

# Modèle de distribution: une approche pour faciliter la distribution de modèles à base d'agents

Lucas Grosjean<sup>1,2</sup>, Alexis Drogoul<sup>1</sup>, Bénédicte Herrmann<sup>2</sup>, Nghi Quang Huynh<sup>1,3</sup>, Christophe Lang<sup>2</sup>, Nicolas Marilleau<sup>1,2</sup> et Laurent Philippe<sup>2</sup>

<sup>1</sup>UMI 209, UMMISCO, IRD, Sorbonne Université, Bondy, France

<sup>2</sup>Université Marie et Louis Pasteur, CNRS, institut FEMTO-ST, F-25000 Besançon, France

<sup>3</sup>Can Tho University, Can Tho, Vietnam

## Résumé

La modélisation à base d'agents est un outil puissant pour comprendre les problèmes complexes. Certains modèles exigent une quantité importante d'agents, de données et de ressources pour fonctionner correctement. La plupart des plateformes de simulation d'agents exécutent les modèles séquentiellement et ne peuvent pas exécuter ces modèles de grande taille dans un délai raisonnable, voire ne peuvent même pas les exécuter. Pour résoudre ces problèmes, les SMA peuvent utiliser le calcul distribué afin de répartir la charge sur plusieurs processus. Cependant, la distribution de SMA introduit de nouvelles problématiques qui sont autant de défis à relever lors de l'exécution des simulations. Cet article présente le concept de *modèle de distribution* visant à simplifier la distribution des SMA. Ce modèle offre un cadre flexible pour modéliser la distribution de modèles en séparant la modélisation thématique du modèle, de la modélisation de sa distribution. Nous illustrons, avec la plateforme GAMA, comment améliorer l'expérience des modélisateurs pour distribuer efficacement leur modèle.

**Mots clés** : Distribution, Modélisation, HPC, Simulation Multi-Agent

## Abstract

Agent-based modeling is a powerful tool for understanding complex problems. Some models require large amounts of agents and data. Most agent simulation platforms run models sequentially and cannot run large models in a reasonable time, or at all. To solve these problems, ABMs can use distributed computing to spread the load over multiple processors. However, the distribution of ABMs introduces new issues that pose challenges for the execution of simulations. This paper presents the concept of *distribution model* to simplify the distribution of ABMs. This model offers a flexible framework for modeling the distribution of models by separating the thematic modeling of the model from the modeling of its distribution. Using the GAMA platform, we illustrate how to improve the experience of modellers in order to distribute their models efficiently.

**Keywords:** Distribution, Modeling, HPC, Agent-Based Model

## 1 Introduction

Les systèmes multi-agent (ou SMA) sont utilisés pour simuler et comprendre des systèmes réels de plus en plus complexes. A mesure que l'échelle de ces modèles et la précision des comportements modélisés augmentent, des problèmes de performance apparaissent. Leur exécution dans une simulation mono-processus, c'est à dire séquentielle, se voit limitée par les ressources de calcul disponibles si bien qu'elle n'est parfois plus possible dans un délai raisonnable voire pas possible du tout. Dans ce cas, l'utilisation de techniques de calcul à haute performance (High Performance Computing ou HPC) ouvre des perspectives intéressantes pour s'affranchir des contraintes de ressources et des limitations matérielles en distribuant une même simulation sur plusieurs processus.

Malgré la nature intrinsèquement "distribuée" des systèmes multi-agents, la plupart de ces systèmes, y compris les plus simples, décrivent un monde unique avec des interactions et des données partagées synchronisées rendant difficile leur distribution. Les simulations distribuées à base d'agents nécessitent alors des opérations de synchronisation fréquentes, telles que la mise à jour de l'état de l'environnement

et des perceptions des agents. La "distribution" d'un modèle basé sur des agents n'est donc pas une tâche triviale, ni facilement généralisable. Certaines plateformes s'attaquent à ce problème en offrant des fonctionnalités qui permettent aux utilisateurs de simuler des SMA sur des architectures distribuées, y compris les architectures HPC. Malgré l'existence de ces plateformes, la distribution d'un SMA reste une tâche complexe. En effet, la maîtrise des plateformes HPC nécessite une expertise spécialisée dans la programmation parallèle et les systèmes distribués. Ainsi, les plateformes permettant la distribution des modèles sont complexes et ne conviennent pas aux utilisateurs ayant peu ou pas d'expérience en informatique ou dans la distribution de SMA. A l'inverse, les plateformes traditionnelles de modélisation à base d'agents sont simples, faciles et accessibles aux utilisateurs de tous niveaux d'expérience et de diverses disciplines.

Étant donné que tous les SMA peuvent rencontrer des problèmes de mise à l'échelle, il nous paraît pertinent de proposer des solutions pour que les plateformes de distribution soient accessibles à tous les utilisateurs, indépendamment de leurs compétences et de leurs expériences dans ce domaine. La problématique suivante émerge alors: Comment construire une plateforme de distribution de SMA qui combine la facilité d'utilisation d'une plateforme SMA classique telle que GAMA ou NetLogo avec les fonctionnalités des plateformes de distribution de SMA telles que RepastHPC ou D-Mason?

Cet article propose le concept de *modèle de distribution*. Ce concept, basé sur le couplage de modèles, permet de distribuer facilement et efficacement un SMA grâce à la modélisation à base d'agents.

Dans la Section 2, nous analysons les lacunes et les faiblesses des plateformes de distribution de SMA existantes sous l'angle d'un modélisateur souhaitant distribuer un SMA. Dans la Section 3, nous présentons le concept de modèle de distribution et donnons les définitions précises des termes liés à ce concept tels que *modèle de distribution*, *modèle thématique* et *agent de distribution*. La Section 4 présente nos travaux pour implémenter les concepts de modèle de distribution et d'agents de distribution dans la plateforme GAMA. Dans la dernière section de cet article, nous revenons sur les premiers résultats présentés et nous ouvrons des perspectives pour des implémentations futures dans le domaine de la distribution de simulations multi-agents.

## 2 Distribution de modèles à base d'agents: perspectives d'amélioration de l'expérience utilisateur

Dans le domaine vaste de la modélisation et de la simulation à base d'agents [2] [16] [25], bien qu'il existe de nombreuses plateformes avec divers objectifs et caractéristiques, peu d'entre elles permettent de répartir facilement les simulations de modèles sur différents processus. Cette étude se concentre principalement sur la distribution de SMA, la parallélisation de modèles n'est pas son objectif principal.

La plupart des plateformes, telles que NetLogo [1], GAMA [24], Hash.ai<sup>1</sup>, Evoplex<sup>2</sup>, Jadex [6], Agentscript<sup>3</sup>, permettent l'exécution de modèles basés sur des agents dans un environnement de calcul centralisé, sans aucune capacité de distribution particulière. À l'inverse, certaines plateformes sont spécifiquement construites à des fins de distribution, comme FPMAS [7], PDES-MAS [23], Pandora [20], MASS [9], FAME-core [21], ou Care HPS [4]. D'autres plateformes, telles que DEVS-Suite [15], AnyLogic[5], Jade [3], SARL [18], bien que centralisées, offrent dès le départ la possibilité d'exécuter des modèles distribués dans des scénarios spécifiques. Enfin, bien qu'initialement conçues pour des environnements centralisés, certaines plateformes ont depuis proposé des évolutions leur permettant de distribuer les calculs d'une simulation, Flame avec FLAME GPU [17], Repast [14] avec RepastHPC [11] et Mason [13] avec D-MASON [12].

### 2.1 Distribution de SMA: tâche complexe et chronophage

À l'heure actuelle, la distribution d'un SMA nécessite des connaissances et une expérience approfondies dans les domaines de la distribution et de la parallélisation d'application. En effet, les plateformes de distribution de SMA imposent des exigences techniques élevées, qui peuvent être difficiles à maîtriser sans avoir les compétences requises. La distribution de SMA avec les plateformes disponibles

---

1. <https://hash.ai/>

2. <https://evoplex.org>

3. <https://agentscript.org>

peut ainsi s'avérer être un processus complexe et chronophage, ce qui peut décourager les modélisateurs de les utiliser pour distribuer leurs modèles.

Alban Rousset et al. [19] analysent différentes plateformes de calcul haute performance (HPC) existantes. Ils se sont intéressés, entre autres, aux langages utilisés. Fait intéressant, dans chacune de ces plateformes le langage de modélisation est identique au langage utilisateur permettant la distribution du modèle.

- RepastHPC [10]: C++/RLogo
- D-Mason [12], JADE [3]: Java
- PDES-MAS [23], FPMAS [8]: C/C++
- Flame GPU [17]: CUDA, C++, python3

Notons également que la complexité intrinsèque de ces langages de programmation peut constituer un obstacle conséquent pour les modélisateurs débutants, limitant ainsi l'accès à ces plateformes.

Bien que ces plateformes facilitent la distribution, une charge de travail conséquente est souvent laissée aux modélisateurs concernant la communication entre les processus, le partitionnement de l'environnement, la synchronisation des données, l'équilibrage de charge et la visualisation des données. Ces problématiques ne sont pas toujours liées à la modélisation à base d'agents, ce qui complique encore plus le processus de distribution pour un modélisateur non-expert en distribution.

Actuellement, ces plateformes HPC sont plus proches de bibliothèques que de réelles plateformes, comme NetLogo ou Gama qui accompagnent les modélisateurs et offrent de véritables IDE de modélisation. Une bibliothèque peut permettre une utilisation efficace des plateformes pour des utilisateurs expérimentés, mais cela peut également être moins intuitif pour les modélisateurs débutants qui n'ont pas une connaissance approfondie du fonctionnement interne de ces plateformes.

Pour faciliter la distribution des SMA, il est donc nécessaire de développer des plateformes plus simples à utiliser et moins exigeantes en termes de compétences techniques. Ces plateformes devraient également, dans la mesure du possible, intégrer davantage de fonctionnalités liées directement à la distribution de modèles, afin de réduire la charge de travail laissée aux modélisateurs.

## 2.2 Modéliser et distribuer: un amalgame?

Les plateformes de distribution de SMA imposent, la plupart du temps, aux modélisateurs de modifier le code source de leurs modèles pour y intégrer les instructions nécessaires à la distribution. Cette enchevêtrement des genres, code du modèle mélangé au code de distribution, rend plus difficile la maintenance et l'évolution des modèles.

L'intégration de fonctions de distribution dans un SMA nécessite de combiner le code source existant du modèle avec de nouvelles fonctions. Si ces dernières ne sont pas correctement utilisées, elles peuvent entraîner une modification des résultats voire des incohérences ou erreurs lors des simulations distribuées. L'évaluation de ce conglomérat hybride SMA/distribution est une tâche difficile qui nécessite une certaine expertise en distribution et en modélisation. En effet, les résultats de la simulation distribuée doivent être analysés avec soin pour identifier toutes les anomalies introduites par les fonctions de distribution.

Les SMA sont conçus pour être adaptables et évolutifs, car leurs propriétés et les comportements des agents peuvent être facilement modifiés pour ajouter de nouvelles fonctionnalités. Par exemple, les propriétés des agents peuvent être modifiées pour refléter des caractéristiques spécifiques du système étudié. Les comportements des agents peuvent également être modifiés pour prendre en compte de nouveaux facteurs, de nouvelles hypothèses ou données. Intégrer des fonctions de distribution directement dans la définition SMA rend la modification du système plus complexe. En effet, toute modification du système, même mineure, peut nécessiter une modification des fonctions de distribution pour prendre en compte ces nouveaux comportements, au risque d'introduire des erreurs.

Il existe donc un besoin réel de plateformes de distribution de SMA qui soient capables de distribuer des simulations sans nécessiter de modifications importantes du système pour y intégrer des fonctions de distribution. Pour cela, il faut définir une frontière claire entre la modélisation et la distribution d'un SMA afin d'éliminer la confusion existante et de faciliter la distribution de SMA.

À notre connaissance, seuls les travaux présentés dans [22] tentent de séparer la modélisation d'un SMA de sa distribution, seulement ces travaux sont surtout focalisés sur l'adaptation du déploiement du modèle à une infrastructure dynamique.

Notre approche se caractérise, par opposition aux autres travaux du domaine, par le découplage total de la distribution d'un modèle thématique et de sa définition. Notre approche se démarque par

l'utilisation du paradigme agent et de la modélisation pour décrire à la fois le déploiement du SMA à distribuer et le traitement des problématiques de la distribution interne au SMA.

Cet article présente une approche pour la distribution de SMA qui permet de découpler la modélisation du thème de la modélisation de la distribution du système. Cette approche offre une plus grande flexibilité dans la distribution de SMA et facilite grandement la distribution de SMA pour les non-experts grâce à la modélisation.

### 3 Modèle de distribution: découplage thème/distribution

Nous introduisons dans cet article le concept de modèle de distribution que nous définissons comme suit: *un modèle de distribution est un modèle composé d'un ensemble d'instances distribuées, collaborant dans le but de distribuer un modèle thématique*. On utilise le terme *modèle thématique* pour qualifier le SMA à distribuer.

Le modèle de distribution est un système multi-agents distribué dans lequel chaque instance collabore pour résoudre les problématiques de la distribution. Les instances du modèle de distribution collaborent en partageant des informations et en coordonnant leurs actions pour assurer la distribution du modèle thématique.

La Figure 1 montre qu'une instance de chaque modèle (distribution et thématique) se trouve dans chaque processus. Sur un processus, le modèle de distribution accède directement aux données du modèle thématique via une relation de co-modélisation. Les instances du modèle de distribution communiquent entre elles pour échanger des données.

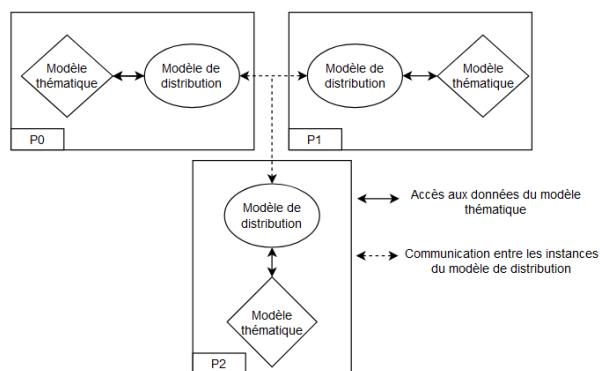


Figure 1 – Représentation de la topologie de la distribution d'un modèle thématique et d'un modèle de distribution. Une instance de chaque modèle se trouve sur chacun des processus

L'objectif premier d'un modèle de distribution est de permettre la séparation entre la définition du modèle thématique et la définition des fonctions liées à la distribution. Le fait d'avoir une distinction claire entre la partie modélisation et la partie distribution suit le principe de la séparation des responsabilités (*separation of concerns*). Cela donne au modélisateur plus de facilité à maintenir son projet avec deux aspects clairement identifiés: la définition de son SMA d'un côté et la distribution de l'autre.

Cet article continue en présentant comment est réalisé l'exécution des modèles, les différents agents du modèle de distribution, les mécanismes préservant la définition du modèle thématique lors de la distribution et les possibilités offertes par cette approche.

#### 3.1 Garder le modèle thématique intact

L'innovation première du modèle de distribution réside dans l'externalisation complète des fonctions de distribution et des algorithmes traitant les problématiques de la distribution qui, jusqu'à présent, étaient implémentés directement dans le modèle. Les solutions ne font plus partie du modèle thématique lui-même, mais sont définies et exécutées indépendamment dans le modèle de distribution, éliminant toute confusion. Cela se traduit par un code plus modulaire et facile à maintenir, avec d'un côté la distribution et de l'autre la thématique.

Du point de vue du modèle de distribution, le modèle thématique est un simple fournisseur de données. Les données ne sont accessibles que par des getters et setters, ce qui signifie que le modèle thématique n'a pas besoin de contenir de fonctions permettant la distribution de la simulation.

L'utilisation d'un modèle de distribution pour déployer un modèle thématique présente un autre avantage majeur: les deux modèles peuvent s'appuyer sur un langage de modélisation commun. L'utilisation d'un langage de modélisation commun pour les modèles thématiques et de distribution offre une solution élégante et efficace pour simplifier le processus de distribution pour les modélisateurs. Leurs compétences en modélisation peuvent directement être mises au service de la distribution de leur modèle.

Dans la section suivante, nous introduisons les agents de distribution en expliquant leurs fonctionnalités, leurs avantages et comment ils peuvent s'intégrer dans le modèle de distribution.

### 3.2 Constitution d'un modèle de distribution: Agents de distribution

Nous appliquons également le principe de la séparation des responsabilités à la définition des types d'agents peuplant les instances du modèle de distribution.

Le modèle de distribution est ainsi composé de différents types d'agents. Chaque type d'agent est responsable d'une problématique particulière dans la distribution d'un SMA.

Ces agents sont appelés *agents de distribution*:

- **Agent de communication:** agent chargé de la communication entre les différents processus dans la simulation distribuée.
- **Agent de partitionnement:** agent responsable du partitionnement de l'environnement/répartition des agents de la simulation.
- **Agent de synchronisation des données:** agent maintenant la cohérence des données lors d'interactions entre les agents dans la simulation distribuée.
- **Agent de visualisation:** agent chargé de transcrire l'état de la simulation distribuée sous une forme compréhensible pour l'utilisateur (sous forme de logs, de snapshot,...).
- **Agent d'équilibrage de charge:** agent permettant l'équilibrage de charge entre les processus disponibles.
- **Agent d'ordonnancement: agent assurant l'exécution et la synchronisation temporelle des instances du modèle de distribution.**

Les travaux de [22] introduisent un concept similaire d'« agents macro de distribution » qui se focalise sur le déploiement du modèle plutôt que de résoudre les problématiques internes de la distribution du modèle thématique comme proposé dans cet article. Notre agent d'ordonnancement se rapproche assez de la définition d'agents macro, cet agent va gérer l'ordonnancement, le déclenchement du modèle de distribution quand nécessaire et permet la synchronisation des différentes instances du modèle de distribution.

La Figure 2 schématisé le fonctionnement et la constitution d'un modèle de distribution capable d'assurer le découplage entre la modélisation et la distribution d'un SMA. Elle montre comment nous pouvons exporter le traitement des problématiques de la distribution en dehors du modèle thématique. Le modèle de distribution s'appuie sur les agents de distribution pour collecter les données du modèle thématique et les utiliser afin de distribuer correctement le modèle thématique grâce à des algorithmes distribués.

Ainsi, dans le modèle de distribution chaque problématique de distribution est gérée par des agents spécialisés. La composition de ce modèle est donc étroitement liée au modèle thématique, car les problématiques à gérer varient, voire disparaissent, selon le modèle thématique. La constitution du modèle de distribution est donc variable. Le modélisateur peut sélectionner les agents de distribution pertinents à utiliser pour distribuer son modèle thématique.

### 3.3 Algorithmes distribués au service de la distribution

Les modèles de distribution, par leur nature, favorisent l'élaboration d'algorithmes distribués. Lorsqu'un type d'agents est défini dans le modèle de distribution, un agent de ce type est créé dans chaque instance du modèle de distribution. Avoir un agent du même type sur chaque instance du modèle de distribution permet de mettre en œuvre des algorithmes distribués dans le but de traiter la problématique spécifique liée à ce type d'agent.

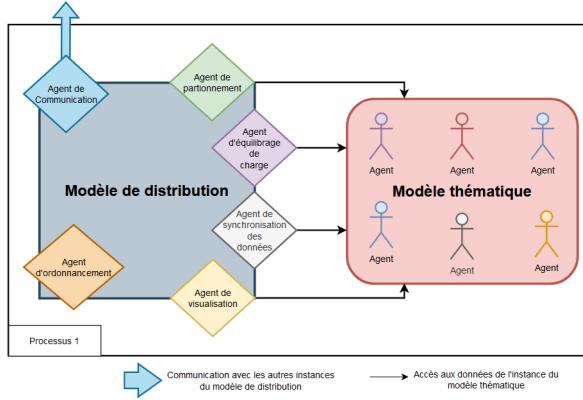


Figure 2 – Fonctionnement interne d'une instance du modèle de distribution

---

#### Algorithme 1 KMEAN distribué

---

```

function KMEANDISTRIBUÉ
    MAJPositionKMEANS()
    calculKMEANLePlusProche()
    echangeLesAgents()
    mettreAJourLaLocalisation()

```

---

L'algorithme 1 donne une implémentation possible d'un algorithme distribué visant à partitionner l'environnement. Cet algorithme peut être déployé directement depuis le modèle de distribution en utilisant les agents de partitionnement pour l'exécution. Chacun de ces agents va exécuter cet algorithme. L'algorithme K-MEANS est un algorithme de clusterisation, il fonctionne en partitionnant un ensemble de données/points en K groupes distincts, appelés clusters, basés sur la proximité des points/données. Cet algorithme peut être utilisé pour regrouper les agents spatialement proches dans le même processus.

Voici l'explication détaillée de l'algorithme 1 exécuté par un agent de partitionnement dans chaque instance du modèle de distribution:

- **MAJPositionKMEANS()** : chaque agent de partitionnement ou K-MEANS envoie sa position à tous les autres agents de partitionnement.
- **calculKMEANLePlusProche()** : pour chaque agent du modèle thématique sur le processus actuel, on détermine le K-MEANS dont il est le plus proche spatiallement.
- **echangeLesAgents()** : chaque K-MEANS envoie les agents du modèle thématique qui ne lui sont plus les plus proches localement au K-MEANS qui est désormais le plus proche d'eux.
- **mettreAJourLaLocalisation()** : le K-MEANS calcule la moyenne des coordonnées de tous les agents du modèle thématique qui lui sont affectés et se repositionne à ce point central.

L'implémentation de ces différents algorithmes distribués est grandement facilitée par le modèle de distribution. En effet, ce modèle offre plusieurs avantages clés:

- Le modèle de distribution masque la complexité de la répartition des données. Les modélisateurs utilisent les getters et setters depuis les agents de distribution pour accéder aux données de l'instance du modèle thématique sur leur processus.
- Le modèle de distribution fournit aux modélisateurs un environnement de travail simplifié pour la création d'algorithmes distribués grâce à la nature distribuée des agents de distribution.
- Le découpage entre la modélisation et la distribution d'un SMA, peut permettre au modélisateur de déléguer le développement d'algorithmes distribués à un expert dans ce domaine.
- La spécification d'un agent de distribution en fonction de la problématique qu'il traite permet de d'identifier clairement les rôles dans le modèle de distribution.

La suite de la section analyse comment le modélisateur peut influencer la définition des agents de distribution et les adapter aux besoins de son modèle thématique

### 3.4 Agents de distribution: flexibilité et collaboration

Nous proposons dans cet article un changement de paradigme avec l'introduction des agents de distribution. Ces agents gèrent le processus de distribution, offrant une approche plus flexible et centrée sur la gestion de la distribution grâce à la modélisation. Cela leur permet un meilleur contrôle sur la distribution des données, autorisant des optimisations et des personnalisations en fonction des exigences uniques de leur modèle thématique.

La nature distribuée des modèles/agents de distribution offre une architecture modulaire capable de s'adapter au modèle thématique. Ainsi le modélisateur, en fonction de ses besoins et compétences, a le choix entre trois niveaux pour son utilisation des agents de distribution:

- **Bibliothèque d'agents:** les modélisateurs ont accès à une bibliothèque d'agents prêts à l'emploi. Ces agents sont directement utilisables sans nécessiter de travail supplémentaire.
- **Personnalisation d'agents:** les modélisateurs peuvent modifier le comportement et les fonctionnalités d'un agent de la bibliothèque pour l'adapter à ses besoins spécifiques, ou créer des agents personnalisés en combinant des modules existants.
- **Développement avancé d'agents:** les modélisateurs expérimentés peuvent personnaliser entièrement la distribution de leur SMA grâce à la modélisation. Ils peuvent créer de nouveaux agents pour concevoir des solutions sur mesure. Ces contributions peuvent être partagées avec la communauté pour enrichir la bibliothèque d'agents.

En conclusion, le concept de modèle de distribution proposé offre une solution flexible pour la distribution de SMA. Les modèles/agents de distribution peuvent être réutilisés et étendus, ce qui, combiné à la possibilité d'utiliser des modèles/agents de distribution depuis une bibliothèque prédéfinie, permet de couvrir un large éventail de cas d'utilisation.

## 4 Implémentation d'agents de distribution dans la plateforme GAMA

Après avoir exploré les concepts fondamentaux du modèle de distribution, il est maintenant temps de s'intéresser à la mise en œuvre concrète de nos agents de distribution. Dans cette optique, la plateforme GAMA s'avère être un outil adapté. En effet, la plateforme GAMA [24] est un environnement de modélisation et de simulation open-source facile à utiliser pour créer des simulations spatialement explicites basées sur des agents.

Cette section de l'article présente les premiers résultats des travaux menés dans la plateforme GAMA pour implémenter le concept de modèle de distribution et d'agents de distribution. Nous proposons des solutions pour résoudre les problématiques de distribution en nous concentrant sur les agents de partitionnement et les agents de visualisation. Ces solutions ne sont pas nécessairement les meilleures ou les seules possibles, car l'adéquation dépend du modèle thématique à distribuer.

### 4.1 Partitionnement de l'environnement avec les agents de partitionnement

Nous présentons dans la suite l'implémentation de deux solutions de partitionnement sous la forme d'agents, un partitionnement avec l'algorithme K-MEANS et un partitionnement avec une grille.

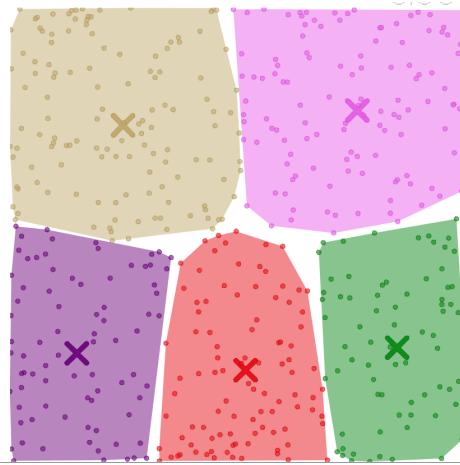


Figure 3 – Partitionnement spatial par l’algorithme distribué K-MEANS (Algorithme 1) partitionnant les agents entre 5 instances de modèle thématique

La figure 3 montre le résultat obtenu en partitionnant les agents d’un modèle thématique en se basant sur un agent de partitionnement qui implémente l’algorithme distribué K-MEANS (Algorithme 1). L’algorithme est appliqué dans un contexte où le modèle thématique (et donc le modèle de distribution) est distribué sur 5 processus distincts.

La figure 4 montre le partitionnement spatial d’un modèle thématique avec une grille à l’aide d’agents de partitionnement. L’environnement du modèle thématique est divisé en cellules, chacune de ces cellules et ses agents associés sont ensuite attribués à un processus. Les bandes rouges et vertes représentent des zones de recouvrement, appelées AOI (Area of Interest) ou OLZ (OverLapping Zone) dans la littérature. Ces zones de recouvrement assurent la cohérence spatiale entre les cellules distribuées. Elles permettent à des agents spatialement proches, mais logiquement éloignés (différents processus), d’interagir entre eux. Ce partitionnement est possible grâce à la modélisation de cette grille à travers les agents de partitionnement. On définit pour chaque agent de partitionnement les coordonnées de la cellule qu’il gère.

Le listing 1 montre la facilité avec laquelle nous pouvons configurer les différents aspects de la grille (largeur/longueur des cellules, nombre de colonnes, nombre de lignes, taille des OLZ...) grâce à la modélisation offerte par le langage GAML associé à la plateforme GAMA. L’originalité de cet exemple réside dans l’utilisation détournée du concept de grille ou **grid**<sup>4</sup> natif à la plateforme GAMA pour le partitionnement d’environnement pour un SMA distribué.

Listing 1 – Code GAML configurant la grille obtenue dans la Figure 4

```
global {
    int grid_width <- 4; // number of columns
    int grid_height <- 4; // number of rows
    int size_OLZ <- 5; // wideness of the OLZ
}
grid cell width: grid_width
height: grid_height neighbors: 4
```

Les deux méthodes de partitionnement, K-MEANS et grille, représentées par les Figures 3 et 4, possèdent chacune leurs points forts et leurs points faibles. Le partitionnement à base de K-MEANS est plus dynamique et s’adapte mieux à une simulation contenant des agents mobiles ne nécessitant pas beaucoup d’interaction. Le partitionnement à base de grille est plus statique et s’adapte mieux à une simulation contenant des agents peu ou pas mobiles nécessitant de pouvoir intéragir spatialement avec leur entourage.

Listing 2 – Code GAML permettant de choisir l’agents de partitionnement à utiliser. Nous pouvons donc choisir l’agent de partitionnement à instancier de cette façon.

```
global {
```

4. <https://gama-platform.org/wiki/GridSpecies>

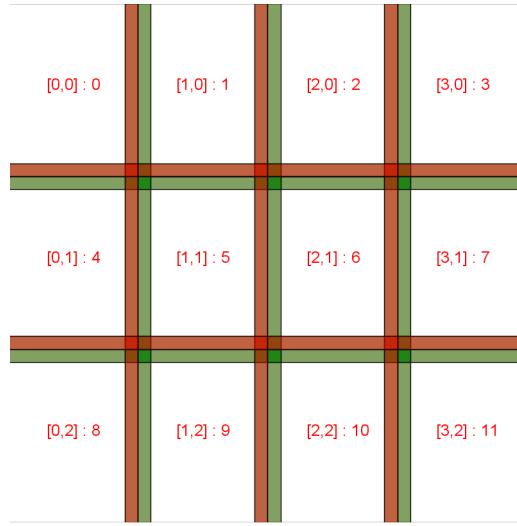


Figure 4 – Partitionnement spatial d'un modèle thématique grâce aux agents de distribution et à la modélisation. Chaque cellule de la grille représente l'espace exécuté par une des instances du modèle thématique. Les zones vertes et rouges représentent des zones de recouvrement ou OLZ.

```

// Choice 1: partitioning by K-MEANS
create partitioning_agent_Kmean;

// Choice 2: partitioning with a grid
create partitioning_agent_grid;
}

```

Le choix de l'agent de partitionnement à utiliser dans le modèle de distribution est laissé au modélisateur, c'est lui qui est le plus à même de savoir lequel de ces agents est le plus adapté à son ABM. Le listing 2 montre le code GAML permettant d'instancier un agent dans le modèle de distribution. C'est à cet endroit que le modélisateur à simplement à choisir d'instancier l'agent `partitioning_agent_Kmean` pour le partitionnement à base de K-MEANS ou l'agent `partitioning_agent_grid` pour un partitionnement avec une grille.

## 4.2 Visualisation de données avec les agents de distribution

La visualisation de données dans un contexte distribué n'est pas toujours une tâche simple du fait de plusieurs défis:

- **La fragmentation des données:** les données sont réparties sur plusieurs processus, ce qui rend difficile l'obtention d'une vue globale.
- **La cohérence des données:** il est important de s'assurer que les données visualisées sont cohérentes, même si elles proviennent de plusieurs processus différents.
- **La complexité de mise en place:** le déploiement d'une infrastructure de visualisation de données distribuées est complexe et exige des compétences avancées en programmation distribuée.

Cette section met en lumière comment les agents de visualisation peuvent résoudre la visualisation de données dans un SMA distribué en traitant les défis précédents.

Le listing 3 donne l'implémentation dans le langage GAML d'un algorithme distribué de visualisation de données. L'objectif de cet algorithme est de calculer la moyenne globale des valeurs dispersées entre les différents agents de visualisation. L'algorithme se déroule comme ceci:

1. Un agent de visualisation est désigné comme agent racine, cet agent est chargé de recevoir les moyennes locales des autres agents de visualisation et de calculer la moyenne globale.
2. Chaque agent de visualisation calcule la moyenne de ses données locales.
3. Chaque agent de visualisation (hors agent racine) envoie sa moyenne locale à l'agent racine.
4. L'agent racine reçoit toutes les moyennes locales.

## 5. L'agent racine calcule la moyenne globale du SMA distribué.

Listing 3 – Implémentation d'un algorithme de calcul de moyenne distribué avec un agent de visualisation dans la plateforme GAMA.

```

species visualisation_agent {
    // MPI rank of the processor
    int rank;
    // values on this processor
    list<int> values;
    // root rank
    int root <- 0;

    init{
        // rank of the processor
        rank <-MPI_RANK;
        // random values
        values <- [rnd(10), rnd(20), rnd(30),
                    rnd(40), rnd(50), rnd(60),
                    rnd(70), rnd(80), rnd(90)];
    }

    reflex compute_result
    {
        // local average for this processor
        float local_average <-mean(values);

        if(rank = root){
            // receive on root
            list<float> average_from_every_proc <-
                MPI_GATHER(local_average, root);
            // compute final value
            float global_average <-
                mean(average_from_every_proc);
        } else{
            // send local_average to root
            do MPI_GATHER(local_average, root);
        }
    }
}

```

Le listing 3, comme le listing 1 reposent entièrement sur les fonctions de modélisation et sur l'architecture de GAMA à travers le langage GAML<sup>5</sup>. Dans notre cas, les méthodes **mean**, **rnd**, **MPI\_GATHER** sont utilisées pour la communication et les fonctions manipulant les données. Les concepts **species**, **reflex** permettent la création d'agents et l'exécution de méthodes. Aucune implémentation supplémentaire dans le langage GAML n'a été nécessaire pour réaliser nos exemples.

L'implémentation du listing 3 montre le potentiel des agents de visualisation. Indépendant des fonctions prédéfinies, l'agent de visualisation pourrait intégrer aisément de nouvelles méthodes de calcul (SOMME, MIN, MAX, etc...) ou tout autre traitement de données spécifique aux besoins du modèle thématique.

La complémentarité entre les techniques de distribution et la modélisation, comme le démontrent ces exemples, facilite le déploiement d'algorithmes distribués pour les problématiques SMA, sans compromettre la transparence du modèle thématique.

## 5 Conclusion

Dans cet article, nous proposons le concept de modèle de distribution. Ce concept permet de découpler la modélisation et la distribution d'un modèle avec d'un côté le modèle thématique (modélisation) et de l'autre le modèle de distribution (distribution). Dans les modèles de distribution, des agents de distribution dédiés s'attaquent aux défis spécifiques de la distribution: visualisation de données, partitionnement, synchronisation des données, etc...

5. <https://gama-platform.org/wiki/Home>

Nous avons validé nos concepts avec une implémentation de plusieurs agents de distribution dans la plateforme GAMA. Un agent de visualisation spécifiques permettant de visualiser/calculer les métriques du modèle thématique distribué. Des agents de partitionnement spécifique, partitionnant l'espace avec une grille ou avec un algorithme K-MEANS distribué.

Notre travail futur portera sur l'implémentation et la définition complète de modèles/agents de distribution au sein de la plateforme GAMA. Grâce à ces modèles, nous constituerons une bibliothèque complète accessible à tous les utilisateurs de la plateforme.

## References

- [1] Wilensky, u.(1999).NetLogo. Center for Connected Learning and Computer-Based Modeling, North-western University, Evanston, IL.
- [2] S. Abar, G. Theodoropoulos, P. Lemarinier, and G.. O'Hare. Agent Based Modelling and Simulation tools: A review of the state-of-art software. *Computer Science Review*, 24:13–33, May 2017.
- [3] F. Bellifemine, G. Caire, and D. Greenwood. *Developing Multi-Agent Systems with JADE*. Wiley, 1 edition, March 2007.
- [4] F. Borges, A. Gutierrez-Milla, E. Luque, and R. Suppi. Care HPS: A high performance simulation tool for parallel and distributed agent-based modeling. *Future Generation Computer Systems*, 68:59–73, March 2017.
- [5] A. Borschkev. *The Big Book of Simulation Modeling: Multimethod Modeling with AnyLogic 6*. AnyLogic North America, 2013.
- [6] L. Braubach and A. Pokahr. The Jadex Project: Simulation. In Maria Ganzha and Lakhmi C. Jain, editors, *Multiagent Systems and Applications*, volume 45, pages 107–128. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013. Series Title: Intelligent Systems Reference Library.
- [7] P. Breugnot, B. Herrmann, C. Lang, and L. Philippe. A Synchronized and Dynamic Distributed Graph structure to allow the native distribution of Multi-Agent System simulations. In *2021 29th Euromicro International Conference on Parallel, Distributed and Network-Based Processing (PDP)*, pages 54–61, Valladolid, Spain, March 2021. IEEE.
- [8] P. Breugnot, B. Herrmann, C. Lang, and L. Philippe. Data Synchronization in Distributed Simulation of Multi-Agent Systems. October 2022.
- [9] T. Chuang and M. Fukuda. A Parallel Multi-agent Spatial Simulation Environment for Cluster Systems. In *2013 IEEE 16th International Conference on Computational Science and Engineering*, pages 143–150, Sydney, Australia. IEEE.
- [10] N. Collier and M. North. Repast HPC: A Platform for Large-Scale Agent-Based Modeling. In Werner Dubitzky, Krzysztof Kurowski, and Bernhard Schott, editors, *Large-Scale Computing*, pages 81–109. John Wiley & Sons, Inc., Hoboken, NJ, USA, April 2012.
- [11] N. Collier and M. North. Parallel agent-based simulation with Repast for High Performance Computing. *SIMULATION*, 89(10):1215–1235, October 2013.
- [12] G. Cordasco, R. De Chiara, F. Raia, V. Scarano, C. Spagnuolo, and L. Vicidomini. Designing Computational Steering Facilities for Distributed Agent Based Simulations. May 2013.
- [13] S. Luke, C. Cioffi-Revilla, L. Panait, K. Sullivan, and G. Balan. MASON: A Multiagent Simulation Environment. *SIMULATION*, 81(7):517–527, July 2005.
- [14] M. North, N. Collier, J. Ozik, E. Tatara, C. Macal, M. Bragen, and P. Sydelko. Complex adaptive systems modeling with Repast Simphony. *Complex Adaptive Systems Modeling*, 1(1), 2013.
- [15] F. Orooji, H. Sarjoughian, and F. Taghiyareh. Modeling & simulation of educational multi-agent systems in DEVS-suite. pages 956–961, January 2011.
- [16] C.-V. Pal, F.n Leo, M. Paprzycki, and M. Ganzha. A review of platforms for the development of agent systems. *ArXiv*, abs/2007.08961, 2020.
- [17] P. Richmond, D. Walker, S. Coakley, and D. Romano. High performance cellular level agent-based simulation with FLAME for the GPU. *Briefings in Bioinformatics*, 11(3):334–347, May 2010.

- [18] S. Rodriguez, N. Gaud, and S. Galland. SARL: a general-purpose agent-oriented programming language. In *the 2014 IEEE/WIC/ACM International Conference on Intelligent Agent Technology*, Warsaw, Poland, 2014. IEEE Computer Society Press.
- [19] A. Rousset, B. Herrmann, C. Lang, and L. Philippe. A survey on parallel and distributed multi-agent systems for high performance computing simulations. *Computer Science Review*, 22:27–46, 2016.
- [20] X. Rubio-Campillo. Pandora: A Versatile Agent-Based Modelling Platform for Social Simulation. 2014.
- [21] C. Schimeczek, M. Deissenroth-Uhrig, U. Frey, B. Fuchs, A. Ghazi, M. Wetzel, and K. Nienhaus. FAME-Core: An open Framework for distributed Agent-based Modelling of Energy systems. *Journal of Open Source Software*, 8(84):5087, April 2023.
- [22] Nicolas Sébastien. *Distribution et Parallélisation de Simulations Orientées Agents*. Theses, Université de la Réunion, November 2010.
- [23] G. Suryanarayanan, V. and Theodoropoulos and M. Lees. PDES-MAS: Distributed Simulation of Multi-agent Systems. *Procedia Computer Science*, 18:671–681, January 2013.
- [24] P. Taillandier, B. Gaudou, A. Grignard, Q.-N. Huynh, N. Marilleau, P. Caillou, and A. Philippon, D. and Drogoul. Building, composing and experimenting complex spatial models with the GAMA platform. *GeoInformatica*, 23(2):299–322, April 2019.
- [25] Z. Wrona, W. Buchwald, M. Ganzha, M. Paprzycki, F. Leon, N. Noor, and C.-V. Pal. Overview of Software Agent Platforms Available in 2023. *Information*, 14(6):348, June 2023. Number: 6 Publisher: Multidisciplinary Digital Publishing Institute.