



HAL
open science

GAMAGraM : Modélisation Graphique sous GAMA

Patrick Taillandier

► **To cite this version:**

Patrick Taillandier. GAMAGraM : Modélisation Graphique sous GAMA. Masyco 2013, 2013, France. hal-00850277

HAL Id: hal-00850277

<https://hal.science/hal-00850277>

Submitted on 6 Aug 2013

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

GAMAGraM : Modélisation Graphique sous GAMA

P. Taillandier^a
patrick.taillandier@gmail.com

^aUMR IDEES - MTG,
Université de Rouen, France

Résumé

Ces dernières années ont vu la multiplication des plates-formes dédiées à la conception et la simulation de modèles à base d'agents. Si ces plates-formes ont permis de démocratiser cette approche de modélisation, elles restent souvent complexes à utiliser par des non-informaticiens car elles requièrent pour la plupart de définir des modèles en écrivant des lignes de codes. Dans cet article, nous présentons GAMAGraM, un nouveau plug-in de modélisation graphique s'intégrant à la plate-forme GAMA. Ce plug-in permet aux utilisateurs de définir un modèle à l'aide d'une interface graphique. Il offre de plus la possibilité de générer automatiquement un modèle graphique à partir d'un modèle GAMA.

Mots-clés : *Modélisation Graphique, Plate-forme GAMA, Génération de modèles graphiques*

Abstract

These last years have seen the multiplication of platforms dedicated to the conception and simulation of agent-based models. If these platforms allowed to democratize this modeling approach, they are often complex to use by non-computer scientists as most of them require to define models by writing code lines. In this paper, we present GAMAGraM, a new graphical modeling plug-in integrated in the GAMA platform. This plug-in allows users to define models from a graphical interface. Moreover, it provides the possibility to generate a graphical model from a GAMA model.

Keywords: *Graphical Modeling, GAMA Platform, Graphical Model Generation*

1 Introduction

La modélisation à base d'agents est aujourd'hui de plus en plus utilisée pour

l'étude des systèmes complexes. Malheureusement, l'utilisation d'une telle approche de modélisation requière de posséder des compétences solides en programmation, en particulier lorsqu'il s'agit de modéliser des systèmes complexes intégrant des aspects spatiaux et sociaux. Cette difficulté a aujourd'hui comme conséquence que la plupart des modèles sont encore développés par des informaticiens et non directement par des thématiciens, ce qui ralentit la diffusion de cette approche et la transmission des connaissances aux acteurs territoriaux.

Il existe aujourd'hui plusieurs plates-formes dédiées à la modélisation agent permettant aux modélisateurs de développer des modèles à partir d'interfaces graphiques. Cependant, ces plates-formes sont souvent complexes à utiliser ou très limités en termes de richesse des modèles développables.

Le plug-in *GAMAGraM* se veut à la fois simple d'utilisation et en même temps capable de définir des modèles très riches. Il se base pour cela sur la plate-forme GAMA [8] qui est particulièrement puissante et permet de définir des modèles très complexes (exemple : [8, 9]).

2 Plates-formes existantes

Il existe aujourd'hui de nombreuses plates-formes dédiées à la modélisation à base d'agents. Ces plates-formes peuvent être classées en 3 catégories (non-exclusives) en fonction du type de langage utilisé pour définir les modèles.

La première catégorie comprend les plates-formes qui requièrent 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 « gros » modèles. Dans cette catégorie, on trouve entre autres les plates-formes SWARM [5] et Repast [6].

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 en général 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 algorithmiques. Dans cette catégorie, on trouve entre autres les plates-formes Netlogo [11] et GAMA [8].

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 algorithmiques. Elle offre de plus l'avantage de faciliter le dialogue entre modélisateurs et thématiciens.

StarLogo TNG [7] fait partie de cette catégorie. Cette plate-forme, qui est avant tout un outil pédagogique, est accessible à tout public. Néanmoins, elle se destine au développement de modèles très simples.

Une autre plate-forme permettant la définition graphique de modèles est Repast Symphony [6]. Cette plate-forme permet en effet de définir des modèles de deux façons différentes : en utilisant directement du Java ou en utilisant un langage de modélisation graphique. Si Repast Symphony offre de nombreuses fonctions, ses outils de modélisation graphique restent complexes à utiliser, en particulier pour les utilisateurs n'ayant peu de notion en algorithmique.

La plate-forme CORMAS [2] comprend dans sa dernière version des outils de modélisation graphique (définition de diagrammes d'activité...). Cette plate-forme, qui est spécialisée dans le développement de simulation participative, n'offre toutefois pas la même richesse en termes de développement de modèles que GAMA ou Repast Symphony.

Enfin, la plate-forme MAGéo [4] offre la possibilité de définir simplement des modèles graphiquement. La version actuelle de MAGéo est néanmoins limitée et ne permet de définir que modèle ayant un environnement de type grille (pas

d'intégration de données SIG ou de gestion de graphes).

GAMAGraM a pour objectif de combler ce manque en plates-formes accessibles au plus grand nombre et permettant le développement de modèles avancés. Nous avons choisi d'utiliser GAMA comme base pour GAMAGraM car cette plate-forme open-source permet déjà la définition de modèles très riches et elle est de plus facilement extensible. Le plug-in GAMAGraM offre la possibilité aux utilisateurs de GAMA de définir leur modèle graphiquement puis de le convertir en modèle *GAML* (le langage de modélisation de GAMA). De même, GAMAGraM permet de générer automatiquement un modèle graphique à partir d'un modèle *GAML*. Cette fonction a pour objectif de plus faciliter les discussions (et communications) autour d'un modèle. GAMAGraM est basé sur le plug-in Graphiti d'Eclipse [3].

2 Modéliser avec GAMAGraM

La modélisation sous GAMAGraM est proche de celle proposée par MAGéo : on définit un diagramme conceptuel correspondant à la structure du modèle, puis on renseigne ces différents éléments à l'aide de boîtes de dialogue. La figure 1 présente l'interface de modélisation sous GAMAGraM. Nous présentons dans le tableau 1 tous les éléments pouvant être ajoutés à un diagramme conceptuel. A noter que celui-ci se base sur le méta-modèle de GAMA, qui permet la définition de modèle multi-niveau [10]. A tout moment, le modélisateur pourra paramétrer ces éléments à l'aide de boîtes de dialogue. Par exemple, pour les éléments de type *species*, le modélisateur pourra entre autres définir le nom de l'espèce, sa géométrie, ses variables (type, valeur initiale, valeur dynamique...), l'ordre dans lequel seront exécutés les réflexes, le constructeur de l'espèce. Pour les éléments de type *reflex*, le modélisateur pourra définir les conditions d'applications ainsi que ce que va faire ce comportement. Ce comportement et la condition d'activation seront décrits en *GAML*. Pour les éléments de type *aspect*, le modélisateur pourra décrire comment les agents de l'espèce seront affichés et en particulier les couches à afficher (géométrie, image, texte...).

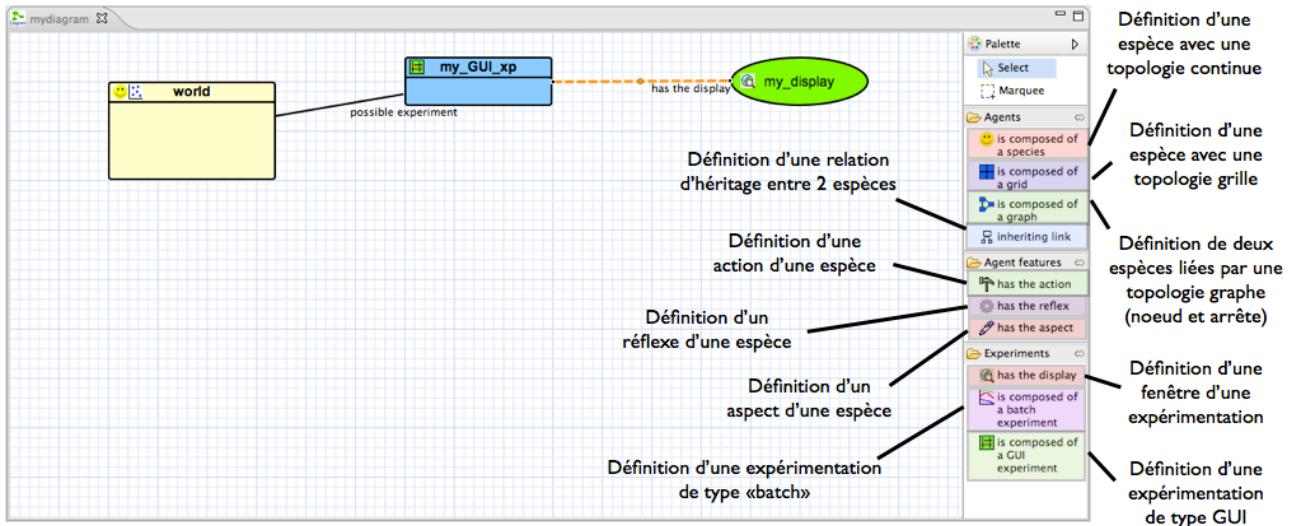


FIG. 1 – Interface de modélisation de GAMAGram

TAB. 1 – Eléments pouvant être intégrés au diagramme conceptuel

Symbole	Source	Description
	Une espèce	Espèce : une espèce représente un archétype (classe) d'agents. Par défaut, un espèce à une topologie continue.
	Une espèce	Grille : une grille est une espèce d'agents dont la topologie est de type grille
	Une espèce	Graphe : un graphe est constitué de 2 espèces d'agents : l'espèce noeud et l'espèce arrête (avec une topologie de type graphe)
	-	Agent monde : agent de plus haut niveau qui contient tous les autres espèces et qui peut contenir des variables et des dynamiques globales
	Une espèce	Action : une capacité que l'agent a (il ne va pas forcément l'appliquer)
	Une espèce	Réflexe : un comportement qui sera appliqué à chaque pas de simulation (suivant certaines conditions)
	Une espèce	Aspect : une visualisation possible pour une espèce d'agents
	Une espèce	GUI Expérimentation : lancement d'une seule simulation
	Une espèce	Batch Expérimentation : lancement d'un ensemble de simulation
	Une expérimentation	Fenêtre d'affichage : fenêtre permettant de visualiser des sorties (cartes, graphiques...)

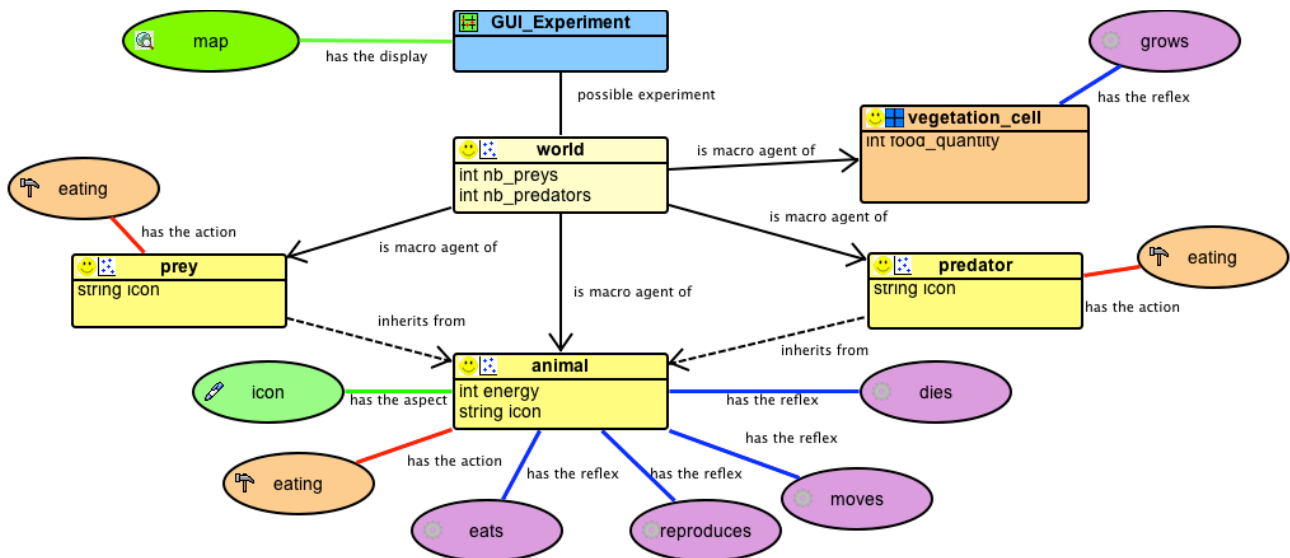


FIG. 2 – Exemple d'un modèle proie-prédateur développé sous GAMAGraM

La figure 2 présente un exemple de modèle développé avec GAMAGraM. Il s'agit d'un modèle de type proie-prédateur dans lequel deux espèces (*predator* et *prey*) héritent de l'espèce *animal*. Cette espèce dispose de différents comportements (*eats*, *reproduces*, *moves* et *dies*), d'une action *eating* et d'un aspect *icon* qui seront automatiquement transmis aux espèces *predator* et *prey* (ces deux espèces spécialisent chacune l'action *eating*). Une grille (*vegetation_cell*) a également définie. Celle-ci aura pour seul comportement un réflexe appelé *grows*.

3 Conclusion

Nous avons présenté dans cet article les grandes lignes du plug-in GAMAGraM qui permet la définition de modèles à base d'agents à l'aide d'un langage de modélisation graphique. Le plug-in, qui est sous licence GPL, ainsi que sa documentation sont disponibles à l'adresse suivante : <http://code.google.com/p/gama-platform/wiki/GraphicalModelingFramework>. Nous prévoyons de nombreuses améliorations pour les futures versions de notre plug-in : une meilleure intégration à GAMA (avec auto-compilation) ainsi que l'ajout de nouvelles fonctions pour suivre les évolutions de GAMA, etc.

Références

[1] A. Banos, N. Marilleau, et MIRO team, Improving individual accessibility to the city: an

agent-based modelling approach. *ECCS*, 2012

- [2] F. Bousquet, I. Bakam, H. Proton, et C. Le Page, Cormas: common-pool resources and multi-agents systems, in *IEA/AIE*, Vol. 2, pp 826-837
- [3] <http://www.eclipse.org/graphiti/>, dernier accès : mai 2013
- [4] P. Langlois, B. Blanpain, et E. Daudé, MAGéo, une plateforme de simulation multi-agents pour tous, *SimTools*, 2013
- [5] N. Minar, R. Burkhart, C. Langton, et M. Askenazi, The Swarm Simulation System: A Toolkit for Building Multi-Agent Simulations, in *SFI Working Paper 96-06-042*, 1996
- [6] M.J. North, E. Tataru, N.T. Collier, et J. Ozik, Visual Agent-based Model Development with Repast Symphony", in *CCISE*, 2007
- [7] M. Resnick, StarLogo: an environment for decentralized modeling and decentralized thinking", in *CHI*, pp 11-12, 1996
- [8] P. Taillandier, A. Drogoul, D. Vo, et E. Amouroux, Gama: a simulation platform that integrates geographical information data, agent-based modeling and multi-scale control. In *PRIMA*, pp. 78–98, 2010
- [9] P. Taillandier, O. Therond et B. Gaudou, A new BDI agent architecture based on the belief theory. Application to the modelling of cropping plan decision-making, *iEMSS*, 2012
- [10] D.A. Vo, A. Drogoul, et J.D. Zucker, An Operational Meta-Model for Handling Multiple Scales in Agent-Based Simulations. *RIVF*, 2012
- [11] U. Wilensky, NetLogo, Center for Connected Learning and Computer-Based Modeling, Northwestern University. Evanston, IL, 1999