

Ingénierie Dirigée par les Modèles (IDM) appliquée à l'observation des modèles socio-environnementaux

Hasina Lalaina Rakotonirainy¹, Jean-Pierre Müller² and Bertin Olivier
Ramamonjisoa¹

¹ LRAM, Université de Fianarantsoa, Fianarantsoa, Madagascar

hasina.rakotonirainy@cirad.fr, bertinram@yahoo.fr

² GREEN, CIRAD, Campus international de Baillarguet, Montpellier, France

jean-pierre.muller@cirad.fr

Abstract

The researchers need to use complex models to understand the socio-ecosystems (SES). However, the observation of a SES model becomes a difficult task and no general framework exists to solve this problem. The aim of this paper is to propose a generic framework to specify the observation of SES models to easily generate the indicators the thematicians want to follow during the simulation. For that, we propose to reformulate observation as a transformation problem between data structures. It allows to use the concepts of Model Driven Engineering (MDE) to implement domain specific languages (DSL). We have designed dedicated languages to allow the specification of what to observe, with which observation strategy, and how to generate the indicators.

Keywords: MDE, DSL, Observation, Complex model, Socio-ecosystem

Résumé

Les chercheurs ont besoin d'utiliser des modèles complexes pour comprendre les socio-écosystèmes (SES). Cependant, l'observation d'un modèle SES est une tâche difficile et aucun cadre général n'a été envisagé pour résoudre ce problème. L'objectif de cet article est de proposer un cadre générique de spécification de l'observation des modèles SES afin d'engendrer facilement les indicateurs que les thématiciens veulent suivre pendant la simulation. Pour cela, nous proposons de reformuler l'observation de modèle en un problème de transformation entre structures de données. Cette reformulation permet d'utiliser les concepts de l'ingénierie dirigée par les modèles (IDM) afin de mettre en oeuvre des langages dédiés (DSL). Nous avons ainsi élaboré des langages dédiés qui permettent de spécifier ce que nous voulons observer, avec quelle stratégie d'observation et comment engendrer les indicateurs.

Mots-clés : IDM, Langage dédié, Observation, Modèle complexe, Socio-écosystème

1 Introduction

Les recherches sur la gestion durable des ressources naturelles renouvelables (RNR) ont montré la difficulté de comprendre les interactions entre dynamiques sociales et dynamiques biophysiques caractérisant un socio-écosystème (SES) donné. Des modèles de plus en plus complexes ont été développés afin de faciliter la compréhension mutuelle des acteurs impliqués directement ou indirectement dans la gestion locale des RNR. A chaque simulation d'un modèle SES comme celui du modèle Mirana [1] dont l'objectif est d'évaluer l'impact d'un ensemble

de systèmes de régulations sur la durabilité écologique, économique et sociale d'un terroir villageois, les données et les interactions entre les éléments du modèle sont nombreuses et très complexes. Toutefois, l'utilisation de ces modèles produit des besoins d'observer ce qui se passe pendant la simulation [1] en termes d'indicateurs sur des données identifiables (revenus, diversité floristique et agronomique, coût du contrôle, etc.) de façon proche du discours des thématiciens et des acteurs locaux. Les indicateurs sont définis par les thématiciens et sont relatifs à des questions de compréhension du modèle, de recherche ou de développement. L'observation de la simulation devient alors plus importante que la simulation elle-même. L'observation consiste à obtenir l'état et l'évolution du modèle pendant la simulation [13] et à en élaborer des indicateurs synthétiques.

Pour ce faire, des outils de modélisation et de simulation comme MIMOSA [11] et JAMES II [8] implémentent le patron de conception "observateur". Certains outils comme OSIF [15] exploitent la programmation orientée aspect. D'autres outils comme PowerDEVs [2] utilisent le mécanisme d'observation réactif ou proactif classique du formalisme DEVs. Dans le cadre des simulations à événements discrets, des chercheurs ont proposé d'étendre le formalisme PDEVs, par un mécanisme générique d'observation [13] n'influençant pas le comportement des modèles pendant la simulation.

Malgré ces efforts, les approches proposées pour l'observation des modèles sont relativement ad hoc. Elles dépendent du formalisme utilisé lors de la construction des modèles et aucun cadre général n'a encore été envisagé. L'objectif de cet article est de proposer un cadre générique de spécification de l'observation des modèles SES afin d'engendrer facilement les indicateurs. Pour cela, nous proposons de reformuler la question de l'observation de modèles en un problème de transformation entre structures de données. Ainsi, la spécification d'observation du modèle revient à décrire quelles structures du modèle on a besoin d'observer avec une certaine stratégie d'observation produisant des trajectoires, puis à spécifier une chaîne de transformation de ces trajectoires en des structures de données qui sont les indicateurs souhaités par les thématiciens.

Cette formulation permet d'utiliser les concepts de l'ingénierie dirigée par les modèles (IDM) [5], [6] afin de faciliter la mise en oeuvre des langages dédiés (ou DSL pour Domain Specific Language) [17] à l'observation des modèles complexes ainsi que les transformations [3] nécessaires pour produire des indicateurs. Dans la section 2, nous introduisons et justifions la méthodologie utilisée pour la conception des DSLs d'observation. Ensuite, nous présentons le processus d'observation des modèles SES (section 3) avec un résultat préliminaire (section 4). Enfin, nous présentons dans la conclusion un aperçu des travaux à venir.

2 Méthodologie

L'IDM est une méthodologie de conception et de développement de logiciels basée sur les modèles [9] qui permet, entre autres, de construire des DSLs. Un des avantages de l'utilisation des DSLs est de permettre aux modélisateurs de se concentrer sur leurs préoccupations et leurs domaines de recherche. Le code d'application est ensuite généré automatiquement. Pour construire des DSLs, l'OMG a proposé la pyramide de modélisation [3] qui s'articule autour de cinq concepts fondamentaux : le système réel, le modèle qui permet de représenter ce système, le méta-modèle qui permet de définir le modèle, le méta-méta-modèle qui sert à spécifier le méta-modèle et les transformations entre modèles. Par ailleurs, nous avons montré dans [14] qu'en utilisant les concepts proposés par l'IDM, nous sommes en mesure de formuler de façon générique le problème d'observation des modèles socio-environnementaux complexes (Fig.1) en spécifiant les méta-modèles pour décrire la partie des trajectoires de simulation du modèle que l'on souhaite observer (les observables - L4), la description des indicateurs que l'on souhaite

construire (L6) et la transformation entre les observables et les indicateurs (L5).

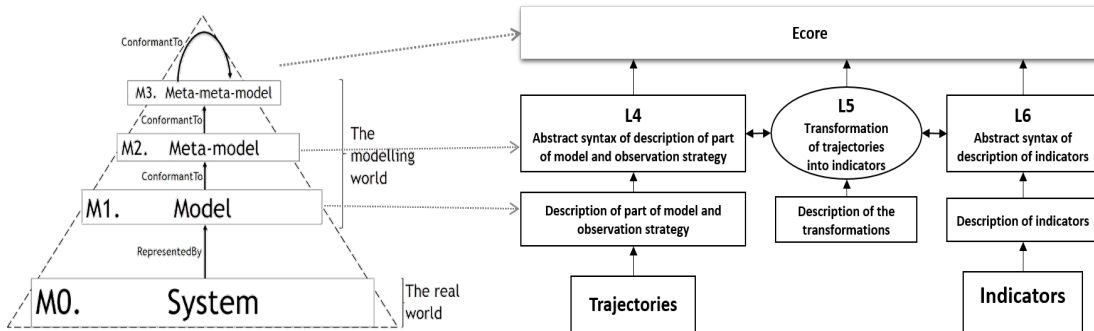


FIGURE 1 – Correspondance entre la pyramide de modélisation [3] et le processus de modélisation

3 Processus d’observation

La Fig.2 présente les étapes globales que nous proposons pour construire les indicateurs souhaités par les thématiciens en partant des structures de données d’un modèle SES. En effet, pendant la simulation, il est possible de suivre l’évolution des structures de données du modèle et d’obtenir des séquences de parties observables de celui-ci en définissant une stratégie d’observation (avec L4). Les séquences obtenues durant la simulation forment des trajectoires qui sont des structures de données indexées par le temps (*séquence dates, structures de données*). Engendrer une trajectoire nous oblige à spécifier une stratégie d’échantillonnage (toujours avec L4). Ensuite, nous devons encore transformer ces trajectoires en des structures de données manipulables (avec L5) afin d’obtenir les indicateurs. Les indicateurs peuvent être enfin soit stockés dans un support de données, soit visualisés selon la spécification faite (avec L6).

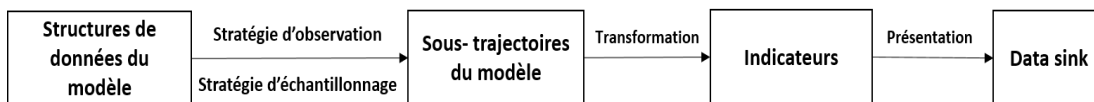


FIGURE 2 – Processus de construction des indicateurs

4 Résultat préliminaire

Pour le moment, nous avons spécifié le langage L4 qui permet de spécifier la partie du modèle que l’on veut observer pendant la simulation, avec une certaine stratégie d’observation.

4.1 Etat de l’art

Suivre tout ce qui se passe dans le modèle coûte très cher en temps d’exécution et en espace de stockage. Pour résoudre le problème de l’exécution, des outils comme SimExplorer [4] ou

OpenMOLE [12] sont capables de distribuer les simulations d'un modèle avec des jeux de paramètres différents sur des grappes de machines. Certains programmes informatiques utilisent la technique d'évaluation par nécessité ou "lazy evaluation" [10] et évitent de calculer des résultats inutiles pendant l'exécution d'un programme. La solution que nous proposons doit être adaptée aux thématiciens et inspirée de l'évaluation de leurs besoins. Pour pouvoir spécifier une partie du modèle, nous devons fournir un moyen qui permette de sélectionner et d'échantillonner, de façon précise, n'importe quelle partie de la trajectoire du modèle à différents niveaux et à différentes échelles (spatiales et temporelles). Le moyen le plus courant en informatique pour spécifier où recueillir l'information est d'utiliser un langage de requête comme SQL pour interroger les bases de données relationnelles, OQL pour les bases de données orientées objet, OLAP pour les bases de données multidimensionnelles, XPath pour parcourir le DOM ou XQuery pour les données XML. Néanmoins, un langage de requête dépend de la structure des données interrogées. Dans la littérature, XML est reconnu comme étant le format de description de données universel [7] et sa structure arborescente permet de représenter la hiérarchie de n'importe quelle structure de données. Ainsi, pour que L4 soit le plus générique possible, nous proposons de créer une syntaxe abstraite inspirée de XPath et de XQuery afin de spécifier ce qu'on veut observer dans la trajectoire du modèle. Cependant, nous ne travaillons pas sur du XML ou du contenu de bases de données, mais sur les structures de données elles-mêmes. Le fait d'interroger une trajectoire plutôt qu'un état nous oblige à ajouter la stratégie d'échantillonnage.

4.2 DSL pour spécifier les stratégies d'observation

Définir une stratégie d'observation consiste à spécifier la façon, le moment ou la fréquence d'observation du modèle pendant la simulation, afin d'obtenir les trajectoires. En effet, deux types de stratégie d'observation peuvent être spécifiés à partir du DSL que nous proposons, à savoir :

- *O_{observer}* : Pendant la simulation, chaque élément du modèle va signaler son changement d'état à un observateur abonné en utilisant le patron de conception "observateur" ; Dans ce cas, le modèle de simulation doit implémenter ce type de pattern et il est nécessaire de décrire les fonctions qui permettent à un observateur de s'abonner à l'élément qu'on voudrait suivre et comment le changement d'état est signalé.
- *O_{timed}* : Pendant la simulation, un observateur interroge l'état d'un élément observable du modèle en spécifiant les dates d'observation (uniques, à pas de temps constant, etc.). Il est alors nécessaire de spécifier comment extraire l'information (*getters*).

Un extrait de la syntaxe abstraite de L4 est donné par la Fig. 3.

Les échantillons du modèle seront ensuite extraits pendant la simulation selon le choix de stratégie d'observation pour former les trajectoires. Par exemple, dans le contexte des modèles SES, pour suivre l'évolution de la quantité des stocks de ressource d'un espace géographique donné, il suffit de spécifier l'élément qu'on voudrait suivre (quantité de stocks) avec une stratégie d'observation.

5 Conclusion et Perspectives

Dans cet article, nous avons montré par la spécification du langage L4 qu'il est possible de formuler une partie du processus d'observation des modèles SES en utilisant les concepts proposés par l'IDM. Néanmoins, pour prouver la généralité dans la construction des indicateurs sur des données identifiables (revenus des foyers, variation de l'habitat, etc.) proches du discours des thématiciens et des acteurs locaux, nous avons encore besoin de spécifier d'une part, un

- [8] J. Himmelspach. James ii : Extending, using, and experiments. In *Proceedings of the 5th International ICST Conference on Simulation Tools and Techniques*, SIMUTOOLS '12, pages 208–210. ICST, 2012.
- [9] J.-M. Jézéquel, B. Combemale, and D. Vojtisek. *Ingénierie Dirigée par les Modèles : des concepts à la pratique...* Références sciences. Ellipses, Feb. 2012.
- [10] T. Johnsson. Efficient compilation of lazy evaluation. *SIGPLAN Not.*, 39(4) :125–138, Apr. 2004.
- [11] J.-P. Müller. Mimosa : Using ontologies for modelling and simulation. In *GI Jahrestagung (1)*, pages 227–231, 2007.
- [12] J. Passerat-Palmbach, M. Leclaire, R. Reuillon, Z. Wang, and D. Rueckert. OpenMOLE : A Workflow Engine for Distributed Medical Image Analysis. In *International Workshop on High Performance Computing for Biomedical Image Analysis (part of MICCAI 2014)*, Boston, United States, Sept. 2014.
- [13] G. Quesnel, R. Trepos, and r. Ramat. Observations of discrete event models. In N. Pina, J. Kacprzyk, and M. S. Obaidat, editors, *SIMULTECH*, pages 32–41. SciTePress, 2012.
- [14] H. L. Rakotonirainy, J. Müller, and B. O. Ramamonjisoa. Towards a generic framework for the initialization and the observation of socio-environmental models. In *Model and Data Engineering - 4th International Conference, MEDI 2014, Larnaca, Cyprus, September 24-26, 2014*, pages 45–52, 2014.
- [15] J. Ribault, O. Dalle, D. Conan, and S. Leriche. Osif : A framework to instrument, validate, and analyze simulations. In *Proceedings of the 3rd International ICST Conference on Simulation Tools and Techniques*, page 56. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2010.
- [16] D. Steinberg, F. Budinsky, E. Merks, and M. Paternostro. *EMF : eclipse modeling framework*. Pearson Education, 2008.
- [17] A. Van Deursen, P. Klint, and J. Visser. Domain-specific languages : An annotated bibliography. *Sigplan Notices*, 35(6) :26–36, 2000.