A declarative modelling language for the design of complex structured agent-based epidemiological models

Vianney Sicard, Mathieu Andraud, Sébastien Picault

A declarative modelling language for the design of complex structured agent-based epidemiological models

Vianney Sicard[1] [0000−0002−4909−5544], Mathieu Andraud[2] [0000−0003−2891−2901], and Sébastien Picault[1] [0000−0001−9029−0555]

[1] INRAE, Oniris, BIOEPAR, 44300, Nantes, France
{vianney.sicard, sebastien.picault}@inrae.fr
[2] ANSES, Ploufragan-Plouzané-Niort Laboratory, Health and Welfare Research Unit,
Ploufragan, France – mathieu.andraud@anses.fr

Abstract. To provide realistic accounts of disease spread in host populations, epidemiological modelling requires ever-finer details, including a complex spatio-temporal structuring. Multi-level agent based systems have proven effective in tackling such epidemiological dynamics, revealing nevertheless the need for coupling multi-level architectures and organizational features. The solution proposed here attempts to overcome this challenging task, through the development of a declarative domain-specific language to facilitate the separation of concerns and thus expert knowledge integration. We illustrate our approach with an application to the spread of the swine Influenza A between the batches and buildings of a pig farm, to demonstrate the impact of control measures based on modifications of the structural farm features.

Keywords: Epidemiological Modelling · Multi-level Agent-based Simulation · Design Patterns · Organizational System · Complex Systems · Knowledge and software engineering.

1 Introduction

Epidemiological modelling has been of growing interest for several years due to its relevance and support to decision-making. Applied to animal health, such models provide quantitative insights to evaluate and anticipate the implementation of control measures to curb the spread of pathogens, and represent decision-support tool for stakeholders. However, for these tools to be operational, they need to remain realistic in regard to field situation, accounting for the structure of contacts between animals, and the different levels of observation (farm, herd, individual, etc.). For this purpose, agent-based simulation (ABS) has proven its real value [12] to consider behaviours, interactions, and the monitoring of individuals. Furthermore, multi-level agent-based system (MLABS) provides solutions to deal with populations evolving at different scales and in different environments [9]. Especially, the introduction of MLABS-specific design patterns [8, 7] allows for an explicit separation of concerns embedded in a generic approach. The EMULSION framework, developed for multi-level epidemiological modelling, is a proof of concept for this approach, offering a separation between a generic simulation engine implementing MLABS design patterns, and a declarative domain-specific language (DSL) [10].

However, to be as close as possible to the situation on the field, and thus implementing the most appropriate control measures, the modelling requires a very fine level of detail, the spatio-temporal structure and dynamics of the population being a key component for anticipating the spread of pathogens. For instance, in the case of pig farming systems, management with batch structuring, i.e. the evolution of groups of individuals according to their physiological statuses, was shown to be pivotal for several pathogens. However, such models present a very high programming complexity, implying difficulties to maintain, adapt or revise them. To tackle this issue, we identified the need to introduce organizational features in MLABS, discussed possible solutions and proposed an organizational multi-level agent-based pattern (OMLABS) [18].

In this paper, we focus on supporting this pattern through a DSL and demonstrate a concrete application to a complex epidemiological model. The paper is organized as follows: section 2 introduces the proposed organizational design pattern with its specifications. Section 3 presents its implementation through a declarative DSL to extend EMULSION, which makes it possible for modellers to focus on the description of the spatio-temporal features of the model (section 3). In section 4, we demonstrate an application to the modelling of the spread of the swine Influenza A virus (SwIAV) within a pig farm.

2 An organizational design pattern for multi-level agent-based systems

To address the coupling between MLABS and organizational concepts, we proposed a solution based on a design pattern architecture [18]. Design patterns provide a generic, reusable and modular solutions to solve a specific problem [5]. Recent developments recommend design principles based on design patterns [7], as in other areas of agent-based simulation [6], to improve the genericity and reusability of conceptual solutions to recurring problems. The OMLABS pattern is based on previous MLABS patterns [7] coupled with organizational concepts [4], transcribed in the notion of organization-groups-atom, with an agentification by each of these levels of agents, which encapsulates an environment [18] (Fig. 1). The introduction of organizational concepts into MLABS allows organizations to have an explicit and flexible representation [18] and aims at addressing three issues: 1) to express and formalize the structural relationship between agents ; 2) to constrain the behaviours and interactions between agents ; 3) to be involved in controlling the environment, both structurally and functionally.

In addition to these general concepts, OMLABS implements specific features to handle multi-level environment dynamics and provide flexible modelling capabilities. The **recursivity**, a specific aspect of the pattern, allows for any sublevel

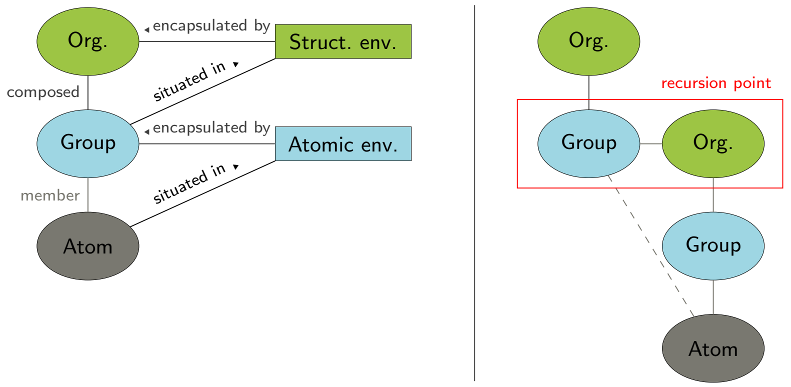

Fig. 1. Structure of the OMLABS pattern [18]. Agent *Organization* encapsulates a *structured environment* composed of *atomic environments*, themselves encapsulated by *Group* agents. Agents involved as *Atoms* belong to an organization by their location in an *atomic environment*, according to the constraints applied by the *Organization*. On the right, the recursive use of the pattern: for instance, a *Group* can itself be a new instance of the pattern. Spaces can be either spatial or social.

(group or atom) to play itself the role of an organization. The **consistent allocation** of atoms into an organization and its groups is managed by the *Organization agent*, according to explicit rules for acceptance, maintaining and location in spaces. **Information** can be stored by each environment and propagated to other environments. This information can be of two kinds: 1) *material*, i.e. with a transfer of information quantity during transmission, involving a loss of this quantity by the source (e.g. pheromone diffusion) ; 2) *informational*, i.e. with a conservation by the source of quantity of information during the transmission (e.g. message broadcast).

This OMLABS design pattern approach has been validated by a proof of concept [18] and proved efficient to interact with modellers. However, going from the conceptual model to its implementation presents real difficulties (allocation rules, recursivity, information propagation, etc.), and it is therefore necessary to provide modellers with an easier way to specify these features. From a model-driven engineering point of view, a convenient solution is to delegate the implementation part to a generic engine, in charge of instantiating several features of the pattern, which are themselves expressed through a convenient DSL, in a declarative way. To do so, we relied on the EMULSION framework, which is specifically built on this separation of concerns.

EMULSION is a generic framework for mechanistic epidemiological modelling, allowing modellers to focus on the modelling issues rather than on computer coding issues. It provides indeed an explicit description of the models through its own declarative DSL, allowing models to be flexible, revisable and modular [10]. The simulation engine of the framework relies on a MLABS architecture [11], based on environment and MLABS design patterns [8, 7], which helps reify every relevant observation level in the pathosystem. In addition, state machines are used at each scale to formalize the processes involved in the epidemiological model (infection, demography, control, etc.). Hence, the framework EMULSION as a MLABS is built as the combination of a structure (the architecture of nested agents and environments) and a function (an explicit description of the processes), allowing agents to represent any kind of entity.

3 A DSL for integrating organizational features into a multi-level agent-based system

Though the OMLABS design pattern is clear to modellers as a conceptual frame, specifying programmatically how to apply the pattern within a MLABS architecture for a given model is quite difficult, thus it requires a specific language to be independent of programming issues. Several agent-oriented programming languages are already used broadly [19, 13, 21], but require programming skills and thorough understanding of concepts specific to agent-based modelling, such as scheduling or behaviours. In contrast, modellers in epidemiology have no need to learn such general purpose languages if they are provided with a language dedicated to epidemiological modelling issues and focused on knowledge elicitation. Hence, we developed a declarative DSL, based on the YAML syntax and, as for EMULSION, built on main sections divided into indented blocks, representing OMLABS concepts and their features (e.g. the state machines, see fig. 2).

This DSL for organizational features integrates with the existing EMULSION DSL [10], as the pattern architecture complements EMULSION metamodel. The coupling of the YAML syntax and the MLABS concepts simplifies the modification of the model structure by adding or removing blocks (e.g. adding states or transitions in a state machine).

3.1 Organization and spaces

The core concepts of the OMLABS pattern, i.e. organization-group-atom with an agentification of environments, are transcribed with the notions of *spaces* corresponding to the *structured space* and *nodes* composed of *atomic spaces* where the atoms are located. (Fig. 3).

In the main block of the organization statement, the section *execute_process* corresponds to the ordering of the organization processes, i.e. the dynamic of the environments (Fig. 3). Processes refer to predefined actions that need to be executed at the beginning of each simulation steps, e.g. *allocate*, *propagate*, *cleaning*, etc. The *spaces* block corresponds to the *group* level of the pattern, which is composed of nodes. The first subsection corresponds to the declaration of the allocation rules, and the second one to the description of the different nodes, i.e. the *atomic spaces*. Each of these spaces has information statements, with the specification of their initial values. The last block is the definition of the weighted graph handling information propagation between the *atomic spaces*.

```
health_state:
 desc: "The state machine which defines the evolution of health states"
 states:
   - S:
     name: 'Susceptible'
     desc: 'suceptible of becoming infected'
   - I:
     name: 'Infectious'
     desc: 'infected and able to transmit the disease'
     duration: 5
   - R:
     name: 'Resistant'
     desc: 'healthy again and resistant to infection'
  transitions:
   - {from: S, to: I, rate: force_of_infection}
   - {from: I, to: R, proba: 1}
   - {from: R, to: S, rate: immunity_loss}
```

Fig. 2. Example of the declaration of the state machine *health_state* with EMULSION DSL. The transition between each state depends on the values of the rates, or after five days for the transition between I to R.

```
farm:
 name: 'farm'
 description: 'organization level of the pattern'
 execute_process:
  - allocate
 spaces:
  name: 'buildings'
  description: 'the farm is composed of buildings'
  allocation:
   - if: 'is_S OR is_R'
    then: main_building
   - if: 'is_I'
    then: quarantine_building
  nodes:
   - main_building:
     informations:
      viral_load:
       value: 0
   - quarantine_building:
     informations:
      viral_load:
       value: 5
  graph:
   - 'main_building ->(viral_load: 0) quarantine_building'
   - 'quarantine_building ->(viral_load:0.6) main_building'
```

Fig. 3. Example of the declaration of an organization *farm*, which is composed of two buildings: *main_building* and *quarantine_building*. The environment can store an amount of viral load, initialized for each environment (0 and 5, resp.). The allocation rules depend on the health states of the individuals: infected ones are located in the *quarantine_building*, the others in the *main_building*. At the end, the graph section define the relation of propagation between each node.

The distinction between material and informational propagation of information is defined by the two keywords *propagate_matter* and *propagate_information*.

3.2 Recursive application of the pattern

One of the specificities of the pattern is its ability to be used recursively: a group can itself be an organization composed of groups. Therefore, the node of the parent organization refers to a child organization (Fig. 1). Propagation between levels – the parent and the child organizations – is notified at the child level with two notions: *from_upper* and *to_upper*. It is thus possible to have a global coverage of the graph, and to know the value of an information at each level (Fig. 4).

```
farm:
  ...
    nodes:
      - main_building:
          reference: main_building

main_building:
  name: ’main_building’
  description: ’sub-organization’
  execute_process:
    - allocate
  spaces:
    name: ’room’
    description: ’buildings are composed of rooms’
    informations:
      viral_load:
        value: 0
        from_upper: 0.5
        to_upper: 0.4
    allocation:
      - ’alternate(spaces=ALL)’
    nodes:
    ...
```

Fig. 4. Example of recursive use. Parent organization *farm* is composed of buildings, and the *main_building* is itself an organization. The child organization *main_building* is composed of rooms. The statement of the child organization is the same as a "normal" organization. The allocation rule specifies that individuals are located in each room alternatively. The transmission between the rooms and the whole building is specified by *from_upper* and *to_upper*, so the graph covers all levels.

4 Application: Spread of swine Influenza A virus in a farrow-to-finish pig farm

To illustrate the complexity of structured systems in epidemiological modelling, more precisely the structuring of populations in space and time, the batch rearing management, commonly practised in pig production herds, is particularly suitable. Indeed, both physiological evolution and the groupings of individuals are closely linked to their location.

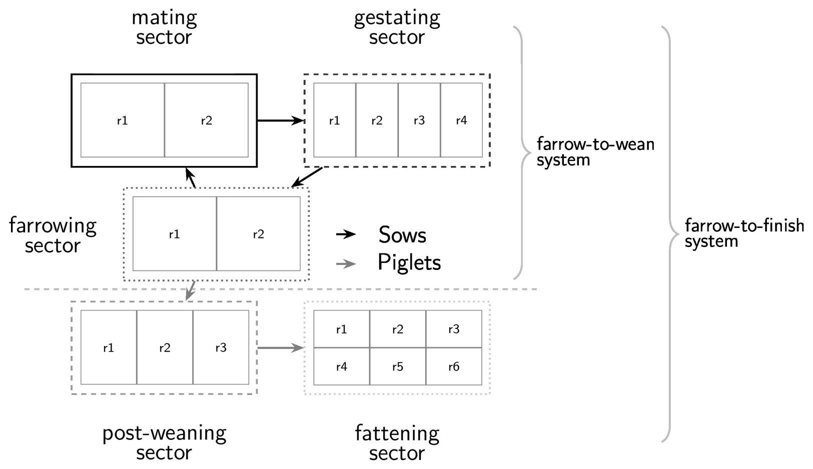

Fig. 5. Flow diagram of sows and piglets in the different sectors, each divided into several rooms.

4.1 Batch rearing management

Batch management is a specific feature of pig breeding, consisting in grouping sows that evolves at the same times. More precisely, a sow evolves through different physiological steps (insemination, gestation, farrowing). After farrowing, each sow returns to insemination while its offspring, i.e. the piglets constituting the litter, evolve separately (weaning, fattening). The grouping of litters and sows evolving at the same time constitutes a batch. Successive batches are separated by a constant delay. In our application case, we modelled a management in 7 batches with a between-batch interval of 21 days.

The farm is composed of sectors, and each sector is divided into rooms. Each sector corresponds to a physiological state for sows or piglets. Individuals are allocated in a room according to their batch, one batch per room at a time (Fig. 5). As batch management is known to impact the spread of pathogens, this complexity of setting up a highly structured population must be modelled explicitly.

4.2 Implementation in OMLABS

The model represents two kinds of pig farms: farrow-to-wean farms which sell weaned piglets, and farrow-to-finish farms which sell fattened piglets directly to slaughterhouses (Fig. 5). A parameter of the state machine *physiological_step* allows to easily switch between both systems. Both are driven according to a 7-batch rearing system, the allocation of sows to a batch is given in the initialization part (Fig. 6). Only piglets (new individuals) have to be allocated after farrowing, so the allocation rules obviously specify that piglets are put in the same batch as their mothers (Fig. 7). The time spent in each physiological step is constant in each stage in a given batch (Table 1). After a gestating period of 112 days, pregnant sows give birth to litters of 12 piglets. At weaning age (28 days), the piglets leave the system for the farrow-to-wean system, or they continue their evolution through post-weaning and finishing sectors in the farrow-to-finish context.

```
sow_batch1:
  animal_type: Sow
  physiological_step: Insemination
  health_state: S
  organization: [batches, housing]
  init_location:
    batches: batch1
```

Fig. 6. Initialization rules for sows in batch 1. Initially, batch 1 is composed of susceptible sows (S) at physiological step *Insemination*. The allocation in the different spaces of the sectors is directly managed by the organization *housing*, and the allocation in batches is explicitly defined in the *init_location*.

```
      allocation:
        - if: ’is_Piglet’
          then: ’with_source()’
          else: ’alternate_by_group(group=batch, spaces=ALL)’
```

Fig. 7. New piglets are placed in the same room as their mother (*with_source*), and sows are alternately placed in rooms according to their batch, with one batch per room at a time.

Physiological step	Duration (days)
Insemination	35
Gestating	77
Suckling	28
Post-weaning	61 (for batches 1,2, and 4)
	54 (otherwise)
Fattening	121 (for batch 5)
	114 (otherwise)

Table 1. Duration for each physiological step [20].

4.3 Epidemiological model

The OMLABS pattern was used to model the spread of a Swine Influenza A virus (SwIAV) in a pig farm with batch rearing management. SwIAVs are widely spreading in pig production units, causing respiratory disorders and recognized as a main agent responsible for the Porcine respiratory disease complex (PRDC) [16]. It has been shown that SwIAV remains endemic in herds, inducing successive and regular infection waves in batches of growing pigs [14, 3].

In what follows, we explain how we extended an existing model [20] by adding both a batch management and states to the epidemiological model to comply with a realistic French pig farm and investigate the impact of batch rearing. We developed the individual-based, discrete-time stochastic model through a MSEIR model with five health states: maternal immunity (M), susceptible (S), exposed (E), infectious (I), and recovered (R). Recovered/Immune sows deliver maternally derived antibodies (MDAs) to their piglets, conferring them a partial protection against infection [1, 2, 20].

Each scenario was run on two different farming systems: a farrow-to-wean system similar to the one described in White et al. [20] with a mating, gestating and suckling sector (piglets then leave the system after suckling), and a typical farrow-to-finish farm, including post-weaning and fattening sectors. Control strategies were based on increasing biosecurity levels, affecting the indirect transmission between rooms and sectors. For this purpose, three scenarios were developed [17]: 1) *Reference scenario* corresponds to the assumptions in [20], i.e. the same transmission level between rooms and between sectors ; 2) *Full isolation* assumes only direct transmission (i.e., no transmission between sectors nor between rooms) ; 3) *Partial isolation* represents a reduction of indirect transmission. Parameters values were modulated through a logarithmic scale from 10^0 to 10^{-5}.

The design of the different scenarios according to the two farming systems was facilitated by the DSL, allowing for the implementation of contrasted scenarios without programming code, and to switch from one scenario to another by modification of a single parameter.

4.4 Results

The simulations were run over 645 days with a burning period of two sow cycles (2×140 days), to reach a demographic equilibrium, after which the transmission process was initiated with the introduction of an infected sow in batch 1 at the very beginning of the third cycle (insemination stage), i.e., on day 280. Each scenario was repeated for 100 times.

The results show a first peak during the first 25 days after introduction of the virus and the persistence thereafter. The different waves correspond to the infection of successive batches, in line with field observations [15]. For the full isolation scenario, the infection was obviously contained into batch 1. For the farrow-to-wean system, the value of the decrease of the transmission parameter

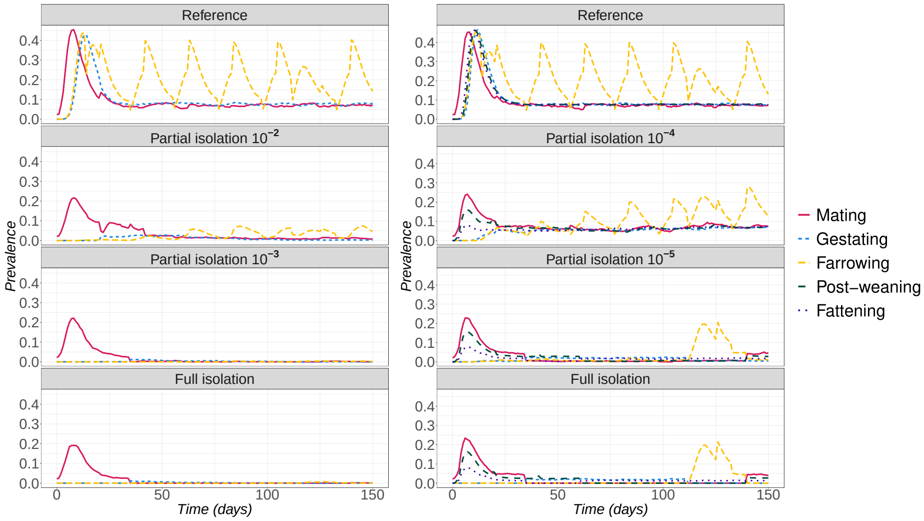

Fig. 8. Average prevalence per sector for the farrow-to-wean system (left), and for the farrow-to-finish (right). Each row represents a scenario.

was between 10^{-2} and 10^{-3}, and for the farrow-to-finish system between 10^{-4} and 10^{-5} (Fig.8).

Not only did the results confirmed the major role of piglets and the airborne transmission on infection persistence – which are in line with White's model – but also that control measures based on the isolation of rooms and sectors should be really effective to prevent disease from spreading in the whole herd (Fig. 8).

5 Conclusion and perspectives

We have presented an organizational multi-level agent-based pattern which, combined to the epidemiological framework EMULSION, makes it is possible to take into account the spatio-temporal structure of the population in mechanistic epidemiological modelling. By extending the DSL of this framework, our method gives the ability to model the complexity of highly structured population. The separation of concerns allows experts to easily express the knowledge integration with modularity, readability and revisability. It is thus possible to explicitly represent organizational aspects of the population (social and spatial), coupled with other processes, such as the spread of pathogens.

Models of SwIAV spread in pig farms had already been developed, but with a very high programming complexity, implying difficulties to maintain, adapt or revise [1, 2, 20]. In contrast, our modelling approach, with the ability to explicitly and easily represent a structured population, provides flexibility, revisability and reproducibility to the model. Assumptions can be explored using parameters (e.g. farming system, buildings division, contact structure, etc.) without compromising the system.

The application of the OMLABS pattern to the modelling of the swine Influenza A virus in batch rearing management highlighted specific behaviours of the pathosystem towards the application of various control measures. Indeed, the OMLABS pattern allowed to represent the dynamics of SwIAV in two different types of farms using a general modelling framework. Although the transmission dynamics were found similar in farrow-to-wean and farrow-to-finish pig farms, the inclusion of post-weaning and finishing sectors required much more biosecurity effort to mitigate the spread of the virus. This result, showing the impact of the structural configuration of the production site, evidences the benefit of the organizational pattern allowing for a direct comparison of selected scenarios through the modification of a single parameter in the DSL. This proof of concept will be applied to the modelling of several European pig farming systems, paving the way for the extension of use cases to broader eco-environmental issues.

Finally, the ability to handle complex systems with a solution coupling flexible MLABS patterns and DSL offers opportunities for the specification of complex multi-level agent-based simulations. This issue is all the more crucial as complex systems modelling often involve scientists from various fields, not necessarily skilled enough in computer science to dive into programming complex simulation architectures. A DSL approach appears then a relevant way to foster efficient interactions.

Acknowledgements

This work is supported by a grant from the Animal Health division of INRAE (French national research institute for agriculture, food, and environment) and the French region Pays de la Loire.

References

1. Allerson, M.W., Cardona, C.J., Torremorell, M.: Indirect Transmission of Influenza A Virus between Pig Populations under Two Different Biosecurity Settings. PLoS ONE **8**(6), e67293 (2013)
2. Cador, C., Hervé, S., Andraud, M., Gorin, S., Paboeuf, F., Barbier, N., Quéguiner, S., Deblanc, C., Simon, G., Rose, N.: Maternally-derived antibodies do not prevent transmission of swine influenza A virus between pigs. Vet.Res. **47**(1) (2016)
3. Chastagner, A., Bonin, E., Fablet, C., Quéguiner, S., Hirchaud, E., Lucas, P., Gorin, S., Barbier, N., Béven, V., Garin, E., Blanchard, Y., Rose, N., Hervé, S., Simon, G.: Virus persistence in pig herds led to successive reassortment events between swine and human influenza A viruses, resulting in the emergence of a novel triple-reassortant swine influenza virus. Vet.Res. **50**(1), 77 (2019)
4. Ferber, J., Michel, F., Baez, J.: AGRE: Integrating Environments with Organizations. In: Weyns, D., Van Dyke Parunak, H., Michel, F. (eds.) Environments for Multi-Agent Systems. pp. 48–56. Lect.n in Computer Science, Springer (2005)
5. Gamma, E., Helm, R., Johnson, R., Vlissides, J.: Design Patterns: Elements of Reusable Object-Oriented Software. Addison-Wesley (1994)
6. Juziuk, J.: Design Patterns for Multi-Agent Systems (Jun 2012)

7. Mathieu, P., Morvan, G., Picault, S.: Multi-level agent-based simulations: Four design patterns. Simulation Modelling Practice and Theory **83**, 51–64 (Apr 2018)
8. Mathieu, P., Picault, S., Secq, Y.: Design patterns for environments in multi-agent simulations. In: Chen, Q., Torroni, P., Villata, S., Hsu, J., Omicini, A. (eds.) Proceedings of the 18th Conference on Principles and Practice of Multi-Agent Systems (PRIMA 2015). vol. 9387, pp. 678–686. Springer (2015)
9. Morvan, G.: Multi-level agent-based modeling - A literature survey. arXiv:1205.0561 [cs] (May 2012), arXiv: 1205.0561
10. Picault, S., Huang, Y.L., Sicard, V., Arnoux, S., Beaunée, G., Ezanno, P.: EMULSION: Transparent and flexible multiscale stochastic models in human, animal and plant epidemiology. PLOS Computational Biology **15**(9), e1007342 (Sep 2019)
11. Picault, S., Huang, Y.L., Sicard, V., Ezanno, P.: Enhancing Sustainability of Epidemiological Models through a Generic Multilevel Agent-based Approach. In: 26. International Joint Conference on Artificial Intelligence (IJCAI). p. np. Melbourne, Australia (Aug 2017)
12. Roche, B., Guégan, J.F., Bousquet, F.: Multi-agent systems in epidemiology: a first step for computational biology in the study of vector-borne disease transmission. BMC Bioinformatics **9** (2008)
13. Rodriguez, S., Gaud, N., Galland, S.: SARL: A General-Purpose Agent-Oriented Programming Language. In: 2014 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT). vol. 3, pp. 103–110 (Aug 2014)
14. Rose, N.: Modélisation de la dynamique d'infection par le PCV2 en naissage-engraissement (Mar 2014)
15. Rose, N., Hervé, S., Eveno, E., Barbier, N., Eono, F., Dorenlor, V., Andraud, M., Camsusou, C., Madec, F., Simon, G.: Dynamics of influenza A virus infections in permanently infected pig farms: evidence of recurrent infections, circulation of several swine influenza viruses and reassortment events. Veterinary Research **44**(1), 72 (2013)
16. Salvesen, H.A., Whitelaw, C.B.A.: Current and prospective control strategies of influenza A virus in swine. Porcine Health Management **7**(1), 23 (Dec 2021)
17. Sicard, V., Picault, S., Andraud, M.: Coupling spatial and temporal structure in batch rearing modelling for understanding the spread of the swine influenza a. In: Society for Veterinary Epidemiology and Preventive Medicine. p. 11. SVEPM (2022)
18. Sicard, V., Andraud, M., Picault, S.: Organization as a Multi-level Design Pattern for Agent-based Simulation of Complex Systems. In: Proceedings of the 13th International Conference on Agents and Artificial Intelligence. pp. 232–241. SCITEPRESS - Science and Technology Publications, Austria (2021)
19. Taillandier, P., Gaudou, B., Grignard, A., Huynh, Q.N., Marilleau, N., Caillou, P., Philippon, D., Drogoul, A.: Building, composing and experimenting complex spatial models with the GAMA platform. GeoInformatica **23**(2), 299–322 (Apr 2019)
20. White, L., Torremorell, M., Craft, M.: Influenza A virus in swine breeding herds: Combination of vaccination and biosecurity practices can reduce likelihood of endemic piglet reservoir. Preventive Veterinary Medicine **138**, 55–69 (Mar 2017)
21. Wilensky, U., Rand, W.: An introduction to agent-based modeling: modeling natural, social, and engineered complex systems with NetLogo. The MIT Press, Cambridge, Massachusetts London, England (2015), oCLC: 911027259