# Design and stochastic modeling strategies for the simulation of large (spatial) distributed systems

Alexandre Muzy* and David R.C. Hill**

* LISA UMR CNRS , Università di Corsica – Pasquale Paoli, B.P. 52, Campus Mariani, 20250 Corti. France.

**Blaise Pascal University, ISIMA/LIMOS UMR CNRS 6158, BP 10125,
Campus des Cézeaux
63177Aubière Cedex France

Currently, as a result of the increase of computational capabilities, internal structures of models describing large (spatial) distributed systems can be refined. This results in a high number of interacting sub-models requiring exploiting efficiently computational resources. Furthermore, refining the structure of a model requires the modeler to deal with design decisions (modularity, identification of model structures, algorithms, etc). To achieve this goal, specification frameworks (like Discrete-Event System Specification) can be used and developed. The more generic and reusable are the structures provided by these frameworks, the more these structures can be implemented easily by modelers. Nevertheless, the more these structures are generic and reusable, the more they produce simulation overhead. To reduce simulation overhead, efficiency is usually considered as an implementation issue, whereas genericity and reusability are discussed at a more theoretical level, the framework one. However, improving efficiency at the implementation level reversely decreases the reusability advantages provided by frameworks. Integrating and comparing, at a theoretical level, interactions between reusability and efficiency better describes the choices to be achieved by modelers. This is the first objective of this study (reusability being reduced to modularity aspects of the sub-models composing the whole simulation model). Another fundamental aspect for the modeling and simulation of large (spatial) distributed systems is related to the stochastic aspect of information diffusion. During the simulation, causal discrete-events are exchanged randomly between distant and neighboring sub-models. The second objective of this study is to discuss the importance of stochastic aspects in the modeling of large (spatial) distributed systems. This point is illustrated through a fire spread application.

**Keywords:** Stochastic modeling, large (spatial) distributed systems, modularity, efficiency.

## Introduction

Developing computational models of large (spatial) distributed systems is usually achieved to implement and simulate: (i) original mathematical models [obtained by other disciplines (physics, biology, etc).], or (ii) fully computational models. In the first perspective, mappings are defined between the structures of the mathematical models and the computational ones. The modeling capabilities of computers are thus constrained by the modeling assumptions of other disciplines or by the discretization methods used to implement the model. This chapter focuses on the second perspective. Only discrete computational models are used to specify and simulate both structure and behavior of systems.

Both structure and behavior of systems can be modeled deterministically or stochastically. Determinism presupposes actual mechanisms to be modeled in such a way that they do not depend on probability. To one input (corresponding to context and initial conditions) corresponds one output. However, human knowledge about large (spatial) distributed systems is far from complete and actually uncertain (notably because human abilities are limited). Under uncertainty, Monte Carlo techniques can be used to design approximated models (Fishman, 2000). Probabilities and distribution laws assigned to parameter values. Investigation of state spaces through parameter changes can be automatically achieved using experimental plans (Kleijnen, 1997; Amblard et al., 2003).

Another aspect of large (spatial) distributed systems relates to the design of simulation models. A fine-grain design of internal structures necessitates then computational resources to be sufficient to account for many message exchanges, computations and systems. Hence, a trade-off between modularity and performances has to be done.

Considering large (spatial) distributed systems, this study discusses:

1. Design strategies as a trade-off between modularity and performances,
2. Stochastic modeling approaches, based on a pure algorithmic specification, over physics-based ones, in the context of discrete and distant interactions with random occurrences. Fire spreading (including distant interactions through firebrands) is used for illustration.

Both objectives are used to discuss further strategies to be included in Discrete Event System Specification (DEVS) (Zeigler et al., 2000) for modeling and simulating efficiently large (spatial) distributed systems.

The study is organized as follows: in section 2, the material and methods are developed; in section 3, the simulation results, modeling and design assumptions are described; finally, a conclusion and perspectives are drawn in section 4.

## Material and methods

Design and stochastic modeling tools are presented here. Evidences are given that in many domains, stochastic and spatial interactions have to be considered. A comparison between modeling and design techniques is given in the context of fire spreading.

***Importance of stochastic aspects in (spatial) distributed systems***

In many domains, stochastic aspects of spatial spreading have to be considered. At the beginning of the millennium, colleagues, like Lewis, were surprised that despite the recognized importance of stochastic factors, recent models for ecological invasions were almost exclusively formulated using deterministic equations (Lewis, 2000). In forestry, the forest growth can rely for instance on the distant spread of seeds. It has been shown that pioneering trees, whose seeds were propagated by heavy winds, do have an impact on the colonization of landscapes by particular species (Coquillard and Fain, 1995; Coquillard, 1995). In oceanography, the spreading of a tropical alga, *Caulerpa taxifolia*, introduced by mistake in the Mediterranean Sea is based on the spatial spreading of cuttings. This spreading is by essence stochastic, and occurs at a stochastic and discrete distance from an original point of settlement. Only stochastic models under spatial constraints were able to reproduce maps of the spreading of this alga (Hill et al., 1998).

In fundamental domains like physics, a stochastic and spatial modeling allows to understand phenomena in which determinism does not offer the same precision [*e.g.*, the spreading of spiral waves (Jung and Mayer Kress 1995)]. In this study, the effects observed can be understood as a generalization of the concept of stochastic resonance in spatially extended systems. In (Falcke et al., 2000) a study presents that intracellular spreading of calcium-induced calcium release with the stochastic DeYoung-Keizer-model. The system under study presents a state characterized by backfiring. The backfiring occurs because the steadily propagating pulse solution undergoes a global heteroclinic bifurcation. The use of spatial stochastic modeling was also used to argue that quantum-gravitational fluctuations in the space-time background give the vacuum non-trivial optical properties that include diffusion and consequent uncertainties in the arrival times of photons, causing stochastic fluctuations in the velocity of light in vacuo (Ellis et al., 2000). In nuclear medicine, we have shown that the precise detection of small tumors with an error less than *10%* can currently only be achieved by spatial Monte Carlo simulations (ElBitar et al., 2006).

After a discussion dealing with the implementation techniques of discrete simulations, a state of the art of fire spread modeling is provided.

***Modularity and performance balance***

There are four common types of strategies to implement the kernels of discrete event simulations (Balci, 1998). These strategies, also called world views, consist of: event-scheduling, activity-scanning (including the three phases optimization), and process-oriented strategy introduced by the Simula language (Dahl and Nygaard, 1966). A strategy makes certain forms of model description more naturally expressible than others. In all of these world-views, an event corresponds to an instantaneous change in the state of a system at a particular time. Event scheduling models work with pre-scheduling of all events without provision for activating events. In contrast, in the activity scanning approach, events can be conditioned on a contingency test in addition to being scheduled. A model is said to be active when its scheduling time has occurred and when its contingency test is satisfied. An optimization of the activity scanning strategy is named the three phases approach (Pidd, 1992; Coquillard and Hill, 1997). The interaction process world view is a combination of both event-scheduling and activity scanning strategies.

An additional classification consists of considering the two kinds of time management: discrete-time and discrete-event simulations. In discrete-time simulations, a clock is advancing the simulation by a fixed time step. At every time step, states are computed. In discrete-event time based management, events drive the simulation. The simulation time advances from on event time-stamp to another according to the events scheduled.

All these techniques and strategies focus on particular concepts (events, activities, processes) and simulation time managements. On the other hand, source systems are also usually described through a system-based decomposition. According to interactions and autonomy of systems, systems are identified and connected. DEVS is the soundest framework aiming at tackling computational systems, discrete-events and simulation time managements. Through its specification hierarchy, according to the observed behavior of the whole system, the internal structure is progressively described until the definition of interfaces (or ports), in a modular way. At the last specification level, a sub-model is fully modular. At the anterior level, the states of sub-models can influence directly the transitions of other sub-models. The choice of specification level is crucial in terms of reusability and efficiency. Currently, three kinds of approaches discuss both modularity and efficiency of DEVS models:

1. At the simulation level, (Wainer and Giambiasi, 2001) propose to flatten the hierarchy of cellular models. (Hu and Zeigler, 2004) aim at taking advantage of spatially distributed causal events in cellular models. (Muzy and Nutaro, 2005) suggest activity tracking mechanisms only focusing on active components, reducing the data structure overheads resulting from discrete-event managements.
2. At the compilation level, (Lee and Kim, 2003) only account for active event paths. Sub-systems which will not receive events or compute transitions during the simulation are not compiled previously.
3. At the modeling level, wrapping and modularity are compared in terms of reusability and efficiency (Muzy et al., 2003; Shighina, 2006; Sun and Hu, 2008). Modularity reduces performances (because of discrete-event managements) and improves reusability. A weaker modularity (reducing state encapsulation) reduces the reusability of atomic systems, while improving performances.

More recently the activity tracking paradigm has been introduced (Muzy and Zeigler, 2008). This paradigm describes and proposes structures focusing on active sub-systems. Usual time flows and world-views are embedded. The formal description of DEVS models is discussed through activity-based simulation algorithms and new structures accounting for dynamic structures.

Using the activity-tracking paradigm, components can be modeled and simulated in two steps:

1. The propagation activity is tracked. Information exchanged between components is routed and computed. The current set of active component is scanned. Events are routed and output transitions are computed. Final receivers are detected in the hierarchy using a recursive routing function (Muzy and Nutaro, 2005). The active set is then updated including imminent components for external transitions. Order of the active set depends on a tie-breaking function of imminent components (Zeigler et al., 2000).
2. According to current states and to new inputs, new states are computed. External and internal transitions (due to external and internal events) of active components are computed. Components changing state significantly are marked to be added to the new

(ordered) active set. In a discrete-event driven simulation, the new active set corresponds to a scheduler and active components are marked to execute further their internal transition function (corresponding to an internal event occurrence). In this case, the current active (ordered) set is a sub-set of the scheduler, which corresponds to components active at the current simulation time.

DEVS is a powerful framework. However, its structures have to be exploited to map activity of systems. Sub-systems are modeled as active or inactive, the whole simulation directly tracking and computing activity, whereas ignoring inactivity. This approach seems obvious; however it allows merging fine-grain usual world views with larger grain system-based approaches, explicitly accounting for reusability and efficiency. Moreover, simulation resources can be characterized through activity. Observing activity allows choosing the level of modularity according to the overhead induced.

### *Fire spread modeling*

Fire spread models are usually categorized into forest and urban fire spread models. This study focuses on forest fire spread models. Forest fire models can be separated into stochastic and deterministic models. Stochastic models aim at predicting the most probable fire behavior in average conditions. On the contrary, in analytical models, the fire behavior is usually deduced from the physical laws driving the evolution of the system. Recently, several sophisticated models were proposed (Barros and Mendes, 1997; Karafyllidis and Thanailakis, 1997; Hernández Encinas et al., 2007; Yassemia et al., 2008) and successfully validated by comparison with real fires. All these models use either simple or Dynamical Structure Cellular Automata (DSCA).

With reference to deterministic models, based on Weber's classification (Weber 1990), three kinds of mathematical models for fire propagation can be identified according to the methods used in their construction. The first type of models are statistical models (McArthur 1966), which make no attempt to include specific physical mechanisms, being only a statistical description of test fires. The results can be very successful in predicting the outcome of fires similar to the test fires. However, the lack of a physical basis means that the statistical models must be used cautiously outside the test conditions. The second category of models is composed of semi-empirical models (Rothermel, 1972) based on the principle of energy conservation but which do not distinguish between the different mechanisms of heat transfer. Rothermel's stationary model is a one-dimensional model, in which a second dimension can be obtained using propagation algorithms (Richards, 1990) integrating wind and slope. Finally, physical models (Albini, 1985) integrate wind and slope effects in a more robust manner by describing the various mechanisms of heat transfer and production. Physical mechanisms are described using a chemical, thermal and mechanical definition of basic fire phenomena. Hence, physical and semi-empirical models use the definition of basic fire phenomena to physically describe fire propagation.

With reference to stochastic models, few works have been achieved. In (Hargrove et al., 2000), authors explore the results of various stochastic experimental designs (according to various classes of moisture and ignition probabilities based on percolation thresholds) of a replicated fire spread simulation. No fuel biomass is taken into account. Besides, authors argue that "each simulation was replicated five times", without demonstrating the choice of the number

of replications although serious studies of the stochastic variability of results have to be done (Kleijnen and Groenendaal, 1992). In (Jimenez et al., 2007), a sensitivity analysis of Rothermel's model using Monte Carlo simulation is provided. In (Gu and Hu, 2008), Monte Carlo simulation is used to evaluate the error of fire spread data acquisition during a fire spread simulation. This error is then integrated in the fire spread simulation. On stochastic fire spread modeling two major approaches need nonetheless to be mentioned. The first is a very precise mathematical framework for stochastic fire spread modeling (Vorob'ov, 1996). In the latter a very clear presentation of ellipse based stochastic model is presented. An interesting concept of random spread process is introduced. However, here too no discussions on the stochastic variability of results, neither on biomass modeling, are provided. More recently, in the reference journal *Combustion and Flame*, a very interesting and promising article (Porterie et al., 2007) has been introduced (and apparently discussed with the Nobel Prize P.G. de Gennes). Small world networks, percolation and stochastic simulations are designed. This demonstrates a new interest of the physicist community for computer-based techniques. Long range spotting is modeled and analytic physics-based equations are used to predict fire brand distances and numbers.
Concerning the discrete-event design of fire spreading models, many works can be cited:

- In (Vasconcelos et al., 1995), a first discrete-event design of Rothermel's model is proposed. Discrete-events correspond to the patch burning times provided by Rothermel's model. An experimental frame is defined. This frame embeds ecological data of the usual fuel classes provided by Rothermel's model.
- In (Barros and Mendes, 1997), dynamic structure cellular automata are used to upload memory only with burning active cells obtained through Rothermel's model. A dynamic structure specification is provided. Memory reductions are discussed.
- In (Ameghino et al., 2001), a high-level specification of (Vasconcelos, 1995) is presented. The specification exemplifies the delay-focused and cell-focused macro-instructions of the Cell-DEVS formalism. As macro-specifications and instructions are automatically embedded in Cell-DEVS, a code reduction is observed.
- In (Muzy et al., 2005), an intensive fine-grained semi-physical model of fire spread has been investigated through discrete-event design. Optimizations (at the implementation and specification levels) are investigated in (Muzy, et al. 2003) to account for dynamic structures and discrete-time modeling. Recently, in (Muzy et al., 2008), the quantization technique has been investigated.
- In (Ntaimo et al., 2008), previous work achieved in (Vasconcelos, 1995) has been detailed and extended to many scenarios including wind, slope and fire fighting. Dynamic structures have been investigated in (Sun and Hu, 2007). A stochastic implementation comparing experimental data and simulation ones (obtained through Rothermel's model) is described.

The complexity of fire spread modeling induces bottlenecks at both physical and computer modeling levels. At the physical level, physicists use concepts (convection, radiation, diffusion) and corresponding usual mathematical models (partial differential equations and ordinary differential equations) to describe fire spread models. After a discretization through usual numerical methods (Euler, Runge-Kunta, etc.), computer-scientists use then design techniques (object-oriented, discrete-events, meta-modeling, dynamic structure, etc.) They discuss design and execution time advantages of the techniques used. However, many scientific disadvantages emerge from this approach. Design issues desperately try to follow the evolutions of discretized

physics-based models. A first drawback is that the computer design is totally dependent to discretization techniques [except perhaps now with quantization techniques: (Zeigler et al., 2000)]. A second drawback is that computer scientists usually do not master physics-based modeling techniques and physicists do not exploit the full advantages of computer-based modeling techniques they sometimes do not know. A third drawback is that the filters constituted by both physics-based concepts and discretization techniques result in multidimensional parameters, which do not directly represent physical, topological, biological, properties of fire fronts and fuels. Modeling levels are not grasped in a single consistent and complete top-down approach (from problem to implementation). Using a consistent stochastic framework, embedding experimental plans, to model (spatial) distributed systems, scientists are able to master and experiment the full modeling and simulation process (even if they need usually expert advises).

## Design and modeling prerequisites with simulation results

Complexity of fire spreading, which necessitates many simulation models interacting in parallel, allows investigating and discussing many design, modeling and simulation perspectives. Every solution presents drawbacks and advantages the modeler can deal with. Efficiency, error, design facilities are the main interdependent parameters the modeler needs to compose with. After, fire propagations occurring at a laboratory small scale are used to precisely discuss advantages and drawbacks of structure choices of models and simulators. A physics-based model is implemented. Lastly, virtual experiments are developed on a personal computer to define a new coherent stochastic model of fire spread.

### *Design of cellular models*

Modularity of models and simulators as well as their ability to efficiently detect activity are discussed here after.

#### *Modularity of models and simulators*

As depicted in Figure 1, various choices of modularity can be done for describing a spatial system in a uniform way through cellular models.

At the modeling level, two levels of modularity can be implemented. A first fully modular model level consists of cells whose state is encapsulated. Cells interact through interface ports and every cell is designed as receiving and sending events. At a previous weak modularity level, cells do not have ports. They directly access the state of influencing neighbors.

At the simulation level, two levels of aggregation can be implemented. At every level, a root can implement the whole simulation loop. At a fully aggregated level, a single simulator is piloted by the root. Another solution is to affect one simulator to every cell. This solution increases the number of messages exchanged (decreasing efficiency) but enhances reusability.

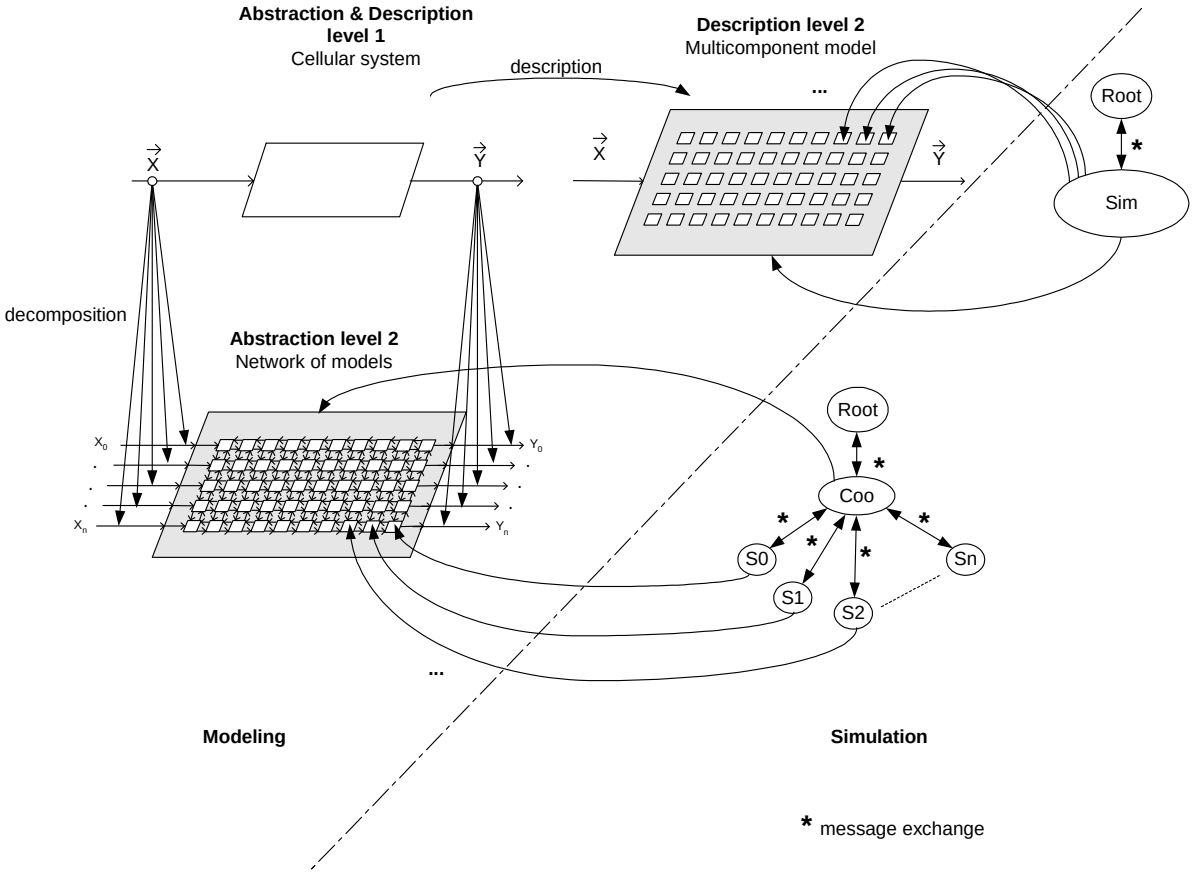



**Figure 1.** Comparison between modular and non-modular modeling and simulation (Muzy et al., 2003)

A first full modular implementation at both model and simulator levels has been implemented through the Cell-DEVS formalism (Wainer and Giambiasi, 2001) and the corresponding CD++ environment (Wainer, 2002). The laboratory experiment consists of a combustion table of *30cm* long and *60cm* wide for a line-ignition, the prediction of spread rate (*2.96mm/s*). For a real propagation of *150s* of a *100*x*100* cell space, the execution time of the Cell-DEVS simulation were about *21h20min* (with a *Pentium III at 500 Mhz*). Even if this modeling and simulation experiment is elegant and grounded formally, the many messages exchanged produce overhead because of data structure management (of schedulers). This results in an execution time much greater than the actual propagation time.

An opposite fully aggregated simulator and non-modular model has been implemented then (Muzy et al., 2003). A point-ignition has been simulated by initializing center cells with a temperature gradient. For a real propagation of *200s*, execution times decreased to *160s*. On the other hand, the whole modeling and simulation model is less reusable.

*Activity tracking*

Models can be designed for tracking activity during the whole simulation loop. Metaphorically, distributed sub-models (constituting the whole model) can be considered as filtering gates orienting and computing activity. As depicted in Figure 2, models are activated or not by the flows of external events or by scheduled internal events (bottom right corner). This activation depends on the adjustment of a sensitivity parameter determining the activation of the model. Activity in a network of models depends on: (i) State transitions or internal events, and (ii) External events received and sent.

Concerning finite state propagations, state changes are discrete and no information is loss. The types of events are defined before the simulation. For every (external or internal) event type, a corresponding state transition is attributed. As in finite state automata, state can then remain unchanged or switch to another one.

Concerning continuous values embedded in discrete events, a threshold filter needs to be set to determine the relevance of the discrete-event received and of changes in states. If the value received is greater than this threshold, the corresponding component will be activated. Otherwise, the event is not taken into account and the component will return inactive. If states do not "change enough" to activate the model, the latter can be designed to inform the mdoels it influences that it turns inactive (or it can simply not send events). If states "change enough" to activate the model, the latter can be designed to inform the components it influences of this change (Zeigler et al., 2000). Using such a design mechanism allows reducing computations and message exchanges. However, according to the amplitude of this filtering threshold, error is induced. The larger is the threshold the greater is the error. Many good results have been obtained using activity tracking modeling and simulation of fire spread (Muzy et al., 2008).

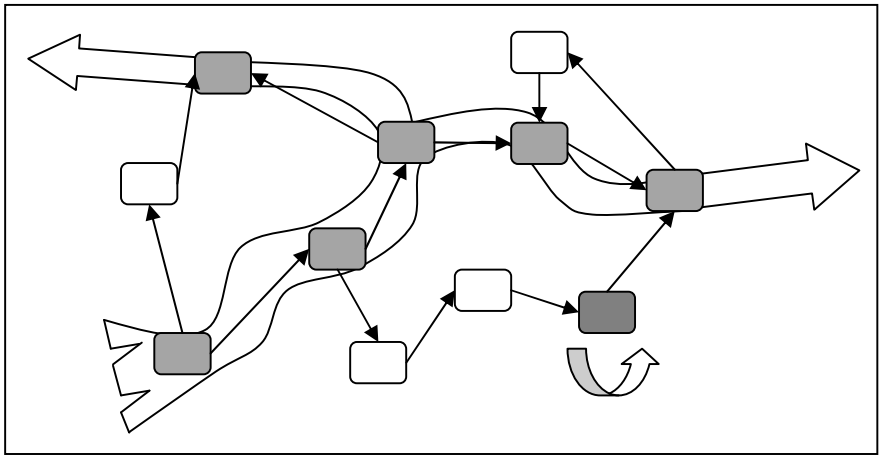


**Figure 2.** Active components

*Design choices*

Many choices can be achieved at the modeling and simulation levels for enhancing efficiency. First according to the degree of reusability requested, modularity can be modulated at the modeling and/or simulation levels. Obviously, reducing the degree of modularity, reduces the degree of reusability of components. Full modular components are designed in an autonomous

way. Reducing this modularity, the internal structure of models depends of other models. Another solution to enhance efficiency is to design the models to automatically track activity. The efficiency here is inversely proportional to the precision and depends on the amplitude of the filtering threshold chosen [more details about the error induced in (Muzy et al., 2008)].

***Fire spread modeling through a virtual laboratory***

In Physics, physicists use equation-based structures to represent fundamental identified physical mechanisms involved in both fire and diffusion processes. Parameters of equations have been (or are) identified through laboratory experiments. These parameters do not correspond directly to the parameters intuitively identified as influencing the fire spread (slope, wind, biomass, humidity). The latter parameters are split in many parameters, which aim at describing very finely physicochemical mechanisms (gas dynamics, temperature radiation, etc.) These models are more adapted to fine-grain physics-based laboratory experiments than actual fire spread.

Computer scientists have a different approach. First their laboratory consists only of one or many computers. They directly build models and simulate them. Then, they verify if the model behavior fits their expectations describing consistent dynamics. They calibrate their model adjusting parameters exploring parameter values through experimental plans. After, the first constructive experiments designed by computer scientists are described.

*Experimental model 1: Near-to-near propagation including firebrands*

Because of the numerous phenomena (entangled and spatially distributed) occurring uniformly and in parallel in a fire spread, cellular automata constitute an appropriate paradigm to model and simulate them. A first very simple program in *C++* with a textual interface is sufficient to draw a first fire spread model.

Cells of the cellular automaton hold a discrete state constituted of symbols as follows:

**'.'** : The cell is unburned,
**'*'**: The cell is burning,
**'x'**: The cell is burnt.

Fire propagation is twofold:

1. Near-to-near propagation: Every burning cell gets one chance over eight (for the eight neighbors) to ignite a neighbor. In case of wind, neighbors are selected according to the strength and direction of the wind.
2. Firebrand propagations: Every burning cell produces one firebrand in the direction of the wind. The distance depends on the wind strength. Distances of the firebrand are calculated as follows:

$$\begin{cases} dx = i + w_x[\theta(\varepsilon + \vartheta)] \\ dy = j + w_y[\theta(\varepsilon + \vartheta)] \end{cases}$$

Where, $dx$ and $dy$ are the distances of firebrand projection in directions $x$ and $y$, $\theta \epsilon [0,1]$ is a random real number, *(i,j)* are the coordinates of cells, $w_x$ and $w_y$ are the wind

directions, $\vartheta$ is the wind strength and $\varepsilon$ is a parameter adjusted to observe modification of fire spread.

This very simple model allows obtaining consistent results as represented in Figure 3.

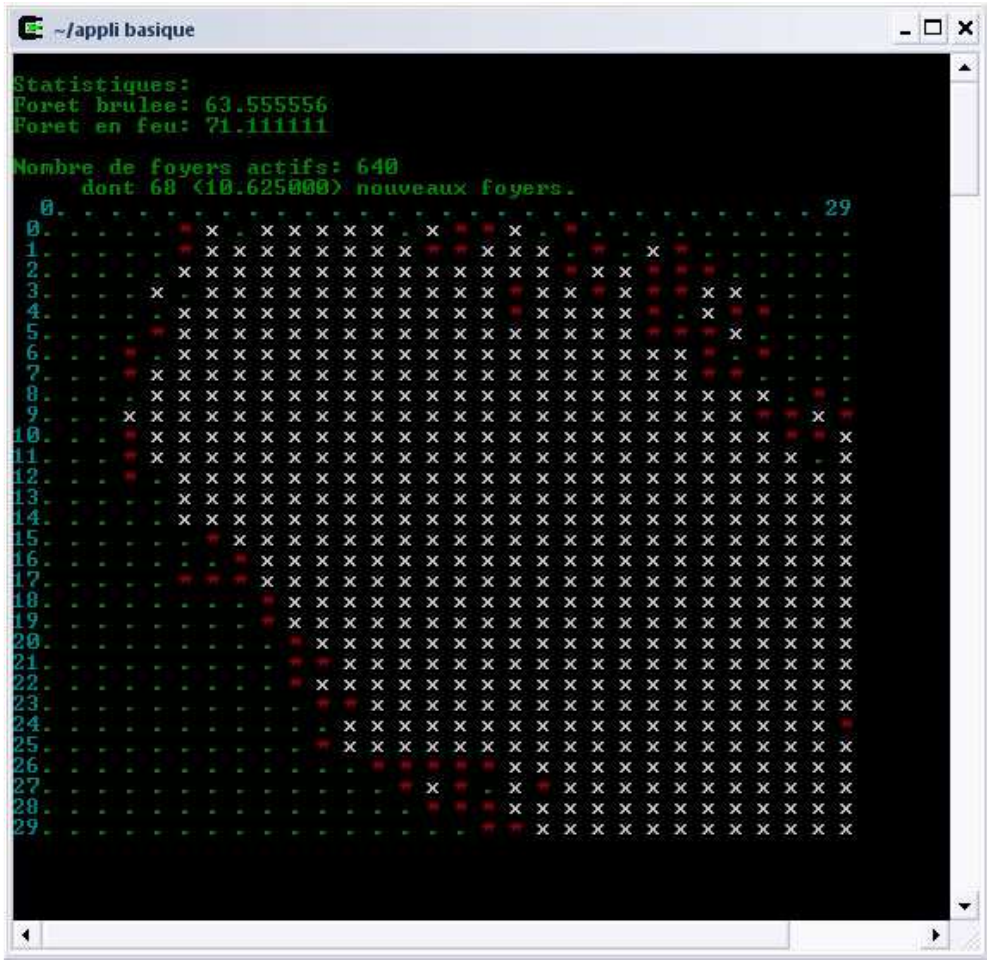


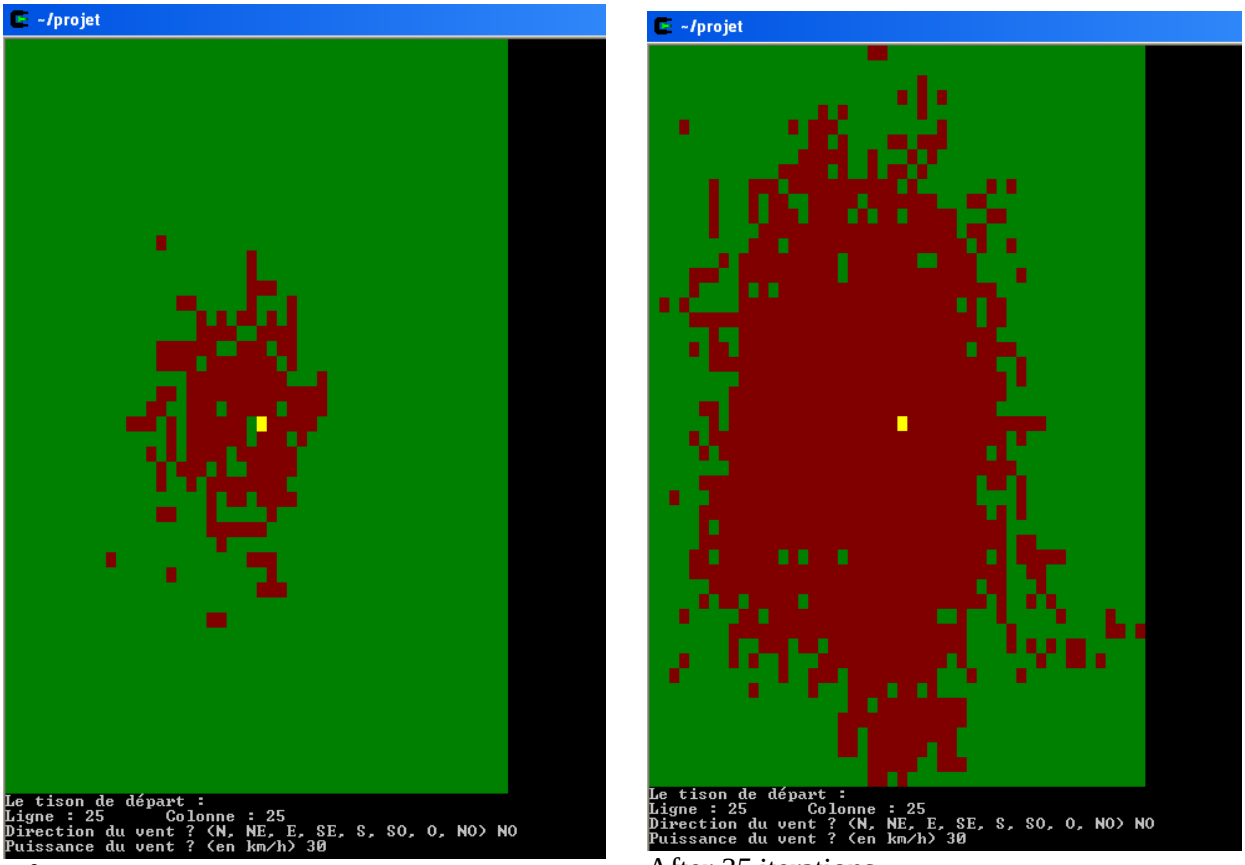

**Figure 3.** Fire spread under a wind of force 2 in South-West direction

*Experimental model 2: Pseudo-random distributions of firebrands*

Here, cells have two states: unburned and burning. Parameters correspond to: (i) The strength of the wind, and (ii) The wind direction. The direction corresponds to the angle value respect to a horizontal wind. Fire propagates only by firebrands. The distance *D* of projection is determined through a negative exponential law whose mean is proportional to wind strength. Thus, the greater is the wind strength, the more distant would be the firebrand. The direction is calculated according to wind direction and to a noise parameter $\beta\epsilon[-1,1]$ through a Gaussian law of mean equal to zero and deviation inversely proportional to *D*. Thus, when the distance is large, the firebrand direction corresponds to the wind direction ($\beta$ is minimum). When the distance is

small, the firebrand is projected in whatever direction. This latter case corresponds to a near-to-near propagation.

Results are presented in Figure 4. Some imperfections of the model can be observed. Some cells inside the fire front remain unburned. However, the firebrand method provides acceptable results. The model is sensible to the proportions chosen between wind strength and the mean of the exponential law.




After *20* iterations

After *35* iterations

**Figure 4.** Fire spread under a wind of 30 km/h in North-West direction

*Final Experiment*

These simple models allowed to design and implement a final simulator. The finite states used can be refined using a biomass decrease in cells. According to the level of degradation of cells, corresponding states are selected (unburned, heated, burning and burned). Another difficulty relates to the number of parameters to calibrate. This number needs to be limited to be able to explore the full state space of the final model. The latter implements the near-to-near propagation of model 1 and firebrand propagation of model 2.

*Implementation*

A design pattern Model-View-Controller is used. This design pattern mixes the Observer and Strategy ones (Gamma et al., 1994). It allows:

- Separating variable aspects from static aspects of the implementation,
- Favoring the composition against inheritance,
- Programing interfaces,
- Reducing couplings between objects.

Entities of the fire spread simulator according to the Model-View-Component are:

1. Model: which contains the whole simulator,
2. View: which provides the window interface,
3. Controller: which manages events from the window interface and transfers changes from view to model.

The pseudo-random generator used is the Mersenne Twister (Matsumoto and Nishimura, 1997) from SSJ java library (L'Ecuyer and Buist, 2005). Figure 5 depicts the main diagram of the whole implementation. The class Simulator is composed of Tree(s) and Shrub(s) and replications are used for representing spectral analyses (Hill, 1997). By replicating simulations, the whole span of exponential and Gaussian laws can be explored. Every replication is saved. At the end of the simulation, spectral analyses are computed. Spectral analyses correspond to a sum of probabilities of fire spread. In the Simulator class a method for image analysis is implemented. This method converts an image in Element(s) (burnable and unburnable).

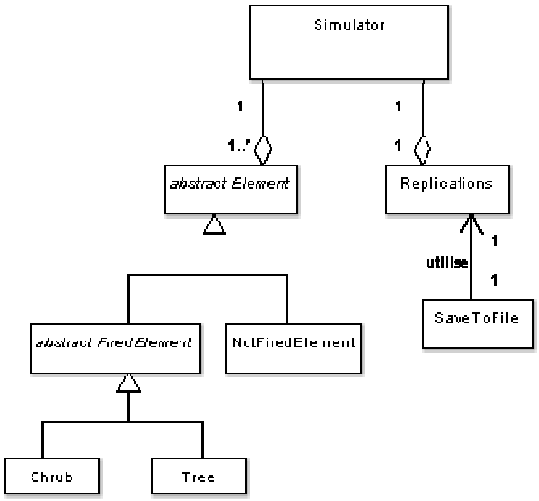



**Figure 5.** Simulator class diagram

*Interface and simulation results*

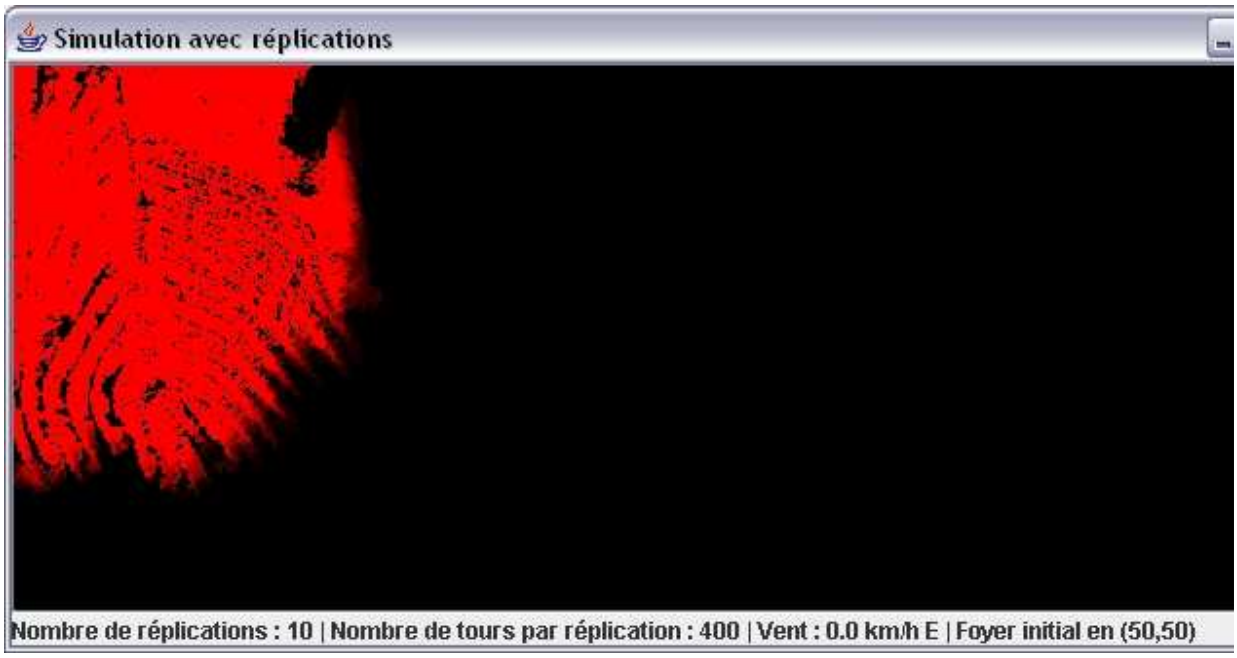
Parameters of the interface are: Strength and directions of the wind, as well as the real-time simulation speed. Actual aerial pictures can be uploaded (*cf.* Figure 6).


**Figure 6.** Interface parameters

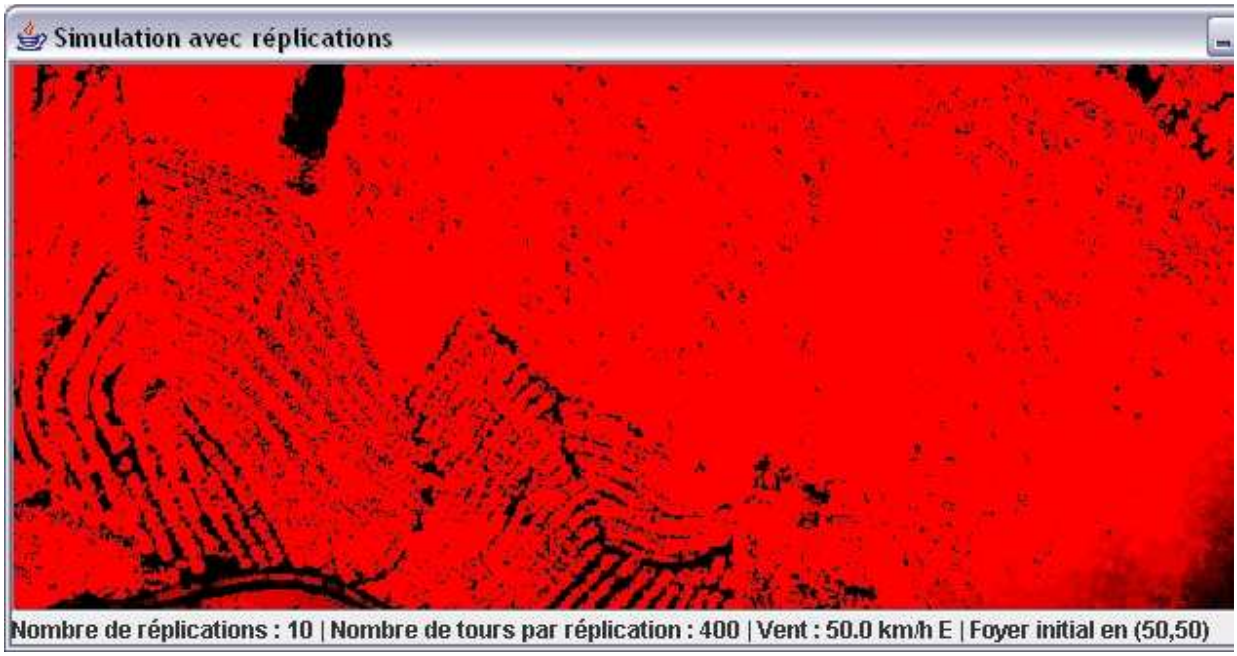
Figure 7 depicts a fire spreading. It can be noticed that the unburnable elements constituted of the road are not burned.


**Figure 7.** After *141* iterations, with a wind set to West direction at iteration *120*.

Figure 8 presents the replication results of fire propagation.




(a)




(b)

**Figure 8.** (a) Ignition on the top left corner after *10* replications of *400* iterations each without wind (b) Ignition on the top left corner after *10* replications of *400* iterations each with a wind of 50 km/h in East direction

Last, Figure 9 represents possible fire propagation in 3D in a virtual environment (near ISIMA building).

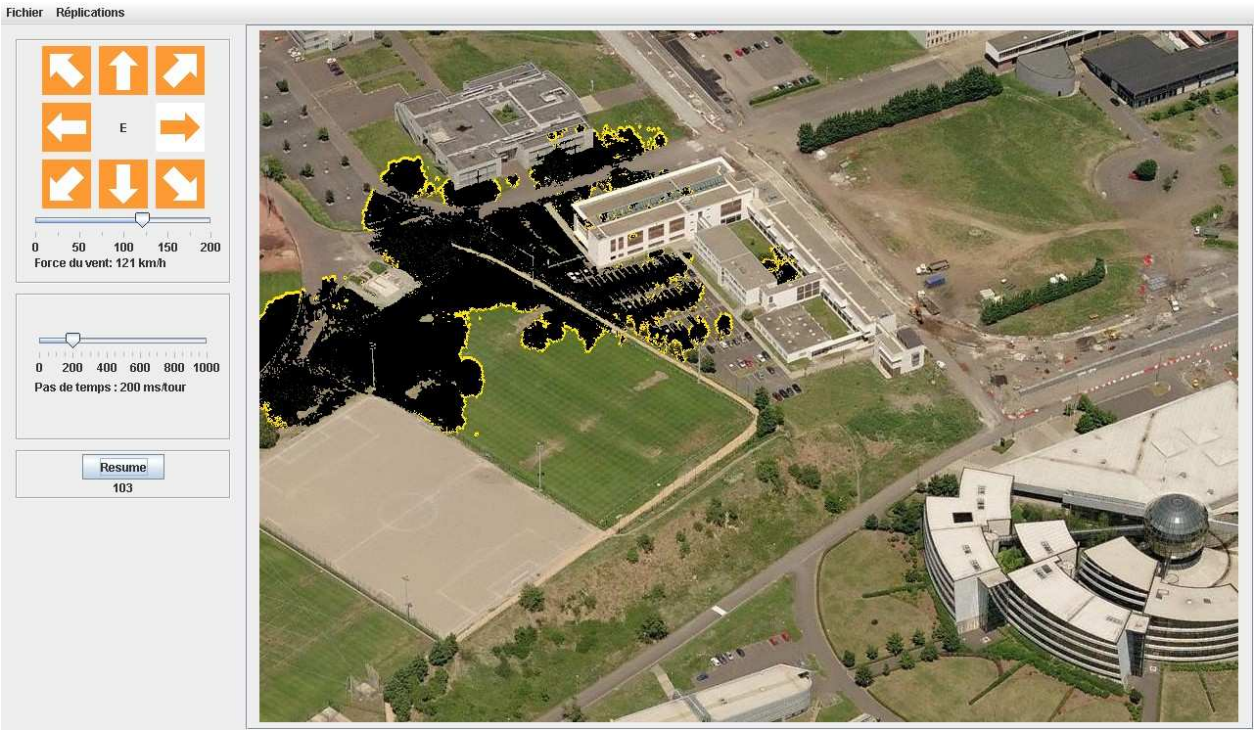



**Figure 9.** Wind in East direction of 121 km/h.

## Conclusion and perspectives

In this study advantages of stochastic modeling for implementing spatial constraints have been presented. In several cases, stochastic modeling is the only strategy available to represent the distant and discrete interactions. The formalization in DEVS of stochastic modeling is possible. In (Zeigler, 1976), a formal specification of pseudo-random generators is described. Pseudo-random generators being deterministic, DEVS describes stochastic models in terms of systems embedding pseudo-random generators. Recently, in (Kofman and Castro, 2006), a formal extension of DEVS has been provided. A DEVS-based description of stochastic models through STDEVS would allow the modeling and simulation community to explicitly explore this powerful domain. DEVS popularity denotes a right level of definition. It is abstract enough to allow the development of new concepts and structures. It is precise enough to avoid ambiguities and guide modelers. Since its first book definition (Zeigler, 1976), many developments have been achieved. A constant critical evaluation of these developments is necessary to enhance the whole framework. However, these developments have to be at the right level of abstraction. Because of the increasing complexity of hardware and software architectures, metalevels (and automatic mappings from higher to lower levels) need to be designed. Considering complex systems as multilevel and highly composed, detecting and mapping simulation structures on activity will be more and more necessary. Activity tracking mechanisms will allow improving DEVS efficiency and abstraction.

Integrating explicitly modularity and efficiency issues, as well as experimental plans and stochastic modeling strategies in DEVS, (computer) scientists will gain freedom and dispose of a more complete and powerful modeling and simulation framework.

## References


Albini, F. A., 1985. A model for fire spread in wildland fuels by radiation. Combustion Science and Technology 42: 229-258.

Amblard, F., D. R. C. Hill, S. Bernard, J. Truffot, and G. Deffuant, 2003, MDA compliant design of SimExplorer, A software to handle simulation experimental frameworks, in Proc. Summer Simulation Conf. (SCSC) 2003, Montréal, QC, Canada, Jul. 20–24, 279–284.

Ameghino, J.; Troccoli, A.; Wainer, G., 2001. Models of complex physical systems using Cell-DEVS. Proceedings of Annual Simulation Symposium, Seattle, WA. U.S.A.

Balbi, J.H., P.A. Santoni, and J.L. Dupuy, 1998. Dynamic modelling of fire spread across a fuel bed, Int. J. Wildland Fire, 1998, 275-284.

Balci O., 1988. The Implementation of four conceptual Frameworks for Simulation Modeling in High-Level Languages, Technical Report SRC-88-009, System Research Center, Virginia State University.

Barros, F. J. and M. T. Mendes, 1997. Forest fire modelling and simulation in the DELTA environment. Simulation Practice and Theory 5(3): 185-197.

Coquillard P., 1995, Simulation of the cyclical process of heathlands. Induction of mosaics structures, evolution to irreversible states, Ecological Modelling, Vol. 80, 97-111.

Coquillard P., Fain J., 1995. Monte-carlo simulation of colonization by pinus sylvestris of heathland on degenerate phase on volcanic substrate. In Bellan-Santini D., Bonin G., C. C. Emig (Eds), Functioning And Dynamics Of Natural And Perturbed Ecosystems,. Lavoisier Pub., Paris, 274-289.

Coquillard, P., D.R.C. Hill, 1997, Modélisation et simulation d'écosystèmes — Des modèles déterministes aux simulations à événements discrets, Collection Ecologie, Masson, Paris, 1997.

Dahl, O. J., Nygaard K., 1966, Simula - An algol based simulation language. Communication of the ACM. Vol. n°9, 671-678.

El Bitar Z., Lazaro D., Breton V., Hill D., Buvat I., 2006, Fully 3D Monte Carlo image reconstruction in SPECT using functional regions. Nucl. Instr. Meth. Phys. Res. 569, 399-403.

Faure, T. and G. Deffuant, 2006, SimExplorer: A software tool for programming and executing experiemental designs on complex models," in Proc. Summer Simulation Multiconf., Calgary, Canada, 218–225.

Fishman G.S., 2000, Monte Carlo. Concepts, Algorithms and Applications, 2000, Springer.

Gamma, E., Helm, R., Johnson, R., and Vlissides, J., 1994. Design patterns. Elements of Reusable Object-Oriented Software. Addison Wesley. ISBN:0-201-63361-2.

Gu, F., X. Hu, 2008, Towards Applications of Particle Filter in Wildfire spread Simulation, Proc. 2008 Winter Simulation Conference (WSC'08).

Hargove, W. W., R. H. Gardner, M. G. Turner, W. H. Romme and D. G. Despain, 2000. Simulating fire patterns in heterogeneous landscapes. Ecological Modelling 135: 243-263.

Hernández Encinas Hoya White S., Martín del Rey, A., Rodríguez Sánchez G., 2007. Modelling forest fire spread using hexagonal cellular automata. Applied Mathematical Modelling 31: 1213–1227.

Hill D., Coquillard P., De Vaugelas J., Meinesz A., 1998, An algorithmic Model for Invasive Species Application to Caulerpa taxifolia (Vahl) C. Agardh development in the North–Western Mediterranean Sea. Ecological modelling, Vol. 109, pp. 251-265.

Hill D.R.C., 1997, Modélisation des processus d'expansion : application à Caulerpa taxifolia", Conférence Internationale sur la " Dynamique des Espèces invasives ", 13-15 mars 1997, Paris, Académie des Sciences. Tec & Doc, 219-230.

Hu, X., and B. P. Zeigler, 2004. A high performance simulation engine for large-scale cellular DEVS models. High Performance Computing Symposium (HPC'04), Advanced Simulation Technologies Conference (ASTC), 2004, p. 3-8.

Hu, X., B. P. Zeigler, 2004. A high performance simulation engine for large-scale cellular DEVS models. High Performance Computing Symposium (HPC'04), Advanced Simulation Technologies Conference (ASTC), Arlington, USA, 3-8.

Innocenti, E., Muzy, A., Aïello, A., Santucci, J., F., 2004. Discrete Event Simulation of fire spread: A modular Approach for the Choice of a Strategy of Active Elements Management, In proceedings International Carpathian control conference ICCC'2004, Zakopane Poland, 69-74.

Innocenti, E., Muzy, A., Hill, D., R., C., Aïello, A., Santucci, J.F., 2004. Active-DEVS : A computational model for the simulation of fire propagation. In proceedings IEEE, SMC 2004, International Conference on Systems, Man and Cybernetics, October 10-13. The Hague, The Netherlands, 1857-1863.

Jimenez, E., Hussaini, M.Y., and Goodrick, S., 2007. Quantifying parametric uncertainty in the Rothermel model, International Journal of Wildland Fire 17(5) 638–649.

John Ellis, N. E. Mavromatos1, and D. V. Nanopoulos, 2000, Quantum-Gravitational Diffusion and Stochastic Fluctuations in the Velocity of Light, General Relativity and Gravitation, Volume 32, Number 1 / janvier, 127-144

Karafyllidis, I., Thanailakis, A., 1997. A model for predicting forest fire spreading using cellular automata. Ecological Modeling 99, 87-97.

Kleijnen J. Groenendaal W. V., 1992. Simulation a Statistical Perspective. Wiley.

Kleijnen J.P.C. ,1997, Statistical tools for simulation practionners. Marcel Dekker Inc. Pub., New York, USA.

Kofman, E. and Castro, R., 2006. STDEVS. A Novel Formalism for Modeling and Simulation of Stochastic Discrete Event System. Technical Report LSD0603, LSD, Universidad Nacional de Rosario. Accepted in AADECA 2006.

L'Ecuyer P. and Buist E., 2005, Simulation in Java with SSJ, Proceedings of the 2005 Winter Simulation Conference, 611-620.

Lee, W. B., and T. G. Kim, 2003. Simulation speedup for DEVS models by composition-based compilation. Summer Computer Simulation Conference, 395-400.

M. Pidd, 1992, Object-orientation and three phase simulation, in: Winter Simulation Conference, 689–693.

M.A. Lewis, 2000, Spread rate for a nonlinear stochastic invasion, Journal of Mathematical Biology, Volume 41, Number 5, 430-454.

Martin Falcke, Lev Tsimring, and Herbert Levine, 2000, Stochastic spreading of intracellular Ca2+ release", PHYSICAL REVIEW E VOLUME 62, NUMBER 2, 2636-2643.

Matsumoto M. and Nishimura T. Mersenne Twister, 1997, A 623-dimensionally equidistributed uniform pseudorandom number generator, Proceedings of the 29th conference on Winter simulation, 127-134.

McArthur, A. G., 1966. Weather and grassland fire behaviour. Australian Forest and Timber Bureau Leaflet 100.

Mell, W.E., Jenkins, M.A., Gould, J. and Cheney, P. 2007. A Physics-based approach to modeling grassland fires. International Journal of Wildland Fire 16(1) 1-22.

Morvan D., Dupuy J. L., 2004. Modeling the propagation of a wildfire through a Mediterranean shrub using a multiphase formulation, 2004. Combustion and flame. 138 (3) 199 – 210.

Muzy A., E. Innocenti, G. Wainer, A. Aiello, and J. F. Santucci, 2002. Cell-DEVS quantization techniques in a fire spreading application, Winter Simulation Conference (WSC) - Exploring new frontiers, IEEE/ACM/SIGSIM/SCS, San Diego, USA, 542-549.

Muzy, A. , E. Innocenti, G. Wainer, A. Aiello, and J. F. Santucci, 2005. Specification of discrete event models for fire spreading, Simulation: transactions of the society of modeling and simulation international, SCS, 81, 103-117.

Muzy, A. , G. Wainer, E. Innocenti, A. Aiello, and J. F. Santucci, 2002. Comparing simulation methods for fire spreading across a fuel bed, Simulation and planning in high autonomy systems conference (AIS), SCS, Lisbon, Portugal, 219-224.

Muzy, A. and B. P. Zeigler, 2008. Introduction to the Activity Tracking Paradigm in Component-Based Simulation."The Open Cybernetics and Systemics Journal 2: 48-56.

Muzy, A., E. Innocenti, D. Hill and J. F. Santucci, 2003. Optimization of cell spaces simulation for the modelling of fire spreading. 36th Annual Simulation Symposium, Orlando, USA, IEEE/SCS/ACM, 289-296.

Muzy, A., J. J. Nutaro, 2005. Algorithms for efficient implementation of the DEVS & DSDEVS abstract simulators. 1st Open International Conference on Modeling and Simulation (OICMS), Clermont-Ferrand, France.

Muzy, A.,J. Nutaro, B.P. Zeigler, and P. Coquillard, 2008. Modeling and simulation of fire spreading through the activity tracking paradigm. Ecol. Model., 219(1-2). p. 212-225.

Natimo L., X. Hu, Y. Sun, DEVS-FIRE: Towards an Integrated Simulation Environment for Surface Wildfire Spread and Containment, SIMULATION, Vol. 84, Issue 4, pp. 137-155, 2008.

Ntaimo, B. P. Zeigler, et al. 2004. Forest fire spread and suppression in DEVS. Simulation. 80 479-500

Peter Jung and Gottfried Mayer-Kress, 1995, Spatiotemporal Stochastic Resonance in Excitable Media, Phys. Rev. Lett. 74, pp. 2130 – 2133.

Porterie P., Zekri N., Clerc J.P., Loraud J.P., Modeling forest fire spread and spotting process with small world networks, 2007. Combustion and Flame. 149 (1-2) 63-78.

Posse, E., Muzy, A., Vangheluwe, H. 2006. “A framework for visual specification and simulation of cellular systems”. In Proceedings of the 2006 DEVS Integrative M&S Symposium (DEVS'06). 2006.

Richards, G. D., 1990. An elliptical growth model of forest fire fronts and its numerical solution. International Journal of Numerical Method Engineering 30: 1163-1179.

Rothermel, R. C., 1972. A mathematical model for predicting fire spread in wildland fuels, USDA, Forest Service Research.

Shiginah, F. A. S. B. 2006. Multi-Layer Cellular DEVS Formalism for Faster Model Development and Simulator Efficiency. PhD Thesis. Electrical and Computer Engineering Dept., University of Arizona.

Sun, Y., X. Hu, 2008, Partial-modular DEVS for improving performance of cellular space wildfire spread simulation, Proc. 2008 Winter Simulation Conference (WSC'08).

Sun, Y., X. Hu, Performance Measurement of DEVS Dynamic Structure on Forest Fire Spread Simulation, Proc.14th AI, Simulation and Planning in High Autonomy Systems (AIS 2007), 2007.

Vasconcelos, M. J., J. M. C. Pereira and B. P. Zeigler, 1995. Simulation of fire growth using discrete event hierarchical modular models. EARSeL Advances in Remote Sensing 4(3): 54-62.

Vorob'ov, O. Yu. , 1996. Random set models of fire spread Fire Technology, Springer Netherlands, Volume 32, Number 2 / avril 1996.

Wainer, G. 2002. CD++: a toolkit to develop DEVS models. Software—Practice & Experience 32 (13).

Wainer, G. and N. Giambiasi 2001. Application of the Cell-DEVS paradigm for cell spaces modeling and simulation. Simulation 76(1): 22-39.

Weber, R. O., 1990. Modelling fire spread through fuel beds. Progress in Energy and Combustion Science 17: 67-82.

Yassemia S., Dragićević S. ,Schmidt M., 2008. Design and implementation of an integrated GIS-based cellular automata model to characterize forest fire behaviour. Ecological Modelling 210 : 71–84.

Zeigler, B.P., 1976. Theory of Modelling and Simulation. Wiley Interscience, first edition.

Zeigler, B.P., Praehofer, H., Kim, T.G., 2000. Theory of Modelling and Simulation, second ed. Academic Press.