

Enhancing Sustainability of Complex Epidemiological Models through a Generic Multilevel Agent-based Approach

Sébastien Picault^{a, b}, Yu-Lin Huang^a, Vianney Sicard^a and Pauline Ezanno^a

^aBioepar, INRA, Oniris, La Chantrerie, 44307 Nantes, France

^bUniv. Lille, CNRS, Centrale Lille, UMR 9189 - CRIStAL, F-59000 Lille, France
firstname.lastname@oniris-nantes.fr

Abstract

The development of computational sciences has fostered major advances in life sciences, but also led to reproducibility and reliability issues, which become a crucial stake when simulations are aimed at assessing control measures, as in epidemiology. A broad use of software development methods is a useful remediation to reduce those problems, but preventive approaches, targeting not only implementation but also model design, are essential to sustainable enhancements. Among them, AI techniques, based on the separation between declarative and procedural concerns, and on knowledge engineering, offer promising solutions. Especially, multilevel multi-agent systems, deeply rooted in that culture, provide a generic way to integrate several epidemiological modeling paradigms within a homogeneous interface. We explain in this paper how this approach is used for building more generic, reliable and sustainable simulations, illustrated by real-case applications in cattle epidemiology.

1 Introduction

The huge development of automated data gathering, together with the spread of computer simulation, has fostered major advances in many areas of life sciences in the last two decades. Now, computational models are broadly used, for providing deeper insights on mechanisms involved in complex systems, and recommendation tools aimed at assessing public policies. This is becoming a common practice in health-related fields, where the explanation of processes and their control are tightly coupled, like epidemiology. This epistemological evolution obviously implies a high confidence in computer programs developed to implement models, process input data, produce simulation outputs, and sometimes identify the best policies with respect to a given purpose. The reliability of codes is a first-order requirement for elaborating sound scientific evidence to support the implementation of control measures, impacting public health and economic activities without possible backtrack. Thus, *reproducibility*, i.e. the capability to recalculate model outputs

from data, was identified early as “an attainable minimum standard” [Peng, 2006] for computational sciences.

Yet, most published models are far from allowing reproducibility, because life scientists are not trained to the arduous task of programming complex models [Merali, 2010]. Reproducibility could be achieved through software development methods aimed at keeping traces from the whole development process: e.g., accurate documentation, software versioning, raw data preservation, standardized tools instead of handmade operations, intensive testing, and finally, diffusion of source codes [Sandve *et al.*, 2013]. But, as highlighted more recently [Leek and Peng, 2015], reproducibility is not enough. Results may be reproducible, and yet erroneous or lead to misinterpretations. Thus they advocate for a “prevention approach” relying upon early education efforts, and call for the identification of methods that can be used by novice programmers.

We believe that a further step can be achieved using Artificial Intelligence techniques, such as the separation between explicit expert knowledge, and procedures to handle it. Enforcing a clear description of assumptions, parameters, contact structures, etc. *outside simulation codes* in computational epidemiology would not only afford a higher intelligibility level of models, but also drastically accelerate their revision, extension and counterfactual assessment. In addition, epidemiological models can be built according to several main paradigms, which are difficult to compare and often lead to quite different software architectures: equation-based compartment models, individual-based models, and metapopulations. A growing stake consists in finding efficient solutions to encompass existing paradigms, both in model design and implementation, hence facilitating their comparison and their combination within the same simulation, as suggested by [Bobashev *et al.*, 2007].

Our proposal to do so is based on the multi-agent paradigm. Epidemiological models are intrinsically multi-scale complex systems, where interactions between individuals belonging to highly structured groups, and living in nested environments, account for disease spread dynamics. Hence, multilevel agent-based systems appear a rather natural solution, offering concepts and architectures to model any organization or observation level as an agent endowed with its own behaviors, and to express relations between entities belonging to the lower or upper level.

This paper is organized as follows. Next section presents an overview of epidemiological modeling paradigms and issues raised by their implementation, leading to a taxonomy of existing works. Then, we propose enhancements to usual modeling techniques, implemented in a framework based on a multilevel agent-based architecture. We explain its application to real diseases, and discuss further extension and long-term perspectives.

2 Issues in Computational Epidemiology

In this section, we intend to identify two main difficulties encountered when modeling and programming epidemiological models: the choice of the modeling paradigm and the lack of methodological guidelines for software implementation.

2.1 The Diversity of Modeling Paradigms

Most epidemiological models are based on one of the three paradigms below, depending on the granularity and scale they tackle, but also on the detail level of control measure they want to assess.

Compartment-based models

Compartment-based models (CBM) are historically the first approach developed for handling diseases spread from a mathematical point of view [Kermack and McKendrick, 1927]. They assume that individuals differ only by their state, hence describe the spread dynamics through several partitions (*compartments*), characterized by the number of individuals in each state, and the transition rate to other compartments. These models are usually represented with a flow diagram (Fig. 1). For instance, the classical “SIR” model assumes three health states: *Susceptible* (S), where individuals can become infected due to contacts with infectious individuals; *Infectious* (I), where they are likely to contaminate others; *Recovered* (R), where they are not contagious anymore and cannot be reinfected. In a deterministic, continuous approach, the state variables S, I, R (amount of individuals in each state) evolve according to an Ordinary Differential Equation (ODE) system directly drawn from transition rates, which enables mathematical analyses. In a discrete, stochastic approach, rates are translated into probabilities used in multinomial sampling.

CBM can incorporate many additional features. For instance, edges from or to “invisible” compartments, with associated input and output rates, are used to represent demographic dynamics. But, the main solution for extending a model consists in adding new compartments, or splitting existing ones, for more precise health states, additional species (e.g. in vector-borne diseases), subgroups (e.g. spatial areas or age groups), or accounting for the role of environmental contamination [Marcé *et al.*, 2011]. When many concerns are introduced in addition to health states, e.g. modeling the individual diversity of biological parameters, behavioral features, or prevention and control measures, the number of compartments is bound to grow again and again, which finally resembles very much individual-based models.

Individual and Agent-Based Models

A way to account more straightforwardly and explicitly for *individual states* is using Individual-Based Models (IBM) or

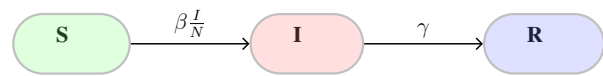


Figure 1: Flow diagram of the classical SIR model. Nodes (compartments) measure the amount of individuals in each health state; flow rates are indicated on the edges.

Agent-Based Models (ABM), which also allow *individual behaviors* and *interactions* [Railsback and Grimm, 2011]. Hypotheses and processes are essentially represented by the choice of individuals, behaviors, and the environment structure where they live [Courcoul *et al.*, 2011; Amouroux *et al.*, 2008; Robins *et al.*, 2015], which makes models potentially indefinitely extensible, but also highly modular. Since model entities are directly mapped to computational ones, the outcomes of simulations also carry more detailed information, providing a deeper understanding of causal mechanisms of epidemiological systems [Marshall and Galea, 2014].

These advantages come with a higher computational cost: not only processing IBM/ABM is slower than CBM, but also, the increased detail level means a larger number of parameters, hence more repetitions in a stochastic context, and a more thorough sensitivity analysis. The possibility of changing hypotheses easily also leads to a multiplication of scenarios aimed at assessing and comparing control measures or public policies.

Metapopulations and multi-scale models

At a regional scale, the ecological concept of *metapopulation* [Grenfell and Harwood, 1997] is usually used to build epidemiological models. A metapopulation is a system of local populations living in isolated patches, each of them endowed with its own epidemic dynamics, and interconnected through a specific contact structure [Keeling, 2005]. Contacts between local populations are indeed caused by several factors: neighbourhood relations, trade movements [Beaunée *et al.*, 2015], 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 modeling of sub-populations dynamics: compared to equivalent IBM simulations, metapopulations models which do not keep individual states may lead to overestimate infections [Ajelli *et al.*, 2010; Keeling *et al.*, 2010]. Conversely, the computational cost of classical IBM is quite high at this scale, all the more when repetitions are required to compare scenarios in a credible way. Unless using a massively parallel platform with simple epidemiological assumptions and a high level of software optimizations [Parker and Epstein, 2011], which is usually out of reach for most epidemiological work, handling millions of agents is an issue.

2.2 The Jungle of Implementations

Independently from the choice of the modeling paradigm, there is no systematic methodology for implementing models, hence a huge diversity of architectures and techniques.

The wide majority of simulation programs are *one-shot tools*, completely *hand-written*, aimed at modeling a *single pathogen in a specific application context* to assess a limited set of control measures. Their reliability highly de-

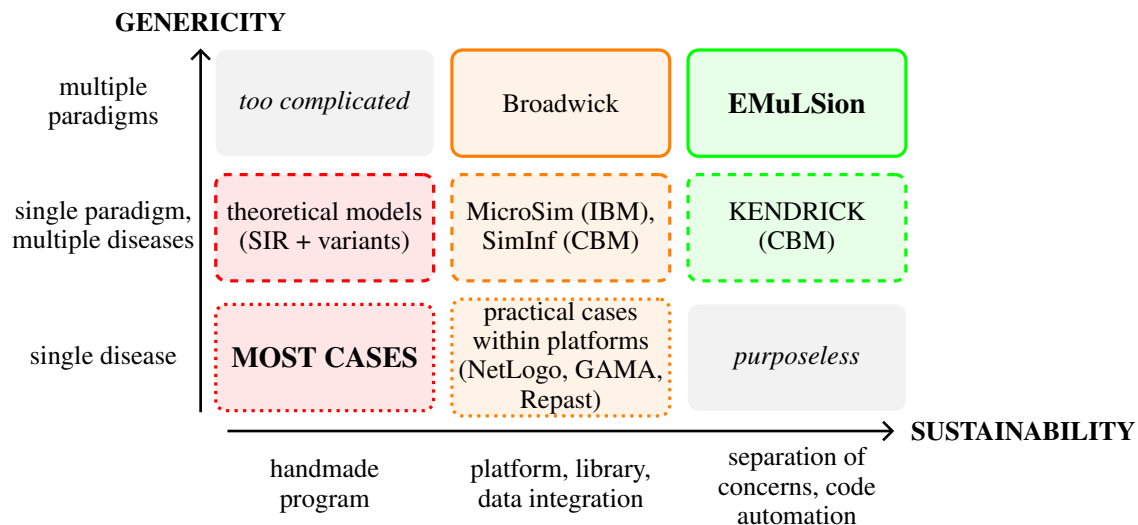


Figure 2: A taxonomy of current approaches in computational epidemiology.

depends on the programming skills of their developers, and their maintainability is necessarily low, especially if low-level languages (e.g. C++) are chosen instead of high-level ones (R, Scilab, Python) for computational efficiency. A few ad-hoc simulators address several diseases, either through theoretical models (SIR or its variants), or sometimes by using abstraction in object-oriented design (e.g. vector-borne disease transmission in [Roche *et al.*, 2008]), but are not capable of handling multiple modeling paradigms.

Instead of writing the entire code from scratch, a wiser attitude to enhance code reuse and sustainability, is to use existing *libraries and platforms*, or write new ones for further work. Several general-purpose platforms have been used to model specific diseases, e.g. NetLogo [Robins *et al.*, 2015], GAMA [Amouroux *et al.*, 2008], or Repast [Collier *et al.*, 2015]. Platforms provide integrated tools for handling core calculations of the model (e.g., in IBM/ABM, reproducible agent scheduling), but also GIS or data integration and easy model outputs. Thus, they allow the developer to spend more time on model design. Their limitation, however, is their genericity as platforms: they are not specifically built for epidemiologic purposes, thus provide no methodological guidelines to guarantee a proper implementation or reusability of models.

Simulation libraries and platforms have also been developed to address multiple diseases within a given modeling paradigm: e.g., SimInf [Widgren *et al.*, 2016], a R library for data-driven stochastic compartment models; MicroSim [Cakici and Boman, 2011], an agent-based platform extended step by step to model several diseases. A small number of works offer several modeling paradigms. The most advanced is Broadwick [O’Hare *et al.*, 2016], which allows the designer to implement either deterministic or stochastic compartmental models, or individuals interacting on networks. However, the development of such tools remains a heavy task (e.g. for MicroSim, “more than 100 person months of implementation work” and “more than 5,000 lines of C++ code”), and still re-

quires a substantial effort for coding disease-specific features.

We believe that the most promising advances will come from a stronger separation of concepts, as proposed in KENDRICK [Bui *et al.*, 2016], an epidemiological-specific language and platform (yet limited to compartmental models). The authors argue for an explicit and separate description of infectious dynamics, spatial distributions and multiple species. The domain expert is able to focus on an accurate description of the characteristics of its model, and pushed to make all assumptions explicit. Then, the tool generates C/C++ code for efficient simulation.

To summarize, existing approaches can be characterized through two dimensions (Fig. 2). First, simulation programs may be more or less generic, depending on their scope, from a single pathogen to several modeling paradigms. Second, they involve more or less computer science skills, from ad-hoc monolithic programs to more and more modular and reusable architectures, and ultimately involving the classical AI separation between procedures and several knowledge domains. It also appears that the next move consists in extending the latter approach towards the integration of multiple modeling paradigms. This is precisely the purpose of the EMuLSion modeling approach, presented below.

3 EMuLSion: Principles and Framework

In this section, we present design principles and a software architecture (called “EMuLSion”¹) to answer above issues. Our proposal relies upon the following assumptions: 1) the model design process should enforce a *strong separation between knowledge domains, and between the model itself and the engine in charge of producing the simulation*, and 2) *epidemiological models have to be designed from the origin as multilevel models* (instead of trying to extend a local model to

¹EMuLSion stands for “Epidemiological Multi-Level Simulation framework”

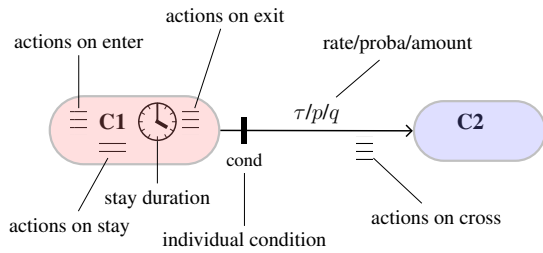


Figure 3: State-machine style flow diagram.

the regional scale or vice-versa), which can be achieved using multilevel agent-based systems, as explained below.

3.1 Separation of Concerns and Knowledge Engineering

Field knowledge (parameters, assumptions, processes, data...) introduced in a model should be split according to the diversity of concerns, which starts with the description of distinct processes involved in the system. To formalize them, modeling habits should be refined with the proper detail level, without being too intrusive. For instance, the ODD protocol proposed in [Grimm *et al.*, 2006] fosters explicit expert knowledge, but remains an ambiguous textual template [Amouroux *et al.*, 2010]. On the contrary, using powerful formalisms originated from physical multi-scale processes [Díaz-Zuccarini and Pichardo-Almarza, 2011], or from molecular biology [Le Novère *et al.*, 2009], would be inappropriate as regards epidemiological concerns. The complexification of flow diagrams proposed by [Perra *et al.*, 2011] is unconvincing and works with compartments only.

To stay as close as possible to existing modeling formalisms, we propose to transform the usual flow diagram into a true finite state machine. We consider that nodes (states) and edges (transitions) can be endowed with additional information (Fig. 3). States are given optional features: 1) a *duration distribution*, which specifies how long an individual is likely to stay in current state, and 2) *actions* performed when entering the state, being in the state, or leaving the state. 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.

When models specified through a classical flow diagrams are implemented, these features are ordinary hard-coded. Making them explicit early in the design process is rather natural, provides a better intelligibility of the whole model, and allows code automation. Moreover, this state-machine diagram can be used in a compartment-based (deterministic or stochastic) or agent-based approach. All information regarding the concerns involved in a model, the corresponding processes and state machines with their states, transitions, conditions, actions, durations and rate, probability or amount values are specified in a YAML configuration file. Parameters can be given either in values, ranges, or expressed as a func-

tion of other parameters. When the YAML file is processed to instantiate the simulation architecture, parameters are parsed using a symbolic mathematics library (SymPy) and the consistency of the model is checked automatically. Besides, this information can also be used for producing technical documentation, figures, comments on sources and assumptions, etc. and thus can be seen as a kind of “literate modeling”.

3.2 Multilevel Modeling with Agents

The architecture we propose to implement the coexistence of concerns at several scales, and encompass multiple modeling paradigms, is a multilevel multi-agent system. This kind of simulation use agents to reify several organization, observation or scale levels at the same time. In epidemiology, the reification of intermediate entities (e.g. age groups, sub-populations...) between the individual and the whole population, as well as the use of associated spatial areas is indeed a major advantage to assess fine-grained control measures.

Among the very few existing generic multilevel meta-models, we embraced the principles defined in [Picault and Mathieu, 2011], which provide many design and implementation features we consider useful for computational epidemiology. Especially, this meta-model enforces a *strong separation between declarative and procedural concerns*, through an independence between agents, behaviors, and a generic simulation engine. 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). Their joint use leads to the *hosting* relation (*a* is host to *b* iff *b* is situated in an environment encapsulated by *a*). Thus, a multilevel multi-agent system is 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, i.e. behaviors of agents). Since agents can represent any kind of entity, they provide a generic and homogeneous, yet polymorphic interface to integrate multiple modeling paradigms.

3.3 Architecture of the Framework

These design principles have been implemented and experimented through a Python framework. In this framework, a model composed of processes (described either as state machines or sequence of actions) occurring at several levels, is translated to multilevel agents endowed with behaviors.

Agents classes are shown on Figure 4. *EmulsionAgents* are situated in at least one environment. They are divided into two categories: *atoms* and *groups*. The *AtomAgent* class represents “individuals” (e.g. persons or animals). Its *EvolvingAtom* subclass represents individuals endowed with state-machine driven behaviors. On the other hand, *Groups* can encapsulate a local environment where other agents can be situated. They are themselves composed of two families: *Compartments* and *Aggregations*. The *Compartment* class intends to aggregate homogeneous individuals into a simple amount, while an *Aggregation* provides a *view* over individuals or other groups, i.e. a representation where agents are gathered

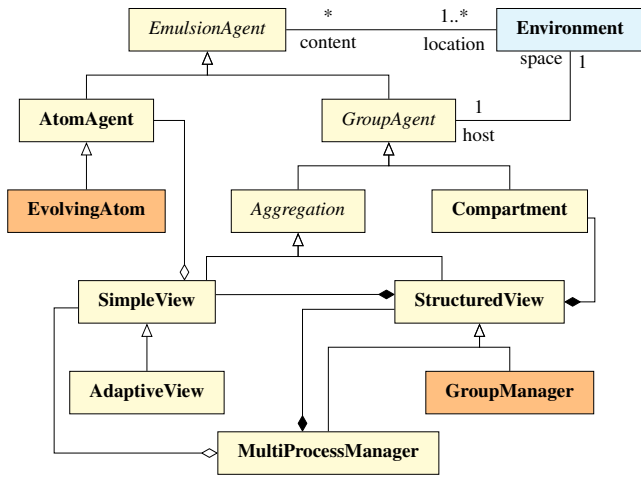


Figure 4: Class diagram of multilevel agents in the EMuLSion framework. Agents with a state-machine driven behavior are represented with orange background.

according to customizable variables, such as health state, age group, etc. More specifically:

- SimpleView agents host individuals (atoms) and schedule their behavior. 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, related to one concern. For instance, in a SIR model, the health_state variable (related to the “infection process” concern) 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 control how individuals (or amounts) change their value for those variables.
- Finally, the MultiProcessManager can handle several concerns in the same simulation. It relies upon at least a SimpleView to control all atoms, and a StructuredView or GroupManager agents to manage the different processes involved in the model.

3.4 Encompassing Existing Paradigms and Beyond

This composition-based architecture allows representing existing epidemiological modeling paradigms. *Compartment-based models* are built using Compartment agents hosted by GroupManager endowed with a state machine for health states and their transitions (Fig. 5a). Compartment agents can adopt a deterministic or a stochastic behavior 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. 5b). Finally, *metapopulations* can be represented using a StructuredView or a MultiProcessManager hosting several agents constructed after one of the latter architectures, with additional processes describing contact structure.

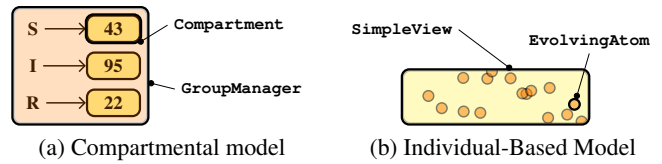


Figure 5: (a) A compartmental model in EMuLSion (with S, I, R states). The GroupManager is endowed with a state machine to determine flows between compartments. (b) An IBM in EMuLSion. EvolvingAtoms are endowed with their own state machines and hosted by a SimpleView agent.

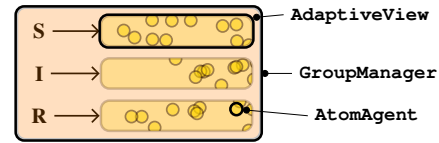


Figure 6: Aggregation of atoms into adaptive views, according to their state. The GroupManager is endowed with the state machine to induce state changes in atoms.

Additional solutions can be built, especially by grouping individuals according to their state for each concern. Using regroupments of individuals provides a straight access to similar individuals, especially for 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. Thus, keeping individual states do not deteriorate performances drastically. This solution is based on a GroupManager agent, endowed with a state machine for one concern, e.g. the infectious process affecting the health_state variable (Fig. 6). 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 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).

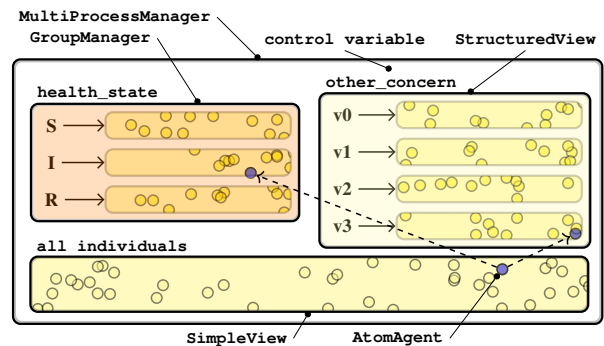


Figure 7: Structure of a multiple-concerns model. Individuals are aggregated according to concern-related variables. Individuals (e.g. the blue one) can be accessed through each concern or through a global list.

A `MultiProcessManager` can handle a combination of such structures. This agent is given the list of processes affecting individuals or groupings, and the variables associated with each concern (“control variables”). It automatically uses a `GroupManager` with the structure described above for handling processes bound to state machines, otherwise it does the same with a simple `StructuredView`. Each individual is thus accessible from a global list (`SimpleView`) and from each of the concern-related groupings (Fig. 7).

4 Experimentation and Validation

This framework has been extensively tested on several variations of the theoretical SIR-like models, to check that all usage patterns above, including those corresponding to classical epidemiological paradigms, produce equivalent results.

To assess the interest of our approach on real cases, we chose two IBM studies, carried out within the same team on the same disease in dairy cattle (Q fever, a zoonosis affecting mainly ruminants) at different scales, thus we can compare our outputs to each model: one for the within-herd spread [Courcoul *et al.*, 2011], and the other for the between-herd spread [Pandit *et al.*, 2016]. The implementation of the within-herd model involves three main concerns: the infectious process (with 6 health states and bacterial shedding in environment), the farming process suited for milk production (with a 4-state life cycle, some states like abortions causing high bacterial shedding), and the culling process (removal of cows from the herd) which depends on animal parity (number of calvings). To benefit from adaptive groupings of individuals, we define a class for individuals (`QfeverCow`, derived from `AtomAgent`), and one for the herd (`QfeverHerd`, subclass of `MultiProcessManager`), built as in Figure 7 but with the three concerns. Thus, animals get automatically grouped by concern-related variables [Picault *et al.*, 2017].

The EMuLSion implementation reproduces the outcomes of the reference model [Courcoul *et al.*, 2011]. In addition, we were able to test in a short time 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 between-herd contamination. This specific model exploration will be published in a separate article. Ongoing work now focuses on the integration of this model to a between-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. Two concerns have thus to be introduced in the intra-herd model: renewal (based on trade data) and bacterial dispersion (based on meteorological data). New procedures are currently elaborated to account for these mechanisms.

5 Conclusion and Perspectives

We have analyzed the difficulties encountered in computational epidemiology regarding sustainability and reproducibility. Our contribution to those issues is twofold. First, we advocate a modeling methodology which enforces a

strong separation between procedural aspects and all knowledge domains, to make all assumptions explicit and easily revisable. Second, we propose an implementation based on a multilevel multi-agent system, to offer a homogeneous way to integrate the many modeling paradigms used in epidemiology. The joint use of both approaches allow to build a comprehensive framework where models are run through a generic, reliable simulation engine.

This work is part of an ongoing collective elaboration of a common modeling frame to foster explicit, comparable and sustainable models in epidemiology, which is a first long-term step to achieve not only reproducibility but also reliability. Besides, the encapsulation of modeling paradigms within agents overcomes their specificities and allows exploring techniques to automatize the substitution of a model by another. The intrinsic multi-scale features of our approach also suggest applications to within-host (immunological) models. From a broader perspective, reducing the duration of design, implementation and testing of models will become a dramatic advantage to answer epidemiological urges raised by environmental changes.

Acknowledgments

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

- [Ajelli *et al.*, 2010] M. Ajelli, B. Gonçalves, D. Balcan, V. Colizza, H. Hu, J.J. Ramasco, S. Merler, and A. Vespignani. Comparing large-scale computational approaches to epidemic modeling: Agent-based versus structured metapopulation models. *BMC Infectious Diseases*, 10(1), 2010.
- [Amouroux *et al.*, 2008] E. Amouroux, S. Desvaux, and A. Drogoul. 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)*, volume 5357 of *LNCIS*, pages 26–33. Springer, 2008.
- [Amouroux *et al.*, 2010] E. Amouroux, B. Gaudou, S. Desvaux, and A. Drogoul. O.D.D.: A promising but incomplete formalism for individual-based model specification. In *RIVF Int. Conf. on Computing and Communication Technologies*. IEEE, 2010.
- [Beaunée *et al.*, 2015] G. Beaunée, E. Vergu, and P. Ezanno. Modelling of paratuberculosis spread between dairy cattle farms at a regional scale. *Veterinary Research*, 46(1), 2015.
- [Bobashev *et al.*, 2007] G.V. Bobashev, D.M. Goedecke, F. Yu, and J.M. Epstein. A hybrid epidemic model: Combining the advantages of agent-based and equation-based approaches. *Winter Simulation Conference*, 2007.

- [Bui *et al.*, 2016] T.M.A. Bui, M. Ziane, S. Stinckwich, T.V. Ho, B. Roche, and N. Papoulias. Separation of concerns in epidemiological modelling. In *15th Int. Conf. on Modularity*, pages 196–200. ACM, 2016.
- [Cakici and Boman, 2011] B. Cakici and M. Boman. A workflow for software development within computational epidemiology. *J. Comput. Sc.*, 2(3):216–222, 2011.
- [Collier *et al.*, 2015] N. Collier, J. Ozik, and C.M. Macal. Large-scale agent-based modeling with repast HPC: A case study in parallelizing an agent-based model. In *Parallel Processing Workshops (Euro-Par)*, volume 9523 of *LNCS*, pages 454–465. Springer Nature, 2015.
- [Courcoul *et al.*, 2011] A. Courcoul, H. Monod, M. Nielen, D. Klinkenberg, L. Hogerwerf, F. Beaudeau, and E. Vergu. 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.
- [Díaz-Zuccarini and Pichardo-Almarza, 2011] V. Díaz-Zuccarini and C. Pichardo-Almarza. On the formalization of multi-scale and multi-science processes for integrative biology. *Interface Focus*, 1(3):426–437, 2011.
- [Grenfell and Harwood, 1997] B. Grenfell and J. Harwood. (meta)population dynamics of infectious diseases. *Trends in Ecology & Evolution*, 12(10):395–399, 1997.
- [Grimm *et al.*, 2006] V. Grimm *et al.* A standard protocol for describing individual-based and agent-based models. *Ecological Modelling*, 198(1–2):115–126, 2006.
- [Keeling *et al.*, 2010] M.J. Keeling, L. Danon, M.C. Vernon, and T.A. House. Individual identity and movement networks for disease metapopulations. *PNAS*, 107(19):8866–8870, 2010.
- [Keeling, 2005] M. Keeling. The implications of network structure for epidemic dynamics. *Theor. Pop. Biol.*, 67(1):1–8, 2005.
- [Kermack and McKendrick, 1927] W.O. Kermack and A.G. McKendrick. A contribution to the mathematical theory of epidemics. *Proc. R. Soc.*, (A115):700–721, 1927.
- [Le Novere *et al.*, 2009] N. Le Novere *et al.* The systems biology graphical notation. *Nat. Biotech.*, 27(8):735–741, 2009.
- [Leek and Peng, 2015] J.T. Leek and R.D. Peng. Opinion: Reproducible research can still be wrong: Adopting a prevention approach. *PNAS*, 112(6):1645–1646, 2015.
- [Marcé *et al.*, 2011] C. Marcé, P. Ezanno, H. Seegers, D. Pfeiffer, and C. Fourichon. 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.
- [Marshall and Galea, 2014] B.D.L. Marshall and S. Galea. Formalizing the role of agent-based modeling in causal inference and epidemiology. *Am. J. Epidemiology*, 181(2):92–99, 2014.
- [Mathieu *et al.*, 2015] P. Mathieu, S. Picault, and Y. Secq. Design patterns for environments in multi-agent simulations. In *18th Conf. Principles and Practice of Multi-Agent Systems (PRIMA)*, volume 9387 of *LNCS*, pages 678–686. Springer, 2015.
- [Merali, 2010] Z. Merali. Computational science: ...error. *Nature*, 467(7317):775–777, 2010.
- [O’Hare *et al.*, 2016] A. O’Hare, S.J. Lycett, T. Doherty, L.C.M. Salvador, and R.R. Kao. Broadwick: a framework for computational epidemiology. *BMC Bioinformatics*, 17(1), 2016.
- [Pandit *et al.*, 2016] P. Pandit, T. Hoch, P. Ezanno, F. Beaudeau, and E. Vergu. Spread of *Coxiella burnetii* between dairy cattle herds in an enzootic region: modelling contributions of airborne transmission and trade. *Veterinary Research*, 47(1), 2016.
- [Parker and Epstein, 2011] J. Parker and J.M. Epstein. A distributed platform for global-scale agent-based models of disease transmission. *ACM Trans. Model. Comput. Simul.*, pages 2:1–2:25, 2011.
- [Peng, 2006] R.D. Peng. Reproducible epidemiologic research. *Am. J. Epidemiology*, 163(9):783–789, 2006.
- [Perra *et al.*, 2011] N. Perra, D. Balcan, B. Gonçalves, and A. Vespignani. Towards a characterization of behavior-disease models. *PLoS ONE*, 6(8):e23084, 2011.
- [Picault and Mathieu, 2011] S. Picault and P. Mathieu. An interaction-oriented model for multi-scale simulation. In *22nd Int. Joint Conf. on Artificial Intelligence (IJCAI)*, pages 332–337, 2011.
- [Picault *et al.*, 2017] S. Picault, Y.-L. Huang, V. Sicard, F. Beaudeau, and P. Ezanno. A multi-level multi-agent simulation framework in animal epidemiology. In *15th Int. Conf. on Practical Applications of Agents and Multi-Agent Systems (PAAMS)*, volume 10349 of *LNCS*, pages 1–13. Springer, 2017.
- [Railsback and Grimm, 2011] S.F. Railsback and V. Grimm. *Agent-Based and Individual-Based Modelling: A Practical Introduction*. Princeton University Press, 2011.
- [Robins *et al.*, 2015] J. Robins, S. Bogen, A. Francis, A. Westhoek, A. Kanarek, S. Lenhart, and S. Eda. Agent-based model for John’s disease dynamics in a dairy herd. *Veterinary Research*, 46(1), 2015.
- [Roche *et al.*, 2008] B. Roche, J.-F. Guégan, and F. Bousquet. Multi-agent systems in epidemiology: a first step for computational biology in the study of vector-borne disease transmission. *BMC Bioinformatics*, 9, 2008.
- [Sandve *et al.*, 2013] G.K. Sandve, A. Nekrutenko, J. Taylor, and E. Hovig. Ten simple rules for reproducible computational research. *PLoS Computational Biology*, 9(10):e1003285, Oct 2013.
- [Widgren *et al.*, 2016] S. Widgren, P. Bauer, and S. Engblom. SimInf: An R package for data-driven stochastic disease spread simulations. *arXiv preprint arXiv:1605.01421 [q-bio.PE]*, 2016.