KENDRICK: a Domain Specific Language and Platform for Mathematical Epidemiological Modelling

Thi-Mai-Anh Bui, Serge Stinckwich, Mikal Ziane, Benjamin Roche, Tuong Vinh Ho

KENDRICK: a Domain Specific Language and Platform for Mathematical Epidemiological Modelling

Bui Thi Mai Anh*†, Serge Stinckwich†§, Mikal Ziane‡, Benjamin Roche† and Ho Tuong Vinh*†
*Institut de la Francophonie pour l'Informatique, Vietnam National University, Hanoi, Vietnam
†UMMISCO 209 UMI, IRD; UPMC, Paris, France
‡Sorbonne Paris Cité, Université Paris Descartes, UMR 7606, LIP6, Paris, France
§Université de Caen Basse-Normandie, Caen, France
Email: thi-mai-anh.bui@etu.upmc.fr, ho.tuong.vinh@vnu.edu.vn, serge.stinckwich@ird.fr
mikal.ziane@lip6.fr, benjamin.roche@ird.fr

***Abstract*—Mathematical modelling of infectious diseases often uses simulation models in order to explore transmission mechanisms and to plan potential management strategies to control the epidemics. However, bridging the gap between the conceptual model of epidemiology and its simulation on computer can lead to some issues related to the lack of expressiveness of implemented models and their reusability and adaptability to new circumstances due to detail computer instructions in General-purpose Programming Languages (GPLs). In this paper, we propose to develop a Domain Specific Language (DSL) for expressively specifying mathematical models of epidemiology and to construct a flexible simulation environment for investigating them. We then use our proposed approach to experiment the measles model in different epidemiological aspects. Our platform is also validated through statistical comparisons between time series generated and theoretical expectations, consequently keeping the link with literature on mathematical epidemiology.**

Keywords*—*epidemiological modelling, compartmental model, domain-specific language

I. INTRODUCTION

Mathematical modelling is a powerful method for studying complex systems that is commonly used in many scientific disciplines. It is widely used to carry out researches on modelling infectious diseases in order to study the mechanisms of transmission [1], explore characteristics of epidemics [2], predict the future course of an outbreak and evaluate strategies to find a best control-program [3]. The first mathematical model of epidemiology was proposed by Daniel Bernoulli in 1766 to defend the practice of inoculation against smallpox [4]. The major contribution to modern mathematical epidemiology was carried out by Kermack and McKendrick who had formulated a compartmental model based on relatively simple assumptions on the rates of flow between different classes categorised by epidemiological status [5]. Epidemiological models largely rely on this so-called SIR framework that basically considers individuals who are *Susceptible* to pathogen (status S), then can be infected, assumed *Infectious* (state I), that can spread infection and *Recovery* (status R) who are immunised and cannot become infected again. From this initial configuration, an infinity of other status could be added in order to represent different transmission cycles to make more complicated compartmental structures such as SEIR or SEIS models [6], noting that the E (*Exposed*) class indicates the individuals in the latent period who are infected but not yet infectious. The transition of status between compartments is represented mathematically as derivatives of compartment size with respect to time [5]. Thus, SIR models are formulated as differential equations. In this paper, we consider mathematical models of epidemiology based on ordinary differential equations (ODEs).

While the mathematical modelling focuses on establishing a conceptual model, the computational modelling helps implement such model as a simulation program. Computational models give out numerical results that are often used to investigate the evolutionary dynamics, estimate parameters and/or compare the effectiveness of different potential control-programs. Firstly, an epidemiological model can be simulated as deterministic in which an ODE solver method such as Runge-Kutta [7] could be used. While the deterministic approach is especially useful to understand the average dynamics without chance, shifting to a stochastic approach, e.g., through Gillespie algorithms [8], is known to be more realistic [1] and can significantly impact the dynamics of infectious diseases, particularly on their seasonality [1]. Finally, an individual-based implementation is sometimes required to reach a more detailed level in which individuals could interact with each other to transmit infection [9].

One of the problems of modelling is bridging the gap between conceptual models and their computer simulation. In fact, going from a conceptual model to a computational one requires modellers to have programming skills. Moreover, the model might be harder to understand and to validate, whithout being overwhelmed by the underlying implementation details. An implemented model is otherwise often tailored to a specific purpose and/or a particular disease and may contain some assumptions concealed within its implementation. Therefore, it is probably difficult to be reused or to adapt to various situations.

Domain Specific Languages (DSLs) address such difficulties by separating two concerns of modelling, specification (conceptual model) and implementation (computational model). As opposed to General-purpose Programming languages (GPLs), DSLs are higher-level languages that provide a more expressive syntax based on abstractions and notations

representing directly the concepts of studied domain [10]. Because of making the implementation details transparent, they increase the readability of the code and facilitate the reuse of domain knowledge and the communication between developers and domain experts [11].

In this paper, we present a DSL that is being developed for modelling mathematical models of epidemiology, together with a simulation platform targeted to study such models. To illustrate the practical usage of our modelling language, we investigate a typical epidemiological model of measles, then, vary this model by changing its specification. Our approach is also carefully validated through statistical comparisons between the simulation results and the theoretical expectations.

II. Design of the Kendrick modelling language

Following the language application development procedures recommended by Parr, T. [12], our modelling language KENDRICK design consists of two parts: the first part provides domain concepts adapted for mathematical modelling of epidemiological models as well as the concrete syntax of the language; the second one aims at establishing a set of semantic operations which either execute a simulation on the specified model or generate its C/C++ version. Figure 1 shows the overview of KENDRICK and the simulation platform around it.

A. Main concepts of KENDRICK

KENDRICK is based on five core concepts: epidemiological model, population, compartment, parameter and equation. Figure 2 represents the abstract syntax of KENDRICK (also called language meta-model) in which the relationship between concepts is described using UML (Unified Modelling Language). This meta-model has been carefully designed to avoid depending on semantic module. It is thus possible to use different semantic modules for different tasks.

1) Epidemiological model: This is the fundamental concept of the language (KEMODEL[1] class) which links to other concepts of domain to complete the representation of a whole mathematical model of epidemiology.

It defines other concepts of the domain as instance variables and provides methods for specifying such concepts as well as semantic operations for translating the model into a running simulation or a C/C++ version that allows more performance.

2) Population: KEPOPULATION class describes the structure of host individuals over which the disease is studied. The individuals of population are categorised into compartments according to their epidemiological status. During the execution of simulation, at each time step, the population records the size of each compartment and the status of all individuals.

3) Compartment: A *KECompartment* is considered as a sub-population in which all individuals have the same epidemiological status. Each compartment is characterised by a label (epidemiological status) and a number of individuals.

4) Equation: KEEQUATION instance represents ordinary differential equation. Equations are textually specified and parsed by a built-in ODE parser.

5) Parameter: KEPARAMETER are used in an epidemiological model and could be specified as an expression[2] of other parameters. Its value is estimated at run-time.

KENDRICK is implemented as an embedded Domain-Specific Language in Smalltalk, a dynamic object-oriented programming language. Its concrete syntax is then a combination of Smalltalk expressions. Table I shows concrete syntax of our modelling language. The concept *Compartment* is specified through *Population* in order to highlight the whole-part relationship between them.

B. Semantic module

The semantic module takes instances of the language meta-model as an input and gives out results [12]. A language, hence, could have many semantic modules, depending on what we want to do with the model. As mentioned above, KENDRICK's semantic modules perform two tasks: simulation in three different modes (deterministic, stochastic and individual-based) and GPL code generation. Figure 3 details the semantic transformations that are done at the moment.

The modellers can switch between the simulation modes by indicating the algorithm used. At the moment, the platform supports the RK4 method [7] for deterministically resolving ODEs. The stochastic simulation converts the ODEs of the model to events and using Gillespie's direct method or Gillespie's explicit tau-leap method [1] to generate stochastic model. The platform also formulates a stochastic individual-based model, reaching event-driven stochasticity at individual level [1], [9]. It is possible to include more simulators in the platform by extending the `KESimulator` class and implementing new algorithm. The code generator transforms KENDRICK model towards a variety of other platform-specific models by using the Visitor Pattern (KEVISITOR class). In the next section of this paper, we will clarify the usage of KENDRICK language and verify the running simulation of an epidemiological model on measles epidemic.

III. Case study and validation

In this section, we choose a well-known childhood infectious disease as an example to figure out the different steps from specifying the conceptual model to running a simulation, using our modelling language KENDRICK. At the end of this section, we will validate the results produced by our platform.

A. Experimenting the measles model using KENDRICK

The most appropriate model representing measles epidemics is the SEIR with demography model [13] in which individuals are categorised in four classes: first, all the newborn individuals are assumed in *Susceptible (S)* class at birth rate μ, then enter in *Exposed (E)* class who are infected but not yet infectious with transmission rate β, become *Infectious (I)* after a latent period given by $1/\sigma$, and finally change to *Recovery (R)* after an infectious period $1/\gamma$. $N = S + E + I + R$. The SEIR system described above can be expressed by the

[1] KENDRICK classes are prefixed with KE

[2] By default, the expression is nil that means no expression

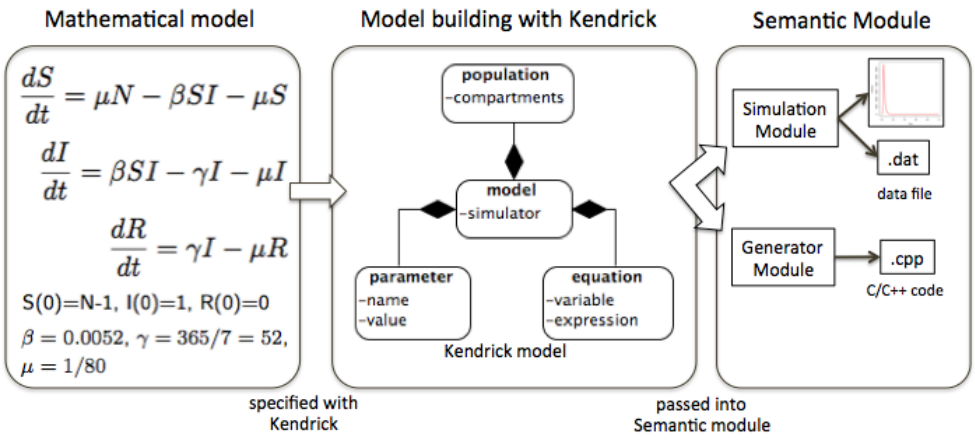

Fig. 1. **Overview of KENDRICK modelling and simulation platform.** Epidemiologists express their conceptual models with KENDRICK language. The KENDRICK model is then passed into the semantic module in order to either perform a simulation or generate a version of model in C/C++.

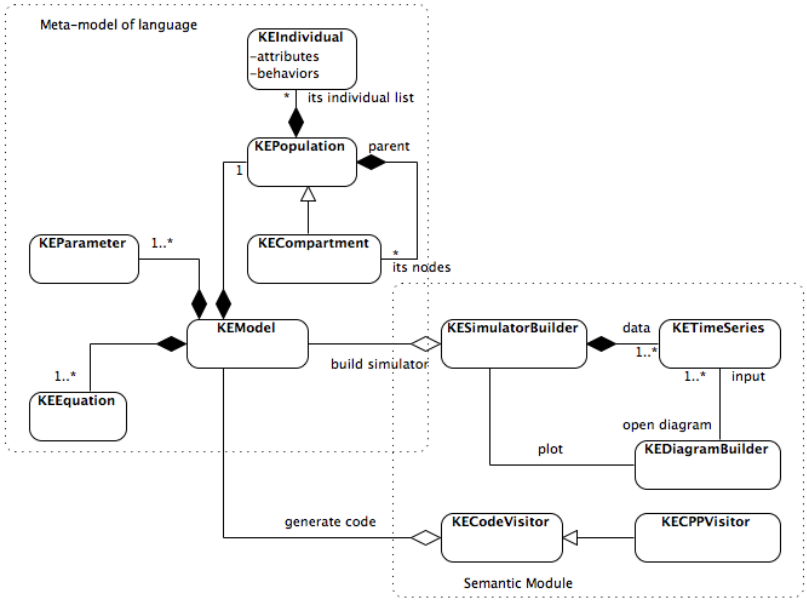

Fig. 2. **The language meta-model with five concepts.** The fundamental concept Epidemiological Model links to others by the UML composition relationship represented by a filled diamond shape pointing to the composing class. The instance of this meta-model expresses the epidemiological model with reference to other domain concepts and could perform semantic operations such as: running simulation, generating GPL code (C+/C++) or visualising the evolutionary dynamics of compartments.

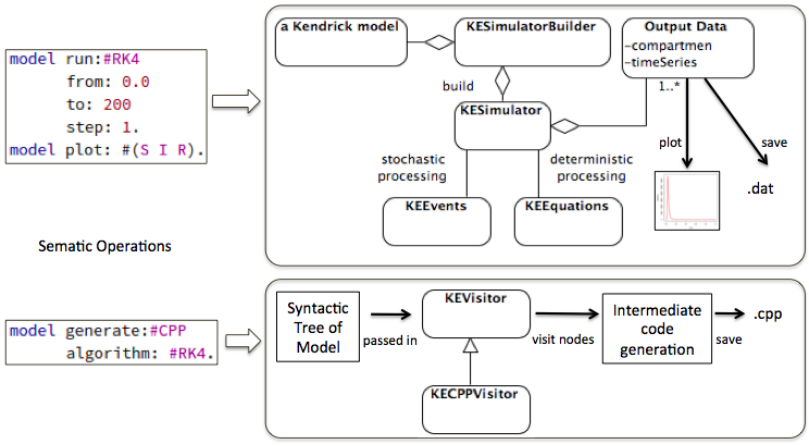

Fig. 3. **Kendrick simulation platform.** A KENDRICK model after being specified is passed into the semantic module to perform the simulation or to analyse the syntactic tree for generating code

TABLE I. CONCRETE SYNTAX FOR SPECIFYING KENDRICK CONCEPTS

Concepts	DSL Syntax

model	`model := KEModel new.`
	`model population: [a population]`
	`model addEquation: [an equation]`
	`model addParameter: [a parameter]`
	`model addParameter: [name] value: [value]`
	`model addParameter: #beta value: 520`
	`model addParameter: [name] expression: [an expression]`

population	`population := KEPopulation new.`
	`population compartments: #(S I R)`
	`population at: [status] put: [number]`
	`population at: #S put: 99999`

equation	`'[variable name] : t = [expression]'parseAsAnEquation` `'S : t = -beta * S * I - gamma * I'parseAsAnEquation`

parameter	`parameter := KEParameter new.`
	`parameter name: [name] value: [value]`
	`parameter name: [name] expression: [an expression]`
	example:
	`anExpression := 'exp(-d * alpha)'parseAsAnExpression`
	`parameter1 name: #alpha value: 0.005`
	`parameter2 name: #gamma expression: anExpression`

following set of ordinary differential equations:

$$\begin{cases} \frac{dS}{dt} &= \mu N - \beta SI - \mu S \\ \frac{dE}{dt} &= \beta SI - \sigma E - \mu E \\ \frac{dI}{dt} &= \sigma E - \gamma I - \mu I \\ \frac{dR}{dt} &= \gamma I - \mu R \end{cases} \quad (1)$$

Using the concrete syntax shown in Table I, we wrote this model with KENDRICK as:

```
|model|
model := KEModel new.
model population compartments: #(S E I R).
model population
      at: #S put: 99999;
      at: #E put: 0;
      at: #I put: 1;
      at: #R put: 0.
model addParameter: #beta value: 0.0000214.
model addParameter: #gamma value: 0.143.
model addParameter: #sigma value: 0.125.
model addParameter: #mu value: 0.0000351.
model addParameter: #N value: 100000.
model addEquation:
      ('S:t=mu*N-beta*S*I-mu*S'
      parseAsAnEquation).
model addEquation:
      ('E:t=beta*S*I-sigma*E-mu*E'
      parseAsAnEquation).
model addEquation:
      ('I:t=sigma*E-gamma*I-mu*I'
      parseAsAnEquation).
model addEquation:
      ('R:t=gamma*I-mu*R'
      parseAsAnEquation).
model run: #RK4 from: 0 to: 100 step: 0.01.
model plot: #(I S E R)
```

The measles epidemics is investigated over a population of 100000 individuals. The model parameters are taken from the literature [1], [13]. Here, we take the basic reproductive $R_0 = 15$ meaning that each infectious individual can infect 15 susceptible on average during the infectious period $1/\gamma = 7$ days. From R_0, the infection rate of model is produced by $\beta = R_0 * \gamma/N$. In Figure 4, we plot the deterministic evolutionary dynamics of the model specified in KENDRICK.

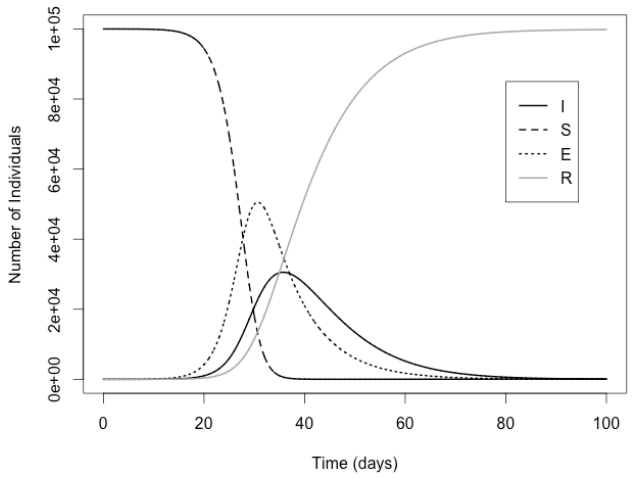

Fig. 4. Modelling results of measles model specified in KENDRICK: $S = 99999$, $E = 0$, $I = 1$, $R = 0$, $\beta = 0.0000214$, $1/\gamma = 7$ days, $1/\sigma = 8$ days, $\mu = 1/(78 * 365)$ in day^{-1}, $N = 100000$. The graph shows the deterministic evolutionary dynamics of the model using KENDRICK language.

Now, we would like to see the impact of vaccination on the measles by vaccinating the population at birth [1]. We have to change the equations of S and R as follows:

```
model addEquation:
      ('S:t=mu*N*(1-p)-beta*S*I-mu*S'
      parseAsAnEquation).
model addEquation:
      ('R:t=mu*N*p+gamma*I-mu*R'
      parseAsAnEquation).
```

Figure 5 shows the epidemic cycles over 100 years in which

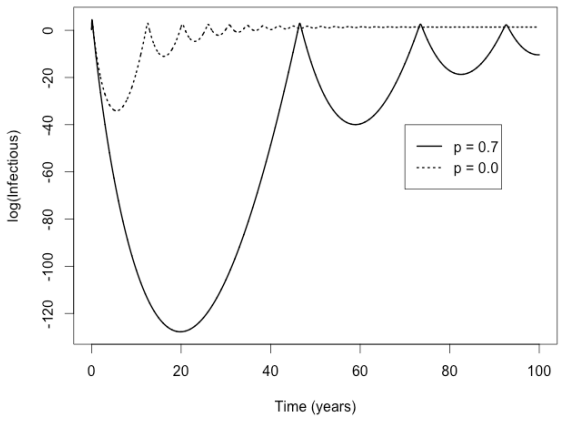

Fig. 5. Infectious dynamics of the measles with vaccination control: $S = 99999$, $E = 0$, $I = 1$, $R = 0$, $\beta = 0.00782$, $1/\gamma = 7/365$, $1/\sigma = 8/365$, $\mu = 1/78$ in $year^{-1}$, $N = 100000$. The black curve shows the deterministic dynamics of infectious in which the population is vaccinated at 70% at birth. The gray curve shows the result with no vaccination. The Y-axis has been log-transformed for readability.

TABLE II. RESULTS OF THE KOLMOGOROV-SMIRNOV TEST FOR THE MEASLES MODEL USING KENDRICK LANGUAGE

Epidemics properties	P values
Peak of epidemics	0.8762
Time at peak of epidemics	0.2471
Epidemics duration	0.5246

the number of epidemics decreases in the case of having vaccination. In short, with this example, we want to demonstrate that with KENDRICK, the modellers could easily vary their model in different ways with a few operations.

B. Validating the modelling language and the platform

We aim here to validate our language in order to suggest that a model executed through KENDRICK language will give identical results to those produced by other well-established platforms used by epidemiologists. We have compared the measles model implemented in KENDRICK and a Matlab program [14] with identical initial values and parameters through the RK4 method. The simulation results (Figure 4) show that KENDRICK produces identical results to those produced by the script written in Matlab.

We then compare the dynamics of the deterministic model and the ones of stochastic model in order to show that these dynamics are similar in two cases (Figure 6), suggesting that our implementation of the stochastic simulation is correct. Finally, in order to validate our implementation of Individual-Based simulation, we compare it with the stochastic simulation outputs in identical configuration. To validate the full process, we apply this validation procedure on the generated code C/C++ of the model. Given that the stochastic and Individual-Based simulation have been executed 150 times, we have applied a Kolmogorov-Smirnov test to explore statistical difference between the distribution generated by the two versions of the model for some global properties of the epidemics: (i) the epidemic peak (in term of infectious population size), (ii) the time at the epidemic peak, (iii) the epidemic duration [9]. Table II shows the test results of the measles model, concluding that these distributions are statistically indistinguishable (all

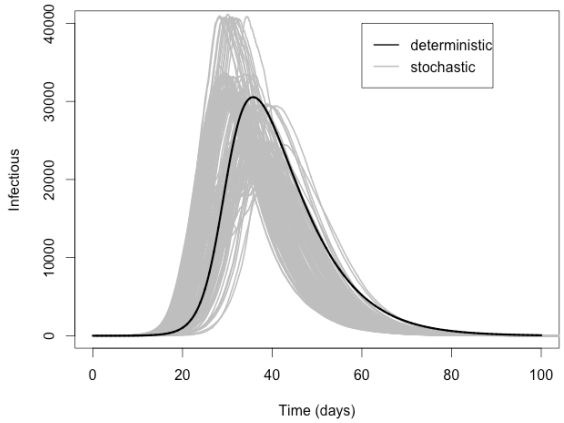

Fig. 6. Comparison between the dynamics of deterministic and stochastic model. The black curve shows the result of deterministic simulation, seeming to be superimposed on the stochastic dynamics (gray curves).

the $P - values > 0.05$). Consequently, we believe that our modelling language and simulation platform give expected results.

IV. RELATED WORK AND DISCUSSIONS

Recently, DSLs were introduced in the context of bioinformatics [15]–[17], but only a few of them focused on epidemiological modelling [18], [19]. *Ronald* [18] is a DSL for studying the interactions between malaria infections and drug treatments but seems not to be developed anymore.

Using a functional reactive programming framework, the work of Schneider and al. [19] is the closest related to our work but their goal is only to specify agent-based epidemiological models while we are targeting multiple representations and simulations modes. From the epidemiological point of view, the agent-based models attract attention of modellers because of the ability to describe a wide range of details.

Our works also support this kind of modelling but is more generic. Instead of specifying all agent states and how agents switch between them (also called *transitions*), we formulate an agent-based model from the mathematical representation meaning that all transitions would be automatically generated, increasing the readability of models and allowing alternatives methods to study them.

It may be argued that Mathematical Modelling Languages (MMLs) such as Matlab [14], Scilab [20] or Modelica [21] are more suitable for implementing models based on ODEs. While such software make deterministic simulation easily doable, stochastic, and in a greater extent individual-based / agent-based simulation requires a significant amount of programming effort. KENDRICK is developed to offer modellers a more accessible way to specify their conceptual models, in the terms of providing a simple and compact syntax and freeing them from arcane programming details. Unlike MMLs, KENDRICK use a model-based engineering approach, that allows the transformation and execution of models in various forms.

An alternative to DSLs are graphical modeling tools like the Spatiotemporal Epidemiological Modeler (STEM) [22] that allows epidemiologists to easily build models but are

usually closed tools. STEM is based on a workflow that is not necessarily the one that is used by epidemiologists. Having a textual language like KENDRICK is also a guaranty for the reproducibility of experiments.

When constructing the KENDRICK modelling language, we aim at taking advantage of the concepts which are closer to the epidemiological domain in order to increase the expressiveness of the language. However, as an embedded language, the modellers may be unfamiliar to the host programming language (Smalltalk) syntax, though, does not spend many times to overcome these difficulties because of the declarative nature of this language.

Finally, together with the extensibility of the language, the ability of modelling more complex situations such as multi-hosts, multi-pathogens, etc. in which a large amount of equations are considered would pose performance challenges to the simulation platform. Part of such problem could be resolved by using code generation. Code generator from DSLs to GPLs could be optimised quite easily and tailored to a variety of semantic operations suitable to the modelling purposes.

V. CONCLUSION

In this paper, we introduce a modelling language and simulation platform to specify and investigate epidemiological models where dynamics are expressed as ODEs. We also highlighted our motivations to develop such language and justify its usage in practice through a typical epidemiological model. In this modelling approach, the mathematical models of epidemiology are specified and varied in a descriptive way that encourages the modellers to be able to focus on their conceptual models. Otherwise, our proposed modelling language, KENDRICK, is easy to use because it lets modellers use ODEs but it is also versatile as it lets them choose among different kinds of simulations: deterministic, stochastic and individual-based.

For the future, we aim to model various concerns of epidemiology: spatial spreading, mobility, risks that deals with heterogeneous populations, evaluation of control strategies, visualisation, etc ... We intend to develop DSLs adapting to different usages on top of the basic infrastructure that we already built. Such a system should allow the composition of the DSLs in an uniform manner, without modifying the core system.

Although more work should be accomplish in order to provide a complete modelling and simulation platform, we believe that our approach is a promising initiative to contribute to the modern modelling of epidemiology.

KENDRICK platform is available as an open source software under the MIT licence: https://github.com/UMMISCO/Kendrick.

ACKNOWLEDGMENT

We gratefully acknowledge the financial support of the European Smalltalk User Group (http://www.esug.org).

REFERENCES

[1] M. J. Keeling and P. Rohani, *Modeling Infectious Diseases*. Princeton: Princeton University Press, 2008.

[2] S. Gandon, M. J. Mackinnon, S. Nee, and a. F. Read, "Imperfect vaccines and the evolution of pathogen virulence," *Nature*, vol. 414(6865), pp. 751–756, 2001.

[3] C. T. Bauch, E. Szusz, and L. P. Garrison, "Scheduling of measles vaccination in low-income countries: Projections of a dynamic model," *Vaccine*, vol. 27(31), pp. 4090–4098, 2009.

[4] K. Dietz and J. Heesterbeek, "Daniel bernoulli's epidemiological model revisited," *Journal Mathematical Biosciences*, vol. 180, pp. 1–21, 2002.

[5] W. O. Kermack and A. G. McKendrick, "Contributions to the mathematical theory of epidemics," *Bulletin of mathematical biology*, vol. 53 (1), pp. 33–55, 1991.

[6] R. M. Anderson and R. M. May, *Infectious diseases of humans: Dynamics and control*. Oxford: Oxford Science Publications, 1991.

[7] D. F. Griffiths and D. J. Higham, *Numerical methods for ordinary differential equations*. Springer Undergraduate Mathematics Series: Springer, 2010.

[8] D. T. Gillespie, "Exact stochastic simulation of coupled chemical reactions," *The Journal of Physical Chemistry*, vol. 81, pp. 2340–2361, 1977.

[9] B. Roche, J. M. Drake, and P. Rohani, "An agent-based model to study the epidemiological and evolutionary dynamics of influenza viruses," *BMC Bioinformatics*, vol. 12, p. 87, 2011.

[10] A. Van Deursen, P. Klint, and J. Viser, "Domain-specific languages: An annotated bibliography," *ACM SIGPLAN Notices*, vol. 35 (6), pp. 26–36, 2000.

[11] M. Mernik, J. Heering, and A. M. Sloane, "When and how to develop domain-specific languages," *ACM Computing Surveys*, vol. 37 (4), pp. 316–344, 2005.

[12] T. Parr, *Language implementation patterns: create your own domain-specific and general programming languages*. Pragmatic Bookshelf, 2009.

[13] R. M. Anderson and R. M. May, *Infectious diseases of humans*. Oxford university press Oxford, 1991, vol. 1.

[14] Matlab, the language of technical computing. [Online]. Available: http://www.mathworks.com/products/matlab/

[15] A. Fall and J. Fall, "A domain-specific language for models of landscape dynamics," *Ecological Modelling*, vol. 141, pp. 1–18, 2001.

[16] P. Degenne, D. Lo Seen, D. Parigot, R. Forax, A. Tran, A. Ait Lahcen, O. Curé, and R. Jeansoulin, "Design of a domain specific language for modelling processes in landscapes," *Ecological Modelling*, vol. 220, pp. 3527–3535, 2009.

[17] R. A. van Engelen, "Atmol: A domain-specific language for atmospheric modelling," *CIT Journal of Computing and Information Technology*, vol. 9, pp. 289–303, 2001, special Issue on Domain-Specific Languages Part I.

[18] T. Antao, I. Hastings, and P. McBurney, "Ronald: A domain-specific language to study the interactions between malaria infections and drug treatments," in *Proceedings of the 2nd ACM SIGHIT International Health Informatics Symposium*, 2012, pp. 785–790.

[19] O. Schneider, C. Dutchyn, and N. Osgood, "Towards frabjous: a two-level system for functional reactive agent-based epidemic simulation," in *Proceedings of the 2008 International Conference on Bioinformatics and Computational Biology, BIOCOMP*, 2008, pp. 747–752.

[20] Scilab, open source software for numerical computation. [Online]. Available: http://www.scilab.org/

[21] Modelica language. [Online]. Available: https://www.modelica.org/

[22] J. K. S.B. Edlund, M.A. Davis, "The spatiotemporal epidemiological modeler," in *Proceedings of the First ACM International Health Informatics Symposium*, 2010, pp. 817–820.