HAL
open science

Description automatique de dynamiques de groupes dans des simulations à base d'agents

Philippe Caillou, Javier Gil-Quijano

To cite this version:

Philippe Caillou, Javier Gil-Quijano. Description automatique de dynamiques de groupes dans des simulations à base d'agents. Journées Francophones sur les Systèmes Multi-Agents (JFSMA 2012), Oct 2012, Honfleur, France. pp.23-32. hal-00738184

HAL Id: hal-00738184

https://hal.science/hal-00738184v1

Submitted on 3 Oct 2012

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.

Description automatique de dynamiques de groupes dans des simulations à base d'agents

P. Caillou[a] J. Gil-Quijano[b]
philippe.caillou@lri.fr javier.gil-quijano@cea.fr

[a]Laboratoire de Recherche en Informatique,
Université Paris Sud – INRIA, France

[b]CEA, LIST, Laboratoire Information Modèles et Apprentissage,
F-91191 Gif-sur-Yvette, France

Résumé

Les simulations à base d'agents (MABS) ont été utilisées avec succès pour modéliser des systèmes complexes dans de nombreux domaines. Néanmoins, un problème des MABS est que leur complexité augmente avec le nombre d'agents et de types de comportements différents considérés dans les modèles. Pour des systèmes de taille moyenne à grande, il est impossible de valider, voire même d'observer simplement les trajectoires des agents individuels lors d'une simulation. Les approches de validation classiques, où seuls des indicateurs globaux sont calculés, sont trop simplistes pour permettre d'évaluer le modèle de simulation avec un degré de confiance suffisant. Il est alors nécessaire d'introduire des niveaux intermédiaires de validation et d'observation. Dans cet article, nous proposons l'utilisation de la classification automatique de données (clustering) combinée à la caractérisation automatisée de clusters pour construire, décrire et suivre l'évolution de groupes d'agents en simulation. La description de clusters est utilisée pour générer des profils d'agents qui sont réintroduits dans les simulations avec l'objectif d'étudier la stabilité des descriptions et des structures des clusters sur plusieurs simulations et de décider de leur capacité à décrire les phénomènes modélisés. Ces outils permettent au modélisateur d'avoir un point de vue intermédiaire sur l'évolution du modèle. Ils sont suffisamment flexibles pour être appliqués à la fois hors ligne et en ligne comme le montrent les analyses réalisées à la fois sur des simulations Netlogo et sur des logs de simulations.

Mots-clés : *Simulation multi-agents, Méthodes et méthodologies multi-agents, Vérification et validation des systèmes multi-agents*

Abstract

Multi agent based simulations (MABS) have been successfully exploited to model complex systems in different areas. Nevertheless a pitfall of MABS is that their complexity increases with the number of agents and the number of different types of behaviours considered in the model. For average and large systems it is impossible to validate the trajectories of single agents in a simulation. The classical validation approaches, where only global indicators are evaluated, are too simplistic to give enough confidence on the simulation's model. It is then necessary to introduce intermediate levels of validation. In this paper we propose the use of data clustering and automated characterization of clusters in order to build, describe and follow the evolution of groups of agents in simulations. The description of clusters is used to generate profiles of agents that are reintroduced in simulations in order to study the stability of the descriptions and structures of clusters over several simulations and decide their capability to describe the modelled phenomena. These tools provides the modeller with an intermediate point of view on the evolution of the model. They are flexible enough to be applied both offline and online, and we illustrate it with both a NetLogo and a CSV-simulation log example.

Keywords: *Simulation multi-agents, Méthodes et méthodologies multi-agents, Vérification et validation des systèmes multi-agents*

1 Introduction

La modélisation par simulation de systèmes complexes est un processus cyclique, le modèle etant progressivement amélioré en introduisant des nouvelles connaissances et en corrigeant des *bugs* et/ou des effets indésirables. Les systèmes multi-agents (SMA) sont particulièrement bien adaptés pour représenter les phénomènes complexes à partir de la description des comportements locaux d'agents. Une fois que les comportements des agents sont définis, le processus de modélisation cyclique lors de

l'utilisation du SMA se concentre généralement sur la détermination de paramètres globaux de la simulation et/ou des états initiaux des agents qui permettent de reproduire les comportements globaux observés dans les phénomènes modélisés. Calibrer les simulations signifie trouver le bon ensemble de valeurs (ou intervalles de valeurs) pour les paramètres globaux et individuels qui permettent de minimiser la différence entre comportements aggrégés simulés et observés. Cette démarche d'optimisation est mise en oeuvre dans plusieurs outils existants comme LEIA [5] et GAMA [15].

Néanmoins, cette approche traditionnelle présente des limites lorsqu'il s'agit de modéliser des systèmes complexes. En effet, dans de tels systèmes, des phénomènes différents peuvent se produire simultanément à différents niveaux intermédiaires et s'influencer mutuellement [6]. Par exemple, des groupes d'agents (vols d'oiseaux, des groupes sociaux, etc) suivant des trajectoires similaires peuvent apparaitre, évoluer et disparaitre. Pour décrire et évaluer l'évolution de ces groupes, l'observation de variables globales n'est pas suffisante. En outre, en raison des propriétés émergentes des systèmes complexes, ces groupes peuvent être inattendus, et, pire encore, leur présence peut passer inaperçue, car aucune variable ou tout autre mécanisme d'observation adapté n'est fourni dans le simulateur. L'importance et l'existence même des groupes peuvent alors être cachées en raison du nombre très important d'informations qui peuvent être observées dans les simulations SMA.

Dans cet article, nous introduisons l'utilisation d'outils statistiques pour aider le modélisateur dans la découverte, la description et le suivi de l'évolution des groupes d'agents. Nous proposons deux outils complémentaires, le clustering de données et le calcul de valeurs-test, utilisés pour détecter automatiquement et décrire des groupes d'agents. Dans la section 3, nous présentons un modèle d'observation qui utilise ces outils pour produire une analyse automatisée de l'évolution des groupes dans des simulations SMA, puis nous l'appliquons en section 4 à deux exemples, le premier en ligne (sur un modèle NetLogo) et le second sur des logs CSV.

2 Etat de l'art

Les simulations à base d'agents ont été largement utilisées pour modéliser des phénomènes économiques, géographiques ou sociaux. Il existe plusieurs plateformes de simulation, certaines plus accessibles, comme NetLogo [7], et d'autres plus ouvertes, comme par exemple MODULECO [13] , GAMA [15] ou Repast [11, 14]. Se basant sur ces plates-formes, certains outils permettent l'analyse automatique de simulations, tels que l'analyseur LEIA [5], SimExplorer [9] et [2].

LEIA [5] est un navigateur des simulations, qui pourvoi l'utilisateur avec une comparaison visuelle instantanée de N simulations qui fonctionnent en parallèle. LEIA fournit un ensemble d'outils de transformation et de génération pour la modélisation, ainsi que des outils pour le parcourt de l'espace des simulations. Des heuristiques permettent de plus de juger de l'intéret d'un jeu de paramètres par rapport aux autres.

SimExplorer/OpenMole [9] est une plateforme actuellement en développement, qui vise à offrir un environnement générique pour la programmation et l'exécution d'expérimentations sur des modèles complexes. Les objectifs sont multiples : (1) externaliser le développement pour l'exploration du modèle, afin de mettre à la disposition des méthodes et des outils génériques qui peuvent être appliquées dans la plupart des cas pour n'importe quel modèle à explorer ; (2) favoriser la réutilisation des composants disponibles, et donc de réduire l'effort de développement visant les applications de bonne qualité pour l'exploration de modèles ; (3) faciliter une approche de garantie de la qualité dans l'exploration du modèle.

[2] est un outil qui permet de générer et d'exécuter des nouvelles simulations jusqu'à ce que les résultats obtenus soient statistiquement validés selon un test de χ^2. Il peut générer des nouvelles simulations et effectuer des tests statistiques sur les résultats, avec une précision qui augmente progressivement à mesure que les résultats sont produits. Il est utilisable sur n'importe quelle simulation basée sur RePast.

Cependant, tous ces outils ont pour objectif d'explorer l'espace des paramètres par l'exécution/l'analyse de plusieurs simulations. Notre objectif principal est d'étudier plusieurs étapes d'une seule simulation. Pour explorer une simulation, les seuls outils existants sont des outils intégrés (tels que les graphes et les logs dans NetLogo), qui se limitent à des groupes d'agents définis par l'utilisateur, et le data mining classique sur des logs de simulation. Nous souhaitons prendre en compte les avantages de l'analyse en ligne et de l'analyse orientée-agent de NetLogo et les combiner avec le potentiel de

flexibilité et de description des outils de data-mining.

3 Modèle d'observation

Notre objectif est de décrire, en ligne ou hors ligne, ce qui se passe dans une simulation au niveau de clusters d'agents. Notre modèle d'observation peut être décrit à travers plusieurs étapes présentées dans la section 3.2. La notation utilisée pour décrire ces étapes est présentée dans la section suivante.

3.1 Notations

Etant donnée l'ensemble de simulations $\mathbb{S} = \{s_1, ..., s_s\}$, l'ensemble d'agents $\mathbb{A}_t^s = \{a_{t,1}^s, ..., a_{t,n}^s\}$ au pas de temps t dans une simulation donnée $s \in \mathbb{S}$ et l'ensemble $\mathbb{V} = \{v_1, .., v_m\}$ de variables utilisées pour analyser les simulations dans $\mathbb{S}$ [1].

L'ensemble $\mathbb{V}$ est donnée par : $\mathbb{V} = \mathbb{V}ag \cup \mathbb{V}init \cup \mathbb{V}calc$ où :

- $\mathbb{V}ag = \{v_{ag,1}, ..., v_{ag,d}\}$ est l'ensemble de variables qui décrivent les états des agents. Pour l'instant, nous ne considérons qu'un seul type d'agent, alors $\mathbb{V}ag$ est identique pour chaque agent.
- $\mathbb{V}init = \{v_{init,1}, ..., v_{init,d}\}$ est un ensemble de variables virtuelles qui contient les valeurs initiales de $\mathbb{V}ag$. $\mathbb{V}init$ peut être considéré comme l'ensemble de paramètres des agents.
- $\mathbb{V}calc = \{v_{calc,1}, ..., v_{calc,q}\}$ est un ensemble de variables statistiques calculées sur chaque agent. Par exemple la moyenne mobile des variables dans $\mathbb{V}ag$.

Nous définissons l'ensemble $\mathbb{V}clust \subset \mathbb{V}$ qui contient les variables à être considérées dans le clustering.

Le sous-ensemble $\mathbb{V}mod^C \subseteq \mathbb{V}init$ contient l'ensemble de variables initiales utilisées pour définir un *modèle d'agent* associé au cluster C.

Nous notons $x_j^{a,t}$ la valeur de la variable v_j pour l'agent a au pas de temps t, et $X_{\mathbb{V}}^{a,t} = \{x_j^{a,t} / v_j \in \mathbb{V}\}$ l'ensemble de valeurs pour toutes les variables dans $\mathbb{V}$.

Nous définissons l'ensemble de clusters d'agents au pas de temps t dans la simulation s comme $\mathbb{C}_t^s = \{C_{t,1}^s, ..., C_{t,c}^s\}$.

1. Pour l'instant, nous ne considérons que des simulations avec des variables numériques et booléennes. Les variables qualitatives seront considérées dans le travail futur.

$\mathbb{C}_t^s$ a une fonction d'intension associée : $intension_{\mathbb{C}_t^s}(a \in \mathbb{A}_t^s, X_{\mathbb{V}_{clust}}^{a,t}) \rightarrow C \in \mathbb{C}_t^s$. Cette fonction permet d'affecter un agent a au cluster C le plus indiqué en tenant compte de ses valeurs pour les variables de clustering $\mathbb{V}clust$. Par exemple dans le cas de l'algorithme classique de clustering k-means $intension_{\mathbb{C}_t^s}(a, X_{\mathbb{V}_{clust}}^{a,t}) = \underset{C}{\operatorname{argmin}} \sum_{C \in \mathbb{C}_t^s} \sum_{j=1}^{c} |x_{clust,j}^a - \mu_j^C|^2$, où μ_j^C est la valeur moyenne de la variable j dans le cluster C. Pour l'algorithme k-means, la fonction de définition affecte l'agent a à son cluster le plus proche.

Nous appelons $extension(C_t^s)$ l'ensemble d'agents contenus dans le cluster C_t^s. $\forall C_t^s \in \mathbb{C}_t^s, extension(C_t^s) \subset \mathbb{A}_t^s$.

Etant donné un cluster C_t^s, nous pouvons définir l'ensemble $\mathbb{V}sig^{C_t^s} \subset \mathbb{V}$ de variables significatives, comme l'ensemble de variables dont la valeur test (VT) est significative (voir section 3.2) : $\forall v \in \mathbb{V}sig^{C_t^s}, |VT(v, C_t^s)| \geq 2$.

En fin, nous définissons l'ensemble $\mathbb{GP}_s = \{gp_{1,s}, ..., gp_{p,s}\}, s \in \mathbb{S}$ de paramètres globaux pour chaque simulation $s \in \mathbb{S}$.

3.2 Description du modèle

Nous présentons ici une description détaillée de chaque étape du processus d'observation.

Sélection du modèle. Notre modèle est générique pour l'analyse de flux de données d'une simulation. Il peut analyser directement des données en provenance d'une plateforme de simulation (par exemple NetLogo) ou extraites d'un fichier de logs (en simulant un flux de données).

Traitement des données. Une matrice de données est générée tous les st pas de simulation. Une ligne dans cette matrice représente l'état d'un agent. Ces données brutes ne sont pas les seules variables intéressantes pour la génération et l'analyse de clusters. A partir de l'ensemble $\mathbb{V}_{ag}$ de variables qui décrivent les agents, nous générons les ensembles $\mathbb{V}calc$ et $\mathbb{V}init$. Dans $\mathbb{V}calc$ plusieurs filtres ou agrégateurs peuvent être considérés pour enrichir le flux de données. Pour l'instant, nous utilisons un seul filtre : pour chaque variable, nous définissons une nouvelle variable qui correspond à sa moyenne mobile. $\mathbb{V}init$ contient les valeurs initiales de $\mathbb{V}_{ag}$ de chaque agent dans la simulation. Par défaut,

ces variables *initiales* ne sont pas utilisées dans le calcul de clusters, mais dans la description a posteriori des clusters. Le sous-ensemble des variables utilisées pour la classification $\mathbb{V}clust$ doit être sélectionné au début de la simulation (il est maintenu constant pour le reste de l'analyse). Par défaut, $\mathbb{V}clust = \mathbb{V}ag \cup \mathbb{V}calc$.

Clustering : est-il possible de retrouver des groupes homogènes ?. Le clustering est effectuée sur l'ensemble de données $\mathbb{V}clust$ afin de générer des groupes d'agents homogènes. Notre objectif n'est pas de proposer un nouvel algorithme de clustering, dans notre modèle n'importe quel algorithme de clustering peut être utilisé (dans notre application, n'importe quel algorithme de la librairie Weka [2] et ses paramètres associés). Par défaut, l'algorithme de clustering XMeans [12] est utilisé avec la fonction de similarité classique basée sur la distance euclidienne.

Description : comment décrire les clusters ?. Une fois que les clusters sont identifiés, et si nous considérons $\mathbb{A}_t^s$ l'ensemble d'agents et $\mathbb{A}_{C_t^s} = extension(C_t^s)$ l'ensemble d'agents dans le cluster C_t^s, il est possible d'obtenir des descriptions claires et précises de chaque cluster en se basant sur le calcul des valeurs-test (VT, équation 1)[10, 8]. Le VT représente l'importance de la valeur moyenne $E(\mathbb{A}_{C_t^s}, v)$ d'une variable v sur l'ensemble $\mathbb{A}_{C_t^s}$ par rapport à la distribution de v sur l'ensemble $\mathbb{A}_t^s$. Grossièrement, le VT peut être décrit comme étant la différence entre la moyenne de v dans le cluster ($E(\mathbb{A}_{C_t^s}, v)$) et la moyenne globale ($E(\mathbb{A}_t^s, v)$) normalisée par rapport à l'écart-type global de v ($\sigma^2(\mathbb{A}_t^s, v)$) :

$$VT(v, C_t^s) = \frac{(E(\mathbb{A}_{C_t^s}, v) - E(\mathbb{A}_t^s, v))}{\sqrt{(\frac{card(\mathbb{A}_t^s) - card(\mathbb{A}_{C_t^s})}{card(\mathbb{A}_t^s) - 1} \times \frac{\sigma^2(\mathbb{A}_t^s, v)}{card(\mathbb{A}_{C_t^s})})}} \quad (1)$$

où $card$ est la fonction de cardinalité qui renvoie la taille d'un ensemble.

v est significative pour l'ensemble $\mathbb{A}_{C_t^s}$ si $|VT(v, C_t^s)| \geq 2$. Si $VT(v, C_t^s) \geq 2$, la valeur de v dans $A_{C_t^s}$ est en moyenne plus grande que sa valeur dans $\mathbb{A}_t^s$ (et plus petite si $VT(v, C_t^s) \leq -2$).

Pour faciliter la comparaison des clusters découverts lors d'une simulation, nous les présentons sous une vue globale qui permet de visualiser les variables les plus significatives dans la simulation (voir par exemple la figure 1).

Evolution des clusters. Afin de décrire l'évolution des clusters, nous considérons deux hypothèses alternatives : soit l'extension, soit l'intension est fixée pour l'analyse.

Evolution des clusters en fixant leur extension

La première possibilité est de fixer l'extension (la population) d'un cluster, à un pas de simulation donné, $extension(C_t^s) = extension(C_{t'}^s), \forall t' > t$. La description du cluster $C_{t'}^s$ est mise à jour à chaque pas de simulation. Etant donné que les agents du cluster restent les mêmes, $VT(v_{init}, C_{t'}^s) = VT(v_{init}, C_t^s)$ (à moins que certains agents sortent, ou que de nouveaux agents entrent dans la simulation). Pour les autres variables (en particulier les variables dans $\mathbb{V}ag$ et $\mathbb{V}calc$), VT peut évoluer car l'état des agents peut changer.

Description de la stabilité. Pour évaluer la stabilité d'un cluster avec un extension fixe, nous définissons la mesure *Desc-stability* (eq. 3). Elle se base sur le calcul de la *stabilité* du cluster C sur une variable v par rapport à un cluster C', comme il suit :

$$\underset{v}{stability}(C, C') = 1 - \frac{|VT(v, C) - VT(v, C')|}{\max(|VT(v, C)|, |VT(v, C')|)} \quad (2)$$

La valeur maximale de stabilité est de 1, ce qui signifie que la valeur de v-test de v est la même pour les deux clusters. L'équation 2 retourne des valeurs négatives lorsque les deux valeurs de v-test ont des signes différents. *Desc-stability* mesure la stabilité moyenne du cluster initial par rapport à toutes les descriptions avec extension fixe :

$$Desc - stability(C^s) = \frac{\sum_{t' \neq t} \sum_{v \in \mathbb{V}} \underset{v}{stability}(C_t^s, C_{t'}^s)}{card(\mathbb{V}) \times card(\{t'/t' \neq t\})} \quad (3)$$

Evolution des clusters en fixant leur intension

L'alternative est de fixer l'intension des clusters. A chaque nouvelle étape, la fonction $intension$ est utilisée pour déterminer quels agents affecter à chaque cluster (certains clusters peuvent être vides) et les nouvelles descriptions sont calculées. Pour un cluster C_t^s, pour chaque pas de temps $t' > t$ un nouveau cluster $C_{t'}^s$ est construit où $\forall a \in extension(C_{t'}^s), intension_{\mathbb{C}_t^s}(a \in \mathbb{A}_{t'}^s, X_{\mathbb{V}_{clust}}^{a,t'}) \rightarrow C_{t'}^s$. La description de chaque

2. http ://weka.wikispaces.com/

cluster $C^s_{t'}$ peut être facilement comparée à celle de C^s_t. Etant donné que les intentions des clusters sont les mêmes, le v-tests des variables utilisées dans le clustering ($\mathbb{V}clust$) sont très enclins à rester les mêmes. Mais les v-tests des autres variables, en particulier ceux des variables initiales ($\mathbb{V}init$) des agents peuvent évoluer (puisque la population du cluster change).

Stabilité de la population La mesure *Pop-stability* permet de calculer la proportion de la population qui reste dans l'extension d'un cluster alors que l'intension est fixe :

$$Pop-stability(C^s) = \frac{\sum_{t' \neq t} \frac{card(C^s_t \cap C^s_{t'})}{max(card(C^s_t), card(C^s_{t'}))}}{card(\{t'/t' \neq t\})} \quad (4)$$

Génération de simulations. Afin de valider les descriptions des clusters, nous étudions d'une part, leur stabilité à travers différentes simulations, et d'autre part, leur sensibilité à la variation de paramètres globaux. Dans ces buts, nous définissons deux méthodes de génération et d'analyse de simulations. La première méthode conserve l'intension du cluster et applique simplement la même intension sur les nouvelles simulations. La seconde méthode suppose que les valeurs significatives initiales identifiées pour un cluster définissent un "*modèle d'agent*". Ce modèle est utilisé dans de nouvelles simulations et l'analyse vérifie si les autres variables des agents générés ont un comportement similaire à celui du cluster initial.

Génération en fixant la définition du groupe. La définition d'un groupe d'agents (l'intension du cluster) est maintenue constante. Pour un groupe cible C^s_t, ns nouvelles simulations sont générées. Au pas de temps t de chaque nouvelle simulation s', un nouveau cluster $C^{s'}_t$ est calculé à partir de l'intension cible :
$\forall a \in extension(C^{s'}_t), intension_{\mathbb{C}^s_t}(a \in \mathbb{A}^{s'}_t, X^{a,t}_{\mathbb{V}_{clust}}) \rightarrow C^{s'}_t$. Les descriptions de ces nouveaux groupes peuvent être comparées à la description cible afin d'évaluer la stabilité des résultats.

Génération en fixant le modèle d'agent. A partir de l'extension d'un cluster cible C^s_t, il est possible de construire une distribution des variables significatives initiales $\mathbb{V}mod^{C^s_t} = \mathbb{V}init \cap \mathbb{V}sig^{C^s_t}$, définissant ainsi un *modèle d'agent* pour C^s_t. Nous émettons l'hypothèse que ces variables suivent des distributions gaussiennes.

Etant donné un cluster C^s_t, son modèle d'agent est défini par la distribution de ses variables significatives initiales comme il suit : $\{v_{mod} \sim \mathcal{N}(\mu(v_{mod}, C^s_t), \sigma(v_{mod}, C^s_t)), \forall v_{mod} \in \mathbb{V}mod^{C^s_t}\}$.

où $\mathcal{N}$ est la distribution gaussienne, $\mu(v_{mod}, C^s_t)$ et $\sigma(v_{mod}, C^s_t)$ sont respectivement la moyenne et l'écart-type de la variable v_{mod} au pas t de la simulation s pour les agents dans $extension(C^s_t)$ (étant donné que les variables $\mathbb{V}_{init}$ sont fixées, le temps t n'est pas important).

Taille des clusters : Nous définissons la *proportion* de la population dans C^s_t par rapport à la taille de l'ensemble de la population en s au pas de temps t comme il suit : $proportion(C^s_t) = \frac{card(C^s_t)}{card(\mathbb{A}^s_t)}$

Génération d'agents : Nous générons ns nouvelles simulations. Pour chaque simulation s' un ratio fixe ($proportion(C^s_t)$) d'agents est choisi au hasard et le modèle d'agent est appliqué pour biaiser les valeurs initiales de ces agents. Le biais est appliqué en changeant les valeurs initiales de toutes les variables $v_{mod} \in \mathbb{V}mod^{C^s_t}$ en suivant le modèle de distribution.

Nous appelons $\mathbb{C}^{s',C^s_t}_{gen}$ le cluster défini par l'ensemble des agents biaisées dans la simulation s' au pas de simulation initial suivant le modèle de C^s_t. Une fois ce groupe défini, il est possible de le décrire dans la nouvelle simulation à l'instant t, afin de le comparer avec la description du cluster original C^s_t. Si la description est stable, cela signifie que le modèle d'agent est suffisant pour expliquer la description du cluster.

Analyses de stabilité et de sensibilité aux paramètres globaux.

Analyse de stabilité : Pour évaluer la stabilité d'un cluster cible $C^{s_0}_t$ par rapport à un cluster équivalent $C^{s'}_t$ dans une nouvelle simulation, nous utilisons la stabilité définie dans l'équation 2. La stabilité d'un cluster $C^{s_0}_t$ sur une variable v sur un ensemble de simulations $\mathbb{S}'$ est donnée par :

$$\underset{v}{stability}(C^{s_0}_t, \mathbb{S}') = \frac{\sum_{s' \in \mathbb{S}', s' \neq s_0} \underset{v}{stability}(C^{s_0}_t, s')}{card(\mathbb{S}') - 1} \quad (5)$$

Sensibilité aux paramètres globaux : Dans le but d'étudier la sensibilité d'un cluster donné $C^{s_0}_t$ par rapport à un paramètre global $gp \in \mathbb{GP}_{\mathbb{S}}$,

nous régénérons des nouvelles simulations en considérant différentes valeurs de gp et en calculant la stabilité de $C_t^{s_0}$ dans ces nouvelles simulations.

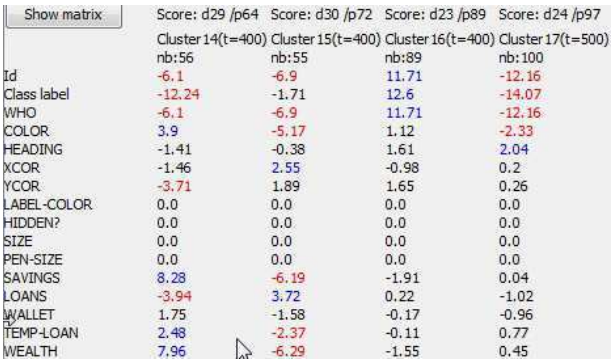
4 Résultats expérimentaux

Pour illustrer notre modèle d'observation, nous présentons quelques expériences effectuées avec notre outil d'analyse. L'outil a été implémenté en Java, utilisant à la fois Weka et l'API de NetLogo. Pour illustrer la généricité de notre modèle, nous avons implémenté des générateurs de flux de données pour NetLogo et pour des fichiers de log au format CSV. Nous présentons les résultats obtenus sur un modèle simple NetLogo et sur l'analyse de fichiers de logs d'une simulation plus complexe.

4.1 Expérimentation sur NetLogo : le modèle *Bank Reserves*

Présentation du modèle. Le modèle *Bank Reserves*, fourni avec Netlogo, est un modèle très simple où lorsque deux agents se rencontrent, un agent tente d'acheter (donner de l'argent) à l'autre. S'il n'a pas d'argent, il tente d'empreunter à la banque. La banque prête de l'argent (et crée de l'argent) tant que le montant total de prêts n'atteint pas le montant total des dépôts ($savings$) multiplié par un paramètre ($1 - Reserves$). Dans ce modèle, des variables globales donnent un aperçu d'une expérience : on peut ainsi observer différents phénomènes, comme par exemple la stabilisation du total d'argent ($money$) lorsque le montant maximal de prêts ($loans$) est atteint, ou encore suivre des groupes fixes d'agents qui dépendent de paramètres globaux (par exemple, 3 groupes : un groupe avec richesse ($wealth$) négative, un avec $wealth > richThreshold$ et le reste). Même si ces informations sont intéressantes, une compréhension plus précise du comportement du modèle ne peut pas être obtenue avec une telle observation globale/locale. Par exemple, il est difficile de savoir *qui sont les agents riches ?* ou encore si *les riches restent riches ?* Ce serait encore plus vrai pour des modèles plus complexes, pour lesquels les variables d'interaction sont beaucoup plus difficiles à déduire que pour une simulation jouet très simple. Pour notre expérience illustrative, on utilise $Reserves = 70$, $People = 200$ et $richThreshold = 20$.

Description des Clusters. *"Qui sont les riches ?"* Nous avons utilisé notre outil d'analyse avec



Show matrix	Score: d29 /p64	Score: d30 /p72	Score: d23 /p89	Score: d24 /p97
	Cluster 14(t=400) nb:56	Cluster 15(t=400) nb:55	Cluster 16(t=400) nb:89	Cluster 17(t=500) nb:100
Id	-6.1	-6.9	11.71	-12.16
Class label	-12.24	-1.71	12.6	-14.07
WHO	-6.1	-6.9	11.71	-12.16
COLOR	3.9	-5.17	1.12	-2.33
HEADING	-1.41	-0.38	1.61	2.04
XCOR	-1.46	2.55	-0.98	0.2
YCOR	-3.71	1.89	1.65	0.26
LABEL-COLOR	0.0	0.0	0.0	0.0
HIDDEN?	0.0	0.0	0.0	0.0
SIZE	0.0	0.0	0.0	0.0
PEN-SIZE	0.0	0.0	0.0	0.0
SAVINGS	8.28	-6.19	-1.91	0.04
LOANS	-3.94	3.72	0.22	-1.02
WALLET	1.75	-1.58	-0.17	-0.96
TEMP-LOAN	2.48	-2.37	-0.11	0.77
WEALTH	7.96	-6.29	-1.55	0.45

FIGURE 1 – Vue d'ensemble des clusters. Les trois premiers groupes correspondent à des clusters d'agents identifiés à l'instant $t = 400$. Les valeurs bleues sont celles significativement plus élevées pour ce cluster ($VT > 2$) et les valeurs rouges celles significativement plus basses ($VT < -2$)

la configuration par défaut : $\mathbb{V}clust = \mathbb{V}ag \cup \mathbb{V}calc$. Le clustering est réalisé tous les $st = 100$ pas de simulation. A chaque étape de clustering, les clusters sont identifiés, visualisés sur NetLogo (avec des couleurs), et leur extension et description sont présentées. Par exemple, dans la figure 1 à $t = 400$, trois groupes sont identifiés : un cluster *pauvre* (55 agents, avec une faible richesse $VT(wealth, pauvre) = -6,29$), un cluster *moyen* (89 agents) et un cluster *riche* : $cluster14$.

"Quelles sont les caractéristiques des riches à chaque étape ?" A la fin de la simulation, une description de tous les clusters obtenus à chaque étape donne un aperçu global de la simulation (la figure 1 est en fait un extrait de cette fenêtre). Dans notre expérience, il est toujours possible d'identifier un cluster *riche* et un cluster *pauvre*, et parfois (comme dans $t = 400$) un cluster *moyen*. A partir de leur description, il est possible de constater que le lien entre la richesse ($wealth$) et la richesse initiale ($T0Wallet$) n'est plus significatif après $t = 800$ ($|VT(T0Wallet, riche)| < 2$ pour $t \geq 800$). Cela peut être liée au fait (observé avec l'observation globale sur NetLogo) que la banque a atteint sa limite max de prêts (la somme totale s'arrête d'augmenter à environ $t = 400$).

Cependant, comparer des clusters de différentes étapes dans cette vue d'ensemble s'avère difficile car ils sont différents à la fois en intension et en extension (voir la section 3.2). Dans un modèle plus complexe, un cluster peut avoir une signification complètement différente à chacune des étapes. Afin de décrire des évolutions, il est

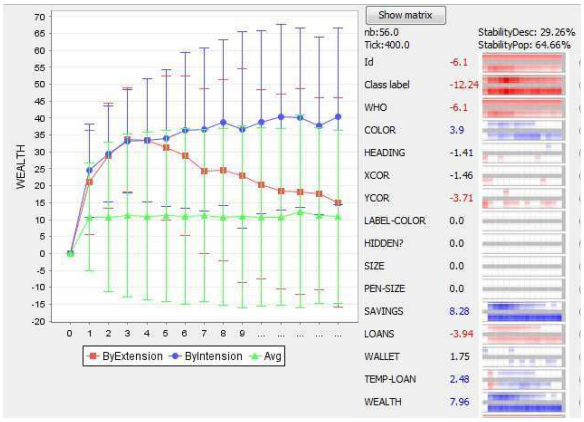


FIGURE 2 – Analyse du $Cluster14$: sur la droite, pour chaque variable, son VT à l'instant t = 400, et deux barres qui correspondent à l'évolution de la valeur de VT pour chaque étape avec extension fixe (barre du haut, la population du cluster reste la même) ou intension fixe (barre inférieure, la définition du cluster reste la même). Sur la gauche, on observe l'évolution des valeurs d'une variable sélectionnée ($wealth$) pour toute la population (en vert), pour le cluster fixé en extension (en rouge) et en intension (en bleu). $t = 1$ dans le graphique représente $t = 100$ pas de simulation.

donc nécessaire de suivre un cluster, défini en extension ou en intension.

Evolution des clusters.

Extension fixe : ***les riches restent-ils riches ?*** La première possibilité est de fixer l'extension (population) d'un cluster, à un pas t de simulation donné. La description du cluster C est mise à jour à chaque étape. Etant donné que les agents du cluster restent les mêmes, $VT(v_{init}, C_{t'}) = VT(v_{init}, C_t) \forall t' > t$ (à moins que certains agents sortent ou des nouveaux agents rentrent dans la simulation).

La figure 2 décrit le $cluster14$ et son évolution, à la fois en extension et en intension. Sur la partie droite de la fenêtre, pour chaque variable, deux barres représentent le VT de la variable avec extension fixe (barre du haut) ou intension fixe (barre inférieure). La couleur est bleue quand $VT > 2$ et rouge quand $VT < -2$. Par exemple, en extension, nous pouvons observer que pour toutes les variables associées à la richesse ($wealth$, $savings$, $loans$), la différence avec les autres agents diminuent : les valeurs absolues des VT des variables $wealth$, $saving$ et $loans$ diminuent (couleur de plus en plus claire et blanche). Cela signifie que, en moyenne, les gens riches de $t = 400$ sont de moins en moins riches. Aucune de ces variables est significativement différent de la moyenne après $t = 1200$. Ceci est représenté à gauche de l'image, où l'on visualise l'évolution de la variable $wealth$: la richesse moyenne de la population est de 10. La richesse moyenne du $cluster14$ en extension est en nette diminution à partir de $t = 400$ ($t = 4$ dans la figure 2 puisque $st = 100$) tout en convergeant vers la moyenne. Cette convergence est également confirmée par le faible score de $Desc - stabilit$ du cluster (29%). Même si elles n'apparaissent pas sur l'image, les valeurs initiales des paramètres ($\mathbb{V}init$) sont par définition stables pour cette évolution.

Intension fixe : ***Les riches restent-ils les mêmes ?*** L'alternative est de fixer l'intension (définition) des clusters. La ligne bleue (graphique de gauche) et les barres inférieures (partie droite) de la figure 2 représentent la description des clusters identifiés à chaque étape avec la fonction d'intension de $t = 400$. Toutes les variables prises en compte dans le clustering ($\mathbb{V}clust$) sont, par définition, à peu prés semblables (comme par exemple $wealth$, $savings$, ...), puisque l'intension des clusters est la même. Toutefois, les autres variables peuvent évoluer (dans notre exemple, les paramètres initiaux des agents $\mathbb{V}init$). Par exemple, le $cluster14$ regroupe à chaque étape les agents riches, mais le nombre et les propriétés initiales de ses agents peuvent évoluer. Il est intéressant de voir que le nombre d'agents riches ($card(cluster14, t)$) reste approximativement constant (56, 53, 52, 48, 50, 53, ...). Mais l'évolution des paramètres initiaux confirme l'observation faite avec la vue d'ensemble : la richesse initiale des agents ($TOWallet$) n'est plus significative après $t = 800$.

Stabilité d'un cluster avec définition de groupe fixe. *"était-ce un effet du hasard et comment réagit le cluster au paramètre $Reserve$?"* Une fois que nous avons identifié un groupe intéressant et sa description (les agents riches du $cluster14$ qui sont initialement riches), il est important de savoir si la description du groupe est stable lorsque la simulation est reproduite avec des paramètres similaires, puis d'étudier l'impact de certains paramètres globaux. Ici, nous pouvons faire l'hypothèse que les gens riches ne peuvent s'enrichir que lorsque la banque est encore capable de faire des prêts. Ainsi, en présence d'une valeur importante du paramètre $Reserves$

TABLE 1 – Génération de simulations : analyse de la stabilité avec définition de cluster similaire (type : Def) et configuration initiale des agents biaisée (type : Modèle)

Type	Var.	Init	Sim1/2/3	Sim4/5/6	Sim7/8/9
Def	Reserves	70	45	70	95
	Size	56	57/46/55	61/51/42	43/41/46
	TOWallet	3,4	3,0/3,3/2,8	3,1/2,7/1,5	-0,7/2,0/1,9
	Stability		0,87	0,69	0,32
Model	Wealth	7,96	0,1/0,2/-1,6	0,7/0,1/-0,6	0,5/-0,2/1,6
	Stability		-0,05	0,01	0,08

le lien entre la richesse initiale et la richesse devrait s'affaiblir. Afin d'étudier à la fois la stabilité et l'effet de paramètres globaux, des nouvelles simulations sont générées automatiquement avec les mêmes valeurs de paramètres globaux, sauf pour le paramètre $Reserves$, qui change de valeur (45, 70 et 95). L'intension du $cluster9$ est fixée et son extension et sa description sont calculées à $t = 400$ pour chaque simulation. Les résultats de l'évaluation de la stabilité sur 9 simulations différentes sont présentés dans le tableau 1 (type Def). Puisque nous avons utilisé toutes les variables, sauf les paramètres initiaux pour le clustering, seulement $\mathbb{V}init$ et la taille du cluster sont intéressants.

Les résultats confirment que les gens riches sont initialement plus riches que la moyenne : $VT(T0Wallet)$ est presque toujours significativement positif ($VT > 2$) pour $Reserve = 45$ et 70. Il n'est cependant pas significatif pour deux des trois simulations avec $Reserve = 95$. Les valeurs de stabilité montrent que notre première simulation avait une valeur de $T0Wallet$ exceptionnelement élevée (stabilité de $TOWallet$ plus importante pour $Reserve = 45$ que pour $Reserve = 70$).

Stabilité d'un cluster avec modèle d'agent fixe. *"est-ce que la richesse peut-être expliquée uniquement à partir de sa valeur initiale ?"* Une autre possibilité pour valider nos résultats est d'essayer de créer des agents à partir des variables significatives initiales et de voir si ils se comportent comme nous pensons qu'ils devraient le faire. Par exemple, avec $cluster14$ nous avons identifié un *profil d'agent riches* caractérisé par sa richesse initiale élevée et sa richesse très élevé à $t = 400$. Pour tester la stabilité de ce cluster, nous pouvons changer les paramètres initiaux de certains agents dans une nouvelle population pour les faire correspondre avec les caractéristiques initiales significatives du $cluster14$ et évaluer en $t = 400$ si leur description est semblable à celle du $cluster14$. Un seul paramètre initial est significatif ($T0Wallet$ qui, nous l'espérons, devrait conduire à une richesse élevée en $t = 400$). Une proportion similaire de la population dans chaque nouvelle simulation (56/200 agents) est changée de manière à obtenir une distribution de $T0Wallet$ similaire à celle du $cluster14$. Pour l'analyse de la stabilité , nous avons également modifié le paramètre $Reserve$ afin étudier son impact sur le comportement des agents. Contrairement à nos attentes, les résultats (tableau 1(type :model)) ne montrent pas de lien significatif avec les variables de richesse ($wealth$, $savings$, $loans$). Notre hypothèse n'est pas validée : définir un agent avec une richesse initiale légèrement plus élevée ne suffit pas pour obtenir une richesse significativement plus grande à $t = 400$. Des expériences complémentaires montrent que considérer des valeurs élevées du seuil de richesse ($rich - threshold$) (qui impliquent une plus grande diversité des richesses) ou effectuer une observation plus rapide (en $t = 200$ par exemple) sont suffisants pour obtenir un effet de causalité significatif.

4.2 Exemple d'analyse sur des logs de simulation

Description du modèle. Pour illustrer la gestion de simulations plus complexes par notre modèle d'observation, nous l'avons également appliqué à un modèle qui suit l'approche de modélisation KIDS [4], ce qui signifie que le nombre de paramètres et des variables observées est maintenu élevé pour être plus descriptif et réaliste plutôt que synthétique. La simulation du marché de Rungis [3] a été développée avec le framework BitBang [1]. Elle reproduit un marché de gros de fruits et de légumes. Un type d'agent vendeur et 4 types d'agents acheteurs sont considérés dans la simulation (avec des nombreux paramètres, 20 paramètres en moyenne par type d'agent) : les *restaurateurs* qui cherchent à acheter rapidement tout ce dont ils ont besoin, les*TimeFree* qui sont à la recherche de bonnes

Show matrix	Score: d48 /p65	Score: d50 /p55	Score: d46 /p65	Score: d48 /p99
	Cluster5(t=1) nb:127	Cluster6(t=1) nb:85	Cluster7(t=1) nb:87	Cluster8(t=1) nb:100
Day	0.0	0.0	0.0	0.0
Id	-6.18	-7.92	3.34	10.95
Money	6.03	3.45	5.76	-15.23
Margin	6.48	8.57	-4.01	-11.23
MarginRate	8.37	8.09	-0.99	-15.7
NewMoney	-1.69	-1.49	-7.76	10.62
NewMargin	7.65	6.2	-4.46	-9.83
NewMarginRate	7.86	6.5	0.76	-15.32
m_pNbPerceivedQuality	0.0	0.0	0.0	0.0
m_pCurrentSeller	-0.97	-0.74	2.68	-0.82
m_pLastListenTime	3.48	-14.89	7.96	2.75
m_pLeaveAfterTime	0.0	0.0	0.0	0.0
m_pArrivalTime	5.24	-9.18	2.97	0.21
m_pFinishedTime	2.64	-15.08	8.52	3.28

FIGURE 3 – Vue d'ensemble des clusters après 10 jours de simulation du "Marché de Rungis". dxx et pxx représentent respectivement la description et la stabilité de la population

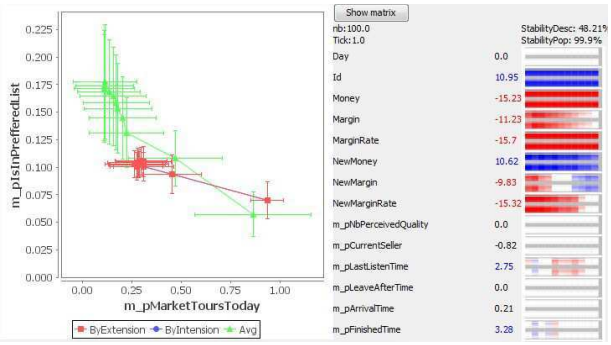


FIGURE 4 – Analyse détaillée du Cluster8. Modèle du marché de Rungis

occasions, les *Barbes* qui cherchent des produit de basse qualité peu chers pour revendre sur leur marché, et les *Neuilly* qui cherchent de la haute qualité. Le fichier csv contient une ligne par couple jour/agent, avec 49 variables de sortie. Ces variables comprennent les marges globale et quotidienne, le nombre de vendeurs visités, les vendeurs habituels, la qualité et la quantité des produits ainsi que les prix. Nous n'analysons ici que les 400 agents acheteurs au cours des 10 premiers jours de la simulation.

Description des clusters

Quatre clusters sont généralement identifiés à chaque étape. Par exemple, au jour 1 de l'analyse, les $cluster5$ à $cluster8$ avec une taille assez homogène (Fig. 3). Un de ces clusters a la particularité d'être parfaitement stable par rapport à sa population ($Cluster8$, 100 agents, stabilité de la population : 99%). Lors de l'examen des identifiants des agents, il apparait que ce groupe contient tous les agents restaurateurs, qui semblent avoir un comportement nettement différent du reste (les 3 autres clusters ont une population mixte). Une de leurs caractéristiques est (par construction) de passer des accords ($TrustAgreements$) avec un unique vendeur pour chaque produit dont ils ont besoin, tandis que les autres agents négocient en général avec plusieurs vendeurs réguliers. La description de ce groupe et son évolution (Fig 4) donne un aperçu très instructif sur leur comportement. Le VT et les graphiques d'évolution sont identiques pour l'analyse en intension et en extension, car la population reste la même. La première propriété très visible est la variable $NewMargin$, qui mesure la marge quotidienne. Cette variable a une valeur très basse au début de la simulation (jour 1, $VT = -9,83$), et devient progressivement non-significative, puis fortement positive après le jour 6 ($VT > 10$). Ceci est représenté sur l'image par l'évolution rouge-blanc-bleu. Cette évolution de la marge (d'abord inférieure à la moyenne, puis supérieure), peut être expliquée en analysant les autres variables significatives représentées sur le graphique figure 4 : $m_pMarketToursToday$ qui mesure le nombre de fois qu'un agent fait un "tour du marché", ce qui signifie qu'il se balade à la recherche d'un produit qu'il n'a pas trouvé chez ses vendeurs habituels, et $m_pIsInPrefferedList$, correspondant au nombre de vendeurs habituels chez qui l'acheteur se rendra en priorité. Les agents $Restorateur$ ont un nombre sensiblement inférieur de vendeurs habituels (la probabilité reste stable autour de 0,1 au bout de 2 jours, alors qu'elle est en constante augmentation pour l'ensemble de la population, le VT passe de -6,49 à -17,1), et une probabilité plus élevée de faire un tour du marché (la probabilité reste plus élevé que 0,25, alors qu'elle décroit pour l'ensemble de la population, VT passe de -0,82 à 14,2). L'interprétation est que la stratégie du restaurateur (quelques vendeurs habituels avec des prix fixes), est moins efficace au début parce que les premiers prix négociés sont élevés. Mais étant donnée qu'il ne visite pas beaucoup de vendeurs (peu de vendeurs habituels), il fera souvent des tours du marché, et trouvera alors des nouvelles opportunités (et des prix mieux négociés), alors que d'autres acheteurs resteront dans leur réseau fermé de vendeurs habituels.

Ce type d'interprétation peut également être faite pour les clusters qui ne correspondent pas à un type d'agent spécifique. Par exemple, le $cluster6$ qui regroupe 85 agents dont la caractéristique principale est d'arriver tôt sur le marché ($m_pArrivalTime$) et de dégager des marges élevées ($NewMargin$). Une analyse plus complète de leurs profil et de leur évolution montre

que c'est effectivement une bonne stratégie. Ceux qui arrivent tôt et qui possèdent un vaste réseau de vendeurs habituels (valeur élevée de $IsInPrefferedList$) vont négocier beaucoup (nombre élevé de visites par vendeur), mais vont voir moins de vendeurs que la moyenne. Et ils vont maintenir une marge plus élevée que la moyenne. Tout ceci peut être interprété à partir de la représentation de l'évolution du cluster, alors que cela aurait été très difficile à identifier à partir d'une analyse classique de logs.

5 Conclusions

Le modèle d'observation des simulations que nous présentons ici fournit au modélisateur des outils génériques qui lui permettent d'avoir une vision synthétique et descriptive des simulations. Il peut être utilisé pour comprendre la dynamique des simulations et pour faciliter leur validation. Grâce au mécanisme de génération des simulations, des agents créés en tenant compte des caractéristiques d'un cluster cible sont réintroduits dans les simulations, les clusters peuvent être reconstruits et les variables globales de simulation peuvent être comparées à leurs valeurs précédentes. De cette façon, la stabilité des clusters et leur "expressivité" peuvent être mesurées sur différentes simulations. Afin de permettre l'analyse d'un nombre important de différents types de simulations, le modèle d'observation est en train d'être adapté afin d'une part de considérer des variables qualitatives et des variables de réseaux, et d'autre part de faciliter l'analyse des simulations à grand nombre d'agents. Cette dernière amélioration se fera par l'intégration de notre modèle au moteur OpenMole, qui prévoit, entre autres facilités, une utilisation facile de la grille de calcul pour les simulations.

Références

[1] T. Baptista, T. Menezes, and E. Costa, 'Bitbang : A model and framework for complexity research', in *ECCS 2006*, (2006).

[2] Philippe Caillou, 'Automated multi-agent simulation generation and validation', in *PRIMA 2010*, p. 16p. LNCS, (2010).

[3] Philippe Caillou, Corentin Curchod, and Tiago Baptista, 'Simulation of the rungis wholesale market : Lessons on the calibration, validation and usage of a cognitive agent-based simulation', in *IAT*, pp. 70–73, (2009).

[4] Bruce Edmonds and Scott Moss, 'From kiss to kids - an 'anti-simplistic' modelling approach', in *MABS*, pp. 130–144, (2004).

[5] Francois Gaillard, Yoann Kubera, Philippe Mathieu, and Sebastien Picault, 'A reverse engineering form for multi agent systems', in *ESAW 2008*, pp. 137–153, (2008).

[6] Javier Gil-Quijano, Thomas Louail, and Guillaume Hutzler, 'From biological to urban cells : lessons from three multilevel agent-based models', in *PRACSYS 2010*, Kolkota, India, (2010). LNCS.

[7] Center For Connected Learning and Computer-Based Modeling, 'Netlogo : http ://ccl.northwestern.edu/netlogo/'.

[8] Ludovic Lebart, Marie Piron, and Alain Morineau, *Statistique exploratoire multidimeensionnelle : visualisation et infï¿½rence en fouilles de donnï¿½es*, Fourth edition, Dunod, Paris, 2006.

[9] Mathieu Leclaire and Romain Reuillon, 'Simexplorer : http ://www.simexplorer.org/'.

[10] Alain Morineau, 'Note sur la caractï¿½risation statistique d'une classe et les valeurs-tests', in *Bull. Techn. du Centre de Statistique et d'Informatique Appliquées*, volume 2, pp. 20–27, (1984).

[11] Michael J. North, Nicholson T. Collier, and Jerry R. Vos, 'Experiences creating three implementations of the repast agent modeling toolkit', in *ACM Transactions on Modeling and Computer Simulation*, volume 16, pp. 1–5, (2006).

[12] Dan Pelleg and Andrew Moore, 'Xmeans : Extending k-means with efficient estimation of the number of clusters', in *17th International Conference on Machine Learning*, pp. 727–734, (2000).

[13] Denis Phan, 'From agent-based computational economics towards cognitive economics', in *Cognitive Economics, Handbook of Computational Economics*.

[14] Steven F. Railsback, Steven L. Lytinen, and Stephen K. Jackson, 'Agent-based simulation platforms : Review and development recommendations', in *Simulation*, volume 82, pp. 609–623, (2006).

[15] Patrick Taillandier, Alexis Drogoul, and Duc-An Vo, 'Gama : a simulation platform that integrates geographical information data, agent-based modeling and multiscale control', in *PRIMA 2010*, (2010).