figure 3 : Chargement des données géographiques (shapefiles) pour les bâtiments et les routes.**



La figure 4 donne un exemple de résultat de simulation obtenu (pas de simulation 0). Nous pouvons observer que des agents route et bâtiment ont bien été créés. Parmi les agents bâtiment, 6 sont de type industriel (en rose). Les cercles correspondent aux agents foyer. Leur couleur représente leur niveau de revenu qui va du rouge pour les plus pauvres au vert pour les plus riches. A noter que les agents foyer sont situés aléatoirement dans le monde car nous n'avons rien spécifié pour leur localisation initiale (comme de les placer dans des bâtiments par exemple).

Figure 4 : Exemple de résultat de simulation obtenu avec le modèle présenté en figure 3.



## Opérateurs spatiaux

GAMA fournit de nombreux opérateurs spatiaux directement accessibles en GAML : requêtes spatiales (agents dont la géométrie se trouve à une certaine distance d'une géométrie, superposition), tests géométriques (intersection entre 2 géométries, croisement, inclusion), transformation (buffer, simplification), opérations (union, différence, intersections). GAMA fournit également des opérateurs de plus haut niveau tels que la triangulation de polygones, la squelettisation ou la création de graphes à partir de géométries. La plate-forme utilise la librairie JTS (JTS, 2013) pour réaliser ces opérations.

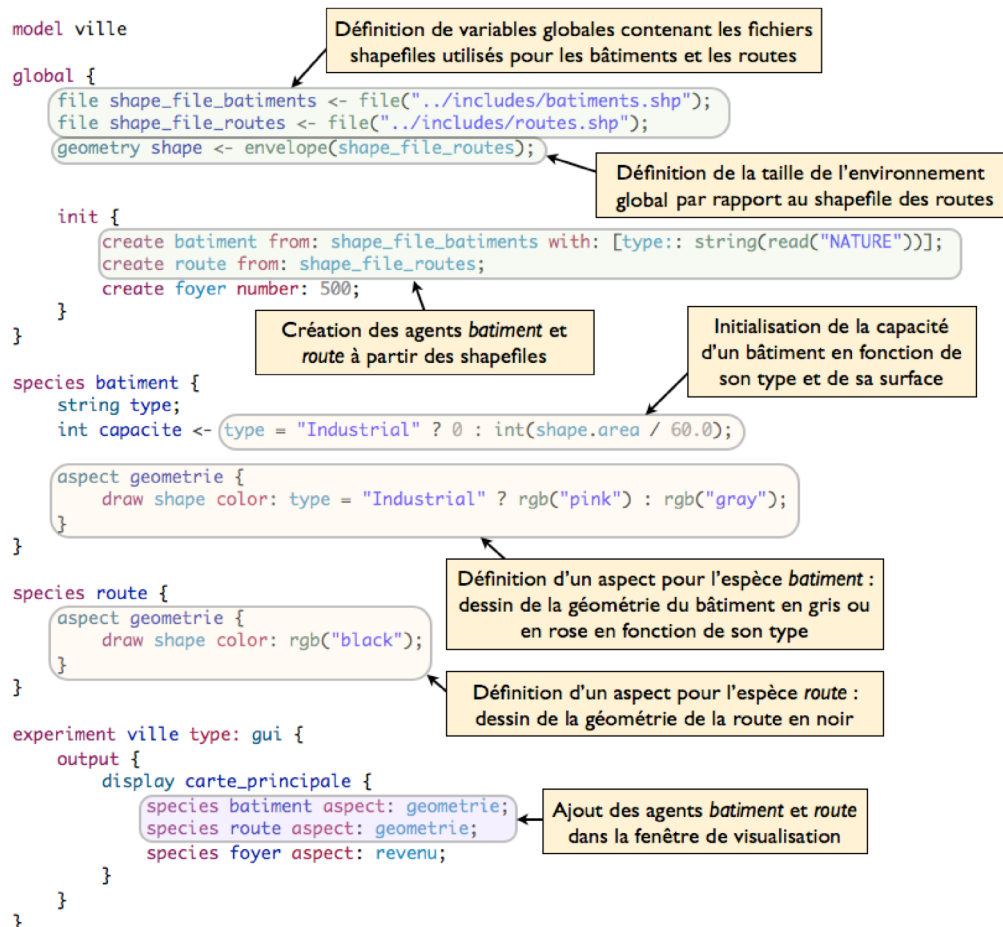
Des outils permettant de gérer et d'interroger des bases de données sont aussi intégrés dans la plate-forme (Truong et al. 2013). Il est ainsi possible d'effectuer au travers du langage GAML des requêtes et des opérations sur des bases de données pouvant être géographiques (par exemple PostGIS).

GAMA fournit également pour les agents de nombreuses primitives de déplacement permettant à un agent de se déplacer de façon transparente dans un environnement continu (borné ou non par une géométrie), dans une grille ou dans un graphe.

La figure 5 présente un exemple d'utilisation d'opérateurs spatiaux pour notre modèle de ségrégation. Dans cet exemple qui se base sur le modèle présenté en figure 3, nous avons défini une nouvelle variable de type booléen appelée « est\_satisfait » pour les agents foyer. Grâce à la facette « *update* » nous pouvons spécifier que cette variable est recalculée à chaque

pas de simulation en faisant appel à l'action `calculer_satisfaction` définie ci-après. Nous avons également défini un bloc d'initialisation pour cette espèce d'agents (ensemble d'instructions qui seront appliquées lors de la création d'un agent foyer). Ce bloc d'initialisation permet de choisir aléatoirement un lieu de travail parmi les bâtiments industriels et d'emménager dans une nouvelle habitation (en utilisant les deux actions « `choisir_habitation` » et « `emménager` »). L'action « `calculer_satisfaction` » va permettre à un foyer de calculer sa satisfaction en comparant son revenu au revenu moyen de tous les foyers se trouvant à une distance de 50m. L'utilisation de l'opérateur spatial `at_distance` permet d'obtenir la liste de tous les agents de l'espèce souhaitée dont la géométrie se trouve à une distance inférieure ou égale à la distance donnée de la géométrie de l'agent appliquant l'opérateur. Si cet opérateur avait été utilisé dans le cadre d'une topologie de type grille, l'opérateur aurait alors renvoyé tous les agents foyer se trouvant à distance de 50 cellules (distances de Moore ou de Von Neumann) et non à 50m. L'action « `emménager` » permet à l'agent foyer d'emménager dans un nouveau bâtiment, c'est-à-dire de décrémenter la capacité du bâtiment et de se déplacer dans le bâtiment. A noter l'utilisation de l'opérateur `any_location_in` qui permet de renvoyer un point choisi aléatoirement dans une géométrie. Modifier la localisation d'un agent (location) permet de la déplacer (translation de la géométrie de l'agent – ici, géométrie par défaut, un point – sur le point souhaité). L'action « `demenager` » ne fait qu'incrémenter la capacité du bâtiment que l'agent foyer quitte. L'action « `choisir_batiment` » permet de choisir aléatoirement l'un des bâtiments dont la capacité est supérieure à 0. Enfin, nous avons défini un réflexe qui est activé uniquement quand l'agent n'est pas satisfait et qui lui permet de choisir un nouveau bâtiment d'habitation, de déménager de l'ancien bâtiment et d'emménager dans le nouveau.

**Figure 5 : Définition du comportement des agents foyer.**



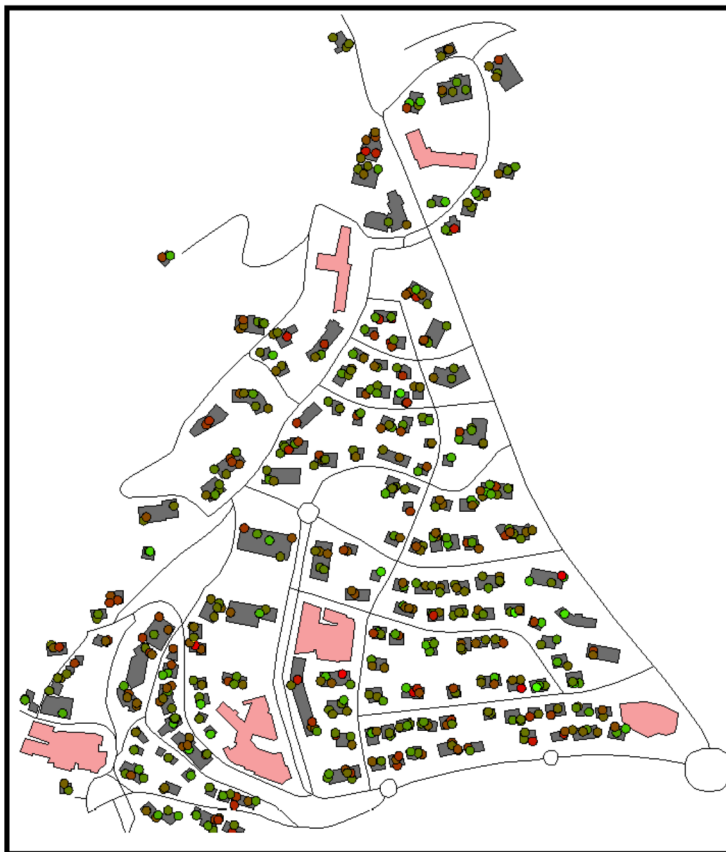
La figure 6 donne un exemple de résultat de simulation obtenu. Débutant sur une distribution aléatoire des foyers (pas de simulation 0), nous pouvons observer l'émergence d'un phénomène de ségrégation sur le revenu dès le pas de simulation 500 qui s'accroît au pas de



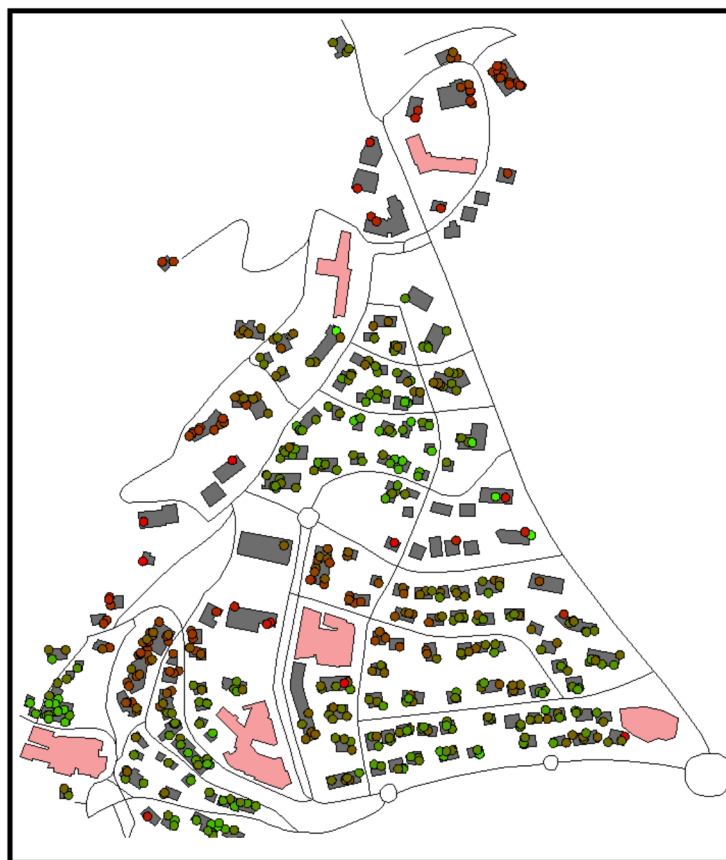
simulation 1000 dans lequel le système atteint un état d'équilibre (tous les agents foyer sont satisfaits) : certains quartiers comme ceux du Nord/Nord-Ouest sont surtout habités par des foyers pauvres, alors que d'autres, par exemple au centre, sont surtout habités par des foyers au revenu élevé.

Figure 6 : Exemple de résultat de simulation obtenu avec le modèle présenté en figure 5 (pas de simulation 0, 500 et 1000).

a) Pas de Simulation 0



b) Pas de Simulation 500

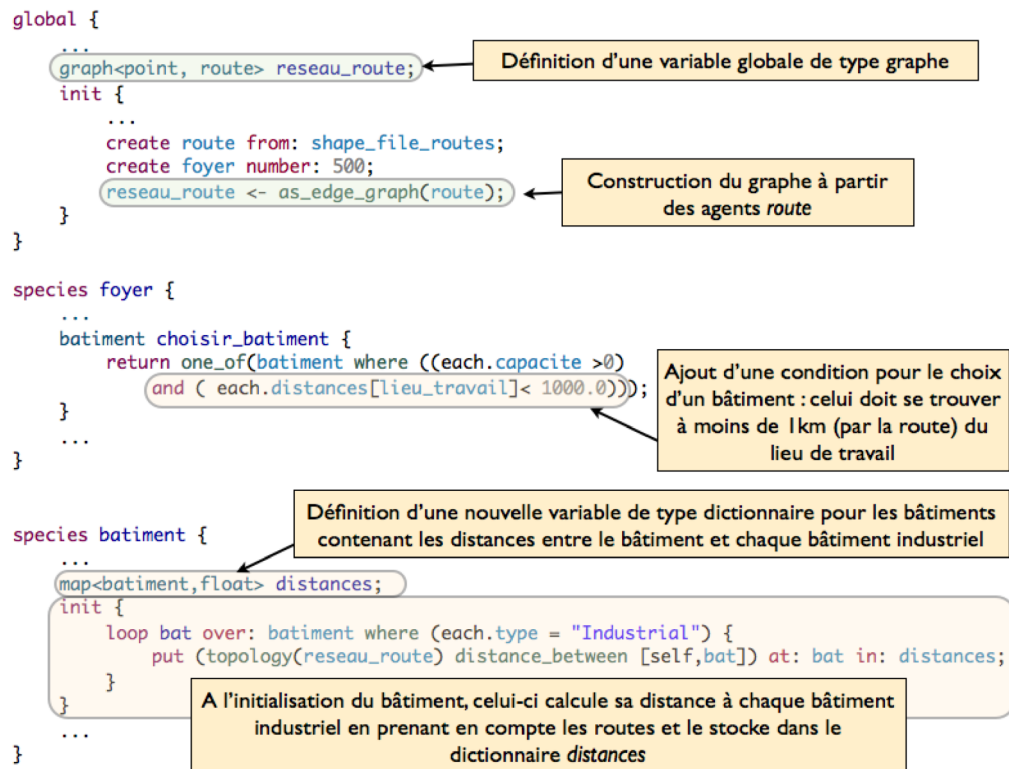


Comme mentionné en partie 3, GAMA permet de créer et de manipuler des graphes. En particulier, GAMA fournit différents opérateurs pour construire un graphe spatial à partir de géométries :

- *as\_edge\_graph* : construction d'un graphe à partir d'un ensemble de polygones (ou d'agents ayant pour géométrie des polygones). Dans ce cas, un arc sera créé par polygone et les nœuds représenteront les extrémités des polygones. Cet opérateur permet par exemple de créer un graphe à partir d'un réseau de routes.
- *as\_distance\_graph* : construction d'un graphe à partir d'un ensemble de géométries (ou d'agents). Dans ce cas, les nœuds représenteront les géométries (ou agents) et un arc sera créé si deux se trouvent à une distance inférieure à un seuil donné.
- *as\_intersection\_graph* : construction d'un graphe à partir d'un ensemble de géométries (ou d'agents). Dans ce cas, les nœuds représenteront les géométries (ou agents) et un arc sera créé si deux géométries s'intersectent.

Dans le cas de notre modèle de ségrégation, nous souhaitons pouvoir calculer la distance entre deux bâtiments en prenant en compte le réseau routier de façon à ne permettre aux foyers d'emménager que dans les bâtiments se trouvant à une distance inférieure à un certain seuil de leur lieu de travail. Nous allons donc pour cela utiliser l'opérateur *as\_edge\_graph* pour construire un graphe à partir du réseau de routes. Nous pourrions ensuite utiliser l'opérateur *distance\_between* à partir de ce graphe pour connaître directement la distance entre 2 bâtiments. A noter que si nous n'avions pas utilisé notre graphe pour le calcul de la distance, l'opérateur *distance\_between* aurait alors utilisé la topologie par défaut (topologie continue) et aurait renvoyé la distance euclidienne entre les 2 bâtiments. La figure 7 présente les modifications apportées au modèle présenté dans la figure 5 pour prendre en compte la distance par les routes. Nous avons choisi pour des raisons d'optimisation de calculer pendant la création des bâtiments la distance entre ces derniers et tous les bâtiments industriels (avec utilisation d'une boucle *loop*) et de stocker les résultats dans un dictionnaire (*map*, clef : bâtiment industriel, valeur : distance entre le bâtiment courant et le bâtiment industriel). Ainsi, quand un agent foyer cherche à déterminer quels sont les bâtiments libres (avec une capacité supérieure à 0) se trouvant à moins de 1000m de son lieu de travail, il peut simplement utiliser ce dictionnaire et n'a pas à recalculer à chaque fois toutes les distances.

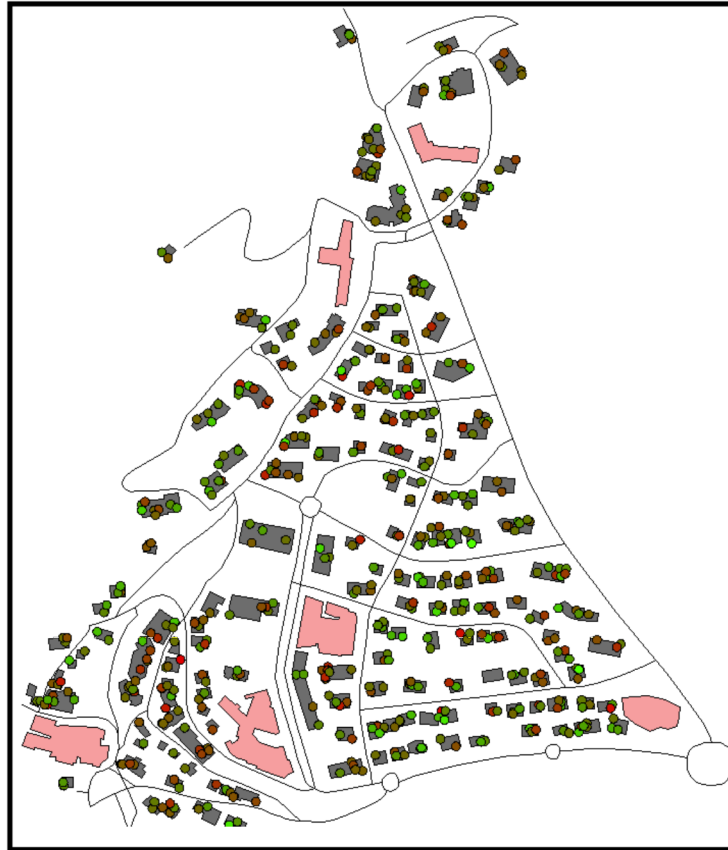
Figure 7 : Ajout de la notion d'éloignement par les routes entre deux bâtiments.



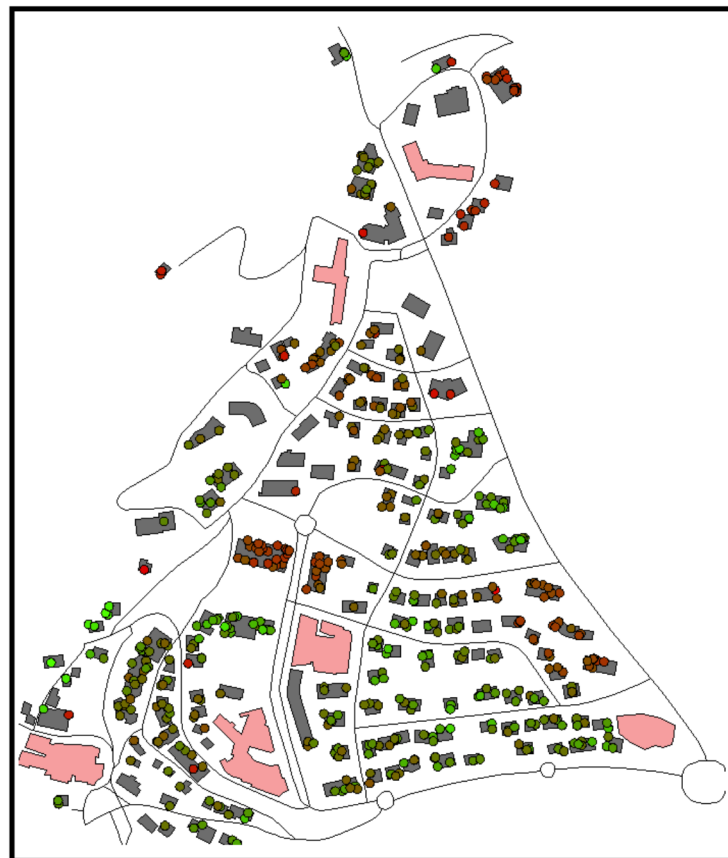
La figure 8 donne un exemple de résultat de simulation (pas de simulation 0, 500 et 1200). Nous pouvons observer le même phénomène de ségrégation sur le revenu que sur la figure 6 ; apparition dès le pas de simulation 500 et accentuation jusqu'au pas de simulation 1200 (état d'équilibre). Mais néanmoins, nous pouvons observer la formation de groupes plus petits et plus dispersés. Par exemple, nous pouvons observer que les foyers les plus pauvres étaient concentrés uniquement dans quelques zones sur la figure 6 (en particulier au Nord), ils sont ici présents dans tous les quartiers de la ville. En effet, la contrainte sur l'éloignement par rapport à leur lieu de travail et le fait que les lieux de travail soient bien distribués dans la ville a entraîné cette répartition plus homogène des foyers.

Figure 8 : Exemple de résultat de simulation obtenu avec le modèle présenté en figure 7 (pas de simulation 0, 500 et 1200).

a) Pas de Simulation 0



b) Pas de Simulation 500



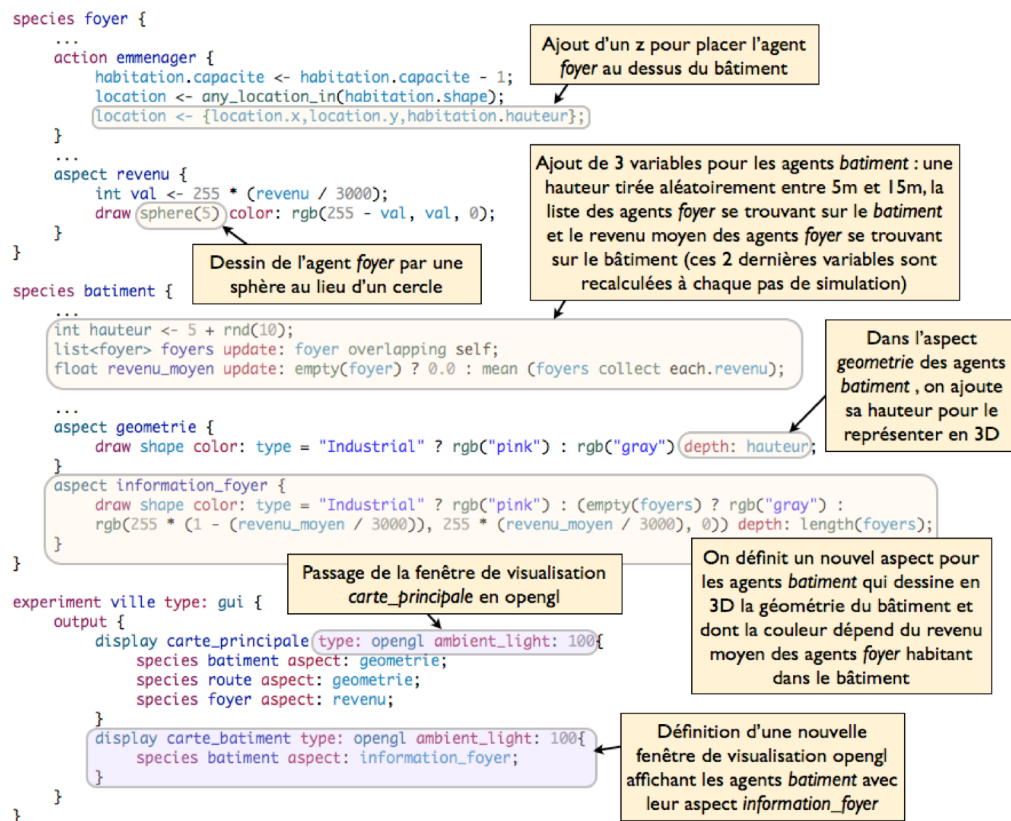
## Gestion de la 3D

GAMA intègre de façon native la 3D puisqu'un point dans GAMA est composé de trois coordonnées x, y et z. La 3D permet de fournir au modélisateur de nombreuses améliorations en termes de visualisation et d'interaction : rendu réaliste et immersif, superposition d'informations supplémentaires en cours de simulation, etc. Pour des raisons de performance, le choix a été fait de ne pas intégrer cette troisième dimension dans certains opérateurs spatiaux (union, intersection).

GAMA utilise OpenGL pour l'affichage de la 3D, permettant ainsi de bénéficier de l'accélération graphique offerte par les cartes graphiques (GPU) et des dernières avancées en termes de rendu réaliste. La 3D dans GAMA n'a pas seulement pour but de rendre l'environnement plus réaliste, mais elle permet d'étudier le modèle de façon plus intuitive et immersive grâce à l'utilisation d'outils interactifs et à une séparation claire de la représentation du modèle et de son exécution. Le modélisateur peut se déplacer facilement dans le modèle, définir différents points de vue ou même suivre un agent en cours de simulation. En termes d'abstraction, la 3D permet aussi de facilement superposer des indicateurs sur la simulation grâce à l'utilisation de calques. Enfin il est possible de définir des agents intégralement dédiés à la visualisation et à l'interaction pour des tâches plus spécifiques (Grignard et al., 2013). Comme mentionné dans la troisième partie, GAMA permet d'intégrer de façon transparente des données géographiques. Ces données peuvent être en 2D ou en 3D. Dans le cadre de données 2D, GAMA permet d'ajouter de la 3D de façon extrêmement simple en ajoutant une hauteur (*depth*) à une géométrie.

La figure 9 propose un exemple d'ajout de la 3D dans un modèle. Dans cet exemple, nous avons modifié le modèle de la figure 7 de façon à ajouter une hauteur aux bâtiments et de remplacer les cercles représentant les foyers par des sphères. Nous avons également ajouté une deuxième fenêtre de visualisation permettant d'afficher uniquement les bâtiments : la couleur de ceux-ci sera directement en lien avec les revenus moyens des foyers habitant le bâtiment et leur hauteur sera proportionnelle au nombre de foyers habitant sur celui-ci.

**Figure 9 : Ajout de la visualisation 3D.**



La figure 10 donne un exemple de résultat de simulation obtenu (pas de simulation 1000). Deux fenêtres de simulation sont disponibles. La première reprend notre fenêtre de visualisation

préalablement définie en ajoutant juste la 3D. La seconde permet d'avoir une vision différente du même espace facilitant certaines analyses (en particulier sur l'émergence du phénomène de ségrégation). Dans cette seconde fenêtre, deux informations sont représentées pour chaque bâtiment : sa hauteur qui dépend du nombre d'habitants et sa couleur qui dépend de la richesse des habitants.

**Figure 10 : Exemple de résultat de simulation obtenu avec le modèle présenté figure 9 (pas de simulation 1000)**



(a) Dans la fenêtre de visualisation du haut, les sphères correspondent aux agents foyer : leur couleur représente leur revenu - du rouge pour les plus pauvres au vert pour les plus riches. Les bâtiments roses correspondent aux bâtiments industriels. (b) Dans la fenêtre de visualisation du bas, la couleur des bâtiments résidentiels représente le revenu moyen des agents foyer habitant le bâtiment : du rouge pour les bâtiments avec les foyers les plus pauvres au vert pour les bâtiments avec les foyers les plus riches.

Aujourd'hui, de plus en plus de modèles tendent à représenter et à utiliser la topographie de la zone étudiée (en altimétrie ou en bathymétrie). En termes de modèles numériques de terrain (MNT), GAMA permet non seulement de représenter un MNT dans un environnement 3D mais aussi d'exploiter les données d'altitude et d'intégrer ces données dans le comportement d'un agent. La figure 11 reprend le modèle présenté en figure 9 en ajoutant pour chaque bâtiment et chaque route une altitude correspondant à celle définie dans le MNT. La figure 12 montre le rendu visuel obtenu par l'ajout de ce MNT.

Figure 11 : Ajout d'un modèle numérique de terrain.

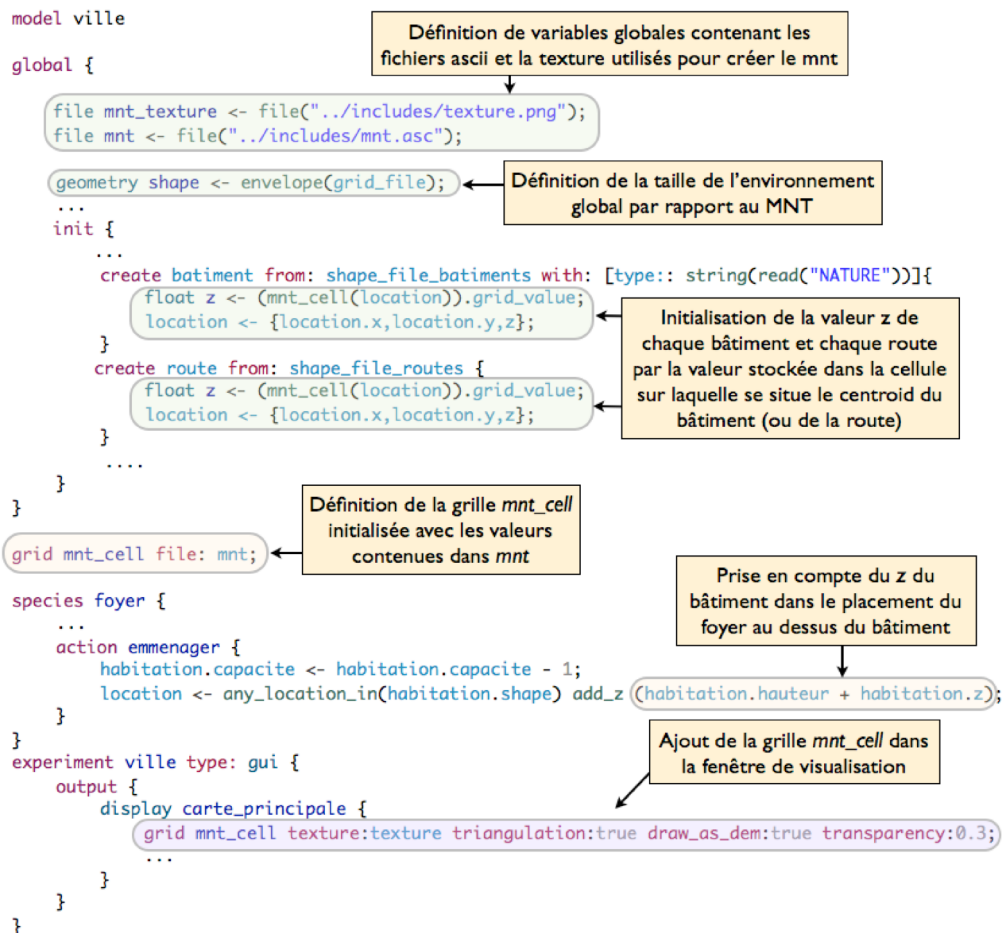
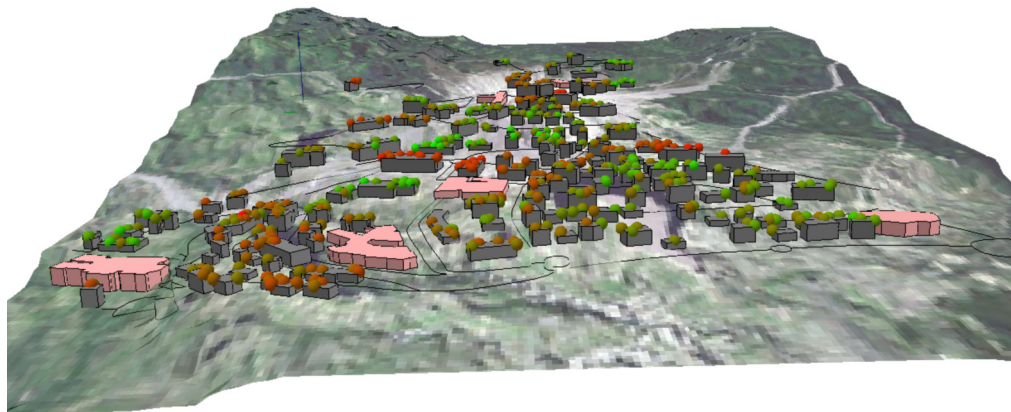


Figure 12 : Exemple de résultat de simulation obtenu avec le modèle présenté en figure 11.



## Conclusion

Depuis le début de son développement en 2007, la plate-forme GAMA a été grandement enrichie. La version 1.6 offre en particulier de nombreux outils puissants pour gérer l'espace. En effet, comme nous l'avons montré au travers de notre modèle fil rouge sur la ségrégation urbaine, GAMA permet de simplement intégrer des données géographiques (shapefile des bâtiments et des routes dans notre exemple). Il permet également de recourir à des outils d'analyse spatiale (comme le calcul du voisinage de nos agents foyer) ou de graphe (le calcul de la distance entre deux bâtiments par les routes dans notre modèle). Enfin, GAMA offre des outils puissants de visualisation 3D nous ayant permis, pour notre modèle, de visualiser les bâtiments et un MNT en 3D.



Ainsi cette nouvelle version permet une gestion intuitive du multi-niveau (Vo et al., 2012). En effet, tout agent peut maintenant être composé d'un ensemble de sous-agents et influencer la nature de ces derniers en leur faisant changer d'espèce (et donc de comportement). Un agent macro a également en charge l'activation de ses sous-agents et peut donc décider quand les faire agir. Dans ce cadre, GAMA fournit de nombreux opérateurs permettant de définir quand et selon quelle condition faire émerger des agents macro, fusionner deux agents macro, décomposer des agents macro, etc.

GAMA offre également la possibilité de définir l'évolution de certaines variables d'un agent au moyen d'équations différentielles ordinaires. Plusieurs méthodes d'intégration sont proposées au modélisateur (Runge Kutta 4, Dormand-Prince). De plus, des primitives GAML permettant d'intégrer directement des systèmes d'équations classiques prédéfinis (modèles épidémiologiques classiques tels que SI, SIR ou SEIR et le modèle proie-prédateur de Lotka-Volterra).

D'autres outils tels que la gestion de forces 3D (moteur physique) ou la possibilité de faire exécuter des scripts par le logiciel R directement dans un modèle sont également offerts par GAMA.

Nous insistons sur le fait que cette version du logiciel comprend également de nombreux autres outils qui n'ont pas été présentés dans cet article. Et de nombreuses autres améliorations et enrichissements sont d'ores et déjà prévus pour les versions futures : par exemple l'enrichissement de la possibilité de définition de plans d'expérience, la possibilité de définir graphiquement des modèles (Taillandier, 2013), la distribution d'instructions sur des GPU, ou encore la possibilité de rejouer des simulations au travers d'une interface web.

---

## **Bibliographie**

Banos A., Maps Team., 2013, « MIRO 2 – Compte-rendu de fin de projet (2013) », *UMR Géographie-Cités*.

Banos A., Marilleau N., 2012, « Improving individual accessibility to the city: an agent-based modelling approach », *Proceedings of ECCS*, Bruxelles.

Banos A., Thevenin T., Chardonnel S., Lang C., Marilleau N., 2005 « Simulating the Swarming City: a MAS Approach » *Computers in Urban Planning and Urban Management (CUPUM'05)*, London.

Barros J., Alves J., 2003, « Simulating rapid urbanisation in Latin American cities », *Advanced Spatial Analysis: the CASA Book of GIS*, Editors P. Longley and M. Batty, ESRI Press.

Batty M., 2005, *Cities and complexity: understanding cities with cellular automata, agent-based models, and fractals*, Cambridge, London, MIT Press, Massachusetts.

Daudé E., 2004, « Apports de la simulation multi-agents à l'étude des processus de diffusion ». *Cybergeo: European Journal of Geography*, article 255. Url: <http://cybergeo.revues.org/3835?lang=en&file=1>

Crooks A.T., Castle C.J.E., 2012, *Agent-Based Models of Geographical Systems*, Springer Netherlands, Dordrecht.

Eubank S., Guclu H., Kumar V., Marathe M., Srinivasan A., Toroczkai Z., Wanz N., 2004 « Modeling Disease Outbreaks in Realistic Urban Social Networks », *Nature*, No 429, 180-184.

GAMA site internet, 2013, <https://code.google.com/p/gama-platform/> (dernier accès : octobre 2013).

GAMA documentation, 2013, <https://code.google.com/p/gama-platform/wiki/GAMAIntro16> (dernier accès : octobre 2013).

GAMA installation, 2013, <https://code.google.com/p/gama-platform/wiki/Installation> (dernier accès : octobre 2013).

GAMA téléchargement, 2013, <https://code.google.com/p/gama-platform/wiki/DownloadGAMA?tm=2> (dernier accès : octobre 2013).

GAMA tutoriels, 2013, <https://code.google.com/p/gama-platform/wiki/Tutorials16> (dernier accès : octobre 2013).

GAMA IJCAI démonstration, 2013, <http://ijcai13.org/video/10> (dernier accès : octobre 2013).

Gaudou B., Sibertin-Blanc C., Therond O., Amblard F., Arcangeli J.P., Balestrat M., Charron-Moirez M.H., Gondet E., Hong Y., Louail T., Mayor E., Panzoli D., Sauvage S., Sanchez-Perez J.,

- Taillandier P., Nguyen V.B., Vavasseur M., Mazzega P., 2013, « The MAELIA multi-agent platform for integrated assessment of low-water management issues », *International Workshop on Multi-Agent-Based Simulation (MABS)*, Saint-Paul, USA, 6-7 May 2013.
- Grignard A., Drogoul A., Zucker J.D., 2013, « Online analysis and visualization of agent based models », *Computational Science and Its Applications–ICCSA*, Ho Chi Minh City, Vietnam, 24-27 June 2013, 662–672.
- JTS, 2013, <http://sourceforge.net/projects/jts-topo-suite/> (dernier accès : octobre 2013).
- Langlois P., Blanpain B., Daudé E., 2013, « MAGéo, une plateforme de simulation multi-agents pour tous », *Simtools Workshop*, Paris, France, 15 Mars 2013.
- Minar N., Burkhart R., Langton C., Askenazi M., 1996, « The Swarm Simulation System: A Toolkit for Building Multi-Agent Simulations », *Technical Report*, Santa Fe Institute, Santa Fe.
- Truong M.T., Amblard F., Gaudou B., 2013, « Combination Framework of BI solution & Multi-agent platform (CFBM) for multi-agent based simulations », *Atelier aide à la Décision à tous les Etages (AIDE 2013)*, Toulouse, France, 29 janvier 2013.
- North M.J., Collier N.T., Ozik J., Tataru E.R., Macal C.M., Bragen M., Sydelko P., 2013, « Complex adaptive systems modeling with repast symphony », *Complex Adaptive Systems Modeling*, vol. 1, No.1, 1–26.
- Railsback S.F., Lytinen S.L., Jackson S.K., 2006, « Agent-based Simulation Platforms: Review and Development Recommendations », *Simulation*, vol. 82, No.9, 609–623.
- Resnick M., 1996, « StarLogo: an environment for decentralized modeling and decentralized thinking », *Conference companion on Human factors in computing systems: common ground*, Vancouver, Canada, 13-18 April 1996.
- Repenning A., 2011, « Making programming more conversational », *Visual Languages and Human-Centric Computing (VL/HCC)*, Pittsburgh, USA, 18-22 September 2011, 191-194.
- Sanders L., Pumain D., Mathian H., Guerin-Pace F., Bura S., 1997, « SIMPOP: a multiagent system for the study of urbanism », *Environment and Planning B: Planning and Design*, vol. 24, 287-305.
- Schelling T.C., 1971, « Dynamic models of segregation », *Journal of Mathematical Sociology*, vol. 1, No.1, 143–186.
- Taillandier P., 2013, « GAMAGraM : Modélisation Graphique sous GAMA », *MASYCO workshop*, Lille, France, 2 juillet 2013.
- Taillandier P., Therond O., Gaudou B., 2012, « A new BDI agent architecture based on the belief theory. Application to the modelling of cropping plan decision-making », *International Environmental Modelling and Software Society (iEMSs)*, Leipzig, Germany, 1-5 July 2012.
- Tisue S., Wilensky U., 2004, « Netlogo: A simple environment for modeling complexity », *International Conference on Complex Systems*, Boston, USA, 16-21 May 2004, 16–21.
- Vo D.A., Drogoul A., Zucker J.D., 2012, « An Operational Meta-Model for Handling Multiple Scales in Agent-Based Simulations », *Conference on Research, Innovation and Vision for the Future in Computing & Communication Technologies (RIVF)*, Ho Chi Minh City, Vietnam, 27 February-27 March 2012, 1-6.
- White R.W., Engelen G., 1993, « Cellular Automata and fractal urban form: a cellular modelling approach to the evolution of urban land use patterns », *Environment and planning A*, vol. 25, No.8, 1175-1193.

---

### ***Pour citer cet article***

#### Référence électronique

Patrick Taillandier, Arnaud Grignard, Benoit Gaudou et Alexis Drogoul, « Des données géographiques à la simulation à base d'agents : application de la plate-forme GAMA », *Cybergeo : European Journal of Geography* [En ligne], Systèmes, Modélisation, Géostatistiques, document 671, mis en ligne le 29 mars 2014, consulté le 11 février 2015. URL : <http://cybergeo.revues.org/26263> ; DOI : 10.4000/cybergeo.26263

---

### ***À propos des auteurs***

#### **Patrick Taillandier**

Université de Rouen, UMR 6266 CNRS IDEES, Maître de conférences en géographie

patrick.taillandier@univ-rouen.fr

**Arnaud Grignard**

Université Pierre et Marie Curie, UMI 209 UMMISCO IRD/UPMC, Doctorant  
agrignard@gmail.com

**Benoit Gaudou**

Université Toulouse 1 Capitole, UMR 5505 CNRS IRIT, Maître de conférences en informatique  
benoit.gaudou@ut-capitole.fr

**Alexis Drogoul**

Institut de Recherches pour le Développement (IRD), UMI 209 UMMISCO IRD/UPMC, Directeur de recherches  
alexis.drogoul@ird.fr

## ***Résumés***

Ces dernières années, la simulation à base d'agents a révélé certains intérêts en géographie. Néanmoins, s'il existe aujourd'hui de nombreuses plates-formes permettant de faciliter le développement de modèle, celles-ci sont souvent trop complexes pour être utilisées par des non-informaticiens (par exemple Repast Symphony) ou trop limitées pour développer des modèles riches intégrant de nombreux agents et données (par exemple Netlogo). La plate-forme GAMA a pour objectif d'offrir aux modélisateurs une plate-forme à la fois puissante et simple d'accès grâce à langage de modélisation dédié. Dans cet article, nous présentons les capacités particulièrement avancées de la nouvelle version de la plate-forme GAMA pour la gestion de l'espace. Cette plate-forme permet en particulier de gérer simultanément plusieurs environnements basés sur trois types de topologie (continue, grille, graphe), d'intégrer très simplement des données géographiques (vectorielles et rasters) et de représenter des géométries dans un environnement 3D. Les capacités détaillées dans cet article sont illustrées par un modèle sur les problématiques de ségrégation urbaine.

## **From Geographical data to agent-based simulation: application of the GAMA platform**

These last years, agent-based simulation has shown its interest for geography. However, if nowadays there are many platforms allowing the development of models, these ones are often too complex to be used by non-computer scientist (e.g. Repast Symphony) or too limited to develop rich models integrating numerous agents and data (e.g. Netlogo). The GAMA platform aims at providing modelers a complete platform that is at the same time very simple to use (thanks to a dedicated modeling language) and powerful. In this article, we present the advanced capabilities of the last version of GAMA concerning the management of space. Indeed, GAMA allows to simultaneously manage several environments based on three types of topology (continuous, grid, graph), to easily integrate geographic data (vector and raster) and to represent geometries in a 3D environment. All the capabilities detailed in this article are illustrated by a model on urban segregation.

## ***Entrées d'index***

**Mots-clés** : Simulation à base d'agents, Plate-forme de modélisation, Données géographiques, Environnement 3D

**Keywords** : Agent-based simulation, Modeling platform, Geographical data, 3D environment