IESIM: SIMULATING AND UNDERSTANDING REAL WORLD EVENTS IN A GAME-LIKE ENVIRONMENT

Joana Simões[a], Chris Skelly[b], João Simões[c]

[a]e-Geo, Lisbon, Portugal; CASA, London UK;
[b]ieSim Org, London, UK;
[c]SIM, FCUL, Lisbon, Portugal; CERN, CH, Switzerland;

[a]j.simoes@ucl.ac.uk, [b]chris.iesim@gmail.com, [c]jalmeida@mail.cern.ch

ABSTRACT
Iesim is a software platform designed to simulate real-world community events in a friendly *game-like* environment. It can be used for evaluating the effectiveness of interventions and predict and understand how events will unfold, and it can also be a valuable tool for training/education.This paper describes an application using IesimLib library and environment, outbreakP2P, which implements an infectious disease model through which the user can simulate an hazard such as Flu or HIV. We will focus on the software mostly as an example of the Iesim technology, but will also discuss its underlying model: an Agent Based Model (ABM); The use of Geographical Information Systems (GIS) concepts and algorithms was fundamental for generating and managing realistic spatial scenarios and it justified the development of an *in-house* GIS library.

Keywords: Simulation Tool, Intervention Strategies, Education, GIS

1. INTRODUCTION

The purpose of this project was to create a modelling environment, that implements models through a set of plugins. For programmers it offers a set of libraries that simplify the task of developing models, and for users it offers a friendly UI that make easier to load data, configure and run the models.

One of the main goals of this project was to "runaway" from the trend on some scientific software of being difficult to use by a non-expert with little knowledge of the project; the models are often difficult to configure, and not rarely that task involves reading extensive manuals. The inspiration for this approach is coming from the computer games, that tend to be very intuitive and rely loosely on reading instructions, since users are typically strongly motivated to "start playing quickly".

Another important concern was to be able to deal with the "gaps" on data, which is a very common situation in many studies. Sometimes there is a dataset, but it does not cover exhaustively all the different variables that need to be considered in the simulation, and therefore it is not used.

In Iesim, there was a strong focus on providing alternative ways of generating data, in an assisted or automatic way, so that the user could follow certain assumptions and generate a dataset. In this sense it is possible to use Iesim to establish realistic scenarios, in the event of good quality detailed data, to "explore" scenarios in the case of having some data and, in the event of no data, it is also possible to produce a completely "fictitious" scenario. Therefore Iesim can be used through a wide range of applications, for instance as a tool for defining policies, to support scientific studies, for education and even as a game.

To demonstrate the potentialities of the Iesim environment, it was developed a package called "outbreakP2P", that we will describe with more detail in the next sections. This is just an example, and the potentialities of Iesim are not in any way, restricted to epidemic modelling. Although it is important to describe the theories behind the model to understand the package, our intention in this paper is mostly to use it as an example of what kind of application can be developed within the logic of Iesim, and not to focus on the model itself. Finally in the last section, we will cover the implementation details describing a bit of the technology.

2. OUTBREAKP2P: A PACKAGE EXAMPLE

Outbreakp2 is a package for simulating the spread of a non-vector infectious disease, that is a disease that spreads directly from one host to another without the intervention of an organism, such as a mosquito, for carrying the disease-causing microorganisms (Simões 2007). The model is split into three sub-models, community, hazard and intervention that are implemented in three plugins: ICom, IHaz, and IInt. On figure 1, we can see the structure of the package that "injects" itself in the UI of the main application (see figure 2).

Each plugin originates an icon in the main window, which gives access to a system of sub-menus that allow the user to parametrize each model, through a friendly set of controls.
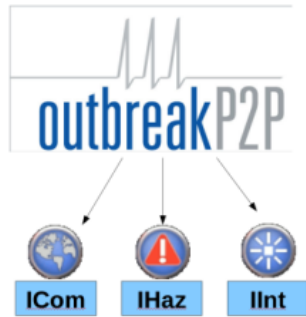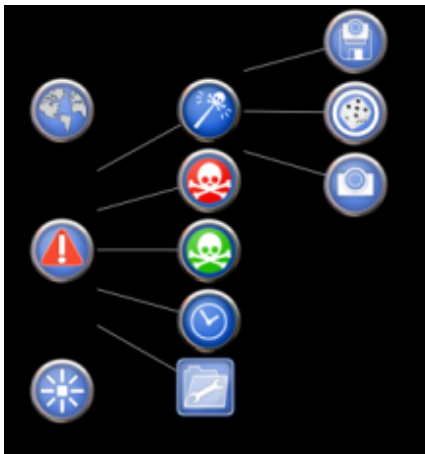
Figure 1: Structure of the package outbreakP2P.

Figure 2: outbreakP2P "injecting" itself into the UI of the application through a set of menus.

In the next sections, we will describe the process of configuring each one of the sub-models and running the simulation, from the perspective of the user; however before that, we will introduce some key concepts to understand the assumptions behind the models.

2.1. Epidemic Simulation and Bottom-up Approach

As in other fields, epidemic studies that incorporate a modelling approach, fall typically into the "Top-down" or "Bottom-up" approach.

Top-down models represent system relationships between aggregate variables in the system as is the case of models based on mean field equations (Lalley 2009), or statistical models based on empirical observations, like the Bayes approach from Chen, White, Caraco, Stratton and Howard (2005). These models are very useful for revealing relationships in the data but unfortunately, they also require a very accurate understanding of the system behavior at a global level; that is not the case when approaching phenomena such as spatial epidemics, which rely on complex social networks, where we have to deal with complicated feedbacks and dependency between variables .

Bottom-up models start from a general understanding of the low-level processes and their elements, and generate aggregate system behavior by simulating the individual entities in the system (Simões

2007); that is the case of Cellular Automata models (Liu, Jin, and Liu 2006), Agent Based Models (Yang 2005) and Network models.

Recently, the largest contribution to epidemic studies, due to the number of published papers and outstanding results, has come undoubtedly from network science, and more specifically from complex network studies, also commonly referred as "new science of networks" (Newman, Barabasi, Albert and Watts 2006). By focusing on the topological properties of networks, scientists were able to abandon the common assumptions of homogeneous mixing and perfect diffusion and come up with some models that exhibit properties similar to real world systems and have been exhaustively applied in all kind of sociological studies (for instance: scale free networks and most notably, Small Worlds). Networks enabled us to establish models based on functional relations between individuals, that are not just random or totally regular (Barabasi and Albert 1999). Many studies, have already emphasized the role of the topological structure of the network on the spread of epidemics (Newman, Barabasi, Albert and Watts 2006). However, few models went a bit further in terms of capturing the hierarchical structures that we find in society. It can also be argued that although space is somehow embedded in these functional relations, network models are not inherently spatial as is the case of CA and not all models explicitly address it (Keeling 1999 is one of the few examples). Besides influencing the propagation of the disease, the network structure also influences its robustness and it has been shown that scale free models in particular, are very sensitive to the detachment of nodes, which is something to have in account when planning vaccination strategies (Newman, Barabasi, Albert and Watts 2006).

Finally, it is important to refer that "bottom-up" simulations provide another big advantage upon differential equation models: their results are a distribution of outcomes, rather than a simple trajectory for each parameter set, and this makes them much more valuable in building scenarios to be used by policy makers (no one wants to base a policy on a single result) (Rahmandad and Sterman 2008).

In outbreakP2P we used a mixed Agent-based/ Network approach, that is described in the community model.

2.2. Running outbreakP2P
2.2.1. Community

The community model relies on a structure of agents and "mixing" or "exposure" environments, where these agents mix and potentially establish infectious contacts (see figure 3). The agents represent households and the mixing environments are functional networks that link the households, grouped through a common activity with a spatial discrimination. The mixing environments considered in the community model were workplaces and schools; however, we can

think of many more (for instance hospitals or shopping malls).

Figure 3: Network of agents linked through mixing environments.

In figure 4, we can see the spatial distribution of households , workplaces and schools in one community scenario. During the configuration of the community model, the user loads/generates this information and aggregates the households around the workplaces and schools, establishing the social networks through which the disease can propagate.
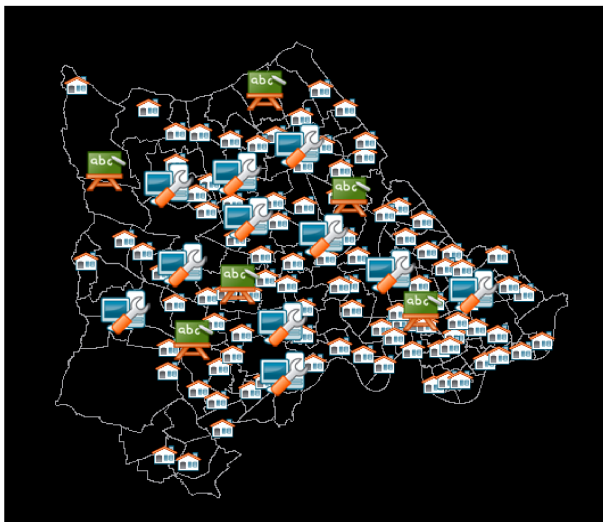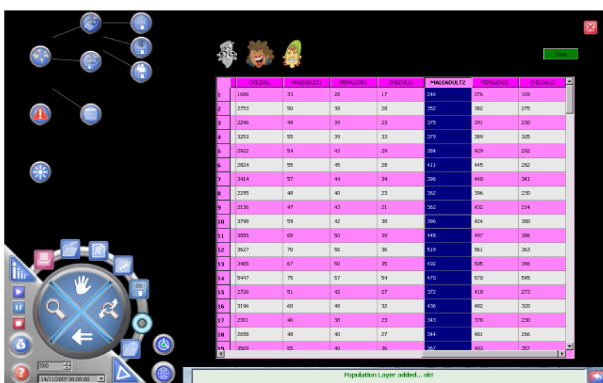

Figure 4: Households and Mixing Environments.

Figure 5: Loading the male, female and child population from the Shapefile.

The process of creating a community, starts with the loading of population data, through a Shapefile (http://www.gdal.org/ogr/drv_shapefile.html), where the attribute table has information on its structure in terms of number of men, women and children (see figure 5). The idea is to use this aggregated data to generate spatially distributed households with a certain structure. For assisting the user in establishing the household size, we use a very valuable tool from risk analysis: the Pert distribution (Vale 2008), that we see in figure 6; this diagram enables us to to pick a minimum, maximum and most likely values, that will be taken in account, when generating the household size; in this way, it is possible to introduce some variability, and at the same time introduce some knowledge of the variable behavior in the system,
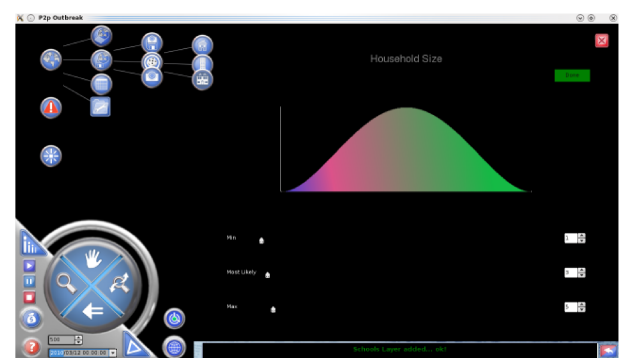

Figure 6: Using the Pert distribution to make "guesses" about the structure of households.

After "generating" the households, the next step is to create the mixing environments, either by loading its location from a file, imputing them manually on the map, or generating them randomly. A spatial buffer with radius defined by the user, is used to aggregate the households "around" the mixing environments (see figure 7); it is also possible to establish a proportion of individuals that are outside this radius and to define its "mixing" degree, using the Pert distribution.
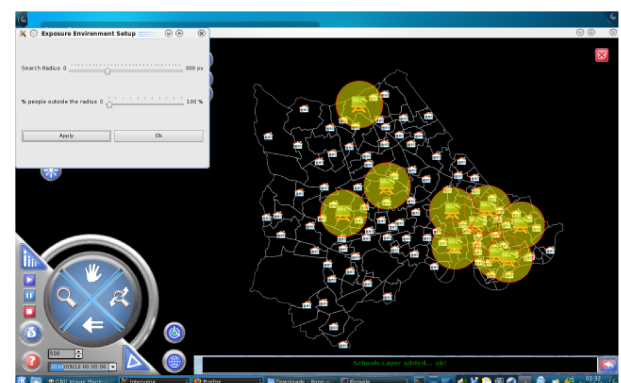

Figure 6: Defining the buffers around schools.

Having defined the spatial structure of the network, it is important to also define its temporal dynamics; this is a somehow complex operation, but it was created a

control for that purpose, which turns it into an easier task. The control is a "calendar", where it is possible to define for each day of the year, which mixing environments will be inactive; for instance it is possible to state that at weekends, both schools and workplaces will be detached from the network (see figure 8). When starting the simulation, it is given an initial date, and the network will follow the established calendar to reconstruct itself at each time step.

Figure 8: Defining the temporal dynamics of the network.

2.2.2. Hazard

In outbreakP2P, the hazard is an infectious disease. The SIR model (Susceptible-Infectious-Removed) formulated by Kermack and McKendrick in 1927 - as cited in Epstein (1997) - expresses the relations between the different population states:

- Susceptibles - individuals capable of acquiring the disease.
- Infected - individuals who can transmit the disease.
- Removed - individuals who are either dead, recovered or become immune from the disease.

There are several variants of this model and in the hazard model, we implemented the SEIR (Susceptible-Exposed-Infectious-Removed), which introduces a fourth state very relevant for certain diseases like HIV or mumps: the Latent or Exposed (see figure 9). In this model, an infected individual might not be yet infectious and therefore it is important to distinguish between these two states.
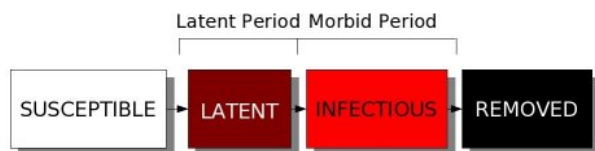

Figure 9: State transitions and definition of the transition periods in the SEIR model.

The hazard model UI uses a diagram similar to the one on figure 9, to assist the user in defining the length of the different periods (figure 10). The "latent period" is designated "incubation period", and it is split into an "asymptomatic" and "symptomatic period", which may be relevant for intervention; the "morbid period" is designated "communicable period", and a natural variability is introduced, again using the Pert Diagram.

The contact parameter β, also known as "infection force", regulates the transition from susceptible to latent and is an essential parameter of the SEIR model. The Pert distribution is also used here, to introduce some variability on the results, based on a "guess".
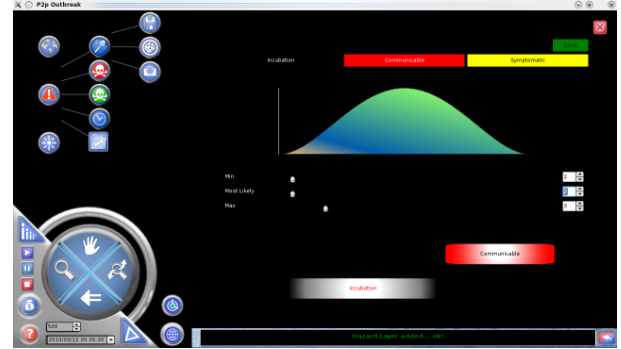

Figure 10: Control to assist the definition of the transition periods in the hazard model.

Finally, it is important to locate the source of hazard, or in this case, the known location of some reported cases of infected individuals. As in other parts of the model, it is possible for the user to either load data, input data manually on the map or generate randomly a certain number of infected cases (see figure 11).
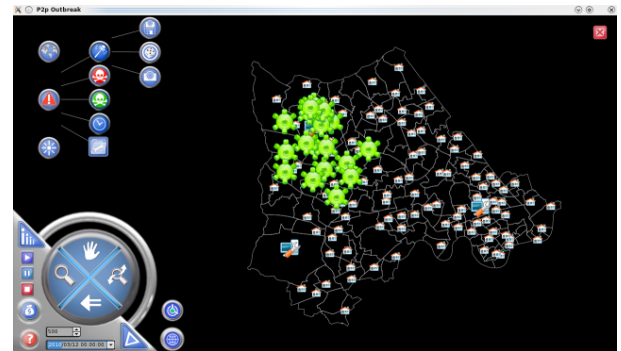

Figure 11: Initial distribution of the infected individuals, which defines the initial spatial extent of the hazard.

2.2.3. Intervention

One of the most useful abilities of this kind of spatial models is to test strategies for the optimal control of an infection in a complex spatial landscape (Simões 2007). In this case, it is relaxed one common assumption in earlier vaccination models: the homogeneity of the population; as most human populations are heterogeneous in space and in their structure the ignorance of this fact can lead to incorrect predictions of the impact of vaccination (Keeling, Woolhouse, May, Davies and Grenfell 2003).

In the intervention model, we planned two vaccination strategies: proactive and reactive, although

only the latter has been implemented. Reactive vaccination assumes the knowledge of infected cases and is a directed effort over the hazard, whereas proactive vaccination is preventive and it starts before the outbreak of the epidemic.

The Pert diagram (see figure 12) assists the user in choosing an immunization degree, in terms of percentage of the population, that will have a random spatial distribution.
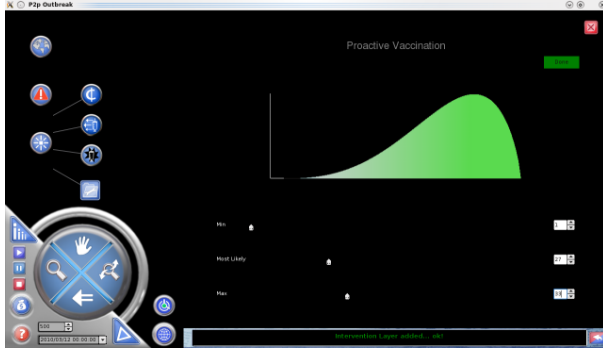

Figure 12: UI for defining the immunization coverage, using the Pert diagram.

Figure 13: UI for defining the costs of the vaccination intervention.
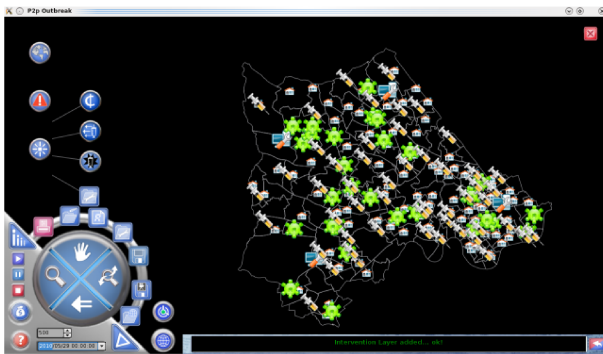

Figure 14: Generated immunized population.

Since costs are an important component in planning intervention strategies, it is also defined what is the cost of each vaccine (in a certain currency), so that we can estimate the total cost of the operation; it is even possible to establish a budget limit, from which the vaccination will cease (see figure 13).

On Figure 14, we can see the spatial distribution of a proactive vaccination intervention.

2.2.4. Simulation: Putting Everything Together

The simulation integrates information from the three models, to create a evolution of the scenario through time. The output of each time step is presented to the user in real time, through a map with spatially distributed pie charts, containing the proportion of each epidemic state (see figure 15); the radius of the pie chart is itself proportional to the population size.
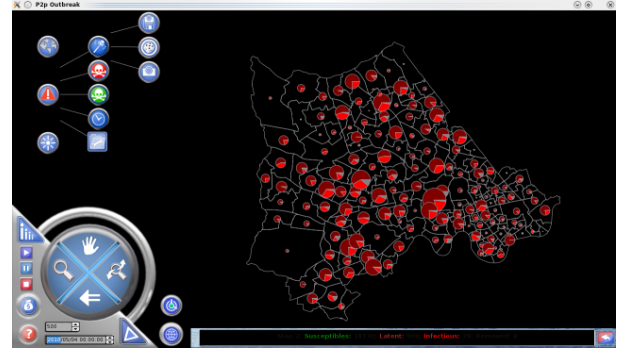

Figure 15: During the simulation, it is presented an animated map, to describe the distribution of the epidemic.

Apart from the map, the user can access other ways of displaying information, such as the log window (alphanumeric) or a line/bar chart (see figure 16). These tools can complement the information in the map and provide a "richer" perception of the scenario in real-time, which would be particularly important if we would want to intervene.
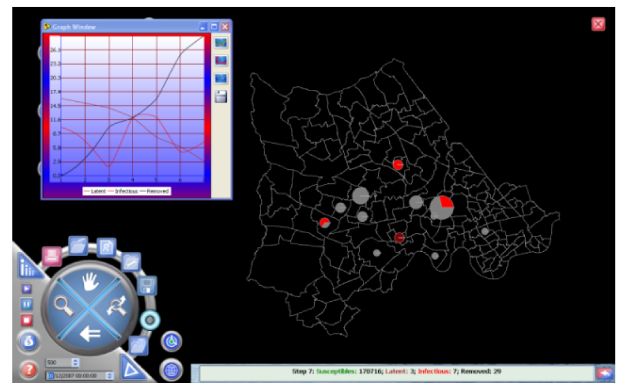

Figure 16: The chart (on the left) and the log window (on the bottom) are complements to the information displayed in the map.

3. THE TECHNOLOGY BEHIND THE MODEL

The choice of language to implement this model was C++, using the Nokia©Qt framework. The motivation for this was to have a native software, that could easily be ported to multiple platforms; in fact, Iesim and

outbreakP2P were released both in Linux and Windows.

The structure of the software is as described on figure 17; at the basis stands the Qt API providing basic mostly (but not only!) UI functionality: this was the only external library used in this project; immediately after, stand the libraries/components that were reused later: the graphic (RTGraph), the calendar (IECalendar) and the GIS: GXLib; GXLib is wrapped inside IesimLib, which is the library that plugin developers have access to, and basically encapsulates all GIS functionalities; finally the main application/ modelling environment is built on top of IesimLib and the other two libraries.
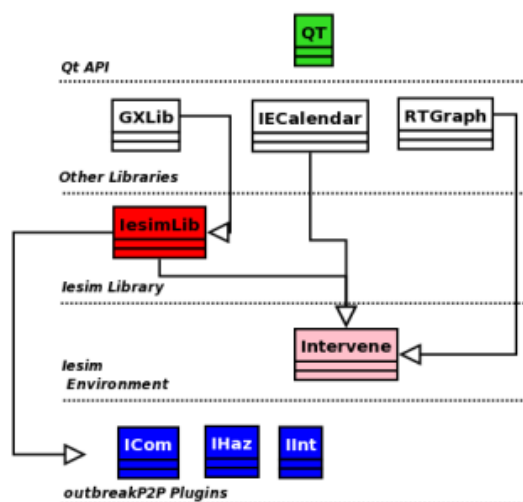

Figure 17: Structure of Iesim software.

Since this modelling environment aims to support the creation of spatial simulation models, the technologies of geographic information were crucial in this matter and they justified the development of a in-house GIS library: GXLib. The ability to load and manipulate geographic data in an efficient way was considered to be essential to the development of this project, and spatial data and algorithms are present in each step of the development of outbreakP2P (see for instance "buffering", on section 2.2.1).

4. SOME FINAL REMARKS

OutbreaP2P is a basic example of what we can do with a "game approach" in scientific modelling; we believe in using simulation models as tools to be used by non-experts, and therefore there was a strong focus in aspects such as UI and portability. We are currently open to collaboration with programmers wanting to develop their own plugins using the Iesim API, and also with end-users interested in solutions in different areas: such as: catastrophe modelling, terrorism, food supply, etc. The product is still under development and far from being complete, but we believe the application to different fields could give us the motivation and knowledge to improve it.

REFERENCES

Barabasi, A. L. and Albert, R., 1999. Emergence of scaling in random networks. *Science*, 286, 509-512.

Chen, H., White, D.J., Caraco, T., Stratton, B. and Howard, H., 2005. Epidemic and spatial dynamics of Lyme disease in New york State, 1990-2000. *Journal of Medical Entomology*, 7, 889–908.

Epstein, J., 1997. *Nonlinear dynamics, mathematical biology and social science*. Reading, MA, USA: Addison-Wesley.

Keeling, M.J., 1999. The effects of local spatial structure on epidemiological invasions. *Proceedings of the Royal Society of London Series B-Biological Sciences*, pp. 859--867. London, UK.

Keeling, M.J., Woolhouse, R. M., May, M. Davies, G. and Grenfell, B. T., 2003. Modelling vaccination strategies against foot-and-mouth disease. *Nature*, 421, 136-142.

Lalley, S.P., 2009. Spatial Epidemics: Critical Behavior in One Dimension. *Probability Theory and Related Fields*, 2, 429–469.

Liu, Q. X., Jin, Z. and Liu, M. X., 2006. Spatial organization and evolution period of the epidemic model using cellular automata. *Phys Rev E Stat Nonlin Soft Matter Phys*, 3.

Newman, M. E. J., Barabasi, Albert L. and Watts D. J., 2006. *The Structure and Dynamics of Networks*. USA: Princeton University Press.

Rahmandad, H. and Sterman, J., 2008. Heterogeneity and Network Structure in the Dynamics of Diffusion: Comparing Agent-Based and Differential Equation Models. *MANAGEMENT SCIENCE*, 54, 998-1014.

Simões, J., 2007. *An agent-based approach to spatial epidemics through GIS*. Thesis (PhD). University College of London.

Yang, Y. and Atkinson, P.M., 2005. An integrated ABM and GIS model of infectious disease transmission. *Proceedings of Computers in Urban Planning and Urban Management – CUPUM'05*. 29 June- 1 July, London, UK.

Vose, D., 2008. *Risk analysis: a quantitative guide*. 2nd ed. England: John Wiley and Sons.

AUTHORS BIOGRAPHY

Joana Simões acquired a PhD in GIS, from the University College of London, in 2007 and after that she moved on to work on Iesim as lead developer. Currently she works as senior software engineer for a GIS company.

Chris Skelly is a professional geographer with a broad science and social science background. He has a considerable IT experience with a specialization in GIS and he was the project manager at Iesim.

João Simões is currently working for CERN and developing his PhD in Physics at the University of Beira Interior, Portugal. He was a developer for Iesim.