



HAL
open science

A Multi-Level Multi-Agent Simulation Framework in Animal Epidemiology

Sébastien Picault, Yu-Lin Huang, Vianney Sicard, François Beaudeau, Pauline
Ezanno

► **To cite this version:**

Sébastien Picault, Yu-Lin Huang, Vianney Sicard, François Beaudeau, Pauline Ezanno. A Multi-Level Multi-Agent Simulation Framework in Animal Epidemiology. 15th International Conference on Practical Applications of Agents and Multi-Agent Systems (PAAMS'2017), Jun 2017, Porto, Portugal. pp.209-221, 10.1007/978-3-319-59930-4_17. hal-01536640

HAL Id: hal-01536640

<https://inria.hal.science/hal-01536640>

Submitted on 12 Jul 2017

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

A Multi-Level Multi-Agent Simulation Framework in Animal Epidemiology

Sébastien Picault^{1,2}, Yu-Lin Huang¹, Vianney Sicard¹, François Beaudeau¹, and
Pauline Ezanno¹

firstname.lastname@oniris-nantes.fr

¹Bioepar, INRA, Oniris, La Chantrerie, 44307 Nantes, France

²Univ. Lille, CNRS, Centrale Lille, UMR 9189 - CRISTAL (SMAC), F-59000 Lille, France

Abstract. In order to recommend better control measures in public or animal health, epidemiologists incorporate ever-finer details in their models, from individual diversity to public policies, which often involve several observation scales. Due to the variety of modelling paradigms, it becomes more and more difficult to compare hypotheses and outcomes, all the more that the increased complexity of simulation programs is not yet counterbalanced by design principles nor by software engineering methods. We propose in this paper to use the multi-level agent-based paradigm to integrate existing methods within a common interface, provide a separation between concerns and reduce the part of code devoted to model designers. We illustrate our approach with an application to the Q fever disease in cattle.

Keywords: Epidemiological Modelling; Multi-Level Agent-Based Simulation; Knowledge and Software Engineering;

1 Introduction

The work presented here takes place within Project MIHMES¹ in the context of cattle epidemiology. To gain a deeper understanding of disease spread processes and to identify efficient control measures, this research studies enzootic livestock diseases at several scales. Various models were developed for either intra-herd or inter-herd spread, using either a compartment-based or individual-based approach, in many programming languages (Scilab, C++, Python, R). They proved relevant to account for field observations and helped designing recommendation strategies to control disease spread. Yet, they lack genericity from a software engineering viewpoint, so that *the exploration of new hypotheses often requires a substantial coding effort*. Besides, the choice of the modelling paradigm is a strong constraint which is the backbone of each simulation program, though it would often be quite useful to replace compartment models by Individual-Based Models (IBM) or vice-versa. This situation is highly representative of the difficulties encountered in epidemiological modelling and thus is taken in the paper as a use-case for testing new software approaches. We believe that Multi-Agent

¹ <http://www6.inra.fr/mihmes>

Based Simulations (MABS) are a convenient solution for providing a homogeneous interface to several modelling paradigms. Besides, the concepts and techniques developed in Multi-Level MABS are fully relevant to implement epidemiological models, because of their connection to the issues raised by the multi-level nature of infectious processes, farming techniques, and cattle trade.

The paper is organized as follows: section 2 discusses the main existing paradigms in epidemiological modelling. The design principles and the framework proposed to answer above issues are presented in section 3, and illustrated with an application to Q fever disease in section 4.

2 An overview of epidemiological modelling paradigms

2.1 Classical methods: equations and compartments

Since the seminal work of Kermack and McKendrick [12], the classical approach to epidemiological modelling relies upon the partition of the population into several *compartments*, each representing a number of individuals sharing a homogeneous state. This strong assumption allows considering only flows of individuals moving among compartments. This is usually described through a flow diagram as the famous “SIR” model (Fig. 1), where three health states are represented: *Susceptible* (S), where individuals can become infected due to the contact with Infectious individuals; *Infectious* (I), where individuals are likely to contaminate others; *Recovered* (R), where individuals are not contagious anymore and cannot be reinfected. The state variables S, I, R , i.e. the amount of individuals in each compartment, are controlled by an Ordinary Differential Equation (ODE) system based on transition rates between compartments. They can be computed either in a continuous, deterministic way, or in a discrete, stochastic way after transforming rates into probabilities and applying multinomial sampling.

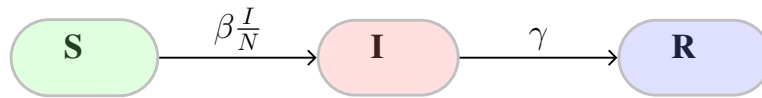


Fig. 1. Flow diagram of the classical SIR model. Nodes are compartments measuring number of individuals in each health state; edges are labeled with the flow rates.

This compartment-based paradigm, in addition to allowing analytical insights, is quite flexible. Input and output rates can be used to integrate demographical dynamics. If needed, compartments can easily be subdivided into finer-grain boxes, for instance to account for spatial areas or age groups. Several species can also be modelled, e.g. in vector-borne diseases. In addition, the role of environmental contamination can be explicitly introduced through one or more dedicated compartments [14].

Yet, it is not very convenient to account for the individual diversity that is encountered in biological parameters (e.g. in the susceptibility to diseases), or to integrate behavioural considerations, such as seasonal moves, or prevention and control measures.

When several concerns are to be taken into account simultaneously in addition to health states, the only possibility offered by compartment-based approaches consists in subdividing compartments again and again (e.g. [14]), which finally resembles very much individual-based models, or to complexify the formalism [18].

2.2 The rise of Individual and Agent-Based Models

Due to this drawback, a growing number of models use Individual-Based Models (IBM), which allow an explicit integration of the diversity of *individual states* (e.g. [7]), or even to Agent-Based Models (ABM), where the diversity of *individual behaviours* and *interactions* between individuals are native features (e.g. [2,20]). The main advantage of IBM/ABM is to be almost indefinitely extensible, since the introduction of new hypotheses essentially leads to introduction of new agents or behaviours. The dynamics of the model can be studied in details, and domain entities are represented very straightforwardly by computational entities. It is also easier to represent multifactorial processes and gain a deeper understanding of causal mechanisms of epidemiological systems [15].

As a counterpart, the computational cost is much higher. Not only processing an IBM is generally slower than a compartment model, but the number of repetitions required to study the model is also greater. Indeed, the detail level is at the cost of additional parameters, thus the sensibility analysis has to be carried out more thoroughly. In addition, the enhanced facility for changing hypotheses naturally leads to the temptation of multiplying the number of scenarios to assess and compare control measures and public policies. Besides, most of these works focus on a practical case and specific pathogens, where Multi-Agent Systems prove a quite convenient tool, but to our knowledge none is dedicated to drawing MAS towards a true paradigm for epidemiological modelling, with a generic methodology and reusable algorithms and architectures.

2.3 The emergence of multi-level simulation

Additional issues are at stake when coming to a regional or national scale, since the spatial dynamics cannot be ignored, and models developed at different scales have to be coupled [4].

A classical approach used to handle epidemiological models at a regional scale is the ecological concept of metapopulation [9]. A metapopulation is a system of interconnected local populations living in isolated patches, each of them endowed with its own epidemic dynamics. Contacts between local populations may occur due to neighbourhood relations, movements, or transport of pathogens via hosts or wind. This method reduces the computational cost that would result from a pure IBM approach, at the expense of a coarse-grained modelling of sub-populations dynamics. This approach is very popular in human epidemiology, since it allows addressing large-scale populations and areas, where contact structures are rather well described. Besides, this approach can be handled either in a deterministic or stochastic way. Yet, compared to equivalent multi-agent based simulations, metapopulations models may lead to overestimate infections [1,11].

Conversely, the computational cost of classical MABS at this scale is quite high. Unless using a massively parallel platform with simple epidemiological assumptions

and a high level of software optimizations [17], handling millions of agents is an issue, all the more when repetitions are required to compare scenarios in a credible way.

As it appears from this diversity of modelling paradigms, there is definitely no “ultimate solution” that would fit every situation. On the contrary, a multi-purpose modelling framework should be able to provide tools for allowing a combination of all these methods depending on the target context.

3 A multi-level modelling method and framework

We first assume that models have to be *designed* from the beginning *as multi-level models*, instead of building for instance an intra-herd model and trying to scale it to the regional scale, or conversely trying to “zoom” within each herd of a metapopulation model by adding finer-grain features. This perspective, far from idealistic, can be achieved through a high degree of modularity concerning two aspects: 1) the structure of models, for which a multi-level agent-based approach seems well suited, and 2) an explicit decomposition of all processes (infectious dynamics, cattle management, trade...) involved in the system. In this section, we present the design principles and the software architecture we propose to answer above issues.

3.1 Structural modularity: multi-level modelling

The number of research works on multi-level agent-based simulations (ML-ABS) is significantly growing in recent years. These simulations use agents to reify organization, observation or scale levels at the same time. In cattle epidemiology, the reification of intermediate entities (e.g. age groups, herds...) between the animal and the population, as well as the use of associated spatial areas (pens, farms, pastures...) is a major advantage to assess fine-grained control measures. Yet, most contributions are designed for specific application fields. Hence, we use one of the very few existing generic multi-level meta-models, PADAWAN [19], which features many design and implementation properties we are seeking for computational epidemiology. Especially, the PADAWAN meta-model enforces a *strong separation between declarative and procedural concerns*, through the independence of behaviours (specified as interaction rules) from the agents that can perform or undergo them. Agents and environments can be associated through two relations: *situation* (agents can interact within several environments) and *encapsulation* (agents can “contain” an environment in order to host other agents). Thus, a multi-level MABS is built as the *combination of a structure* (the architecture and organization of nested agents and environments) *and a function* (an explicit and intelligible description of the processes involved in the system). Since agents can represent any kind of entity, they provide a homogeneous, yet polymorphic interface to integrate multiple modelling paradigms.

3.2 Functional modularity: knowledge engineering

Besides, we argue that field knowledge (parameters, assumptions, processes, data...) introduced in a model must be split with respect to the diversity of concerns, as pointed

out by [6]. Therefore, the first task in model design is the identification of distinct processes. We also believe that a multi-level modelling frame intended for epidemiology experts should be *as little intrusive as possible* with respect to their habits, so as to induce only refinements and clarifications of existing methods. Thus, we propose the following principles:

1. *Enhance existing formalisms so as to make them unambiguous, but not abstruse.* For instance, the ODD protocol proposed in [10] is a first step towards the elicitation of expert knowledge, but remains an ambiguous textual template [3]. On the contrary, using powerful formalisms originated from physical multi-scale processes [8], or from molecular biology [13], would be inappropriate as regards epidemiological concerns.
2. *Provide methodological guidelines aimed at identifying typical issues and at answering them with convenient, standardized solutions,* as Design Patterns do in software engineering.
3. *Automate as much as possible the common tasks involved in the simulation process,* so as to let the final users develop only specific, tiny pieces of code.

To stay as close as possible to existing modelling formalisms, we propose to transform the usual flow diagram into a true finite state machine. To do so, we consider that nodes (states) and edges (transitions) can be endowed with additional information (Fig. 2). States are given optional features: 1) a *duration distribution*, which specifies how long an individual is likely to stay in current state, and which is helpful to describe e.g. demographical effects, and 2) *actions* performed when entering the state, being in the state, or leaving the state, e.g. to handle shedding in infectious states. In addition to their label which can represent either a rate, a probability, or an absolute number of individuals, transitions can be enhanced with: 1) *crossing conditions*, so as to determine which agents from source state are allowed to move towards destination state, and 2) *actions* performed by individuals when crossing the edge, i.e. after leaving the source state and before entering the destination state.

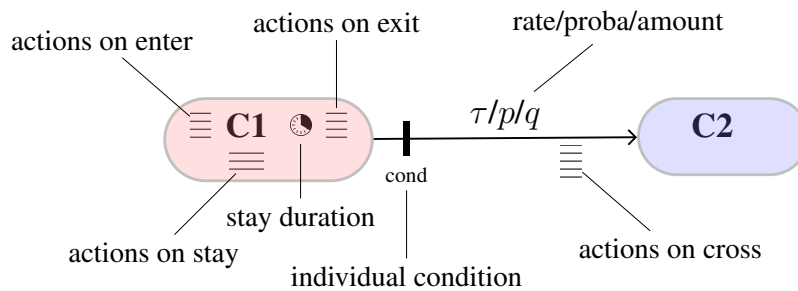


Fig. 2. Enhancements of the flow diagram in a state-machine style.

These additions correspond to features that are ordinary hard-coded when coming to the implementation of a model specified through a classical flow diagram. Moving these

elements very early in the design process is not much intrusive, contributes to a more accurate view of the whole model, and allows including them in a code-generation process. Moreover, the state-machine diagram can be used indifferently in a compartment-based (deterministic or stochastic) or agent-based approach. All information regarding the processes involved in a model, the corresponding state machines with their states, transitions, conditions, actions, durations and rate/probability/amount parameters can be specified in a YAML configuration file, which is processed to generate the simulation architecture, but can also be used for producing a technical documentation, figures, commenting the sources and assumptions, etc.

3.3 Architecture of the EMuLSion framework

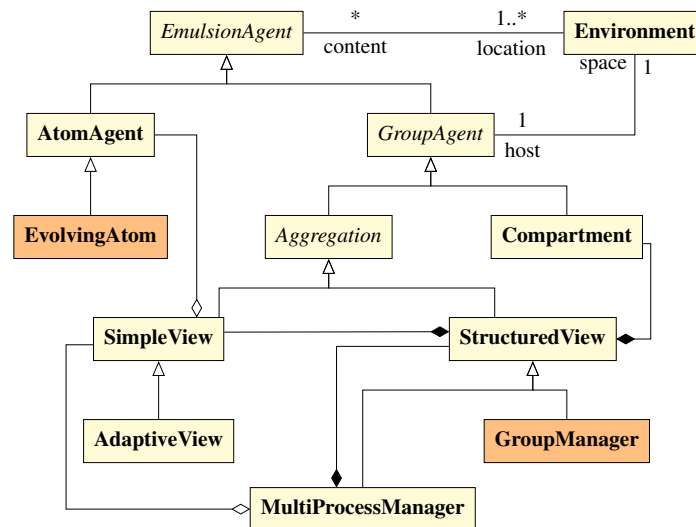


Fig. 3. Class diagram of the multi-level agent hierarchy in the EMuLSion framework. Agent classes with an orange background have a state-machine driven behaviour.

These design principles have been implemented and experimented through a Python framework, called “EMuLSion”². In this framework, a model is composed of processes (a sequence of actions upon agents), some of them driven by state machines, and using parameters. Parameters include rates, probabilities, amounts, distributions... given either in values, ranges, or expressed as a function of other parameters. They are parsed using a symbolic mathematics library (SymPy) and the consistency of the model is checked automatically.

Class related to agents and to their model-specified behaviours are shown on figure 3. All Multi-Level Agents are situated in at least one environment. They are divided

² EMuLSion stands for “Epidemiological MUlti-Level SimulatIOn framework”.

into two categories: *atoms* and *groups*. The `AtomAgent` class represents “individuals” (e.g. animals). Its `EvolvingAtom` subclass represents individuals endowed with state-machine driven behaviours. On the contrary, groups of agents can encapsulate a local environment where other agents can be situated. They are themselves composed of two families: *Compartments* and *Aggregations*. The `Compartment` class represents the situation where individuals are homogeneous enough to be aggregated into a simple amount. An `Aggregation` provides a *view* over individuals or other groups, i.e. a representation where agents are gathered according to customizable variables, such as health state, age group, etc. More specifically:

- `SimpleView` agents host individuals (atoms) and schedule their behaviour. An `AdaptiveView` also detects individuals which have a different value of specific variables than others, and ask their own host to put them in the proper place.
- The purpose of `StructuredView` agents is to associate `SimpleView` agents or `Compartment` agents with possible values of specific variables. For instance, in a SIR model, the `health_state` variable can take three values (S, I, R), hence is associated to three `Compartment` agents or to three `SimpleView` agents. The `GroupManager` does the same, but using a state machine to determine how individuals (or amounts) change their value for those variables.
- Finally, the `MultiProcessManager` can handle several processes in the same simulation. It relies upon at least a `SimpleView` to control all individuals, and a `StructuredView` or `GroupManager` agents to manage the different processes involved in the model.

3.4 Usage patterns

This composition-based architecture allows representing the modelling paradigms found in computational epidemiology. *Compartment-based models* are built using instances of the `Compartment` class, hosted by a `GroupManager` endowed with a state machine specifying the health states and their transitions (Fig. 4a). `Compartment` agents can adopt a deterministic or a stochastic behaviour on demand. *Individual-based models* are composed of `EvolvingAtom` agents, each endowed with their own state machine(s), and hosted by a `SimpleView` agent (Fig. 4b). Finally, *metapopulations* can be represented using a `StructuredView` or a `MultiProcessManager` holding several agents constructed after one of the latter architectures, with additional processes describing contact structure.

Additional solutions can be built, especially by grouping individuals according to their state. The main advantage of using regroupments of individuals is to provide a straight access to similar individuals, especially when determining which individuals have to change state. For instance, in a stochastic approach, one multinomial sample per group can be used instead of one Bernoulli trial per individual, which is significantly more efficient. To do so, the first step consists in building a `GroupManager` agent, endowed with a state machine for one concern, e.g. the infectious process affecting the `health_state` variable (figure 5). Each value of the `health_state` is associated with an `AdaptiveView`, containing `AtomAgents`. Atoms move from one state to another according to the state machine. If an external process affects `health_state`

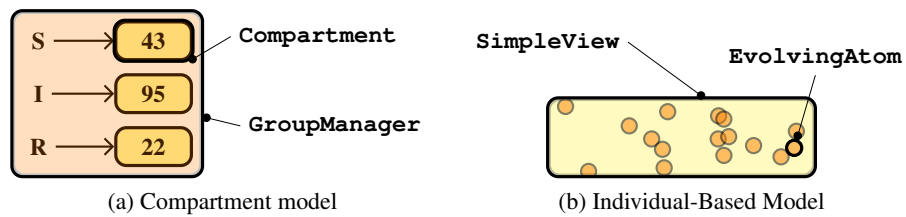


Fig. 4. (a) Structure of a pure compartment-based model. The GroupManager is endowed with a state machine to determine flows between compartments. (b) Pure IBM simulation structure. EvolvingAtoms are endowed with their own state machines and hosted by a SimpleView agent.

of agents, for instance a cure changing “I” into “R”, the “I” AdaptiveView detects the change and asks the GroupManager to put modified atoms in the right place (“R” AdaptiveView).

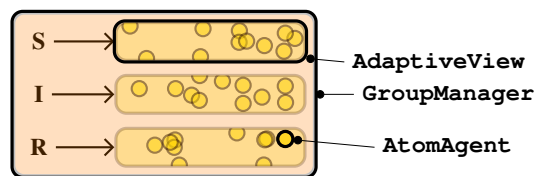


Fig. 5. Structure of an aggregation of individuals into adaptive views. Atoms are grouped according to their common states. The GroupManager is endowed with the state machine to induce state changes in atoms.

The MultiProcessManager agent handles a combination of several analogous structures (figure 6). This agent is given the list of processes affecting individuals or groupings, and the variables associated with each grouping. If a process is bound to a state machine, it automatically uses a GroupManager with the structure described above, otherwise it does the same with a simple StructuredView. Each individual is thus accessible from a global list (SimpleView) and from each of the groupings.

The current version of EMuLSion counts about 3,200 lines of code (in Python). The framework has been extensively tested on several variations of the theoretical SIR-like models, to check that all usage pattern described above produce equivalent results. In the next section, we show how these usage patterns are deployed for the study of a real disease.

4 Application to Q fever in cattle

To illustrate the interest of our approach, we explain here how it is applied to one of the diseases studied in project MIHMES: Q fever, a zoonosis affecting mainly ruminants. Two specific individual-based models have been already developed for this disease in

cattle and are used for comparison, one for the within-herd spread [7], and one for the inter-herd spread [16].

4.1 Q fever model

Our implementation of the Q fever model in dairy herds is based on [7] with 6 health states: susceptible (S), infectious without immune response (I-), infectious with immune response (I+) and bacterial shedding in milk (I+m), and carrier with (C+) or without (C-) antibodies. Contaminations occur through bacterial shedding in the environment. Besides, only adult female cows are taken into account. They follow a farming process (“life cycle”) suited for milk production where cows are inseminated, get pregnant (P state), calve (PC) or abort (A) depending on their health state, and wait before new pregnancy (BP). Calving and abortion bring a high level of bacterial shedding.

During each time step, the processes involved at the herd level are the following: 1) bacterial decay in the environment (exponential decrease); 2) culling process to remove cows depending on their parity (i.e. number of calvings); 3) replacement process to introduce new animals in the herd; 4) infection process based on the state machine affecting health state; 5) farm management based on the state machine affecting life cycle; 6) actualization of the grouping of animals by parity.

4.2 Implementation, tests, and future work

To implement this model, we could use a classical IBM approach, but to benefit from the adaptive grouping of individuals, the most suited architecture consists in defining a class for individuals, namely `QfeverCow` which extends `AtomAgent`, and one for the herd (`QfeverHerd`) as a subclass of `MultiProcessManager`.

The processes involved in the model require that individuals are grouped according to three criteria: parity (for culling), life cycle and health state (both driven by a state machine). The amount of bacteria shedded by infectious animals also depends if they recently calved or not, thus using an additional boolean variable to control the grouping for the infection process is more efficient, all the more than those groups are created only when needed. The resulting architecture is shown on figure 6.

The implementation of the Q fever model within the EMuLSion framework was first tested to reproduce the outcomes of the reference intra-herd model [7]. Figure 7 shows the comparison between both on two main outputs (disease prevalence, i.e. proportion of infectious animals, and amount of bacteria excreted in environment). In addition, we are able to test rapidly alternative hypotheses, especially regarding the simplification of the intra-herd model, in order to keep the core mechanisms responsible for the disease spread before introducing inter-herd contamination. This exploratory work will be published in a separate article.

The original intra-herd [7] and inter-herd [16] models were respectively composed of 1,000 and 2,500 lines of assumption-specific code (in R and Python). The EMuLSion implementation involves a small amount of classes developed for Q fever (circa 250 lines), and a YAML configuration file for describing the state machines, processes, and parameters composed of about 300 key-value pairs, including textual comments.

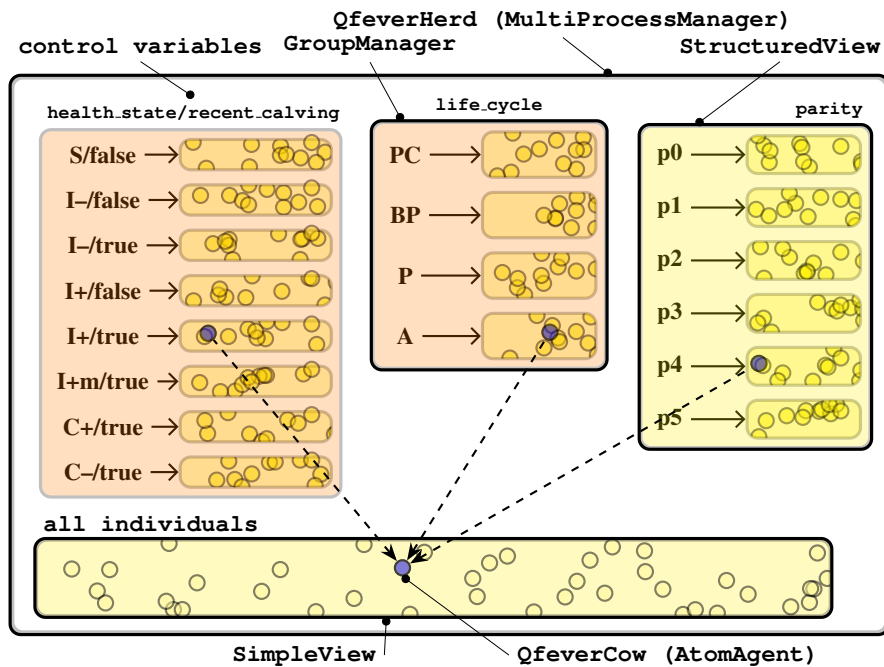


Fig. 6. Structure of the intra-herd model of the Q-Fever disease. Individuals are aggregated according to concern-related variables. Individuals (e.g. the blue one in health state I^+ , life state A and parity 4) can be accessed through each concern or through a global list.

Ongoing work now focuses on the integration of this model to an inter-herd simulation, using again a `MultiProcessManager` to represent the metapopulation of herds. In Q fever, contamination between herds occurs because of airborne dissemination of bacteria and because of the introduction of infected animals through animal trade. This only affects two processes of the intra-herd model: the renewal process (based on sales and purchases) and the bacterial dispersion. New procedures are currently elaborated to account for these mechanisms, on the basis of existing trade and meteorological data.

5 Conclusion and perspectives

We have presented a generic approach based on the organization of multi-level nested agents to design and implement epidemiological models. Our method offers a homogeneous way to integrate quite easily classical methods such as compartment-based and individual-based models. Besides, it relies upon a separation of declarative and procedural concerns, first to keep the underlying hypotheses intelligible and revisable, and second to make the simulation engine generic and reliable. Though the development of the framework itself requires advanced skills in computer science, its use for designing and simulating models is made easy for epidemiologists, who have very little code (and

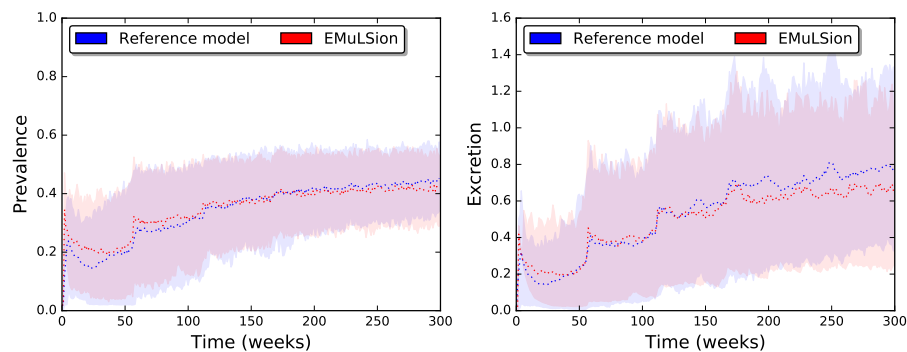


Fig. 7. Average (200 repetitions) and 5–95% percentiles for the prevalence and environmental excretion of bacteria in reference model [7] and in EMuLSion implementation of Q fever.

simple) to write, and conversely forced to write down their assumptions in detail. This framework has been already applied to re-design the intra-herd model of Q fever and test possible simplifications, in order to build an efficient and relevant between-herd model. Other models developed during the MIHMES project will be re-engineered in a similar way to ensure the sustainability and extensibility of the modelling experience.

More generally, this work is a first step towards the collective elaboration of a common modelling frame to allow an accurate comparison of epidemiological models. The ability to make all assumptions explicit and describe precisely all the processes and interactions involved in a system, and the confidence in a generic, yet versatile simulation engine, are a crucial requirement for scientific reproducibility. Besides, the capability of using indifferently and transparently one paradigm or another paves the way for exploring automatic switch techniques between them, as suggested in [5].

In a wider perspective, we also believe that reducing the duration of design, implementation and testing of new models will be a key stake to answer epidemiological urges (avian or human flu, mosquito-borne diseases), increased by global warming which could lead to a multiplication of unexpected situations.

Acknowledgements

Project MIHMES is funded by the French Research Agency, Program Investments for the Future (ANR-10-BINF-07) and the European fund for the Regional Development (FEDER) of Pays-de-la-Loire. The research work presented here is also funded by the Animal Health Division of the French National Institute for Agricultural Research (INRA).

References

1. Ajelli, M., Gonçalves, B., Balcan, D., Colizza, V., Hu, H., Ramasco, J.J., Merler, S., Vespignani, A.: Comparing large-scale computational approaches to epidemic modeling: Agent-

- based versus structured metapopulation models. *BMC Infectious Diseases* 10(1) (2010)
2. Amouroux, E., Desvaux, S., Drogoul, A.: Towards virtual epidemiology: An agent-based approach to the modeling of H5N1 propagation and persistence in north-vietnam. In: 11th Pacific Rim Int. Conf. on Multi-Agents (PRIMA). LNCS, vol. 5357, pp. 26–33. Springer (2008)
 3. Amouroux, E., Gaudou, B., Desvaux, S., Drogoul, A.: O.D.D.: A promising but incomplete formalism for individual-based model specification. In: IEEE RIVF Int. Conf. on Computing and Communication Technologies. IEEE (2010)
 4. Beaunée, G., Vergu, E., Ezanno, P.: Modelling of paratuberculosis spread between dairy cattle farms at a regional scale. *Veterinary Research* 46(1) (2015)
 5. Bobashev, G.V., Goedecke, D.M., Yu, F., Epstein, J.M.: A hybrid epidemic model: Combining the advantages of agent-based and equation-based approaches. *Winter Simulation Conference* (2007)
 6. Bui, T.M.A., Stinckwich, S., Ziane, M., Roche, B., Ho, T.V.: KENDRICK: A domain specific language and platform for mathematical epidemiological modelling. In: IEEE RIVF Int. Conf. on Computing and Communication Technologies. pp. 132–137. IEEE (2015)
 7. Courcoul, A., Monod, H., Nielen, M., Klinkenberg, D., Hogerwerf, L., Beaudéau, F., Vergu, E.: Modelling the effect of heterogeneity of shedding on the within herd *Coxiella burnetii* spread and identification of key parameters by sensitivity analysis. *J. Theor. Biol.* 284(1), 130–141 (2011)
 8. Díaz-Zuccarini, V., Pichardo-Almarza, C.: On the formalization of multi-scale and multi-science processes for integrative biology. *Interface Focus* 1(3), 426–437 (2011)
 9. Grenfell, B., Harwood, J.: (meta)population dynamics of infectious diseases. *Trends in Ecology & Evolution* 12(10), 395–399 (1997)
 10. Grimm, V., et al.: A standard protocol for describing individual-based and agent-based models. *Ecological Modelling* 198(1–2), 115–126 (2006)
 11. Keeling, M.J., Danon, L., Vernon, M.C., House, T.A.: Individual identity and movement networks for disease metapopulations. *PNAS* 107(19), 8866–8870 (2010)
 12. Kermack, W.O., McKendrick, A.G.: A contribution to the mathematical theory of epidemics. *Proc. R. Soc. (A115)*, 700–721 (1927)
 13. Le Novère, N., et al.: The systems biology graphical notation. *Nat. Biotech.* 27(8), 735–741 (2009)
 14. Marcé, C., Ezanno, P., Seegers, H., Pfeiffer, D., Fourichon, C.: Predicting fadeout versus persistence of paratuberculosis in a dairy cattle herd for management and control purposes: a modelling study. *Veterinary Research* 42(1), 36 (2011)
 15. Marshall, B.D., Galea, S.: Formalizing the role of agent-based modeling in causal inference and epidemiology. *Am. J. Epidemiology* 181(2), 92–99 (2014)
 16. Pandit, P., Hoch, T., Ezanno, P., Beaudéau, F., Vergu, E.: Spread of *Coxiella burnetii* between dairy cattle herds in an enzootic region: modelling contributions of airborne transmission and trade. *Veterinary Research* 47(1) (2016)
 17. Parker, J., Epstein, J.M.: A distributed platform for global-scale agent-based models of disease transmission. *ACM Trans. Model. Comput. Simul.* pp. 2:1–2:25 (2011)
 18. Perra, N., Balcan, D., Gonçalves, B., Vespignani, A.: Towards a characterization of behavior-disease models. *PLoS ONE* 6(8), e23084 (2011)
 19. Picault, S., Mathieu, P.: An interaction-oriented model for multi-scale simulation. In: 22nd Int. Joint Conf. on Artificial Intelligence (IJCAI). pp. 332–337 (2011)
 20. Robins, J., Bogen, S., Francis, A., Westhoek, A., Kanarek, A., Lenhart, S., Eda, S.: Agent-based model for Johne’s disease dynamics in a dairy herd. *Veterinary Research* 46(1) (2015)