Proceedings of 2003 Summer Computer Simulation Conference (SCSC'03) Montréal, Canada, July 20-24, 2003.

MDA Compliant Design of SimExplorer A Software Tool to Handle Simulation Experimental Frameworks

Frédéric AMBLARD*✉, David R.C. HILL, Stéphan BERNARD*, Jérome TRUFFOT*, Guillaume DEFFUANT*,**
***Laboratory of Engineering for Complex Systems, Cemagref**
24, avenue des Landais
F-63172 Aubière Cedex, France
****LIMOS – UMR CNRS 6158**
Campus des Cézeaux, B.P. 125,
F-63172 Aubière Cedex, France
Email: frederic.amblard@cemagref.fr

Keywords: Experimental design, Simulation, XMI, MDA, PIM.

Abstract
SimExplorer can manage the exploration and the execution of simulation experiments by means of experimental designs on several simulation models. This paper first discuss the need to handle experimental designs in a generic way. This leads to the proposition of a PIM for the management of simulation experiments following the MDA. And to illustrate the feasibility of our approach, we present design solutions to develop an efficient tool using XMI protocols between software components. Finally the implementation of a prototype is exposed.

1 INTRODUCTION

Since the beginning of computer simulation, experimental design has been an active research field for efficiently testing and analyzing the behavior of computer models, and it still remains a valuable technology in the decision-making context [Zeigler 1976] [Balci and Sargent 1981] [Kleijnen 1987] [Hill 1996] [Sarjoughian and Cellier, 2001].

Following Balci's arguments [Balci et al., 2000], the accreditation of simulation models passes through rigorous standard verification and validation processes of the model. During these processes, one needs to conduce a considerable number of simulation experiments aiming at exploring the behaviors of the model. Moreover, in some cases, conducing simulation experiments is the only way to analyze and understand the behavior of a complex model. This is the case for instance of individual based models, representing social or ecological dynamics, and for which the cumulative effect of interactions can lead to unexpected effects, generally impossible to formalize analytically [Deffuant et al., 2002]. Thereafter, there is actually a need to automate both the run of a huge number of simulation experiments and to manage the corresponding experimental designs, i.e. the organization of the simulation experiments, in a more evolved and valuable fashion than a simple batch process implementation. The previous concern includes both an ergonomic interface for the specification of the experimental designs and the distribution of computations to launch the corresponding simulation experiments on parallel machines when possible. In addition, a true independency from targets simulation programs is almost impossible.

In order to conduce efficiently and rigorously simulation experiments, we propose a Platform Independent Model (PIM) and its implementation prototype, named SimExplorer. This software can be applied to handle both deterministic and random simulation. The case of stochastic distributed simulation is particularly targeted, which leads to discuss recent research results concerning the parallelization of pseudo-random number generators for stochastic simulation [Fujimoto 1990] [Traore and Hill 2003]. The PIM terminology comes from the Object Management Group (OMG: *www.omg.org*), and is part of the Model Driven Architecture (MDA) which proposes a new approach for developing applications and writing specifications. This paper will present an MDA specification of Simexplorer consisting of a platform-independent base model in UML™ (Unified Modelling Language), plus a platform-specific model (PSM) showing how the base model is implemented. The MDA approach for developing Simexplorer helps to focus first on the functionality and behavior of our distributed application which will handle experimental frameworks for sequential or parallel simulators, without the common distortions that are often implied by the technologies in which it will be implemented. With this separation between implementation details and the essential functions of Simexplorer, we expect the system to evolve (with the progressive inclusion of newly available technologies) without having to repeat the process of modeling the application or system's functionality and behavior. Architectures proposed in the last decade, including the Object Management Architecture (OMA, also from the OMG) were generally tied to a particular technology. For instance, our design solution, exposed in a PSM, is to use XMI (e**X**tended markup language **M**etadata **I**nterchange) for the communication between Java software components. However with the choice of MDA, the functionality and behavior of Simexplorer is modeled once and for all.

2 REQUIREMENT MODEL

The automation of model exploration and testing as exposed previously implies imposing a minimal set of non rigid guidelines, that a simulation software have to follow in order to be able to adapt the same procedures to each concrete experimentation with any implemented model. Basically, with our approach the simulation executable model is considered as a simple program with a parametric structure containing a set of input and output variables. Thus the minimum requirements are given in figure 1:

- The initialization of a single file containing the simulation input parameters (this file could then hierarchically refer to other data files that can be handled by the simulation program).
- The ability to launch the execution of the corresponding simulation program as another process.
- The description of an exploration zone. Instead of giving a single value to each parameter, the user specifies a set of values.
- The definition of an exploration method on an exploration zone by the choice of an existing experimental design or the building of a new one.
- The definition of the outputs of the simulations: the user can select particular variables or combinations of variables, at particular time steps or at particular system states (convergence for instance), to be analyzed
- The execution of the simulations corresponding to a set of parameters and a chosen experimental design [Kleijnen et Groenendaal 1992].
- The visualization of the simulation results in a post-processed and synthetic way to interpret them. It could correspond to the visualization of the simulation trace for a single simulation or to more evolved visualization to synthesize an exploration over a large number of simulation experiments.

From this set of minimum requirements, it can be observed that Visual Interactive Simulators are not directly manageable. Indeed, explorations done with interactive environments are obviously not feasible in such environments. However, scientific simulation tools with interactive visualization outputs, often propose a batch version with synthetic trace files to run many experiments, particularly when they deal with stochastic models.

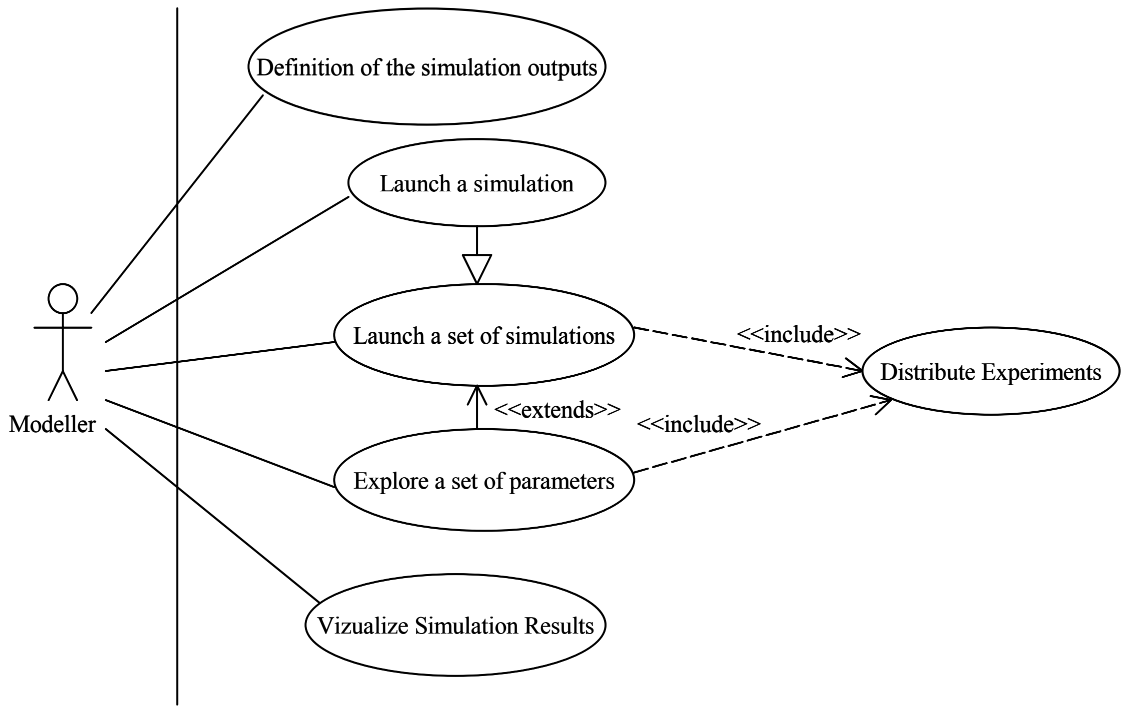

Figure 1. UML use cases for SimExplorer

3 DESIGN MODEL

3.1 Organization of the components

In order to provide the independence of the different software components, especially between the simulation model and the exploration interface, we decided to use XML (extended Markup Language) files corresponding to XMI standard to communicate between components. Similar work is proposed in [Syrjakow et al., 2002] (see fig. 2). Thereafter the exploration interface reads the model inputs and outputs from an XML description file written by the modeler to describe his model and generates for each simulation experiment an XML file containing the values for the different simulation parameters. Moreover the XML file describing the simulation experiment contains also XML description of indicators, elaborated by combining the simulation outputs, to observe.

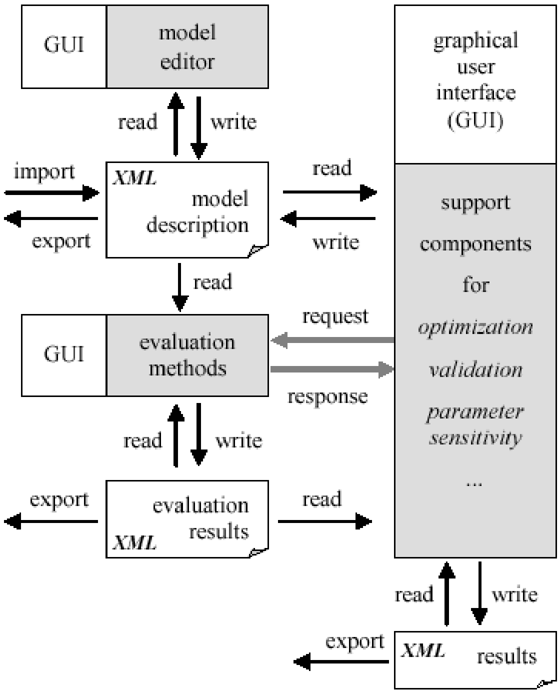

Figure 2. M&S tools and their interdependencies [Syrjakow et al., 2002]

The organization of the exploration interface follows a cut-down into packages that is quite natural. There are then three packages: the "Exploration" package that is in charge of what concerns the exploration specification, which will be detailed hereafter. The "Distribution" package takes in charge the distribution and the execution of the simulation. At last the "Result" package manages the recuperation and the treatment of the simulation results, moreover it manages also the visualization of these results. In the same way, the objectives of the exploration package impose a cut-down into sub-packages (see fig. 3). The first one, the constraints package manages all the constraints on the input and output of the model. The parameter package manages the parameters key part that is the recuperation of the model variables from the XML description file, the possibility to define an exploration zone and some scenarios on it. Finally, the experimental design package manages the choice and the execution of a particular experimental design.

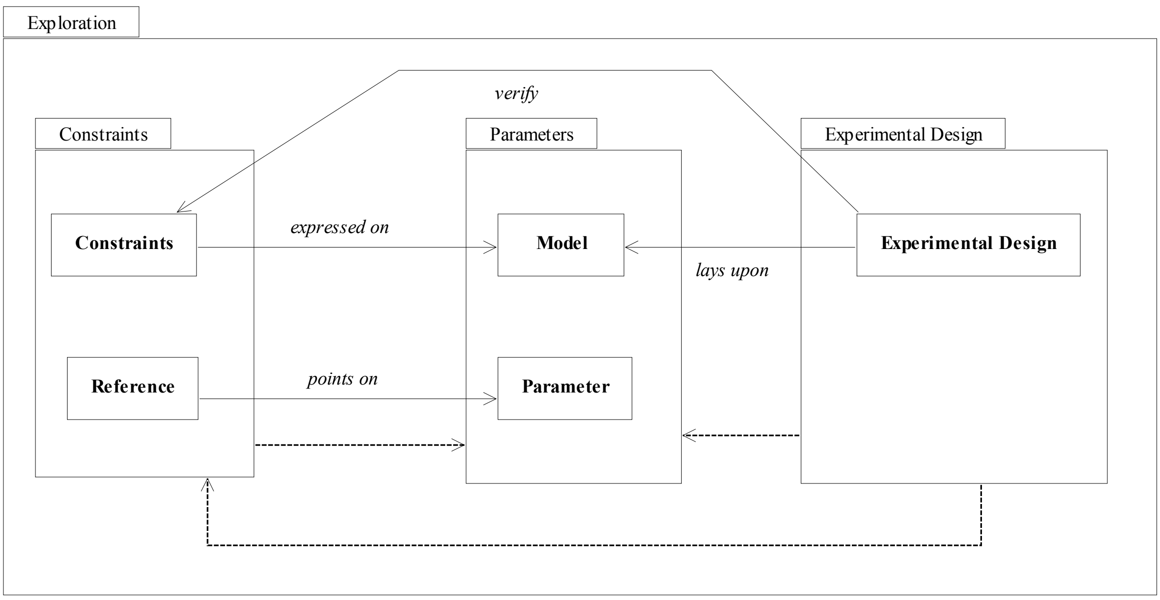

Figure 3. UML class diagram to illustrate the interactions between the sub-packages of the exploration package

3.2 Management of the parameters

Included within the definition of the exploration zone, we have to manage the parameters and the way they are organized. This procedure is independent from the classical classification of simulation parameters into state variables, dynamics parameters and scenarios, as the present classification aims at taking into account as generally as possible the type of these variables and not their status relatively to the model. The model is then considered, from the interface point a view, as a parametric structure aggregating parameters and subsequently we isolate two kinds of parameters: simple parameters corresponding to classical computational types (integer, float, double, string and so on) and group of parameters that corresponds to an aggregation of parameters. We then obtain the following class diagram concerning the organization of the parameters (see fig.4) containing the classical composite design pattern [Gamma et al. 1995]. We have to precise that this classification of parameters is then applicable to each one of the variable status described above (state variable, dynamics parameters and scenarios). Then, including different available types and documentation for the different parameters can extend the different objects treated by the exploration interface to describe the model and define its exploration. It includes name of the parameters, remarks and description for an easier manipulation of the interface (see figure 4).

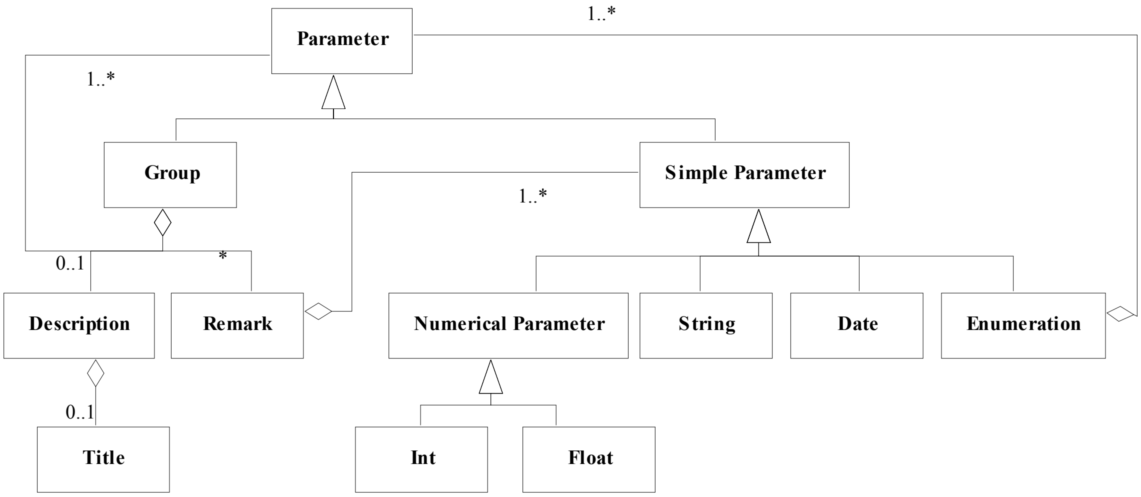

Figure 4. PIM in UML taking into account the different types available and the documentation for each parameter.

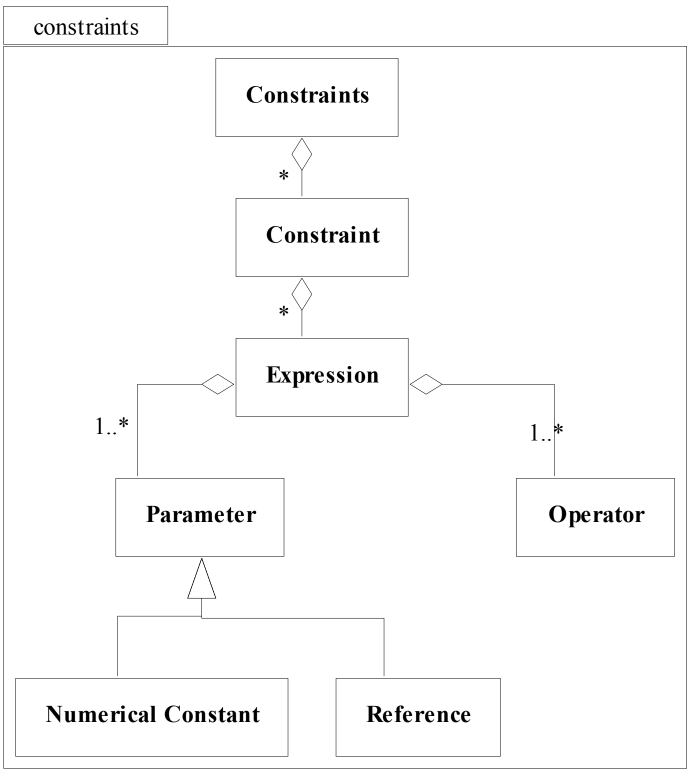

Figure 5. UML class diagram for the constraints package

3.3 Management of constraints

Concerning the constraints applied on the simulation experiment for the manipulation of the model (modeler's constraints, exploration constraints, output constraints), we only have to consider a basic grammar to express all of them. This grammar enables to express formulas linking variables, operators (in particular logical operators) and numerical values. The relations between the objects are expressed in figure 5.

3.4 Management of the experimental design

Once the user expressed the different values related to each one of the parameters of the models he wants to explore, and once the constraints on the parameters or even between the parameters are expressed, he may want to apply this design. The application of experimental designs is not new [Kleijnen 1987] but its generalization to the whole field of modeling and simulation is not yet acquired since an experimental design is often difficult to elaborate. Moreover, the experimental design is very dependent on the kind of models and on the kind of analysis the user want to execute. However, we do not propose a tool that will choose the best experimental design in function of the model or the desired analysis. The intervention and experience of the user is needed here to choose a good experimental design to apply for an exploration. The tool will only provide a library of experimental designs, and allow the user to add new experimental designs not yet included in the library. The different kind of experimental designs we wish to manage are given hereafter. Any experimental design will have to take into account replications of stochastic simulations :

- Complete experimental designs (factorial designs), able to achieve all the simulations characterized by the combination of the specified parameters values and taking into account the specified constraints linking these parameters.
- Incomplete experimental designs, that will run only an organized subset of the experiments proposed by a complete experimental design.
- Random experimental designs, propose a random set of experiments within the parameter space or one of its sub-region.
- Dynamical experimental designs that could be envisaged sometimes as optimization tools. The general principle is to launch a first set of simulation within the defined parameter space (often a regular discretization of this space or a random set within the continuous space), the response of this set is then analyzed and subsequently another set of simulation experiments is determined automatically to optimize an indicator or more generally a variable. It is the case for instance of response surface experimental designs. Dynamical experimental designs are a useful tool in case of refinement of specific zones of the parameter space.

3.5 Design of the functioning

In way to summarize the functioning of the exploration process, we have a parameter package that enables to read a description file of the model to explore written in XML. This package enables to define the parameter space for the exploration by specifying for each parameter of the model, the different values to explore. Then, once defined, the constraints package enables to express constraints on this exploration space by linking parameters and numerical values within formulas. At last, the experimental design package proposes different means to explore the so defined constrained exploration space. Depending on the file taken as input, the interface displays two different forms. Concerning the global structure, they are organized as a tree at the left of the interface. The leaves having the same direct global structure are then grouped on a form at the right of the screen (see figure 6).

CONCLUSION

The design presented within this paper enables to establish the first step towards the implementation of a generic tool. The application of the MDA enables the reuse of analysis and design elements presented in UML, the PIM presents platform independent tools to manipulate simulation models. The use of XMI for the software components to communicate enables also a reuse since XMI is a largely extendable model. The prototype achieved is really independent from the models under study. At last, the distribution of the simulation experiments over a local network or a grid is envisaged to execute the defined experimental design in an efficient way. The perspective of using this tool on the European Data Grid is also under consideration [Hey and Trefethen 2002] since our University is a node of this Grid.

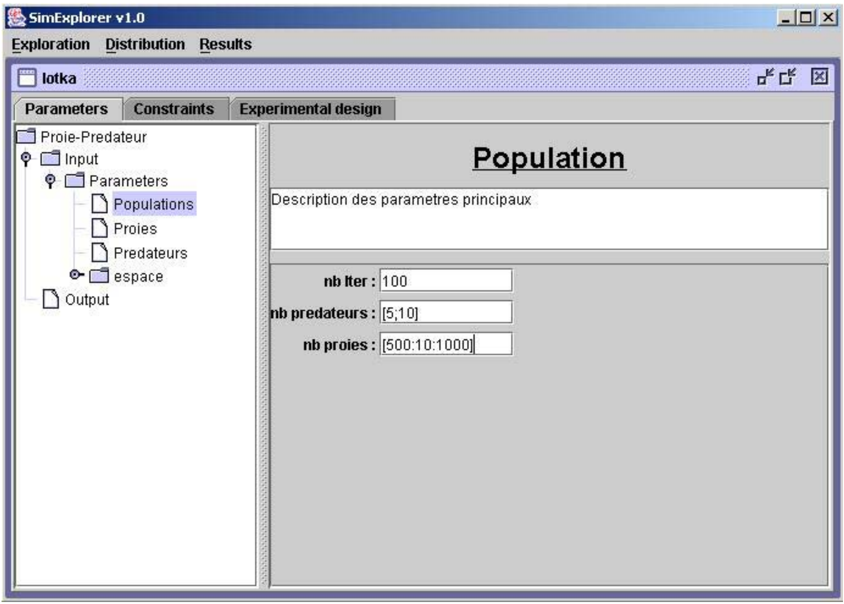

Figure 6. Human Computer interface, a tree organizing groups of parameters and a form on the right

References

[Balci et al. 2000] Balci, O., Ormsby, W.F., Carr, J.T. and Saadi, S.D., 2000. "Planning for verification, validation and accreditation of modeling and simulation applications." In Proceedings of the 2000 Winter Simulation Conference, J.A. Joines, R.R. Barton, K. Kang and P.A. Fishwick (eds), pp. 829-839.

[Balci et Sargent 1981] Balci O, Sargent R., 1981, "A Methodology for Cost-Risk Analysis in the Statistical Validation of Simulation Models ", Communication of the ACM, Vol. 24, N°4, pp. 15-29.

[Deffuant et al. 2002] Deffuant, G., Huet, S., Bousset, J.P., Henriot, J., Amon, G., Weisbuch, G. - 2002. "Agent based simulation of organic farming conversion in Allier département". in Complexity and Ecosystem Management. M.A. Janssen editor. Edward Elgar Publishers, pp 158-189.

[Fujimoto 1990] Fujimoto R., 1990, "Parallel Discrete-Event Simulation", Communication of the ACM, Vol 33, N°10, pp.31-53.

[Gamma et al. 1995] Gamma E., Helm R., Johnson R. and Vlissides J., 1995, "Design Patterns : Elements of Reusable Object-Oriented Software". Addison-Wesley, reading, MA.

[Hey and Trefethen 2002] Hey T., Trefethen A.E., 2002, "The UK e-Science Core Programme and the Grid", Future Computer Generation Systems, Vol 18, pp. 1017-1031.

[Hill 1996] Hill D., 1996, "Object-Oriented Analysis and Simulation", Addison-Wesley.

[Kleijnen 1987] Kleijnen J.P.C., 1987, "Statistical Tools for simulation practitioners", Dekker New-York.

[Kleijnen et Groenendaal 1992] Kleijnen J., Van Groenendaal W., 1992, "Simulation: A Statistical Perspective", John Wiley, Chichester.

[Sarjoughian and Cellier 2001] Sarjoughian, H.S. and Cellier, F.E., 2001. "Towards a Unified Foundation Simulation-Based Acquisition" in Discrete Event Modeling and Simulation Technologies: A Tapestry of Systems and AI-Based Theories and Methodologies, H.S. Sarjoughian and F.E. Cellier (eds.), Springer-Verlag, pp. 1-14.

[Syrjakow et al. 2002] Syrjakow, M., Syrjakow, E. and Szczerbicka, H., 2002. "Towards a Component-Oriented Design of Modeling and Simulation Tools." In Proceedings of 2002 AI, Simulation and Planning in High Autonomy Systems, F.J. Barros, N.Giambasi (eds.), pp. 59-64.

[Traore and Hill 2003] Traore M.K., Hill D.R.C., 2003, "The use of Random Number Generation for stochastic distributed simulation : application to ecological modelling", System Analysis – Modelling and Simulation, to be published.

[Zeigler 1976] Zeigler B.P., 1976, " Theory of Modeling and Simulation", Wiley Interscience, New York.

Frédéric Amblard got his MSc in Computer Science in 1998 from Blaise Pascal University and his French DEA from Montpellier University in 1999. He obtained a fellow to begin his PhD studies in the Laboratory of Engineering for Complex Systems at Cemagref. His research activities deal with the elaboration of multi-agent models for social networks dynamics.

David R.C. Hill received his Ph.D. degree in Object-Oriented Simulation in 1993 from Blaise Pascal University. D. Hill is currently Full Professor and Head of the Soft. Engineering Department at ISIMA Modeling Institute. His current application domain concerns Life Science Simulation. Prof. Hill has authored or co-authored three text books in O-O Simulation and Ecological Modelling.