Agent-based Spatial Model Coupling Using a Coordination Unit

Jean B. Dinaharison[1,2], Nicolas Marilleau[2], Hasina Rakotonirainy[1], Nathalie Corson[3], Laetitia Bernard[4] and Jean-Pierre Müller[5]

[1]*Informatique Géomatique Modélisation et Applications (IGMA), University of Fianarantsoa, Fianarantsoa 301, Madagascar*

[2]*Unité de Modélisation Mathématique et Informatique des Systèmes Complexes (UMMISCO) - Institut de Recherche pour le Développement - Sorbonne Université, 93143 Bondy, France*

[3]*Laboratoire de Mathématique Appliqué du Havre, Le Havre University, 76063 Le Havre, France*

[4]*Eco&Sol, Institut de Recherche pour le Développement, 34060 Montpellier, France*

[5]*UMR SENS - Cirad, 34398 Montpellier Cedex 5, France*

Abstract

Modeling complex systems like soil functioning turns out to be a challenge due to its heterogeneous structure, the number of interactions and processes entangled with each other at various levels and scales. Its heterogeneous structure requires a spatially explicit account, by coupling models through the space by using its resources, as proposed by the spatial model coupling approaches. However, these approaches raise the problem of resource conflicts : interacting models can modify resources simultaneously while the amount available cannot supply models demands. As spatial model coupling approaches do not provide proper resource distribution methods to handle that matter, inconsistencies might occur if resource conflicts are poorly managed. In order to solve this problem, we propose a novel coupling approach based on agent paradigm in which the space, associated with a coordination unit supports model exchange and conflict management. This architecture is illustrated by an UML Class Diagram that we implemented into the GAMA platform. To prove its effectiveness, we test the approach on a simple soil model in which numerous models compete for resources scattered in the space. Comparative results suggest that handling conflicts using the coordination unit shows more precise indications of resource use, compared to the model in which conflicts are mismanaged.

Keywords

Agent based, Coordination, GAMA, Resource conflicts, Spatial model coupling

WOA 2022: 23rd Workshop From Objects to Agents, September 1–2, Genova, Italy

*Corresponding author.

†These authors contributed equally.

dinaharison.jb@gmail.com (J. B. Dinaharison); nicolas.marilleau@ird.fr (N. Marilleau); hasina.rakotonirainy@emit.mg (H. Rakotonirainy); nathalie.corson@univ-lehavre.fr (N. Corson); laetitia.bernard@ird.fr (L. Bernard); jean-pierre.muller@cirad.fr (J. Müller)

00000-0002-9247-5307 (J. B. Dinaharison); 0000-0002-1004-4352 (N. Marilleau); 0000-0001-8678-7526 (H. Rakotonirainy); 0000-0002-9970-8230 (N. Corson); 0000-0002-7057-6182 (L. Bernard); 0000-0003-4614-0108 (J. Müller)

1. Introduction

Space is of main interest when modeling complex systems such as soil functioning [1, 2] or urban dynamics [3, 4]. In all these models, a variety of dynamics interact through their environment. For example in soil science, worms stimulate microbial activities by ingesting an important amount of soil, while microbes decompose the organic matter into nutrients to feed plants. To describe this type of phenomenon properly, "Spatial Model Coupling" approaches such as [5, 3, 4] propose to encapsulate the different dynamics individually (e.g worms, microbes, plants, soil...) as models (referred as "*micro-model*") and to combine them to build a single model (referred as "*macro-model*"). This combination is made possible by the interactions of micro-models through the space.

However, such coupling technique causes inconsistencies when shared resources are poorly managed in a resource conflict scenario. A resource conflict arises when micro-models simultaneously access a limited resource that cannot meet all requests. For example, when the roots and several types of microbes consume nutrients from the soil at the same time when the available amount is insufficient to feed these consumptions. Inconsistencies may arise if a conflict is mismanaged given that micro-model behaviors depend on the availability of a resource. For example, roots and microbes are supposed to compete for soil nutrients, if the available amount does not meet their needs, how will they behave? leaving the resource as it is may affect simulation results and letting the wrong micro-model using the resource may affect other resources and impact other micro-models (inconsistencies). As Spatial model coupling approaches does not provide rules for handling such scenario, resources have to be managed in the event of a conflict for a consistent macro-model.

To address this problem, we propose a novel coupling approach based on the agent paradigm in which the space, together with a coordination unit supports model exchange and conflict management: the agent paradigm [6] is used to encapsulate the micro-models; the space is used as an exchange media; and the coordination unit defines how a resource is shared among micro-models in conflict scenarios.

This paper is structured as following : In section 2, we discuss the issues of model coupling as well as the coupling techniques. In section 3, we provide more insights of our coupling approach and contribution. In section 4, we present the simulation results for a simple model of soil functioning using the proposed approach and discuss its limitations.

2. Model Coupling

2.1. Coupling Issues

The challenge of model coupling is to ensure that a micro-model can interact and evolve with each other. Modelers have to deal with coupling issues such as interaction, time scale and space scale management.

Interaction Micro-models can communicate directly or indirectly [7]. They interact directly by using their respective inputs and outputs such as Discrete Event System Specification (DEVS) [8], in which actomic-DEVS (micro-models in DEVS) communicate through their input and

output ports. They exchange indirectly by using a third as communication media. For instance, Functional Mock-up Inteface (FMI) [9] and High Level Architecture (HLA) [10] uses respectively the Functional Mock-Up Unit (FMU) and Run Time Infrastructure (RTI) buses as communication mediator. The environment is also used as a communication mediator in agent-based approaches such as [3, 4], in which micro-models communicate through the space where they evolve.

Time scale Model time scale is deducted from the dynamic of the represented phenomena. Thus, micro-models operate at different time scales and embed their own *scheduler* (either time step based (continuous time) or event-based). A global *scheduler* ensures the chronological micro-model execution (time consistency). In event-based approaches such as DEVS formalism[8], external (from input ports) and internal (after some delay) events are passed to their respective function to compute the next system state. Approaches such as [4] use specific algorithms that are similar to calendars to ensure models executions.

Spatial scale This constraint is present when the spatial representation of a phenomenon is multi-leveled (e.g. a region divided into multiple micro-regions that are divided into atomic-regions). This type of representation can be seen in models of urban dynamics [3, 4] or soil science [1, 2].

2.2. Coupling models by the environment

Our coupling strategy consists of using the space as a coupling media for micro-models, i.e, micro-models communicate indirectly by using resources of the space.
This proposition creates a resource competition problem among micro-models. Therefore, the lack of resource conducts to a conflict (when a unique and restricted resource is used by micro-models simultaneously) and lead to inconsistencies during the simulation (if mismanaged).
To address this problem, we introduce the *Coordination Unit* concept that permits resource conflicts management in a space centered model coupling approach.

Agent Paradigm as formalism to wrap micro-models We use the agent paradigm [6] to encapsulate micro-models. Therefore, it allows to create a distributed architecture in which agents (containing micro-models) interact through the environment (space).

Space as an exchange media The micro-models communicate through the space, as proposed by [3, 4], by modifying available resources: a micro-model interact with each other through the space by perceiving resource changes and acting accordingly. As a result of such coupling strategy, micro-models are not aware of each other's states.

To handle the simultaneousness of modeled processes, a coordination unit records and supervises the interactions between the space and micro-models, and so manages conflicts. Interactions between the space and micro-models are formalized by using the "Influence Reaction" principle [11]. Influence is depicted as micro-model's intention to change the space's state, as for Reaction, it is the space's response to the influences produced by the micro-models.

A scheduler to manage micro-models executions Time scale consistency is ensured by a *scheduler*. It releases time step calculations at both macro-model and micro-models levels. The macro-model advances to its next state for each iteration of the *global time step*. Following the same principle but at a lower scale or a larger one, micro-models advance to their next state for each iteration of their respective time steps.
The scheduler's role is to ensure that micro-models and the macro-model are executed according to their respective time step. To do so we revisited an algorithm proposed by [4], in which micro-models executions are performed in a discrete way all along the macro-model's.

Spatial scale management Multi-level models describe interacting processes (each represented by a micro-model) evolving at different space scale. To couple these micro-models, aggregating or disaggregating rules are required to up-scale or down-scale the impacts done by the actions of micro-models. Such functionality is done by a scale transfer feature.
This feature will appear in our approach as a placeholder for future implementation, but will not be treated in this paper.

3. Our approach : UML class diagram and Implementation

In this section, we portray our approach's architecture by using an UML (Unified Modeling Language) class diagram to represent each component then we explain how it works.

3.1. UML class Diagram

We summarize previously detailed positioning in figure 1, which represents the architecture of our approach as an UML class diagram.

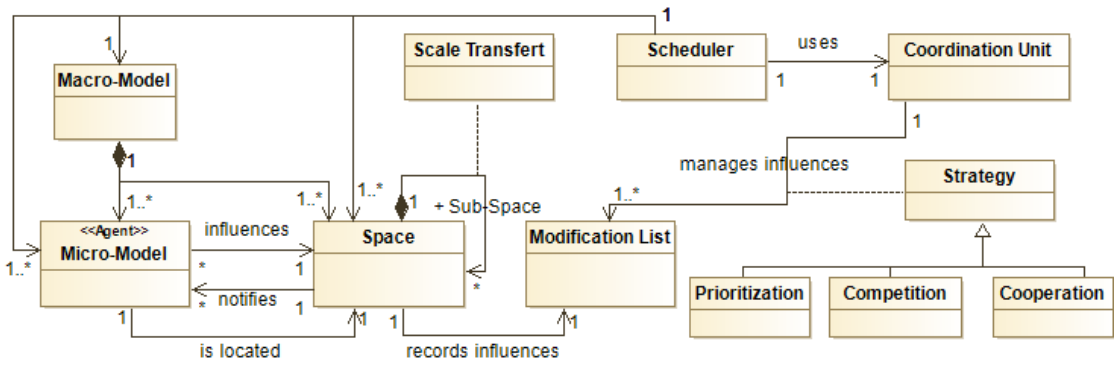

Figure 1: the architecture of our approach as an UML Class diagramm

Macro-Model represents the phenomenon to be modeled. It is composed of *Micro-Model* which are agents, representing the processes that operate inside an environment represented by *Space*.
Space is a multi-leveled structure composed by *Sub-Space* and its scale can be adapted by *Scale Transfer*.

The *Scheduler* manage how often *Macro-Model*, *Micro-Models* and *Space* are executed. Upon execution, Micro-Models send influences to the Sub-space where it is located. These influences are recorded in *Modification List* and the *Coordination Unit* manages them by using *Strategy* that can be *Prioritization*, *Competition* or *Cooperation*.

When the Modification List is altered by the Coordination Unit (by using a selected strategy), the Space notify each micro-model located inside itself of how much resources it used.

3.2. Mechanism

To implement our approach we use Gama platform [12]. It is an open source software dedicated for agent-based modeling and simulation. Gama uses a domain specific language (DSL) called *gaml* or *Gama Modeling Language* , allowing it to be more accessible for non-programmers and domain experts. New features can be integrated in its core by implementing a component called *skill*. A skill is a set of attributes and functions that can be attached to the definition of an agent called *Species*, allowing aforesaid agent to have new actions and attributes.

Connect micro-models to the space We implemented a skill called "Model". It grants new actions to the micro-model such as requesting the use of a resource (i.e sending influence to the space), getting the actual amount of a resource available or how much of a resource it is allowed to use. Each resource request is recorded within the "Modification List", unique to each sub-space.

Scheduling process The scheduler ensure that micro-models, the space and the macro-model are executed according to their respective execution date. We implemented a scheduler inspired from an algorithm proposed by [4].

The *Scheduler* process summarized in figure 2 interact with the *Coordination Unit* (Figure 3) according to the followed algorithm.

The scheduler is initialized by inserting *Micro-models execution Date* (MD) into a list of events called "*Calendar*" (1). The initial value of MD is the micro-model's time step. Then the calendar is sorted by an ascending order of MD (2).
The condition $SD \geq MD$ is checked (3): if it's false, the scheduler proceed to the next simulation cycle (SD + Macro-Model's time step) (9) or stops if the user decide to do so (8). If it's true, every micro-model fulfilling that condition are executed (4); every influences are treated by the Coordination Unit(5, 5.1 to 5.5); Then each executed micro-model's next execution date is computed (MD + micro-model's time step) and the algorithm goes back to execute steps 2 to 9.

Coordination Unit as conflict handler Conflict resolution is realised by a conflict detection and conflict solving. A conflict scenario appears when few micro-models want to access and consume simultaneously an insufficient resource. A conflict happens when the sum of all modifications is greater than the availability of the resource.

Simultaneous changes are detected as the scheduler inquires the coordination unit when micro-models are executed(in figure 2). In figure 3 we summarize the conflict solving process. *E.g* two models M_1 and M_2, with respective resource use α_1 and α_2, and execution time t_1 and

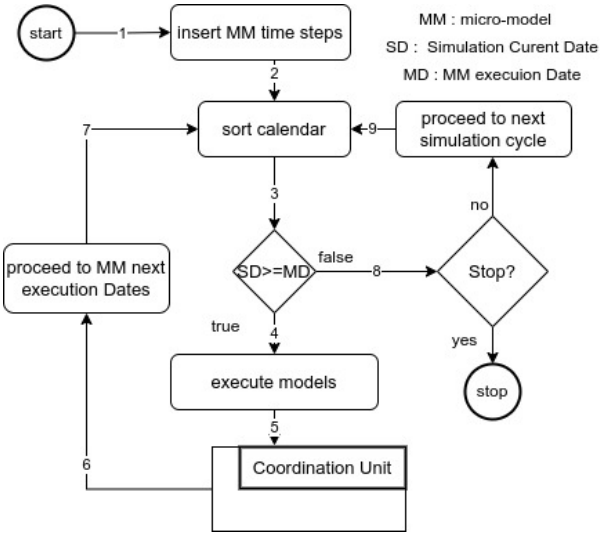

Figure 2: Scheduler Algorithm

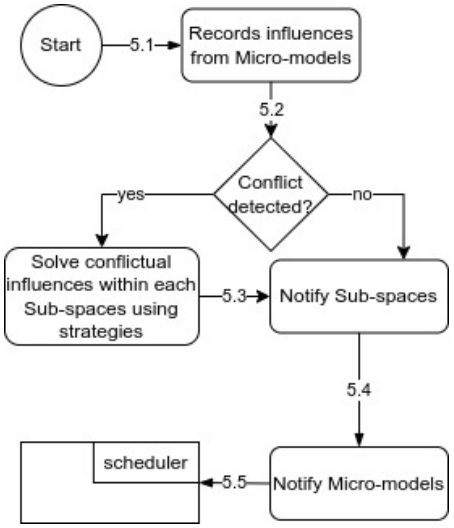

Figure 3: Coordination Unit Algorithm

t_2, modifying a resource R with an amount r. A conflict appears when t_1 = t_2 and [r - (α_1 + α_2)] < 0.

Due to the way micro-models are coupled (through the space), they are not aware of each other state. However, by consulting the modification list, a sub-space can be informed about which micro-models are using its resources.

Before applying any changes to the space, the Coordination Unit filters the Modification List of each Sub-Space by resource: the influences recorded in the Modification List are sorted by resource. If a conflict occurs, a strategy (defined for each resource) is applied to the corresponding set of influences and filter which micro-model can change the resource value. These strategies are implemented into functions described as follows:

- *Prioritization*: for each micro-model, we added a parameter called *priority*. Each influence follows an ascending order of *priority* and consume the resource until depletion or leaves it intact if the resource is not sufficient to support a single influence.
- *Competition*: favors influence according to a selected micro-model. The selected micro-model consumes the available resource first and leave the leftovers to the others.
- *Cooperation*: the resource will be distributed so that each micro-model receive a portion calculated according to their consumption and the available resource. Each amount will be computed according to the formula (1) :

$$\alpha_i^{'} = \frac{\alpha_i r}{\sum_{i=1}^{n} \alpha_i} \tag{1}$$

$\alpha_i^{'}$ is the new resource consumption of a model M_i. α_i is the given resource consumption of a model M_i. r is the amount of resource available. n is the number of micro-modes

using the resource. The new value of the resource is then computed by the formula (2) :

$$r = \sum_{i=1}^{n} \alpha_i^{'} \tag{2}$$

When a resource is conflict-free, the corresponding set of influences recorded inside the modification list can be applied to the space and notified to the micro-models without inquiring the coordination unit to use any strategy.

4. Application: Simple Soil Functioning Model

A soil is naturally a complex multi-scale system in which an enormous amount of entities interact at a different scale. The whole interaction conducts to offer ecosystem services for the society. In this case study, three main processes are outlined : engineering services done by earthworms, carbon decomposition done by bacteria and plant services symbolized by the roots.

The macro-model is fully deterministic, therefore we could compare the effect of using the Coordination Unit or not on simulation results.

4.1. Model Description

Space : Soil model The soil space used in this paper borrows soil properties as proposed in [1]; i.e a soil subspace can be a "pore", a "mineral", an "organic" or a "decomposable". However the decomposable part is ignored as it requires multi-level manipulation features. Thus, the soil we used here is composed of sub-spaces of the same size.
Organic particles represent raw organic matter (`raw_OM`). Depleted organic particles (consumed by worms) become pore particles and activate a given number of microbes inside it.

Pore particles contains living beings (worms, roots and microbes) as well as decomposed organic matter (`decomp_OM`), soluble organic matter (`soluble_OM`) and soluble mineral matter (`soluble_MM`). Mineral particles cannot be decomposed and has zero value of organic matter.

Each organic particle has the same initial amount of `raw_OM`.

`decomp_OM`, `soluble_OM` and `soluble_MM` initial amount are zero.

Three embedded micro-models : Worms, Microbes and Roots Micro-models are the living beings(worms, microbes and roots) inside the soil. They have two actions in common : `eat` allows micro-models to consume a resource and `deject` let them expel transformed resource inside the soil.
The *Worm* micro-model is responsible of changing the soil structure by decomposing `raw_OM` and transforming it into `decomp_OM`. A Worm moves after depleting an organic soil particle.
Microbe micro-model transforms `decomp_OM` into `soluble_OM` and mineralize `soluble_OM` into `soluble_MM`. Then `soluble_MM` are assimilated by microbes and roots.
Roots micro-model feed on `soluble_MM` in order to create a competition (with microbes) on this resource.

Interactions between Micro-models and Soil are defined as following : Competition is defined for the resource `soluble_MM` between microbes and roots. Prioritization is defined for `raw_OM`

(prioritization between worms). Cooperation is defined between microbes for `decomposed_OM` and `soluble_OM`.

4.2. Results

Simulation initial parameters are summarized in table 1.

Table 1
Initial parameters

Parameter	Value
Soil size	10x10, 6x6, 4x4
Raw OM	40g/Organic particle
Global step	2 days
Worm number	1 unit/ pore
Worm step	1 day
Raw OM decomposition	1.3g/worm
Microbe number	5 units/pore
Decomposed OM consumption	0.03g/microbe
Soluble OM consumption(Microbe)	0.007g/microbe
Soluble MM consumption(Microbe)	0.009g/microbe
Microbe step	4 hours
Root number	2 units/pore
Soluble MM consumption(Root)	0.008g/root
Root step	12 hours

Simulation Results We monitor the variation of the available resources (`raw_OM`, `decomp_OM`, `soluble_OM` and `soluble_MM`) over 60 days, in multiple space configuration, then we compare for each configuration how the simulation model behave when using or not the coordination unit's strategies. Different configurations are used to check the result's differences when the number of micro-models and sub-space are rising.

In figure 4 and 7 we compare the results of a 4x4 space, in figure 5 and 8 the results of 6x6 space and in figure 6 and 9 we compare the results of a 10x10 space.

It is also possible to consult the modification list at a sub-space level. A log file records all the computation done by the coordination unit while handling resource conflicts. Some sample in figure 10, 11 and 12. Model is the agent's name, parameter the resource's name, alpha the amount of resource to be used, T is the execution time, r is the amount of resource available (lower than the sum of resource demanded by the models, i.e there is a resource conflict), cycle is the number of simulation steps done when the conflict occurred.

Discussion The results show the effect the Coordination Unit on a simple deterministic model in which micro-models compete for resources. The Coordination Unit limits micro-models behaviours by using strategies to filter their actions in a conflict scenario.
On the 4x4 configuration, the results (figure 4 and 7)are slightly different, and the difference

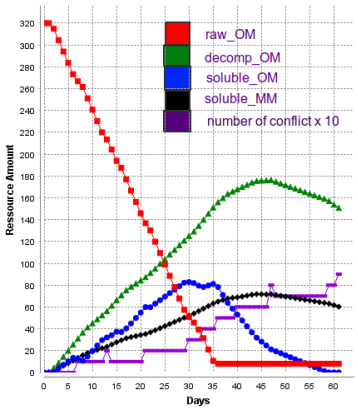

Figure 4: 4x4 sub-spaces : not using the coordination unit

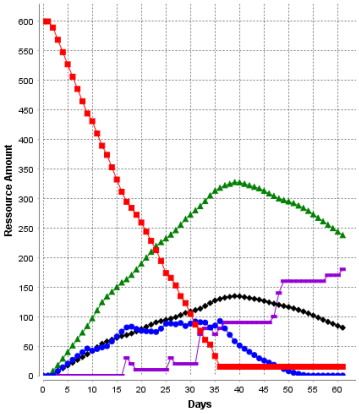

Figure 5: 6x6 sub-spaces : not using the coordination unit

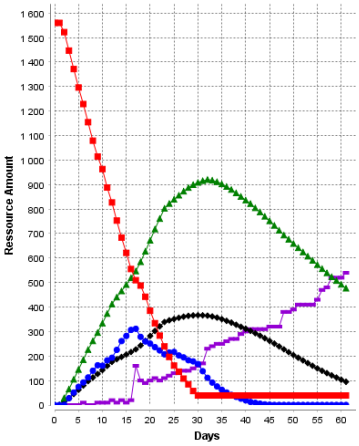

Figure 6: 10x10 sub-spaces : not using the coordination unit

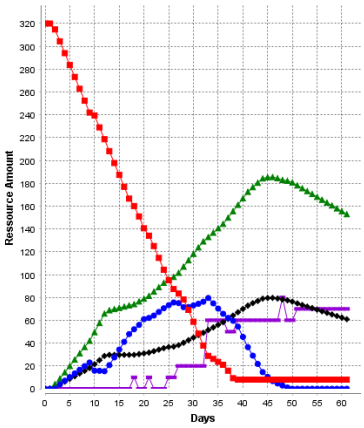

Figure 7: 4x4 sub-spaces : using the coordination unit

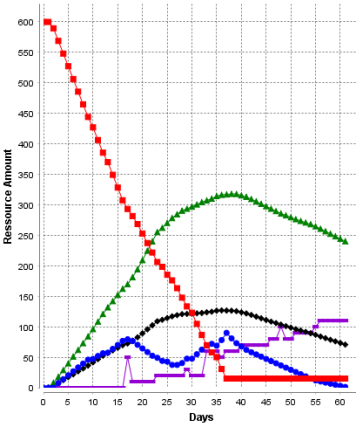

Figure 8: 6x6 sub-spaces : using the coordination unit

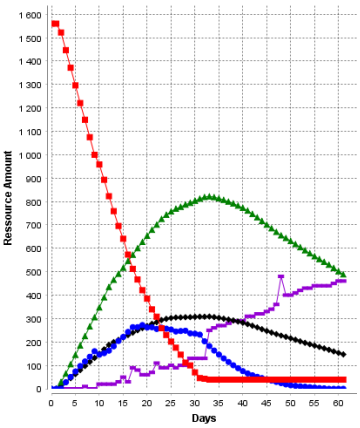

Figure 9: 10X10 sub-spaces : using the coordination unit

between the results of the same configuration becomes greater as you move to a larger space (6x6 and 10x10).

When conflicts occurs (purple) potentially exploitable resources can be leftover by micro-models if no strategies are applied to handle conflicts. In figure 4, 5 and 6 available resources (`raw_OM`, `decomp_OM`, `soluble_OM`, and `soluble_MM`) are always slightly higher than the other results in which strategies are applied (figure 7, 8 and 9) due to the leftovers resources caused by mismanaged conflicts. Thus, using the coordination unit's strategies, leads to more precise results.

These comparisons show that using the coordination unit can greatly affect the simulation results even if changes are made at the micro-model level. For instance in figures 10, 12 and 11, operations computed by the coordination unit affect how resources are exploited by micro-models and consequently affect the general behavior the macro-model (figure 7, 8, 9).

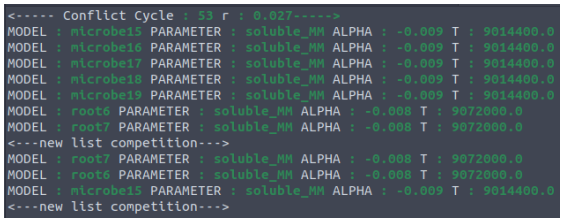

Figure 10: Sample of conflict handling using the competition strategy

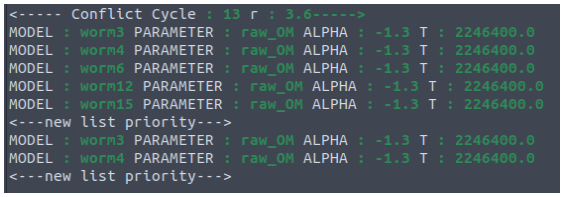

Figure 11: Sample of conflict handling using the prioritization strategy

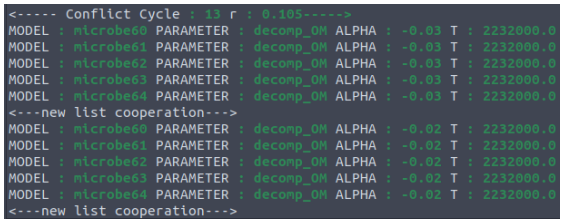

Figure 12: Sample of conflict handling using the cooperation strategy

Spatial model coupling approach such as [5, 4] does not take into account conflict handling, we found that poorly managed conflict situations can lead to inconsistencies. Therefore, it is advised to add a conflict management tool for such coupling strategies. Since the studies in the area of Multi-Agent Systems have a dedicated branch for that matter, it is possible to improve the basic strategies proposed in this paper in order to get more correct results.
Using the Influence and Reaction principle [11] allows this model coupling approach to handle simultaneous actions, however, additional rules have to be defined in order to specify how influences are applied to the space. The Coordination Unit provides strategies to define these rules. Additional strategies can be added in order to define new rules by manipulating influences inside the Modification list.
For now, the applicability of our approach is limited to model coupling approaches in which micro-models communicates through their environment by using its resources. The basic strategies used in the coordination unit can also be improved by using more advanced techniques related to agent coordination. Heterogeneous and multi-scale aspects of the space [1] could be dug for future improvement to show how this approach can behave in such spatial configuration.

5. Conclusion

In this paper, we propose an agent-based spatial model coupling approach to solve resource conflict problems in a system where micro-models are scattered in the space and interact through its resources. This study highlights the importance of handling conflicts in spatial model coupling as it improves the behaviour of micro-models, thus correct the overall behaviour of the simulation. The approach enhances micro-model interactions by solving resource conflicts. The proposed approach allows more flexibility to the modeler as it brings multiple functions to coordinate micro-models in the event of a conflict. Observed results suggest that using the coordination unit changes the simulation results by forcing micro-models to adopt restrained behaviours in conflict scenario, enhancing the coherency and the correctness of the results when using the space's resources. As a continuation of our work, more coordination functions could be added in the future, in order to improve the effectiveness and versatility of our approach. We also intend to use this approach in a multi-level space that can incorporate granulometic data to represent soil space more realistically.

References

[1] N. Marilleau, C. Cambier, A. Drogoul, J.-L. Chotte, E. Perrier, E. Blanchart, Multiscale mas modelling to simulate the soil environment: Application to soil ecology, Simulation Modelling Practice and Theory 16 (2008) 736–745. doi:10.1016/j.simpat.2008.04.021.

[2] E. Blanchart, N. Marilleau, J.-L. Chotte, A. Drogoul, E. Perrier, C. Cambier, Sworm: An agent-based model to simulate the effect of earthworms on soil structure, European Journal of Soil Science 60 (2009) 13 – 21. doi:10.1111/j.1365-2389.2008.01091.x.

[3] D. David, D. Payet, A. Botta, G. Lajoie, S. Manglou, R. Courdier, Un couplage de dynamiques comportementales: le modèle ds pour l'aménagement du territoire., JFSMA 7 (2007) 129–138.

[4] I. Hassoumi, Approche multi-agents de couplage de modèles pour la modélisation des systèmes complexes spatiaux: application à l'aménagement urbain de la ville de Métouia, Ph.D. thesis, Paris 6, 2015.

[5] D. Payet, R. Courdier, N. Sebastien, T. Ralambondrainy, Environment as support for simplification, reuse and integration of processes in spatial mas, in: 2006 IEEE International Conference on Information Reuse Integration, 2006, pp. 127–131.

[6] J. Ferber, Les sma: vers une intelligence collective, Inter Editions, Paris (1995).

[7] A. Banos, N. Corson, C. Lang, N. Marilleau, P. Taillandier, Multiscale modeling: Application to traffic flow, in: Agent-based Spatial Simulation with NetLogo, Volume 2, Elsevier, 2017, pp. 37–62.

[8] B. Zeigler, S. Vahie, Devs formalism and methodology: Unity of conception/diversity of application, in: Proceedings of 1993 Winter Simulation Conference - (WSC '93), 1993, p. 573–579.

[9] T. Blockwitz, M. Otter, J. Akesson, M. Arnold, C. Clauss, H. Elmqvist, M. Friedrich, A. Jung-

hanns, J. Mauss, D. Neumerkel, et al., Functional mockup interface 2.0: The standard for tool independent exchange of simulation models, 2012, p. 173–184.
[10] M. U. Awais, P. Palensky, A. Elsheikh, E. Widl, S. Matthias, The high level architecture rti as a master to the functional mock-up interface components, in: 2013 International Conference on Computing, Networking and Communications (ICNC), 2013, pp. 315–320.
[11] J. Ferber, J.-P. Müller, Influences and reaction: a model of situated multiagent systems, in: Proceedings of second international conference on multi-agent systems (ICMAS-96), 1996, pp. 72–79.
[12] P. Taillandier, B. Gaudou, A. Grignard, Q.-N. Huynh, N. Marilleau, P. Caillou, D. Philippon, A. Drogoul, Building, composing and experimenting complex spatial models with the gama platform, Geoinformatica 23 (2019) 299–322. URL: https://doi.org/10.1007/s10707-018-00339-6. doi:10.1007/s10707-018-00339-6.

A. Online Resources

Github - Soil Organic Matter Decomposition Model - Gama source code
Github - Spatial Model Coupling Gama plugin.