

Chapitre 9

Spécification conceptuelle *MR POTATOHEAD - Property Market Edition* du système complexe d'un territoire touristique à deux marchés : application au territoire corse

Eric Innocenti, Claudio Detotto, Corinne Idda, Dawn Cassandra Parker
et Dominique Prunetti

1

9.1 Introduction

Nous abordons dans ce chapitre la problématique de la simulation informatique des systèmes complexes de *changement d'utilisation et de couverture des sols* (*Land-Use and Land-Cover Change - LUCC*). L'étude des *LUCC* a pour principal objectif d'améliorer la compréhension des problèmes et conséquences liés aux changements d'occupation et d'usage des sols sur des territoires en interaction avec les sociétés humaines. Cette problématique constitue un enjeu majeur pour la communauté scientifique dans le contexte du changement climatique global que nous subissons actuellement. Nous œuvrons dans ces travaux de recherche :

DOI : <https://doi.org/10.58110/estate-bg32>

1. Ce chapitre est une version remaniée de l'article suivant : Innocenti, E., Detotto, C., Idda, C., Parker, D.C., Prunetti, D., 2020. « An iterative process to construct an interdisciplinary ABM using MR POTATOHEAD : an application to Housing Market Models in touristic areas », *Ecological Complexity*, 44, December 2020, 100882. (<https://doi.org/10.1016/j.ecocom.2020.100882>).

- A soutenir l'*économie computationnelle*², un nouveau domaine de recherche interdisciplinaire qui se développe depuis de nombreuses années maintenant en science économique ;
- À produire des bibliothèques génériques objets de calcul (modèles exécutables) fondées sur des formalismes et des concepts novateurs reconnus, dédiées à l'expérimentation par la simulation informatique.

Le contexte applicatif de ces travaux de recherche est la Corse qui est l'objet de fortes pressions sur les prix des marchés foncier et immobilier. En effet, une économie touristique s'y développe depuis les années 1970 et se caractérise par la construction d'infrastructures destinées à accueillir un nombre important de touristes (hôtels, bars, restaurants, etc.), ainsi que par la construction de résidences secondaires qui ne cessent de progresser. Depuis l'année 2012, on observe également une forte croissance démographique due au solde migratoire. La géographie bornant de fait les possibilités de construction à moyen terme, cela conduit inexorablement le territoire de la Corse vers des mutations d'usage de son espace insulaire, ainsi que vers des recompositions sociales.

Dans ce chapitre, nous spécifions un modèle conceptuel de type *Agent-Based Model / Land-Use and Land-Cover Change* - (*ABM/LUCC*). La spécification formelle *MR POTATOHEAD - Property Market Edition* - (*MPH-PME*) est utilisée (Parker et al., 2019 ; Parker, 2014 ; Parker et Filatova, 2008). Il s'agit d'une spécialisation de la spécification formelle originale *Model Representing Potential Objects That Appear in The Ontology of Human - Environmental Actions & Decisions* - *MR POTATOHEAD* - (*MPH*) pour les *Agent-Based Model / Land Market Model* (*ABM/LMM*), une spécialisation croissante des *ABM/LUCC* qui intègre une dynamique de marchés. Nous présentons dans ce chapitre comment il convient de l'utiliser pour simuler le système complexe d'un territoire touristique à deux marchés (foncier et immobilier) dans un *ABM/LMM*.

Dans une deuxième partie, nous présentons le processus de modélisation itératif simplifié en quatre étapes *4-Step Simplified Iterative Modelling Process* (*4-SSIMP*) articulant la mise en œuvre de ce travail interdisciplinaire de simulation informatique. Nous rappelons les concepts-clés du *paradigme agent*, ainsi que les formalismes de spécifications conceptuels *MPH* et *MPH-PME* à la base de la construction de ce type de modèles conceptuels.

Dans une troisième partie, nous présentons un exemple de modélisation conforme à la spécification conceptuelle *MPH-PME* pour les *ABM/LMM*. Nous l'utilisons pour spécifier le système complexe des marchés foncier et immobilier du territoire insulaire corse. Le modèle informatique déduit est organisé conformément au *Design Pattern NetLogo*, l'objectif ici étant de produire rapidement un premier prototype de calibration (modèle exécutable), facilement vérifiable et validable sur l'infrastructure de simulation *NetLogo*.

Enfin, dans une quatrième et dernière partie, nous concluons ce travail préliminaire de simulation informatique, avec pour principal objectif de procéder ultérieurement à des plans d'expériences.

2. *Agent-based Computational Economics* (*ACE*) en anglais.

9.2 Paradigmes, formalismes et concepts

9.2.1 Le processus de modélisation 4-SSIMP

Le processus de modélisation d'un système complexe a pour objet de produire un modèle exécutable de la dynamique d'un phénomène (économique) observé et de l'exécuter sur l'outil informatique (ordinateur, supercalculateur, grappes de calcul, etc.). A cette fin, le modèle exécutable est l'expression numérique d'un ensemble de constituants, de leur organisation et de leur interaction au sein de différents niveaux d'abstraction d'un modèle conceptuel. Le processus de modélisation 4SSIMP (*4-Step Simplified Iterative Modelling Process*) a été bâti selon cette base philosophique. Celle-ci pose le questionnement des échelles descriptives et de l'organisation des constituants d'un système complexe hiérarchisé, comme préalable dans un modèle conceptuel de composition et multi-échelle. Eu égard au contexte interdisciplinaire de ce travail, la formulation du modèle conceptuel doit se faire selon un cadre méthodologique scientifique suffisamment simple, néanmoins précis. Celui-ci prend corps dans un processus de modélisation réduit à quatre étapes (*4-SSIMP*) que nous avons développé et qui constitue la pierre angulaire articulant les tâches de ce travail. La figure 9.1 est une illustration du processus *4-SSIMP*.

Dans le processus de modélisation *4-SSIMP* de la figure 9.1, la complexité est introduite de manière progressive et à chaque itération. Le but est de produire des codes objets (modèle exécutable) dont le fonctionnement est conforme à la réalité du système complexe observé. L'étude numérique portant sur le système complexe initial s'effectue lors de l'ultime étape de simulation (phase de simulation) pendant laquelle des expériences scientifiques sont menées *in silico* dans le modèle exécutable. Les données obtenues lors de la phase de simulation sont utilisées à des fins d'études et de synthèses, notamment pour aider les prises de décision d'ordre politique. Cela fait maintenant de nombreuses années que le débat portant sur le processus de modélisation existe dans la communauté scientifique, notamment à cause de son caractère critique. En effet, devant l'infinité des choix possibles quant aux hypothèses simplificatrices de modélisation qu'émettent les modélisateurs, il peut exister une infinité d'abstractions, c'est-à-dire autant de modèles conceptuels que de méthodes de conceptualisation différentes, et cela pour l'observation d'un même système complexe. Ainsi, comme le souligna Robinson (2015) lors d'une conférence sur la simulation, la modélisation est encore aujourd'hui davantage un « *art, plutôt qu'une science* »³. C'est dans ce contexte que la science de la simulation informatique acquit ces dernières années davantage de méthode et de rigueur. Elle peut être maintenant considérée comme une science à part entière, fédérant des chercheurs voulant aborder les sciences de manière transdisciplinaire, en utilisant des outils issus des mathématiques, de la physique et de l'informatique, mais aussi des sciences sociales et du vivant.

3. « *conceptual modeling is more an art than a science.* »

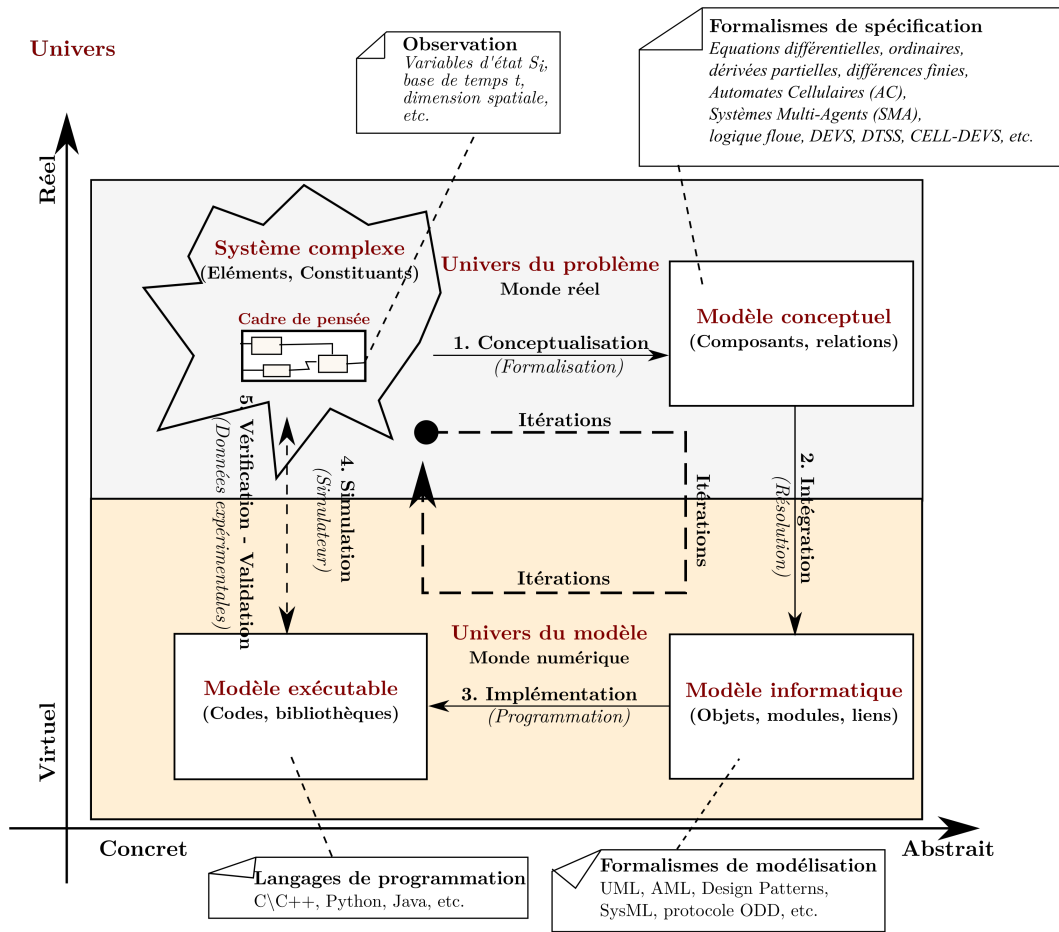


FIGURE 9.1 — Le processus de modélisation 4-SSIMP d'un système complexe. Ce processus est une simplification des travaux précurseurs de Siegfried (2014) ; Chwif et al. (2013) ; Robinson (2013) ; Balci (2011) ; Mitroff et al. (1974)

La phase de conceptualisation

La première étape du processus de modélisation simplifié en 4 étapes (*4-SSIMP*) consiste à définir une *représentation simplifiée* du système complexe sous la forme d'une *collection d'entités* et de *constituants* caractérisant le *phénomène* d'intérêt du projet de simulation (Menner, 1995 : p.7.). Il s'agit d'un objet mathématique constituant le modèle abstrait⁴ (conceptuel) qui comme le souligna *Marvin Lee Minsky*⁵ en 1965, correspond davantage à un *paradigme général de pensée* faisant consensus dans le domaine. À ce sujet, celui-ci énonça que « pour un observateur B , un objet A^* est un

4. En général, un modèle n'existe pas en tant que tel, mais correspond plutôt à l'une des multiples constructions définies dans différentes phases du processus de modélisation. Ici, nous faisons clairement le choix de ne considérer que la seule phase de conceptualisation. Voir à ce propos Siegfried (2014) : p.13.

5. *Marvin Lee Minsky* (1927-2016) est notamment connu pour ses travaux sur l'intelligence artificielle.

modèle d'un objet A s'il permet à B de répondre à une question qu'il se pose sur A »⁶ (Minsky, 1965). Cette définition limite de fait les constituants du système complexe à considérer à ceux d'intérêt pour la problématique de l'étude. Cela implique en préalable la formulation d'hypothèses simplificatrices, c'est-à-dire d'*hypothèses (descriptives) de modélisation*. Ainsi, le modèle conceptuel constitue ce préalable qui sert à formaliser des constituants pertinents du système complexe, leurs relations et leur organisation au sein d'une entité abstraite consensuelle, dans le but de répondre à une question de simulation.

La phase d'intégration

La *phase d'intégration* traite de la prise en compte des contraintes techniques liées à l'usage des outils numériques (ordinateurs, calculateurs, etc.). Les modélisateurs interviennent dans cette phase en tant qu'*intégréteurs*. Leur travail consiste ici à construire et interconnecter des objets et des ensembles d'objets (*modules*) au sein d'une entité unique appelée *modèle informatique*, conformément aux composants et aux relations définis préalablement dans le modèle conceptuel. Pour cela, les modélisateurs qui interviennent dans le *4-SSIMP* doivent en général posséder des compétences en *informatique appliquée*⁷. Dans le contexte des systèmes complexes spatialisés, le *modèle informatique* est qualifié de modèle de composition, hiérarchique et multicomposant (multi-échelles). Les méthodes d'intégration associées à la phase d'intégration servent à traduire le *modèle conceptuel* et ses états issus du *monde réel* (variables continues), en un modèle informatique à états discrets adapté au *monde numérique* (variables discrètes) et à ses circuits électroniques. Ainsi, la formulation du *modèle informatique* repose la plupart du temps sur les fondements des *mathématiques discrètes*⁸ et nécessite d'être associée au paradigme informatique de la *Programmation Orientée Objet*⁹. C'est la raison pour laquelle, la *science mathématique* initia la première ce travail d'intégration des modèles conceptuels car elle fournit notamment les équations différentielles comme premiers instruments d'intégration (Zill, 2012). Il s'agit alors d'utiliser des *méthodes de résolution* pour traduire les objets mathématiques du *modèle conceptuel* en un *modèle informatique* numérique.

La phase d'implémentation

La *phase d'implémentation* est la dernière étape traitant de la modélisation du système complexe. Il s'agit de produire le code du *modèle exécutable* à l'aide d'un

6. Dans sa version originale : *To an observer B , an object A^* is a model of an object A to the extent that B can use A^* to answer questions that interest him about A .*

7. En informatique appliquée l'accent est mis sur la mise en pratique des connaissances plutôt que la recherche (voir, entre autres, Hoganson (2008)).

8. Voir à ce propos l'ouvrage de Biggs (2008).

9. Aussi appelée « O.O.P. » (*Object-Oriented Programming*) en anglais.

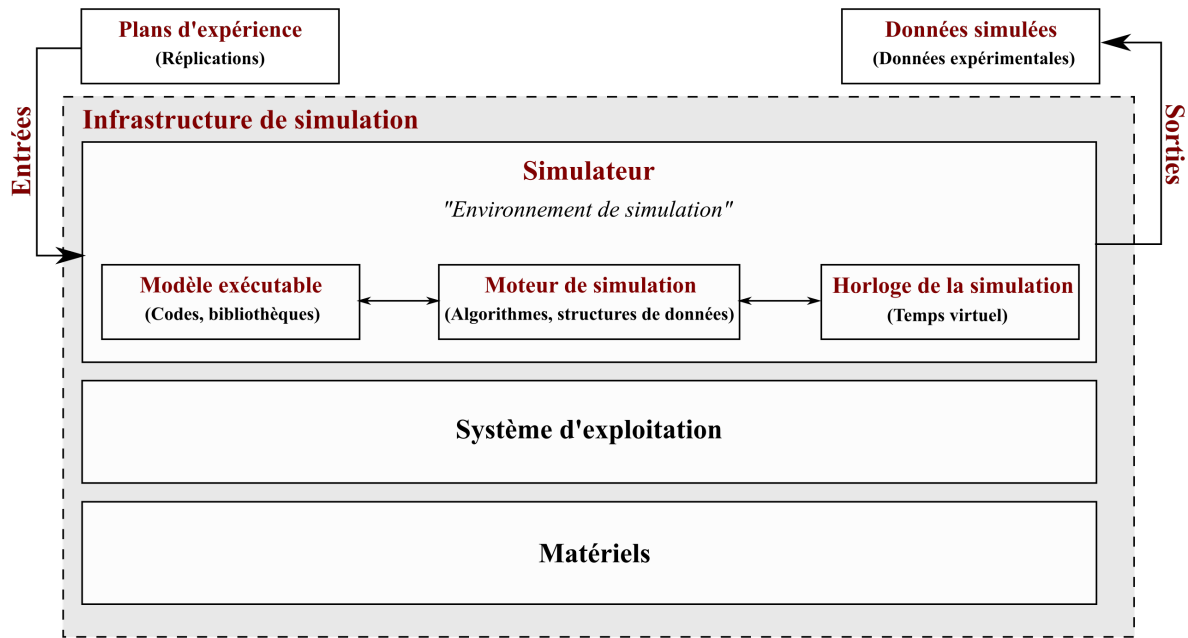


FIGURE 9.2 — Les éléments constitutifs d'une infrastructure de simulation.

langage de programmation (orienté objet la plupart du temps¹⁰), en respectant le schéma numérique des objets et des modules décidés dans le *modèle informatique* lors de la phase d'intégration. Le *modèle exécutable* est le programme informatique en charge de produire les *données de simulation* qui sont utilisées pour répondre à la question de recherche posée initialement. Le code du *modèle exécutable*, conforme au *modèle informatique*, est *modulaire* ce qui garantit l'amélioration progressive des codes effectuée lors des différentes itérations du processus de modélisation ; ainsi que la *réutilisabilité* et le *partage* des mêmes codes informatiques entre différents projets de simulation. Un *modèle exécutable* ainsi conçu de manière modulaire est également particulièrement utile quand il s'agit de réaliser des bibliothèques de modèles.

La phase de simulation

La *phase de simulation* ou *simulation* a pour objet la mise en œuvre de la *dimension temporelle* du processus de modélisation. Il s'agit d'intégrer le *modèle exécutable* dans un *simulateur* et de générer des *données simulées* ou *expérimentales*. Pour cela, un *moteur de simulation* doit être sélectionné. Celui-ci implémente des modules objets adaptés en charge de faire évoluer le *modèle exécutable* dans le *temps virtuel* $t \in T$ de l'*horloge de la simulation*. Les éléments constitutifs d'une infrastructure de simulation sont illustrés sur la figure 9.2.

10. En fait, même si le langage de programmation n'est pas orienté objet, il est toujours possible de mettre en œuvre techniquement les principes de la P.O.O.

Il existe dans la littérature différents *moteurs de simulation* qui dans le cas de la mise en œuvre de *simulations parallèles et distribuées*, savent notamment exploiter de manière optimale les matériels informatiques sous-jacents. Dans la littérature, une description rigoureuse des moteurs de simulation basée sur l’usage de formalismes de spécification, prouve la *portabilité* ainsi que l’*interopérabilité* de nombreux codes de simulation (Zeigler et al., 2000). Le lecteur intéressé trouvera une description détaillée des *moteurs de simulation* les plus courants dans les ouvrages Coquillard et Hill (1997) ; Erard et Déguénon (1996) ; Jerry (1984). Le simulateur, le système d’exploitation et les matériels informatiques constituent l’*infrastructure de simulation* qui sert à expérimenter au sein d’un projet de simulation. Cette infrastructure est fortement dépendante des choix conceptuels effectués par les modélisateurs lors de la phase de conceptualisation.

Nos choix conceptuels sont fondés sur le *paradigme agent*. Nous détaillons ses principes de modélisation dans la section suivante.

9.2.2 Modèles conceptuels à base d’agents

Un *modèle à base d’agents* (ABM : *Agent-Based Model*) est un modèle conceptuel construit selon le paradigme générique du *Système Multi-Agents* (SMA¹¹) (Klügl et Bazzan, 2012 : entre autres). Un *ABM* consiste en une représentation abstraite du système complexe dans lequel de nombreuses entités clairement identifiées, appelées « *agents* », interagissent entre elles (niveau micro), ainsi qu’avec un environnement global (niveau macro), et selon un ensemble de règles comportementales bien précises (voir Clarke (2018) ; Ferber et Weiss (1999)).

Ce regroupement d’entités constitue un système à base d’agents (ABS : *Agent-Based System*), dont les exemples les plus courants sont les *modèles individus-centrés* (IBM : *Individual Based Models*) développés pour des applications en sciences du vivant (voir Grimm et Railsback (2006) pour plus de détails). La définition d’un *ABM* nécessite de conceptualiser dans le détail, des agents, leurs interactions au niveau micro, et les interactions avec l’environnement au niveau macro (cf. figure 9.3).

En Science Economique, les formulations d’*ABM* complètent les approches traditionnelles de la modélisation économique en permettant de mieux relier les échelles micro et macro du système complexe étudié (Hamill et Gilbert, 2016). Au cours de ces dernières années, les simulations informatiques à partir d’*ABM* et d’*ABM/LUCC*, ont énormément évolué en termes de paradigmes et de concepts, en se fondant également sur les développements récents du matériel informatique. Cela encourage d’autant plus les modélisateurs à développer des modèles conceptuels à base d’agents toujours plus complexes (Grignard et al., 2013). Ainsi, Parker et Filatova (2008) soutiennent que de tels modèles devraient également inclure des représentations des marchés foncier et immobilier, arguant que les *ABM/LUCC*, ainsi spécialisés, sont en capacité de mieux saisir la complexité de la réalité que les modèles économiques analytiques traditionnels. Elles actent ainsi la spécification d’*ABM/LUCC* spécialisée dans la description de dy-

11. *Multi-Agent Systems (MAS)* en anglais.

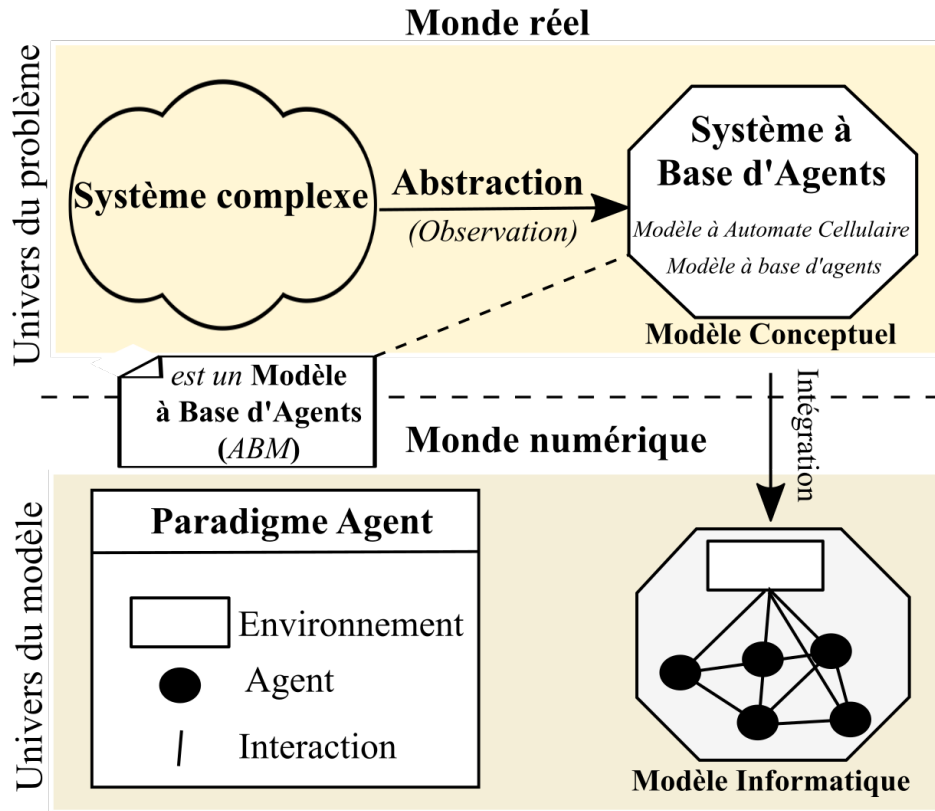


FIGURE 9.3 – Formulation d'un modèle à base d'agents selon le point de vue SMA.

namiques de marchés économiques. La spécification conceptuelle *MR POTATOHEAD* s'inscrit dans ce contexte d'étude.

9.2.3 La spécification conceptuelle MR POTATOHEAD

La spécification conceptuelle *MR POTATOHEAD* - (*MPH*) fut initialement proposée par Parker et al. (2008a). Celle-ci sert à caractériser précisément les composants et les relations que l'on rencontre habituellement dans un *ABM/LUCC*. Elle sert à définir précisément les concepts-clés qu'il faut manipuler par la pensée, pour construire le modèle conceptuel d'un *ABM/LUCC* lors la phase de conceptualisation (cf. Figure 9.4).

En simulation informatique, elle prend forme d'une *ontologie informatique*¹² pour les *ABM/LUCC*, dans le but de remédier à la problématique de la formulation des hypothèses simplificatrices, lors de la phase de conceptualisation (Robinson, 2008).

La spécification conceptuelle *MR POTATOHEAD* vise ainsi à faciliter la conception des modèles conceptuels et informatiques, préalable indispensable au développement de

12. « Une ontologie informatique est une représentation de propriétés générales de ce qui existe dans un formalisme supportant un traitement rationnel. » (Gandon, 2006 : p.2)

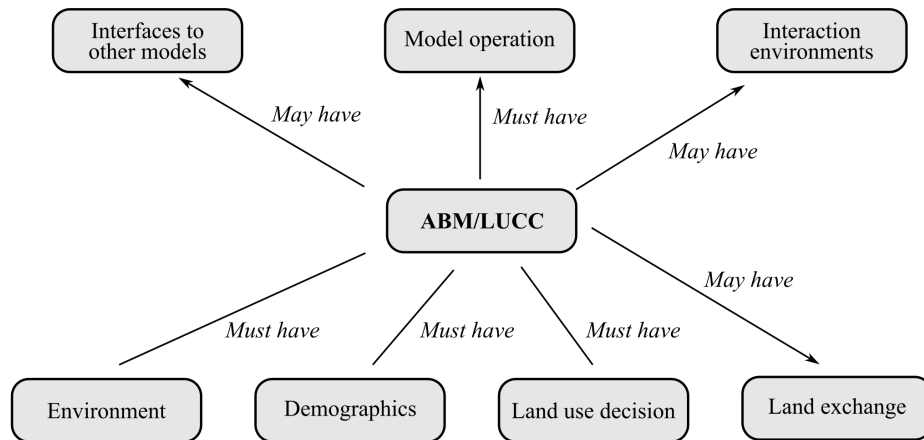


FIGURE 9.4 — Les principaux concepts-clés haut niveau d'un ABM/LUCC se conformant à la spécification MR POTATOHEAD (Parker, 2014 :p. 407).

modèles exécutables fiables (vérifiables et validables). Pour cela, elle fournit les ontologies adéquates pour formuler les composants d'un modèle conceptuel (*ABM/LUCC*) que l'on traduit sous la forme d'objets spécialisés dans le modèle informatique. La spécification conceptuelle *MR POTATOHEAD* complète en outre les protocoles de description normalisés existant dans le domaine de la simulation informatique, tels que *Unified Modeling Language (UML)* (Ambler, 2005), utilisé en *Programmation Orientée Objet (P.O.O.)* ; *Agent-Based Modelling Language (AML)* pour les *ABM* ; et *Overview, Design concepts, and Details (ODD)*, ce dernier protocole étant davantage une méthode de description pour les projets de simulation d'*ABM* (voir notamment Grimm et al. (2010)).

Dans le cadre d'un projet de simulation d'un système *LUCC*, ces outils de formalisation à destination des modélisateurs sont beaucoup trop génériques. Ils ne fournissent pas d'éléments de documentation adaptés au contexte interdisciplinaire de la conceptualisation d'un système complexe de type *LUCC*, ce qui nuit fortement à la communication entre les modélisateurs intervenant dans le projet de simulation. En outre, ils ne fournissent pas d'éléments de documentation pouvant garantir la reproductibilité des codes objets qui sont produits. La comparaison des modèles *ABM/LUCC* et des expériences de simulation associées n'est ainsi pas possible.

Dans ce contexte d'interdisciplinarité impliquant des *ABM/LUCC*, la spécification conceptuelle *MR POTATOHEAD* permet de surmonter ces lacunes. Elle offre un protocole de documentation générique et efficace, garantissant la communication entre les modélisateurs des différentes disciplines. Son usage facilite, en outre, la comparaison des modèles *ABM/LUCC* et la reproductibilité des expériences de simulation associées. À titre d'exemple, dans Parker et al. (2008b), les auteurs comparent différentes implémentations d'*ABM/LUCC* via l'analyse de cinq études de cas.

9.2.4 MR POTATOHEAD - Property Market Edition

Comme décrit plus en détail dans Parker et al. (2019), la spécification conceptuelle *MR POTATOHEAD* a récemment été mise à jour et étendue afin de pouvoir intégrer la dynamique particulière des marchés économiques. Les modèles conceptuels formulés sur la base de la spécification formelle *MR POTATOHEAD - Property Market Edition* (*MPH-PME*) sont qualifiés d'*Agent-Based Model/Land Market Model* (*ABM/LMM*). Il s'agit d'une évolution de la spécification *MPH* issue de deux ateliers internationaux, auxquels nous avons participé. Durant ces travaux, les ontologies servant à formuler les composants génériques d'un *ABM/LMM* ont été définies par les participants : l'objectif ultime étant de pouvoir spécifier un *ABM/LMM* générique et son modèle exécutable correspondant. Il faudra pour cela fournir dans un premier temps une base de code générique pour les systèmes complexes de marchés foncier et immobilier de territoires touristiques.

9.3 Spécification des marchés foncier et immobilier du territoire insulaire corse

Nous présentons dans cette partie la manière d'utiliser la spécification conceptuelle *MPH-PME* pour modéliser le système complexe du territoire touristique à deux marchés de la Corse.

9.3.1 Conceptualisation

Les *ABM* présentent divers avantages pour le développement de modèles plus proches des systèmes économiques du monde réel. Notamment, ils permettent de prendre en compte des agents hétérogènes, des interactions spatiales et entre agents, ainsi que des dynamiques hors équilibre.

Le modèle développé par Filatova et al. (2009a) comprend des interactions entre plusieurs types d'agents, le processus de formation des prix acheteur-vendeur, et des modèles de décision des agents combinant l'optimisation déductive et des modèles inductifs de formation des anticipations de prix.

En s'appuyant sur ce domaine de recherche, nous proposons un modèle économique basé sur deux marchés, l'un pour les terrains et l'autre pour les habitations. Le programme de maximisation de l'utilité des ménages est basé sur les modèles de Parker et Filatova (2008), Filatova et al. (2009a) et Filatova et al. (2009b). Nous nous inspirons également d'autres travaux, notamment pour la modélisation des développeurs, et pour la modélisation des fonctions de consentement à accepter (WTA) des propriétaires de terrains agricoles. Cette littérature fait généralement référence à Magliocca et à ses co-auteurs dans les modèles CHALMS (Magliocca et al., 2011 ; Magliocca et al., 2012 ; Magliocca et al., 2015).

Enfin, deux nouveaux types d'agents économiques sont introduits dans notre modèle

afin de mieux représenter le contexte spécifique des zones touristiques caractérisées par un développement résidentiel intense. D’une part, les « propriétaires fonciers non agricoles » qui ont un comportement sensiblement différent de celui des « propriétaires fonciers agricoles ». D’autre part, un agent « Investisseur » est également inclus. Ce dernier cherche à réaliser des gains en achetant des résidences et en les transformant en investissements locatifs touristiques.

Les modèles de description des modèles de décision des agents sont mis en œuvre pour chaque type d’agent dans chaque rôle différent (acheteur ou vendeur). De nombreux détails supplémentaires nécessaires au développement du code sont également inclus dans les modèles.

Notre étude de cas est caractérisée par un développement résidentiel intense, composé de deux marchés : un marché foncier et un marché immobilier. Sur le premier marché, on considère deux types de propriétaires fonciers (propriétaires de terres agricoles et propriétaires de terres non agricoles) et un type d’acheteurs (développeurs). Sur le second marché, après avoir construit un type de maison sur une parcelle de terrain donnée, le développeur devient vendeur sur le marché immobilier. La maison est alors mise en vente sur le marché immobilier sur lequel il y a deux types d’acheteurs différents : les ménages et l’investisseur.

La structure globale de notre modèle économique est décrite dans la figure 9.5 et le comportement des différents agents économiques ainsi que leurs interactions sont décrits en utilisant le formalisme économique traditionnel ¹³.

Nous utilisons des modèles pour détailler tous les éléments de notre modèle conceptuel économique. Huit modèles sont réalisés : un pour chaque marché et un par agent en fonction de son rôle (acheteur ou vendeur).

Les modèles correspondant à notre modèle conceptuel économique sont illustrés dans la figure 9.6. Deux exemples de ces modèles sont fournis ci-après ¹⁴ : un modèle de décision d’agent (pour l’acheteur promoteur sur le marché foncier) et un modèle de classe d’échange (pour le marché des habitations).

13. Le modèle mathématique est détaillé dans le chapitre suivant.

14. Les modèles basés sur le MR POTATOHEAD sont conçus pour être suffisamment modulaires pour traiter de manière exhaustive les différents ABM/LUCC. Dans les deux exemples ci-dessous, nous ne fournissons que les parties pertinentes des modèles correspondants pour notre modèle conceptuel économique. Les autres parties sont présentées dans un matériel supplémentaire en ligne : <https://ars.els-cdn.com/content/image/1-s2.0-S1476945X20301628-mmc2.pdf>.

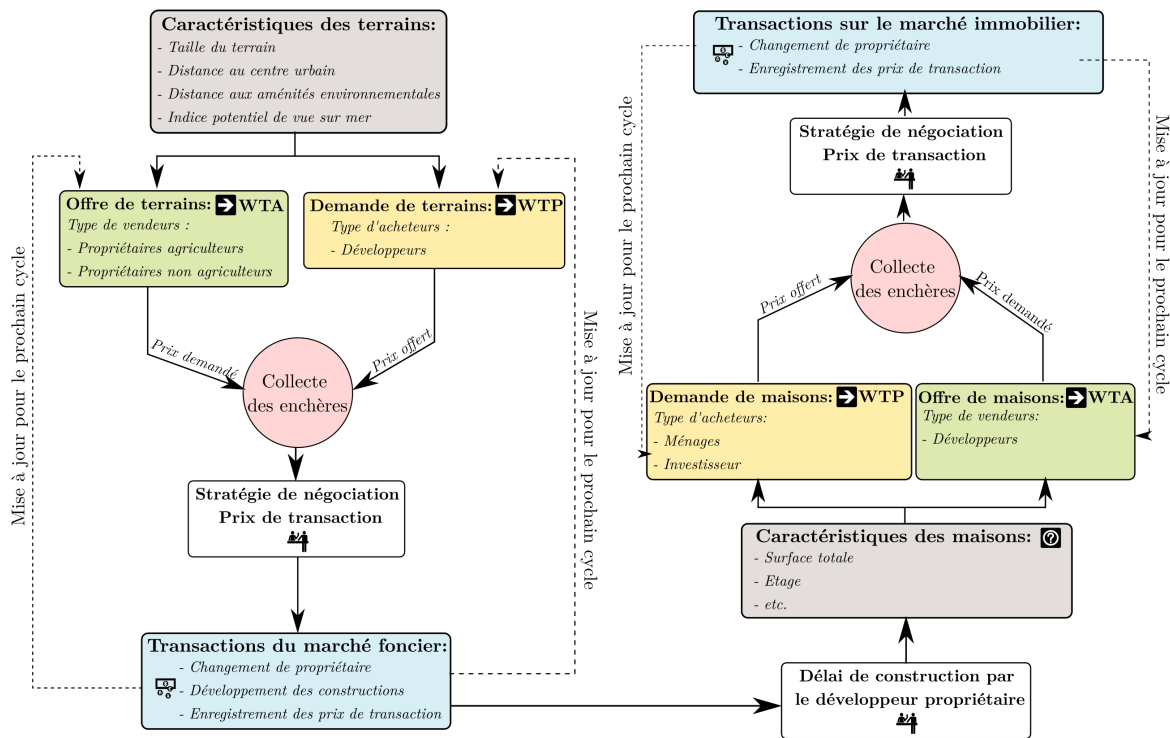


FIGURE 9.5 – Modèle conceptuel, d'après Filatova et al. (2009a).

MPH-PME : spécification de la décision d'utilisation des terres

MR POTATOHEAD Real Estate Market Framework - Modèle de décision pour les agents - Promoteurs/acheteurs de terrains

Classes démographiques (Ce sont des types d'agents. Chaque type d'agent peut acheter, vendre, louer ou mettre en location).

1. Acheteurs

(a) Développeurs – Acheteurs de terrains

Classe de prise de décision (remplie pour chaque type d'agent du marché foncier, 1-4 à partir du haut dans les modèles actuels).

Quatre types correspondent aux concepts de NetLogo : classe (race) C ; attributs, (variable propre) A ; liens/relations (liens directs (parent), liens non directs (ami, voisin, peut avoir des variables propres pour les liens dans Netlogo, non binaire) L ; processus (go, set-up) P ; documentation D.

Les auteurs sont encouragés à remplir les questions spécifiques à la prise de décision de l'agent dans le protocole ODD+D en complément.

1. A / P – Attributs (méthodes/fonction and paramètres) qui affectent les décisions (Ces éléments peuvent être appelés à plusieurs endroits - par exemple, le budget peut affecter la décision de déménager, puis le prix de l'offre. Précisez soigneusement toute source d'incertitude/de rationalité limitée pour chaque élément.)

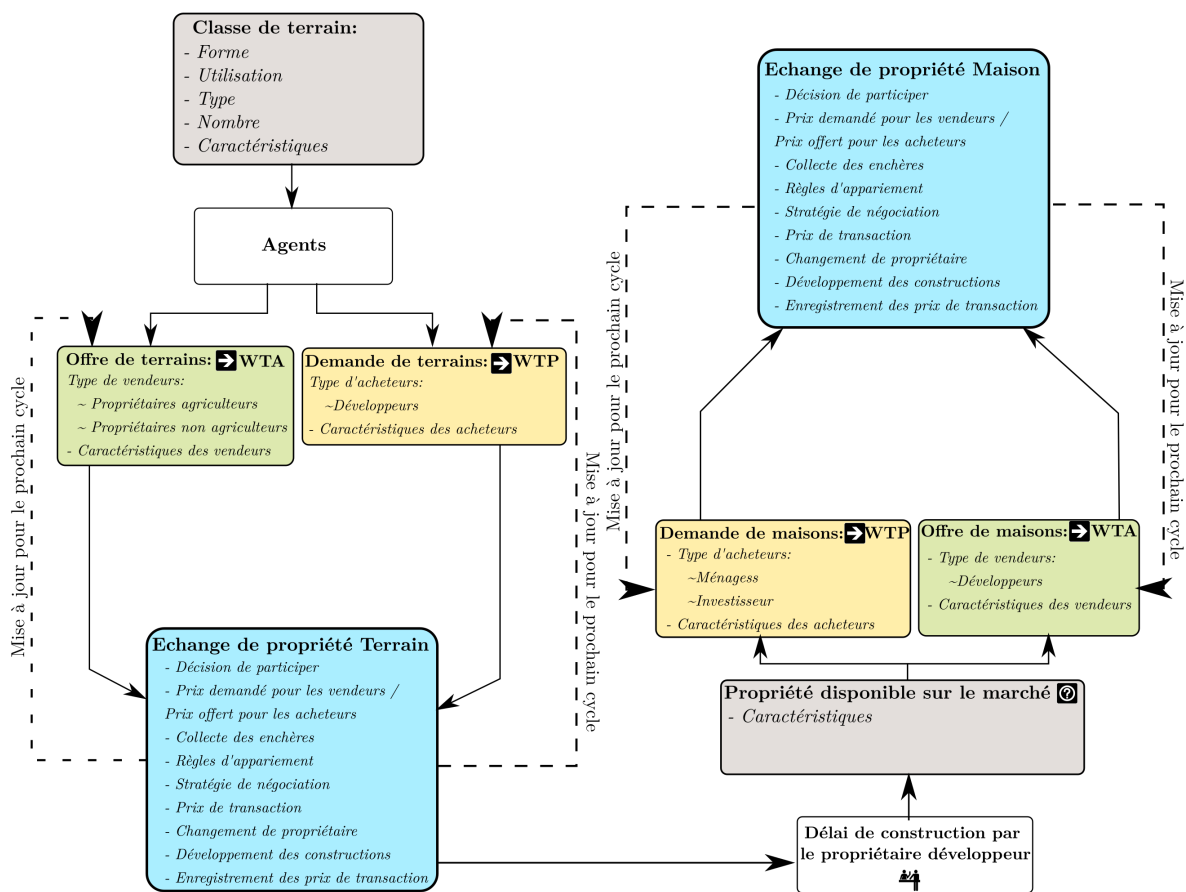


FIGURE 9.6 – *Modèle conceptuel*

- (a) Anticipations de prix : $P_{H,i,t-1}(z_i, LivA_m)$ est le prix moyen des ventes antérieures pour une maison avec des caractéristiques de terrains z_i et des caractéristiques de maisons h_m . Il est donné par les résultats du modèle économétrique.
 - (b) Taux de marge du développeur π_m
 - (c) Fonction de coût (Développeurs) : $C_{LivA}(z_i) LivA_m$ est le coût de construction pour construire une maison de type m .
 - (d) Ce qui sera construit (développeur acheteur de terrain) : Il y a M développeurs, chacun d'eux achète une parcelle et construit une maison de type m , avec $m = 1, \dots, M$.
 - (e) Consentement à payer (WTP) : Ils déterminent leur WTP en exigeant que le prix de vente leur permette de dégager une marge π_m sur le coût total de la maison construite sur la parcelle i (somme du prix payé pour l'achat de la parcelle de terrain i et du coût de construction d'une maison). On suppose qu'ils ne construisent qu'une maison par parcelle. Le WTP pour un développeur m pour une parcelle donnée avec des caractéristiques z_i , est donné par l'expression suivante :
$$WTP_m(z_i) = \frac{P_{H,i,t-1}(z_i, LivA_m)}{1+\pi_m} - C_{LivA}(z_i) LivA_m$$
 - (f) Offre d'enchère : Elle est donnée par $Bidprice_m = (1 - \psi_m)WTP_m(z_i)$
 - (g) Marge de manœuvre ψ_m : Le développeur propose un prix inférieur à ce qu'il est prêt à payer pour se laisser une marge de manœuvre.
 - (h) Les sources d'information (c'est-à-dire l'agent immobilier, les annonces, les voisins) : Ils disposent d'informations sur les prix passés des parcelles et des maisons comme une connaissance commune.
 - (i) Contraintes : Les développeurs ont une contrainte de capacité de construction qui signifie qu'ils ne peuvent construire que n maisons pendant une période t . Ainsi, un développeur m ne pourra enchérir que sur $n_m = n - \sigma_m$ parcelles, où σ_m est le nombre de sites de construction en cours de ce développeur.
2. P – Décision de participer au marché (devenir actif en tant qu'acheteur, vendeur, investisseur ou locataire)
- (a) Décision de participer au marché en tant qu'acheteur : Ils sont actifs sur le marché s'ils n'ont pas atteint leur capacité de construction.
3. P – Participer au marché en tant qu'acheteur
- (a) Quelles propriétés sont évaluées (stratégie de recherche) ? : Toutes les propriétés sont évaluées.
 - (b) WTP – voir ci-dessus
 - (c) Offre d'enchère – voir ci-dessus
 - (d) Sur quelle(s) propriété(s) enchérir ?

- i. Plusieurs par acheteur : Dans le respect de leurs contraintes de capacité de construction.
- (e) Stratégie de négociation : Les agents économiques sont toujours actifs sur le marché. Lorsqu'ils ont tous déterminé leurs offres et demandes d'enchère, un mécanisme d'appariement et d'échange débute. Afin de déterminer ce mécanisme, nous avons mené des entretiens auprès d'agents et experts immobiliers. Les développeurs enchérissent sur toutes les maisons en fonction de capacité de construction et les vendeurs évaluent toutes les offres reçues pour leur terrain. Après cela, le processus de décision commence pour chaque parcelle de terrain i :
 - (a) **Processus de transaction directe** (DTP^{15}) : Les vendeurs examinent toutes les offres et déterminent, le cas échéant, le meilleur $Bidprice$. Si le $Bidprice_{Max}$ d'un développeur est plus élevé que le $AskPrice_i$ du vendeur, la vente se réalise. Le prix de transaction est alors égal au $AskPrice_i$ du vendeur sans négociation de prix.
 $\bar{P}(z_i) = AskPrice_i$.
 - (b) **Transaction avec processus de négociation** (TNP^{16}) : S'il n'y a pas d'offre pour laquelle le $Bidprice_{Max}$ est supérieur au $AskPrice_i$, il n'y a pas de vente directe. Cependant, si pour un terrain, le WTP_m d'au moins un développeur est supérieur au WTA_q d'un propriétaire foncier, un mécanisme de négociation est enclenché. Le propriétaire détermine le WTP_{Max} des développeurs et l'appariement se fait entre eux. Le prix de la transaction est alors négocié et établi entre le WTP_{Max} et le WTA_q selon l'état du marché. Plus il y a de demande potentielle pour un bien, plus le prix de transaction sera élevé et plus le pouvoir de négociation du vendeur sera important, et vice versa. Le prix de la transaction est donné par : $\bar{P}(z_i) = WTA_q(z_i) + \frac{ND_i}{\chi} [WTP_{Max} - WTA_q(z_i)]$. ND_i est le nombre de demandes potentielles et χ est le nombre de développeurs pour lequel $n_{m'} \geq 0$, avec $m' = \{1, \dots, \chi\}$ pour $ND_i \leq \chi$.
 - (c) **Pas de transaction** : Enfin, si le WTA des vendeurs pour une parcelle i est supérieur au WTP des développeurs ou si les développeurs ont atteint leur limite de capacité de construction pour la période ($\chi = 0$), aucune transaction n'a lieu.
- (f) Règles s'ils ne parviennent pas à acheter
 - i. Rester sur le marché.

MPH-PME : spécification de l'échange de propriété

MR POTATOHEAD Real Estate Market Framework - Modèle de classe d'échange de propriété

15. *Direct Transaction Process.*

16. *Transaction Negotiation Process.*

1. Quel marché est modélisé ?
 - (a) Maisons
2. Qui sont les acteurs du marché ?
 - (a) Acheteurs
 - i. Investisseur
 - ii. Ménages
 - (b) Vendeurs
 - i. Développeurs
3. Quels produits sont achetés et vendus ?
 - (a) Grille générique, tessellation uniforme (défaut Netlogo)
 - (b) Nombre de types de conteneurs : un par parcelle.
 - i. Caractéristiques de l'entité spatiale
 - A. Localisation(s) spatiale(s)
 - B. Surface totale du terrain
 - C. Surface au sol du bâtiment
 - D. Accessibilité des transports
 - E. Aménités du voisinage (exemple : indice potentiel de vue sur mer)
 - F. Nuisances du voisinage (exemple : risque d'inondation)
 - ii. Nombre de catégories d'unités différentes dans le conteneur : une par conteneur
 - iii. Descriptions des catégories d'unités :
 - A. Nombre d'unités de ce type
 - B. Surface de l'unité
 - C. Nombre de pièces/chambres
 - D. Nombre de salles de bain
 - E. Taille de la parcelle/lot
 - F. Garage
4. Initialisation des emplacements des acheteurs et des vendeurs
 - (a) Un agent par conteneur spatial non construit
5. Règles de prix d'achat et de vente pour une allocation des terres basée sur le marché ?
 - (a) Comment les prix sont-ils affichés ? (*Un agent immobilier peut être mentionné ici, ou d'autres moyens de rendre les informations sur les annonces disponibles.*)
 - i. Information publique

- (b) Soumission des enchères : Sur combien de biens un acheteur peut-il enchérir ? Les ménages enchérissent sur une seule maison par pas de temps et l'investisseur peut enchérir sur toutes les maisons.
 - (c) Acceptation des offres : Combien d'offres sont acceptées par maison ? Toutes les offres.
6. Algorithme d'appariement : Comment les acheteurs sont-ils appariés avec les unités ?
- (a) Comment les unités sont appariées au vendeur : *Voir le vendeur ADM(s)* : modèle de propriété unique (un à un).
 - (b) Comment les offres des acheteurs sont-elles appariées aux unités ? *Voir l'acheteur ADM(s)* : Une seule.
 - (c) Comment les offres gagnantes sont-elles sélectionnées ? (*Voir le vendeur ADM*) : L'acheteur qui a le *Bidprice* le plus élevé (ou à défaut le *WTP* s'il est supérieur au *WTA*) est apparié au vendeur. (Si 2 acheteurs ou plus ont le même $Bidprice_{max}/WTP_{max}$, l'acheteur gagnant est choisi au hasard.)
 - (d) Comment le prix de la transaction est-il défini ? Voir les ADM(s) des acheteurs ou vendeurs.
7. Quel changement a lieu dans votre modèle ?
- (a) Vérifier les nouveaux logements.
 - (b) Changement de propriétaires.
 - (c) Enregistrement des prix de vente pour le prochain cycle.
 - (d) Règles pour les acheteurs qui n'ont pas trouvé de maison (Voir les fonctions ADM) : Les acheteurs qui n'ont pas trouvé de maisons restent sur le marché et continuent à faire des offres au pas de temps suivant.
 - (e) Identifie les acheteurs et vendeurs actifs pour le prochain cycle.
8. Quelles sont vos règles d'arrêt pour l'activité du marché ?
- (a) Le processus s'arrête lorsque : (1) toutes les maisons sont vendues ; (2) tous les ménages ont une maison, (3) il n'y a plus de transaction, (4) fin du temps de simulation.

9.3.2 Calibration

Dans cette section, nous traitons de la calibration du modèle informatique de l'*ABM/LMM* et l'implémentation d'un premier prototype (modèle exécutable) pour la calibration, sur l'infrastructure de simulation de la plateforme *NetLogo*.

Afin de formuler rapidement un prototype fonctionnel d'objets et de modules rapidement exploitables de l'*ABM/LMM*, nous faisons le choix d'intégrer le modèle conceptuel conformément au *Design Pattern NetLogo (DP-NetLogo)*. Le *DP-NetLogo* est un design pattern d'architecture largement reconnu par la communauté des utilisateurs de la

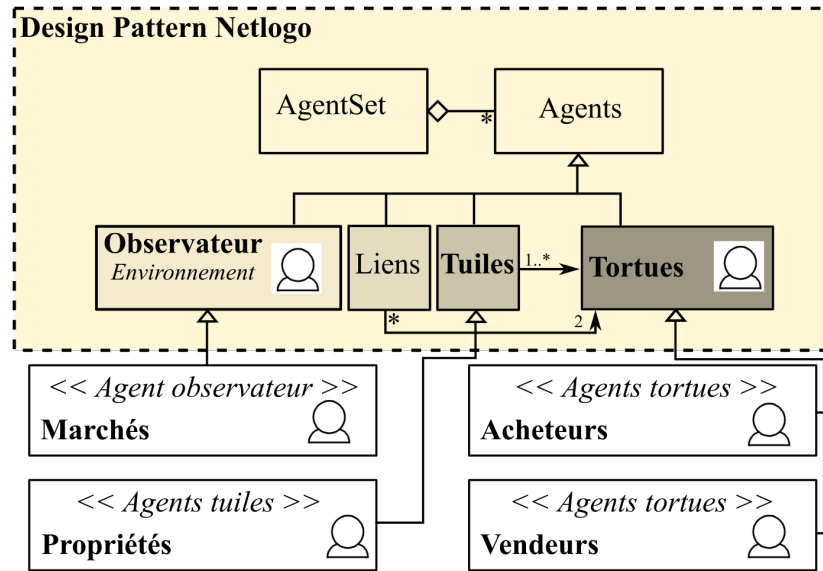


FIGURE 9.7 — Les objets spécifiques du modèle informatique de l'ABM/LMM sont conformes au DP-NetLogo.

plateforme. Les objets du modèle informatique ainsi organisés sont de fait facilement compréhensibles et comparables. A titre d'exemple, comme cela est illustré sur la figure 9.7, les agents précédemment formalisés dans l'ABM/LMM ont été intégrés comme des spécialisations croissantes de quatre catégories génériques d'objets : l'*Observateur* (objet *Observer*), les *Tuiles* (objets *Patch*), les *Tortues* (objets *Turtle*) et les *Liens* (objets *Link*).

L'agent *Observateur* est un agent unique omniscient qui observe l'environnement et qui donnent des ordres aux autres agents si nécessaire. Les agents *Tuiles* sont des agents qui modélisent les emplacements fixes du plan ; en général, ils correspondent aux cellules de l'espace cellulaire des modèles conceptuels de type automates cellulaires (CAM : *Cellular Automata Model*) (Innocenti et al., 2016). Les *Tortues* sont les agents mobiles du monde numérique. Leur localisation est déterminée à partir de celle des agents *Tuiles* sur lesquels ils se déplacent, c'est-à-dire une localisation dans l'espace cellulaire.

Les agents *Tortues* peuvent se déplacer, et il peut y avoir plusieurs agents *Tortues* sur une même *Tuile*. Les *Liens* sont des agents particuliers qui servent à relier deux agents *Tortues* entre eux.

Le rôle de l'agent *Property* est crucial dans l'ABM/LUCC. Ces agents sont placés sur chaque parcelle de terrain du monde (*Tuiles*). Ils sont utilisés pour implémenter l'évolution de la parcelle dans le temps, d'un état de base à un état de construction qui correspond au dernier état de la maison. Ils sont constitués de plusieurs propriétés immobilières. En résumé, l'utilisation des agents *Property* dans Netlogo facilite efficacement les relations de type « un à plusieurs » patches sur une propriété (comme un immeuble d'habitation sur une seule parcelle de terrain), et des relations de « plu-

sieurs à un » entre les propriétés et les agents économiques (comme un investisseur ou un promoteur qui possède plusieurs propriétés). Ce type d'agent reste fixe et n'a pas de règles de comportement, bien qu'il hérite des caractéristiques des agents acteurs fonciers. Leur tâche consiste à stocker des informations limitées pour le processus de marché. A $t = 0$, plusieurs agents sont placés sur le monde constituant les conditions initiales de la simulation. Les agents Property sont placés sur les parcelles du monde.

Bien que l'utilisation du *DP-NetLogo* cible préférentiellement la plateforme *NetLogo* quant à l'implémentation du modèle exécutable, le modèle informatique proposé constitue une spécification de code objet standardisée qu'il est possible d'implémenter dans un autre langage de programmation orientée objet, tel que Python, C++, etc.

Cette étape du processus de modélisation *4-SSIMP* est relativement aisée même pour des non-informaticiens. Tout d'abord, parce que les objets du modèle informatique étant spécifiés selon le *DP-NetLogo*, ceux-ci sont maintenant rapidement implémentables sur l'infrastructure de simulation de la plateforme *Netlogo* (Wilensky, 1999). Ensuite, comme le langage de programmation de la plateforme est un dérivé du langage de programmation *Logo*, les règles de syntaxe sont réduites au minimum et le vocabulaire se rapproche énormément du langage naturel.

9.4 Conclusion

Dans ce chapitre nous avons abordé la problématique de la simulation informatique des *systèmes complexes de changement d'utilisation et de couverture des sols* (*Land-Use and Land-Cover Change - LUCC*). Nous avons construit sur la base du processus de modélisation simplifié en quatre étapes *4-SSIMP*, un modèle conceptuel à base d'agents (*ABM/LUCC*) de type *Agent-Based Model / Land Market Model - ABM/LMM*. Nous avons utilisé la spécification conceptuelle *MR POTATOHEAD - Property Market Edition* pour formuler deux marchés : un marché foncier et un marché immobilier. Le marché foncier décrit la dynamique et les interactions entre des agents propriétaires fonciers vendeurs et des agents développeurs acheteurs (*Dev-agents*). Le marché immobilier décrit la dynamique et les interactions entre des agents ménages (*Hse-Agents*), modélisant de futurs propriétaires immobiliers acheteurs, et un unique agent investisseur (*Inv-Agent*) acheteur. Les agents acheteurs (*Hse-Agents* et *Inv-Agent*) sont en concurrence pour l'achat des biens immobiliers, construits par les agents développeurs entrepreneurs (*Dev-agents*), devenus les vendeurs sur ce marché. Le modèle informatique est fondé sur le *Design Pattern NetLogo*. Cela nous permet d'implémenter rapidement le prototype d'un modèle exécutable dans un but de calibration. Il nous faut maintenant procéder à la vérification et à la validation des structures de données et des logiques algorithmiques des processus de décision de l'*ABM/LMM*. Nous présentons ce travail dans le chapitre suivant.

Bibliographie

- Ambler, Scott W (2005). *The elements of UML (TM) 2.0 style*. Cambridge University Press.
- Balci, Osman (2011). « How to successfully conduct large-scale modeling and simulation projects ». In : *Proceedings of the Winter Simulation Conference*. Winter Simulation Conference, p. 176-182. DOI : 10.1109/WSC.2011.6147749.
- Biggs, Norman L (2008). *Discrete mathematics*. Oxford University Press.
- Chwif, Leonardo, Jerry Banks, João Patrício de Moura Filho et Bruno Santini (2013). « A framework for specifying a discrete-event simulation conceptual model ». In : *Journal of Simulation* 7(1), p. 50-60. DOI : 10.1057/jos.2012.18.
- Clarke, Keith C (2018). « Cellular automata and agent-based models ». In : *Handbook of regional science*, p. 1-16.
- Coquillard, Patrick et DRC Hill (1997). *Modélisation et simulation d'écosystèmes: des modèles déterministes aux simulations à événements discrets*. Paris : Masson.
- Erard, Pierre-Jean et Pontien Déguénon (1996). *Simulation par événements discrets*. PPUR presses polytechniques.
- Ferber, Jacques et Gerhard Weiss (1999). *Multi-agent systems: an introduction to distributed artificial intelligence*. T. 1. Addison-Wesley Reading.
- Filatova, Tatiana, Dawn C. Parker et Anne Van der Veen (2009a). « Agent-based urban land markets: agent's pricing behavior, land prices and urban land use change ». In : *Journal of Artificial Societies and Social Simulation* 12(1), p. 3.
- Filatova, Tatiana, Anne Van Der Veen et Dawn C. Parker (2009b). « Land Market Interactions between Heterogeneous Agents in a Heterogeneous Landscape-Tracing the Macro-Scale Effects of Individual Trade-Offs between Environmental Amenities and Disamenities ». In : *Canadian Journal of Agricultural Economics/Revue canadienne d'agroeconomie* 57(4), p. 431-457. DOI : 10.1111/j.1744-7976.2009.01164.x.
- Gandon, Fabien (2006). « Ontologies informatiques ». In : *Interstices. Journal en ligne de l'INRIA*. URL : <https://hal.inria.fr/inria-00080851>.
- Grignard, Arnaud, Patrick Taillandier, Benoit Gaudou, Duc An Vo, Nghi Quang Huynh et Alexis Drogoul (2013). « GAMA 1.6: Advancing the art of complex agent-based modeling and simulation ». In : *International Conference on Principles and Practice of Multi-Agent Systems*. Springer, p. 117-131.

- Grimm, Volker, Uta Berger, Donald L. DeAngelis, J. Gary Polhill, Jarl Giske et Steven F. Railsback (2010). « The ODD protocol: A review and first update ». In : *Ecological Modelling* 221(23), p. 2760-2768. DOI : 10.1016/j.ecolmodel.2010.08.019.
- Grimm, Volker et Steven F Railsback (2006). « Agent-based models in ecology: patterns and alternative theories of adaptive behaviour ». In : *Agent-based computational modelling*. Springer, p. 139-152.
- Hamill, Lynne et G Nigel Gilbert (2016). *Agent-based modelling in economics*. Wiley Online Library.
- Hoganson, Kenneth E (2008). *Concepts in computing*. Jones & Bartlett Learning.
- Innocenti, Eric, Dominique Urbani, Paul-Antoine Bisgambiglia et Pierre-Regis Gonsolin (2016). « A Multicomponent Modeling Approach for Fishery Simulations ». In : *2016 IEEE International Conference on Agents (ICA)*. IEEE, p. 108-109. DOI : 10.1109/ICA.2016.036.
- Jerry, Banks (1984). *Discrete-event system simulation*. Pearson Education India.
- Klügl, Franziska et Ana LC Bazzan (2012). « Agent-based modeling and simulation ». In : *AI Magazine* 33(3), p. 29-29. DOI : 10.1609/aimag.v33i3.2425.
- Magliocca, Nicholas, Virginia McConnell et Margaret Walls (2015). « Exploring sprawl: Results from an economic agent-based model of land and housing markets ». In : *Ecological Economics* 113, p. 114-125. DOI : 10.1016/j.ecolecon.2015.02.020.
- Magliocca, Nicholas, Virginia McConnell, Margaret Walls et Elena Safirova (2012). « Zoning on the urban fringe: Results from a new approach to modeling land and housing markets ». In : *Regional Science and Urban Economics* 42(1-2), p. 198-210. DOI : 10.1016/j.regsciurbeco.2011.08.012.
- Magliocca, Nicholas, Elena Safirova, Virginia McConnell et Margaret Walls (2011). « An economic agent-based model of coupled housing and land markets (CHALMS) ». In : *Computers, Environment and Urban Systems* 35(3), p. 183-191. DOI : 10.1016/j.compenvurbsys.2011.01.002.
- Menner, William A (1995). « Introduction to modeling and simulation ». In : *Johns Hopkins APL Technical Digest* 16(1), p. 6-17.
- Minsky, Marvin (1965). *Matter, mind and models*. Project MAC, Massachusetts Institute of Technology.
- Mitroff, Ian I, Frederick Betz, Louis R Pondy et Francisco Sagasti (1974). « On managing science in the systems age: two schemas for the study of science as a whole systems phenomenon ». In : *Interfaces* 4(3), p. 46-58. DOI : 10.1287/inte.4.3.46.
- Parker, Dawn C. (2014). « An economic perspective on agent-based models of land use and land cover change ». In : *The Oxford Handbook of Land Economics*. Sous la dir. de J. Duke et J. Wu. Oxford University Press. Chap. 16, p. 402-429.
- Parker, Dawn C., Daniel G Brown, J Gary Polhill, Peter J Deadman et Steven M Manson (2008a). « Illustrating a new conceptual design pattern for agent-based models of land use via five case studies—the MR POTATOHEAD framework ». In : *Agent-based modelling in natural resource management*. Universidad de Valladolid, p. 23-51.

- Parker, Dawn C., Adrian Carro, Claudio Detotto, Tatiana Filatova, Jiaqi Ge, Yu Huang, Corinne Idda, Nicholas Magliocca, J. Gary Polhill et Dominique Prunetti (2019). « MR POTATOHEAD: Property Market Edition-Development of a Common Description Template and Code Base for Agent-Based Land Market Models ». In : *“ESSA at Work” Session of the Social Simulation conference*. Mainz, Germany, 23-27th Sept.
- Parker, Dawn C. et Tatiana Filatova (2008). « A conceptual design for a bilateral agent-based land market with heterogeneous economic agents ». In : *Computers, Environment and Urban Systems* 32(6), p. 454-463. DOI : 10.1016/j.compenvurbsys.2008.09.012.
- Parker, Dawn C. et al. (2008b). « Case studies, cross-site comparisons, and the challenge of generalization: comparing agent-based models of land-use change in frontier regions ». In : *Journal of Land Use Science* 3(1), p. 41-72. DOI : 10.1080/17474230802048151.
- Robinson, Stewart (2008). « Conceptual modelling for simulation Part I: definition and requirements ». In : *Journal of the operational research society* 59(3), p. 278-290.
- Robinson, Stewart (2013). « Conceptual modeling for simulation ». In : *2013 Winter Simulations Conference (WSC)*. IEEE, p. 377-388. DOI : 10.1109/WSC.2013.6721435.
- Robinson, Stewart (2015). « A tutorial on conceptual modeling for simulation ». In : *Proceedings of the 2015 Winter Simulation Conference*. IEEE Press, p. 1820-1834. DOI : 10.1109/WSC.2015.7408298.
- Siegfried, Robert (2014). « A framework for Efficient Agent-Based Modeling and Simulation ». In : *The Oxford Handbook of Innovation*. Sous la dir. de Springer Fachmedien Wiesbaden. Springer Vieweg. Chap. 2, p. 11-26.
- Wilensky, Uri (1999). *NetLogo*. Evanston, IL: Center for Connected Learning and Computer-Based modeling, Northwestern University.
- Zeigler, Bernard P., H. Praehofer et T.G. Kim (2000). *Theory of Modeling and Simulation: Integrating Discrete Event and Continuous Complex Dynamic Systems*. Academic Press.
- Zill, Dennis G (2012). *A first course in differential equations with modeling applications*. Cengage Learning.