OBJECT-ORIENTED FRAMEWORK FOR MODELLING AND SIMULATION OF PROPAGATION PROCESSES : APPLICATION TO A FIRE SPREADING

Alexandre Muzy*, Eric Innocenti*, David R.C. Hill**, Antoine Aïello*,
Jean-François Santucci*

*University of Corsica
SPE – UMR CNRS 6134
B.P. 52, Campus Grossetti, 20250 Corti FRANCE
e-mail: a.muzy@univ-corse.fr
URL http://spe.univ-corse.fr/simffweb/debut.htm

**ISIMA/LIMOS UMR CNRS 6158
Blaise Pascal University
Campus des Cézeaux BP 10125, 63177 Aubière Cedex FRANCE
e-mail: drch@isima.fr

ABSTRACT

An important class of ecological problems concerns propagation processes. In ecological modelling, these phenomena generally occur on large scales and are generally difficult to efficiently simulate because of the number of entities and connections between entities. Studies of this kind of phenomena lack genericity and reusability because they are often presented through the point of view of a single domain expert. Simulations made by domain experts seem to lack genericity for computer science specialists and simulations developed by computer science specialists seem not to grasp modelling terminology and problems of the domain experts. We propose here a general object-oriented framework for modelling and simulation of propagation processes. Object-oriented techniques allow to provide genericity and reusability. From modelling to simulation, the Unified Modeling Language (UML) provides a common means of communication between computer science specialists and domain experts. The Model Driven Architecture (MDA) is used to improve object-oriented methodology. Simulation optimisations are defined for implicit and explicit models of propagation. The approach is applied to the modelling and simulation of fire spread. Starting from wildland fire problems, specification levels are used to gradually specify a fire spread simulator. Each level of the study is specified in UML and thus can be reused in another wildland fire problem.

Keywords: Fire spread modelling and simulation, propagation processes, object-oriented techniques, discrete event simulation, Model Driven Architecture.

1 INTRODUCTION

Ecosystems show a high degree of heterogeneity in space and time (Jørgensen and Bendoricchio, 2001). Their study often necessitate to develop models taking into account the system evolution in both space and time. These models are spatially distributed models. Among the phenomena studied through spatially distributed models, propagation processes where a lateral transfer of energy, mass or information occurs (fire spread, oil spills, insect infestation, watersheds etc.) represent an important class of problems.

Modelling propagation processes on large scales generally present two main difficulties. First the high number of intricate phenomena influencing the propagation leads to collaborate with specialists of different disciplines (ecology, physics, mathematics, biology etc.). Then the large volume of data and the number of operations the models have to manage lead to the use of computer simulation. From system analysis to programming, object-oriented methodology can be used to facilitate these two tasks.

Object-oriented approach is now well recognised to benefit ecological modelling directly (Silvert, 1993) and has been used in different ecosystem simulations (Sequeira et al., 1991; Baveco and Lingeman, 1992; Chen and Reynolds, 1997; Holst et al., 1997; Hill et al., 1998; Neil et al., 1999; Alfredsen and Saether, 2000; Spanou and Daoyi, 2000; Yang et al., 2002). The benefits of this approach for ecological modelling can be summarised as follows: (1) the safe reuse of code increases productivity; (2) clear and understandable code is open to scientific review; (3) through easier maintenance, simulation models remain useful research tools for long periods and; (4) common modelling concepts facilitate communication within and across research groups.

Simulation of propagation processes is performed through cell space models. In ecosystem propagation, landscape is divided into parcels of land (cells) interacting together. According to Tobler (1979), a cellular (or cell) space model consists of an infinite two-dimensional array of polygons (cells), each of which is, at any time, in a state determined by the states of a set of "neighbor" cells according to some uniform location-independent rules.

Cell space dynamic is generally achieved using Cellular Automata (CA) (Von Neuman, 1966). Standard CA are simple mathematical idealisations of natural systems (Wolfram, 1994). They have been widely applied in ecological modelling (Spencer, 1997; Dunkerley, 1999; Sirakoulis et al., 2000; Dumont and Hill, 2001; Matsinos and Troumbis, 2002; Soares-Filho et al. 2002; El Yacoubi et al. 2003; Loibl and Toetzer, 2003). CA consist of an infinite lattice of discrete identical sites, each site taking on a finite site of, say, integer values. The values of the sites evolve in discrete time steps according to deterministic rules that specify the value of each site in terms of the values of neighboring sites. CA are models where space, time and states are discrete (Jen, 1990).

The originality of CA is to deal with relationships between parts of a system producing macro behavioral complexity with simple local rules (Wolfram, 2002). Nevertheless, as models of real-world spatial phenomena, basic CA are restricted by those background conventions the simplicity of which makes the richness of their behavior so unexpected (Couclelis, 1985) (neighborhood and rules uniformity of the cells, one discrete state per cell, closure of the system to external events and infinite lattice). Therefore, CA often need to be modified for simulation purposes (Worsch, 1999; Berjak and Hearne, 2002; De la Fuente et al., 2003) . These problems can be overcome by using object-oriented techniques and discrete event simulation (DES) (Zeigler, 1976; Fishwick, 1995; Hill, 1996). Discrete events allow to focus the simulation on the active cells of a propagation domain thus optimising the simulation. DES has been recently used in the last decade for ecological modelling purposes (Turner et al., 1982; Huston et al., 1988; Pukkala, 1988; Auger and Faivre, 1993; Baveco and Smeulders, 1994; Breckling and Müller, 1994; Hill et al., 1994; Maxwell and Costanza, 1994; Coquillard, 1995; Laughlin et al. 2003).

Among the disturbing propagation processes for ecosystems, fire and its influences on ecosystems remain a vast research field for numerous disciplines (biology, ecology, economy, physics, computer science etc.). Modelling such a phenomenon needs to well identify all the behavioural categories and all the interactions between these categories. The collaborative field is so large that, starting from the phenomenon, different abstraction levels are needed to well define the problem of interest and its relationships with other ones. Nowadays there is no generic method allowing to represent and guide scientists in such a field.

The object technology revolution has allowed the replacement of more than twenty years old stepwise procedural refinement paradigm by the more fashionable object composition paradigm. Currently, this evolution seems to be triggering another even more radical change towards model transformation. As a concrete trace of this, the Object Management Group (OMG) is rapidly moving its previous Object Management Architecture vision (OMA) to the newest Model Driven Architecture (MDA) (Bézivin, 2001). Although the object-oriented and component-based engineering are useful and proved to be adequate for many applications, they also proved to be limited when dealing with concrete interoperability in large software systems. Such techniques can still be efficiently used, but they must be considered as belonging to the past. The MDA is proposing a model engineering approach, not to suppress OO modelling or component-based modelling but to enhance what has proved to be limited.

The paper is organised as follows. First, a background is provided on the MDA and the simulation of fire spread. Secondly, a complete step by step modelling and simulation process is proposed thus providing a first MDA level architecture. Then, each step of the process is specified. Starting from a wildland fire phenomenon, we gradually focus on the study of fire spread through a mathematical model. After, a design framework for cell space simulation is exposed. Next, two generic architectures for simulating efficiently explicit and implicit models of neighbour-to-neighbour propagation are presented. Using these simulation architectures, simulations of the mathematical model of fire spread are then depicted. Finally, we sum up our contribution and define new perspectives.

2 BACKGROUND

Modelling and simulating a phenomenon involves using different abstraction levels. From the problem definition to the simulation system, the MDA proposed by the OMG helps to specify each modelling step. In a fire spread problem, different simulation approaches have been used until now. Each of these approaches has been achieved within a particular frame of interest (improvements of simulation models and ready-to-run software. Defining a higher abstraction level of design for this kind of problem is interesting for the specialists of the domain under study or for specialists of domains interacting with fire spread.

2.1 Meta-modelling concepts

A specific model usually simulates a special aspect of a complex system with a particular modelling and simulation technique (Partial Differential Equations, Cellular Automata, Neural Networks, Individual Based Model with or without Genetic Algorithms, etc.). A multi-model of a complex or huge system can be achieved with a composition of different specialized models of arbitrary kinds and with several abstraction levels (cf. Fig. 1). In addition, model components can be simulated separately, thus the principle of multimodelling is to support the hierarchical refinement of heterogeneous models through functional coupling. The computational refinement or derivability of a model is a tough problem that should not be hushed up. Thorough discussions dealing with model derivation can be found in (Zeigler, 1984; Fishwick, 1995).

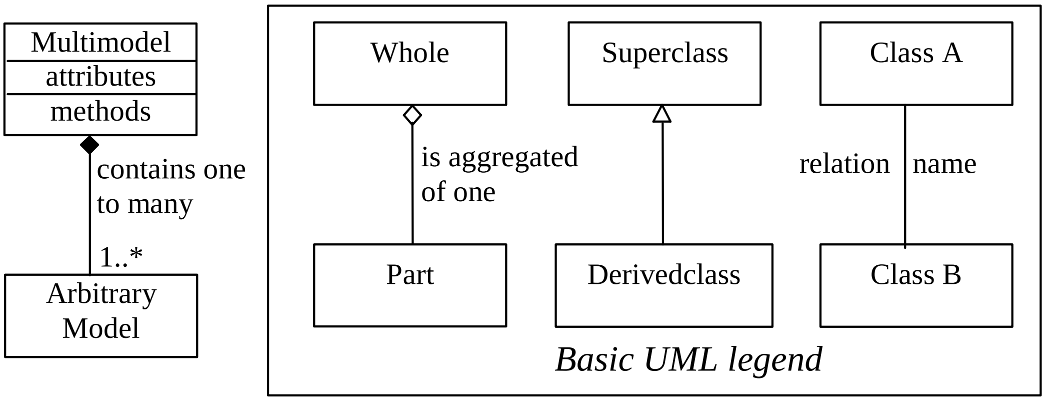


Figure 1: Simple UML multimodel and basic UML legend

The analogy between multimodeling and object-oriented programming has been demonstrated by (Cubert, 1997; Frick, 1997; Fishwick, 1998). Furthermore multi-models can be specified by Zeigler's DEVS formalism (Discrete Event system Specifications), which presents a mathematical ground helping to handle the well known aggregation problem (Zeigler 1976) encountered by biologists and simulationists. DEVS has been successfully applied to various ecological problems with multiple aggregation levels over the last decade. There is a close link between multi-formalism as discussed by (Zeigler 1979) and multimodelling: they are both founded on theoretic system concepts but they can be distinguished since they rely on different abstraction levels. In order to meet multimodelling requirements, Fishwick introduced a new methodology called Object-Oriented Physical Modelling (Fishwick 1996) to extend the classical object-oriented analysis and design methods in use in the simulation community (Hill 1996).

The history of Object-Oriented Analysis and Design methods has led to UML (Unified Modeling Language) in 1997. UML is the subject of a lot of research works and UML 2.0 has recently been adopted by the OMG as the new defacto standard. It provides a sound basis for MDA and represents the next evolutionary step in our ability to express and communicate system specifications. In our paper we retained UML for our multimodel designs since they will be more widely read using this graphical notation. It proposes to various domain experts an unified view of a system and helps solving communication problems (Baskent et al., 2001).

Since the adoption of the Meta Object Facility (MOF) recommendation by the OMG in 1997, the importance of model engineering in the information system and the software development process has rapidly increased (Breton and Bézivin, 2002). A key role is now played by the concept of meta-model in new software organisations like the OMG meta-model stack architecture. At the top of this architecture there is the MOF that provides a language for defining meta-models. This meta-meta-model aims at describing a particular domain of interest by defining a set of concepts and relations between these concepts.

A four-layer architecture has progressively taken shape. It is organised as follows:

- M3: the *meta-meta-model level* (contains only the MOF);
- M2: the *meta-model level* (contains any kind of meta-model described in UML);
- M1: the *model level* (contains any model description in UML with a corresponding meta-model in M2);
- M0: the *concrete level* (contains any real situation, unique in space and time, described by a given model).

MDA is providing both an environment to define platform-independent models (PIM) and generation services to map these models to a particular platform producing a platform-specific model (PSM).

2.2 Simulation of fire spread

Today, most of ready-to-run software for fire spread simulation (Albright and Meisner, 1990; Veach et al. 1994; Finney, 1995; Coleman and Sullivan, 1996; Lopes et al. 2002) and simulations of fire spreading on large-scale (Wu et al., 1996; Hargove et al., 2000; Miller and Yool, 2002) are based on Rothermel's model (Rothermel, 1972). A lot of effort has been placed in improving simulation of Rothermel's model. In the CA field studies pinpoint the need for developing new classes of CA for fire spreading applications (Karafylidis and Thanailakis, 1997; Berjak and Hearne, 2002). Based on discrete event formalisms and object oriented programming many applications have been proposed to improve CA capabilities for fire spread simulation (Vasconcelos et al. 1995; Ameghino et al. 2001; Barros and Ball, 1998; Muzy et al., 2002; Muzy et al. 2003). Unlike CA, these models can receive external updated information, and the fire perimeter can be updated at any moment due to the continuous time nature of the discrete event specifications and active cells can be dynamically created and removed to save memory for large cell spaces.

Nevertheless, all these last advances in the field of fire spread simulation do not deal with real-time constraints. As far as we know, if these approach improve simulation model reusability and simplify simulation model modifications, only one of them allows to simulate fire spread models requiring more computer resources under real time deadlines (Muzy et al. 2003). However, every approach has been achieved only at a simulation level without putting the study in a larger context.

3 THE MODELLING AND SIMULATION PROCESS

The development of a simulation system is performed through a software development life cycle. Numerous works have conceptualized the whole of a part of this development cycle (Zeigler, 1976; Fishwick, 1995; Zeigler et al. 2000). Based on these studies, two approaches proposed a complete software development life cycle (Hill, 1996; Traoré and Hill, 2001). We used these two approaches to develop a new generic software development life cycle. This cycle is presented as a meta-model of level M0 (Fig. 2). It is composed of three phases: *identification*, *engineering* and *integration*. The aim of the cycle is to provide a decision support for a large application domain. This meta-model can then be reused for many simulation model development.

The identification phase is the beginning of a simulation study. This phase consists in defining a problem concerning a phenomenon. Then formulating the problem a system is identified within an application domain. Objectives are formulated corresponding to the question the system has to answer about the phenomenon.

The engineering phase consists of iterative processes through which informal knowledge is transformed into an ultimate executable form: the programmed model. To achieve this goal, the system is specified at different levels of abstraction through a PIM. Each level of abstraction has to be verified regarding to the last level of specification.

First a domain analysis has to be conducted together with the specialist of the domain and the software designer. The objects, the relations and the behaviours brunt out are those of interest for the domain specialist. This work constitutes the *domain class model*. After this phase, the object-oriented analysis (OOA) consists in defining the real system with the *system class model* which instantiates a subset of the domain class model. Then, an object-oriented design proposes the structure and the behaviour of a *conceptual model*. Different abstraction levels can be used to build the conceptual model. The last level is coded in the *implemented model* corresponding to the PSM. Finally, the implemented model outputs are compared to the system objectives.

The integration phase exploits the simulation results to provide information to a decision support (humankind and/or a computer system). The decision-makers can interact with the simulation system (1) proposing new plans of experiments an (re)use the simulation results to solve the problem; (2) improving the system knowledge looking at the conceptual model; (3) sending commands to the physical system (if it exists) or to the conceptual system (if the system has to be (re)configured).

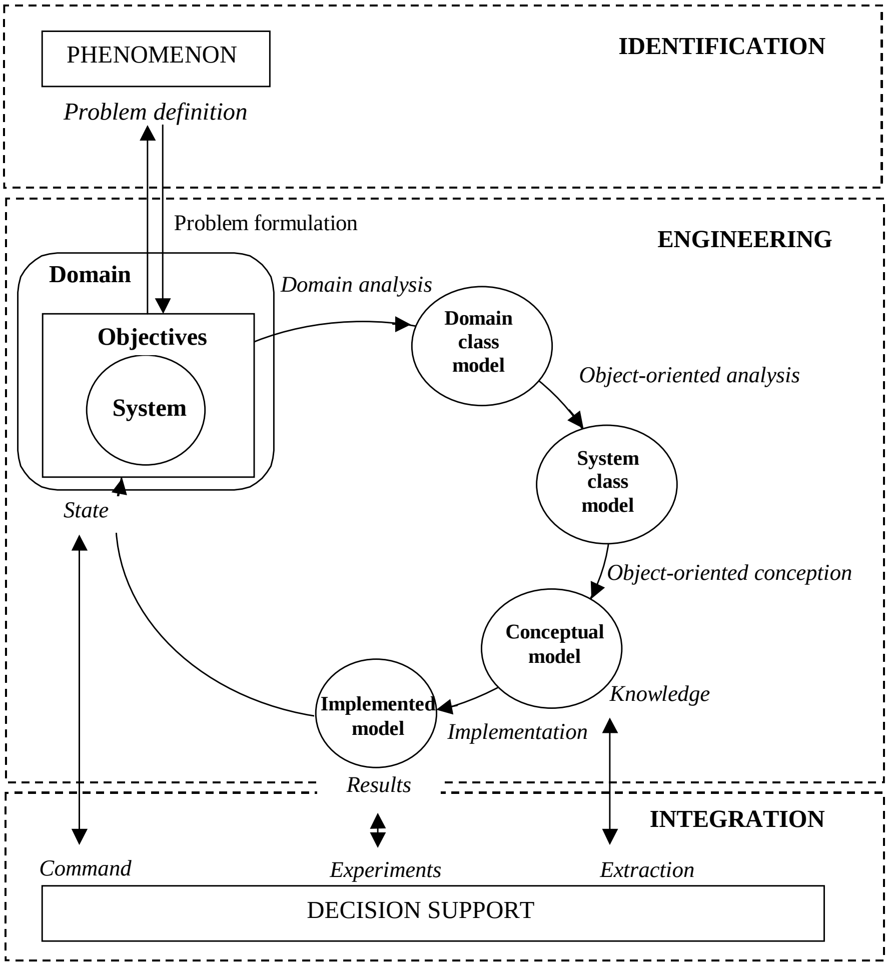


Figure 2: Modelling and simulation life cycle

All the abstraction phases of Figure 2 can be specified by a meta-model. Using UML, this meta-model describes in a precise manner the interactions between the different models and entities defined above (Fig. 3). An *experimental frame* can be added to establish the set of experiments for which the model is valid (Zeigler, 1976, Jørgensen 2001). Separating the model description from the experiment limits the changing of experiment description for new experiments (Lorek and Sonnenschein, 1998).

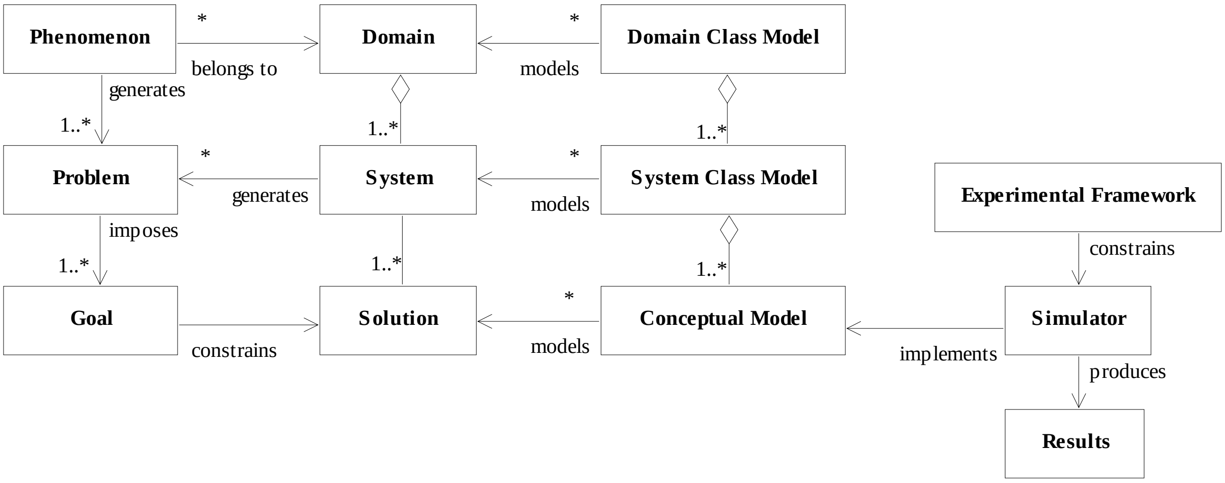


Figure 3: UML meta-model of the modelling and simulation process

4 APPLICATION TO FIRE SPREAD MODELLING

Considering wildland fire phenomena, defining a conceptual model consists in gradually specifying a particular domain of interest. Here, starting from a domain class model of wildland fire we focus on the fire behaviour dynamics. The system class model consists then in a fire propagation experiment. Finally, a solution is proposed through a conceptual mathematical model of fire spread. All these phases constitute a PIM of the model level of MDA and can be reused by other domain specialists.

4.1 The domain class model

Study of a wildland fire phenomenon can be broken down into four packages (Fig. 4):

1. The *Human Action* package encompasses development of wildland fire prevention strategies including land management and equipment policies as well as the elaboration of technical or regulatory solutions for reducing the number of wildland fire ignitions and fighting fires;

2. The *Environmental Conditions* package evaluates wildland fire risks, describes geographic and climatic conditions and provides vegetation models;
3. The *Ecological Impact* package includes the knowledge on fire consequences;
4. The *Fire Dynamics* package provides a description of the fire behaviour dynamics.

Figure 5 details the fire dynamics package. The study of the fire behaviour dynamics consists in describing chronologically the ignition, the propagation and finally the extinction. These phases can be mathematically described. Mathematical models provide the flame front position using the temperature distribution in complex fuel.

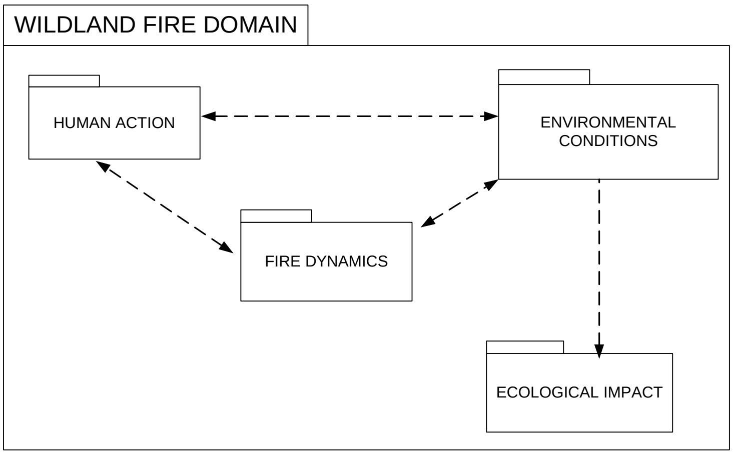


Figure 4: Package diagram of fire domain breakdown

To achieve this goal an experiment is used to provide input data or information and to validate the model. A fire spread experiment obviously modifies the real world to collect data of interest. Using an experiment, the propagation domain is described through the geography, the vegetation and the climate. Depending on the scale of the experiment and of the degree of precision the modeller has, different types of mathematical models can be used. To achieve real-time simulation, complexity of fire spread and data volume require having simple mathematical models capable of predicting the main behavioral features of fire.

Based on Weber's classification (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 at involving physical mechanisms, being merely a statistical description of test fires. The results can be very successful in predicting the outcome of similar fires 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 incorporates *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 empirically wind and slope. Finally, *physical* models (Albini, 1985) integrate wind and slope effects in a more robust manner 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.

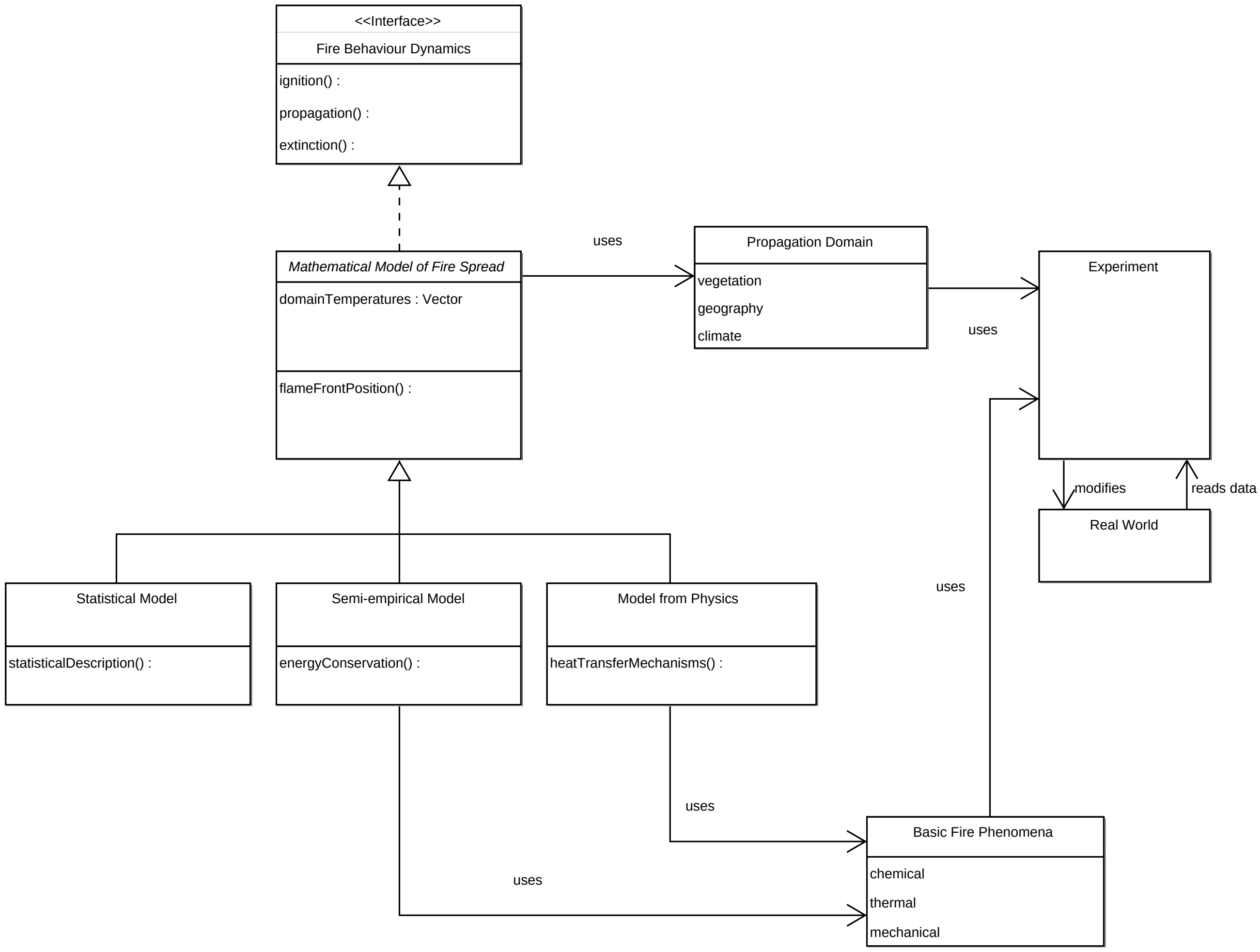


Figure 5: Excerpt of the fire domain class model: the Fire Dynamics package

4.2 The system class model

We used data of experimental fires conducted on *Pinus Pinaster* litter, in a closed room without any air motion, at the INRA (Institut National de la Recherche Agronomique) laboratory near Avignon, France (Balbi et al. 1999). Rigorous experiments were performed in order to observe fire spread for point-ignition fires under no slope and no wind conditions. The experimental apparatus was composed of a one square meter aluminium plate protected by sand. A porous fuel bed was used, made up of pure oven dried pine needles spread as evenly as possible on the total area of the combustion table in order to obtain a homogeneous structure. The experiment consisted in igniting a point using alcohol. The resulting spread of the flame across the needles was closely observed with a camera and thermocouples.

4.3 The Conceptual model

Once the laboratory experiment has been defined, a mathematical model has to be designed to simulate the fire spread. The large number of data and phenomena leads to a model simplification. Nevertheless, the description of the laboratory fire spread is a simplified case of fire spread. Actual fire spread often necessitates to combine different mathematical models.

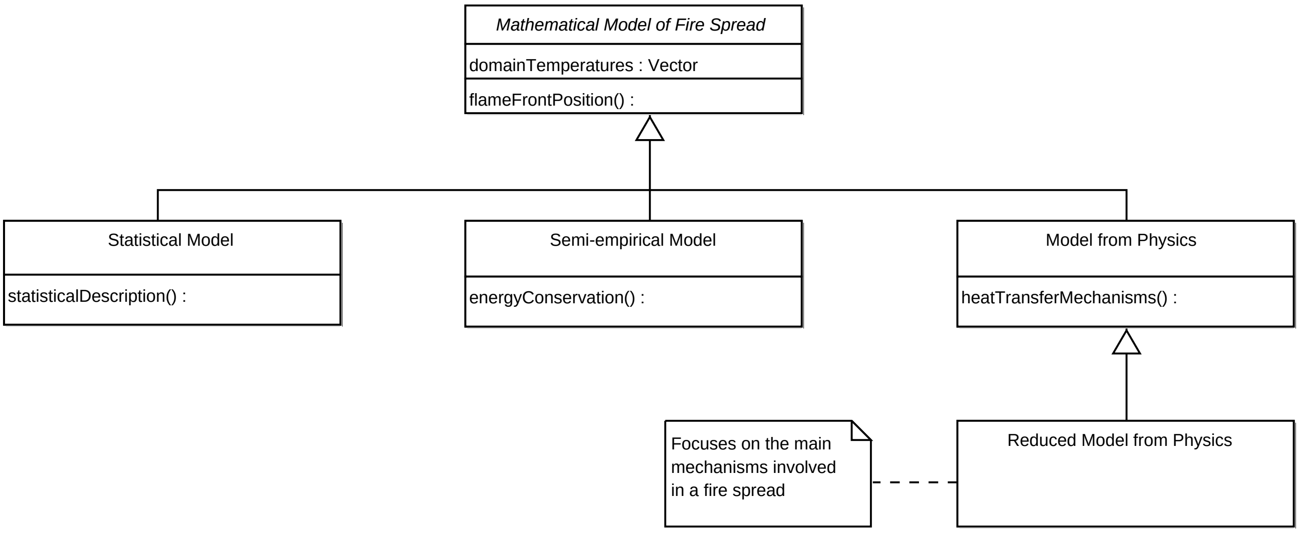

Mathematical Model of Fire Spread
domainTemperatures : Vector
flameFrontPosition() :
Statistical Model
statisticalDescription() :
Semi-empirical Model
energyConservation() :
Model from Physics
heatTransferMechanisms() :
Focuses on the main mechanisms involved in a fire spread
Reduced Model from Physics

Figure 6: Mathematical modelling of fire spread

Among the models from physics (Fig. 6), the multiphase approach which takes into consideration the finest mechanisms involved in fire spreading is the most complete modelling that has been developed so far (Grishin, 1997; Larini et al., 1997). Although the simulation of such models requires very long calculation time, the multiphase approach can be used to improve or develop simpler models dedicated to fire spread simulators (Giroux, 1997; Dupuy and Larini 2001). To achieve such a simple model, we have developed a strategy based on the reduction of multiphase models. This leads to a reduced physical model focusing on the main mechanisms involved in fire spreading (Balbi et al. 1999). This last model is non-stationary and two-dimensional.

This model uses elementary cells of earth and plant matter. Under no wind and no slope conditions, the temperature of each cell is represented by the following PDE:

$$\frac{\partial T}{\partial t} = -k(T - T_a) + K\Delta T - Q\frac{\partial \sigma_v}{\partial t} \quad \text{in the domain} \qquad (1a)$$

$$\sigma_v = \sigma_{v0} \quad \text{if } T < T_{ig} \qquad (1b)$$

$$\sigma_v = \sigma_{v0}.e^{-\alpha(t-t_{ig})} \quad \text{if } T \geq T_{ig} \qquad (1c)$$

$$T(x,y,t) = T_a \quad \text{at the boundary} \qquad (1d)$$

$$T(x,y,t) \geq T_{ig} \quad \text{for the burning cells} \qquad (1e)$$

$$T(x,y,0) = T_a \quad \text{for the non burning cells at } t=0 \qquad (1f)$$

Where, considering a cell, $\boldsymbol{T_a}$ (27 °C) is the ambient temperature, $\boldsymbol{T_{ig}}$ (300 °C) is the ignition temperature, $\boldsymbol{t_{ig}}$ (s) is the ignition time, $\boldsymbol{T}$ (°C) is the temperature, $\boldsymbol{K}$ (m².s^{-1}) is the thermal diffusivity, $\boldsymbol{Q}$ (m².°C / kg) is the reduced combustion enthalpy, $\boldsymbol{\Delta}$ is the Laplacian in two-dimensional Cartesian coordinates, $\boldsymbol{\alpha}$ (s^{-1}) combustion time constant, $\boldsymbol{\sigma_v}$ (kg.m^{-2}) is the vegetable surface mass, $\boldsymbol{\sigma_{v0}}$ (kg.m^{-2}) is the initial vegetable surface mass (before the cell combustion).

The model parameters are identified from experimental data of temperature versus time. The heat transfer of the model is sketched in Figure 7.

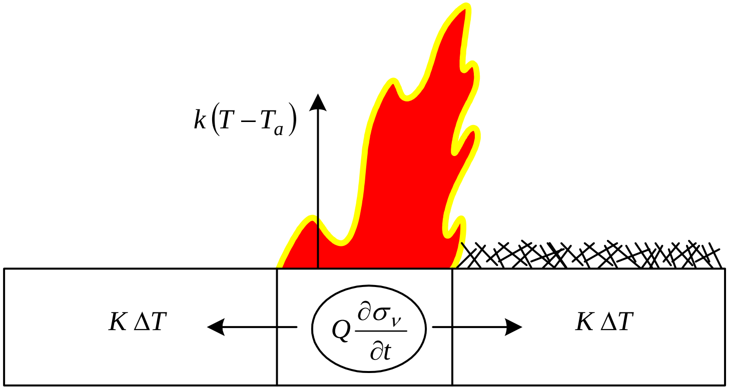


Figure 7: Heat transfer of the semi-physical model

The equation (1a) has to be numerically solved. Although analytical solutions exist for simple mathematical models of fire spread (Weber, 1989), generally, numerical methods are necessary to solve complex mathematical models of fire spread. Two numerical methods can be used to discretise the model: the Finite Element Method (FEM) and the Finite Difference Method (FDM). In a previous study, we applied both methods (Santoni, 1997). Although they provided the same results, the FEM appeared more complex to implement, and involved longer execution time. Thus, the FDM was chosen because of its simplicity and equally good performance.

The propagation domain is divided into a rectangular grid. With the FDM, the equation (1a) can be discretised with both explicit and implicit schemes. Simulation models obtained with an explicit scheme are easier to implement while those obtained with an implicit scheme are numerically more stables.

Using an explicit scheme, a discrete time model, well suited for computer simulation, is retained. The study domain is meshed uniformly with cells of 1-cm² and a time step of 0.01 s. The following algebraic equation is obtained:

$$T_{i,j}^{k+1} = a(T_{i-1,j}^{k} + T_{i+1,j}^{k}) + b(T_{i,j-1}^{k} + T_{i,j+1}^{k}) + cQ\left(\frac{\partial\sigma_v}{\partial t}\right)_{i,j}^{k} + dT_{i,j}^{k} \qquad (2)$$

Where T_{ij} is the grid node temperature. The coefficients *a*, *b*, *c* and *d* depend on the time step and mesh size considered. At each time step *k*, every temperature is calculated for the next time step *k+1*.

Using an implicit scheme, the equation (1a) is discretised onto the domain propagation and leads to a linear system. The system is solved using the Jacobi iterative method (Sibony and Mardon 1998). The study domain is meshed uniformly with cells of 1-cm² and a time step of 0.1 s.

$$T_{i,j}^{k+1} = a'(T_{i-1,j}^{k+1} + T_{i+1,j}^{k+1}) + b'(T_{i,j-1}^{k+1} + T_{i,j+1}^{k+1}) + c'Q\left(\frac{\partial\sigma_v}{\partial t}\right)_{i,j}^{k} + d'T_{i,j}^{k} \qquad (3)$$

Every temperature is calculated at the same time step. A convergence condition is used to pass at next time step.

5 DESIGN FRAMEWORK FOR CELL SPACE SIMULATION

Different choices of software architectures can be selected to simulate propagation processes. It is not often so easy and time consuming to develop efficient, reusable programs which are also easy to maintain. However, for a class of systems, different programming techniques and principles can be identified to propose a reusable software architecture. A framework allows to guide the modeller in the implementation phase. It is an abstract idea of an application structure, *i.e.* a reusable design for solutions to problems in some particular domain, than can be a system or a part of a system (Campos and Hill, 1998).

5.1 Conceptual model for cell space simulation

To achieve a simulation model of a cell space we distinguish different objects (Fig. 8). First, we separate the simulation model into a *Simulator package* and a *World modelling package*. The Simulator package controls and schedules events. The World modelling package represents the model of real world to simulate. Separating computer model and simulator allows to reuse a simulator for many models thus reducing development time (Zeigler, 2000).

Modelling with object-oriented approaches is based on the concepts of hierarchy and taxonomy which are the two organising principles in ecology (Sequeira et al. 1991). By composition, the simulation of a cell space can be constructed in a hierarchical way (Fig. 9). We use the system entity structure (SES) to represent the simulation model composition (Zeigler, 1984). SES allows to refine object-oriented hierarchical trees generally provided in ecological modelling literature (Chen, 1997; Baveco and Lingeman, 1992) differentiating single and multi-decomposition. In a SES a single bar (|) represents single decomposition and a triple bar (|||) represents multi-decomposition.

The Simulator class is composed of object classes such as the synchronization kernel (Synchro. Kernel) that uses the time (Time) to sort and schedule environmental events (Events) that are heaped in an event list (Event List). Using the simulation kernel, the Simulator manages the simulation of the World modelling package. The former is also composed of the Cell Space class and the Experimental Frame one. Here, the Experimental Frame can be seen as an object selecting data from real world and validating simulation results. In the World Modelling Package, behavior and structure are separated. In ecological propagation, structure concerns parcels of landscape. More precisely structure concerns the value of the model's state variables in each parcel (biomass, presence of an organism, food resources, etc.) (Laval, 1996; Congleton et al., 1997). Each state variable of a parcel (which generally corresponds to rasters of Geographical Information System (GIS)) is embedded in an array of object cells (Parcel Attributes) thus containing multiple state variables. Dynamics of the parcels changing one of their state variables is achieved through a Vector of active cells.

Modelling of ecological systems needs different abstraction levels communicating together. More generally, holistic and reductionistic views can be used to complementarily specify a model (Goguen and Varela, 1979; Jørgensen, 2001). Besides the cell's rules used to represent the system dynamics at a low level (Local Rule), rules can be used at a higher level (Global Rule) in the Cell space to represent the evolution of the global state of the system.

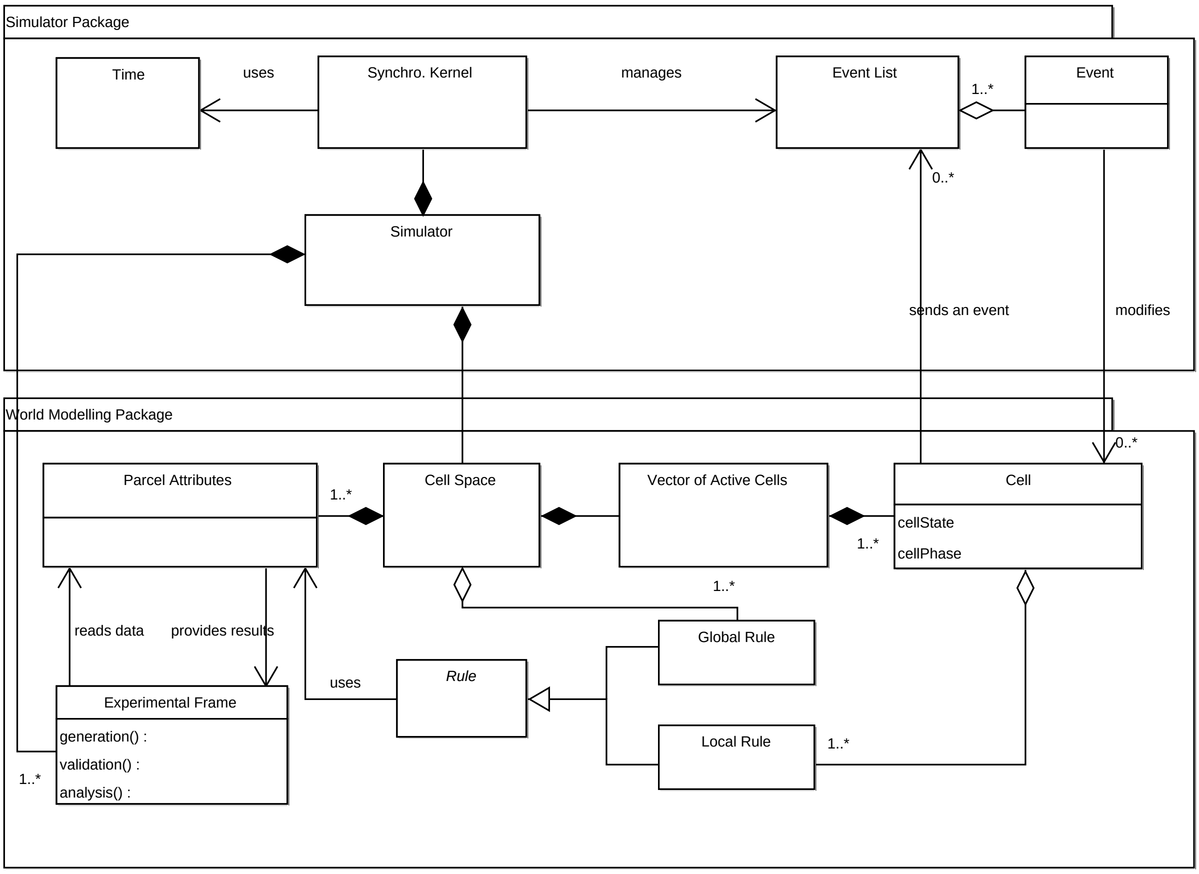


Figure 8: Proposition of a design framework for cell space simulation

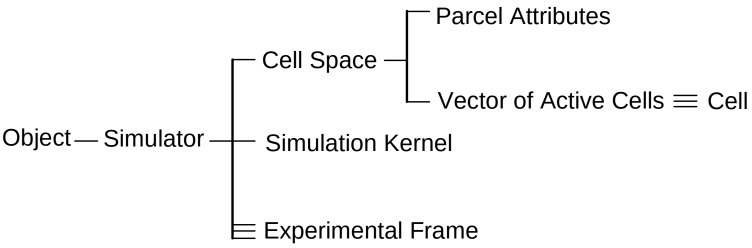


Figure 9: The system entity structure for the cell space simulation

5.2 Coupling of world model and data landscape

Lets now focus on the coupling of the World model and data landscape. The World object can be considered as the object container possessing all entities in the simulation and being coupled with the data landscape. In large propagation phenomena, the latter is generally provided by a GIS. This case can be handled easily by means of the Experimental frame (Vasconcelos et al., 1995). The relevant issues we retain are: the number of active and passive cells, the spatial environment dimension and its scale of representation, the velocity performance and the memory capacity required.

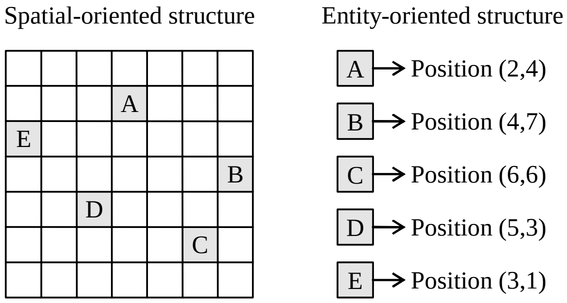


Figure 10: Choices of world data structure for DES

In accordance with these issues, two basic approaches can be retained to encapsulate and to manipulate the world data structure (Campos and Hill, 1998) (Fig. 10). The first is spatial-oriented and the second is entity-oriented. In a spatial-oriented approach we can see the world like a grid providing a matrix of positions where a position can be assigned to an entity. This approach is suitable for simulations with a large number of entities and where the computer time performance is more important than memory constraints. The second approach is for simulations with few simulation entities and when we do not want to use the large memory space required by a spatial localisation table (matrix of positions). Thus, localisation information is saved inside the entity instead of having a matrix of position where a position points to the entity. As a consequence, little memory space will be lost in the simulation implementation. However, the computer time performance will decrease since to get an information about environmental position we may have to consult all entities in the worst case.

For large-scale cell space simulation, a combination of the two first approaches can be achieved using a spatial-oriented representation for the Grid of state variable and an entity-oriented representation for the Vector of active cells (Fig. 8). To respect real time deadlines, dynamic allocation has to be suppressed for these classes. Indeed, for significant numbers of object instantiation/deletion dynamic allocation is inefficient and a specialised static allocation have to be designed (Stroustrup, 2000). A pre-dimensioning via large static arrays and vectors can be easily achieved thanks to current modern computer memory capabilities.

The Vector of active cells is updated using a DES. A start-index and an end-index are delimiting the current calculation domain on the vector. Thus initial active cells that return in a quiescent state during a simulation run can be dynamically ignored in the main loop. At each time step, by modifying the value of the indexes, new tested cells can be added to the calculation domain and cells that return in a quiescent state are removed from the former.

6 SIMULATION MODELS FOR NEIGHBOUR-TO-NEIGHBOUR PROPAGATIONS

The design framework previously introduced is used here to implement an optimized simulation of a neighbour-to-neighbour propagation. In section 4 we saw that the numerical resolution of the PDE *(1a)* leads to two algebraic equations *(2)* and *(3)*. The difference between equation *(2)* and *(3)* are (1) the condition used to increment the discrete time base added for equation *(3)*; (2) a modification of the value of the equation parameters; (3) a different value of time step. Using the previous design framework, most of the classes previously defined can be reused and few modifications have to be achieved on some classes. Modification (1) leads to test the event occurrences in the Synchro. Kernel (Fig. 7). Modifications (2) and (3) induce little modifications of the Time and Cell classes. This phase represents the last abstraction level of a PIM.

6.1 The pure event scheduling approach

For neighbour-to-neighbour propagation, a generic rule can be implemented in the Cell space class to control the evolution of the calculation domain (corresponding to the set of active cells). To control the domain evolution, propagation states have been added to the cells (*testing* for the cells at the edge of the propagation, *non testing* for the cells whose state is not tested at each state transition and *quiescent* for the inactive cells in quiescent state).

A neighbour-to-neighbour propagation example is sketched in Figure 11 for a cardinal neighborhood. In our algorithm, only the bordering cells test their neighborhood, this allows to reduce the number of testing cells. The test depends on the cells' state. If the tested cell fulfils the requested condition, the cell becomes an active cell and new tested neighboring cells are added to the set of active cells.

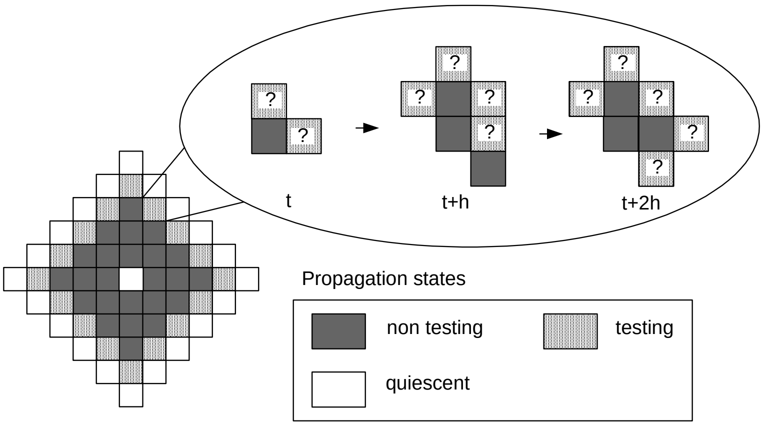


Figure 11: Evolution of the calculation domain

In the case of fire spread, two grids of state variables are used. One to store the propagation states, the other to store the temperatures of cells. As the phase of a cell (*unburned*, *burning* and *burned*) only depends on the temperature of the cell, the phases are directly embedded in the cell objects.

6.2 The activity scanning modification

Activity scanning (also known as the two-phase approach) was first introduced by (Buxton and Laski, 1962). An activity is what transforms the state of an object over a period of time. An activity is initiated by the occurrence of an event and is ended by the occurrence of another event. Under the activity scanning conceptual framework, the modeller describes an activity in two parts: condition and actions.

In order to try to remedy the execution inefficiency of activity scanning (Tocher, 1963) suggested the three-phase approach which combines activity scanning and event scheduling. A classification is made to separate unconditional and conditional activities. The three-phase approach executes sequentially (Pidd, 1984) (1) the time-scanning (A Phase); (2) the **b**ound-to-occur or **b**ook-keeping activities that represent the unconditional state changes (unconditional events) which can be scheduled in advance (B phase); (3) the conditional or co-operative activities that represent the state changes which are conditional upon the co-operation of different objects or the satisfaction of specific (compound) conditions (C phase).

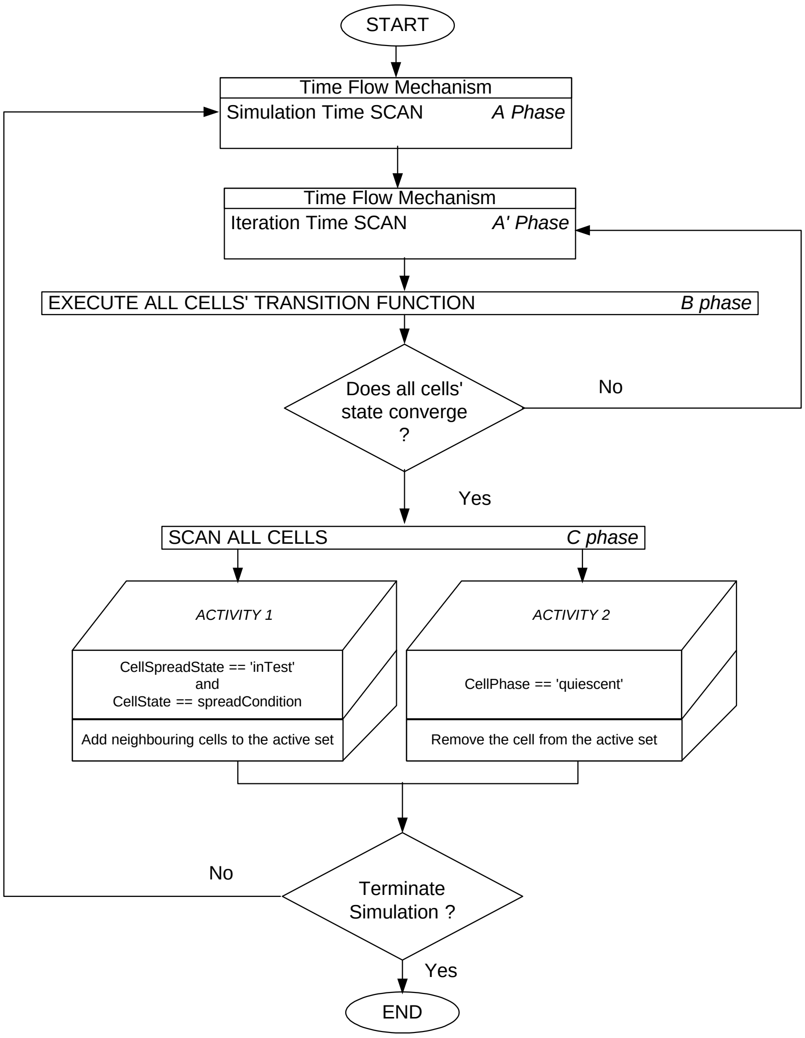

START
Time Flow Mechanism
Simulation Time SCAN
A Phase
Time Flow Mechanism
Iteration Time SCAN
A' Phase
EXECUTE ALL CELLS' TRANSITION FUNCTION
B phase
Does all cells' state converge ?
No
Yes
SCAN ALL CELLS
C phase
ACTIVITY 1
CellSpreadState == 'inTest' and CellState == spreadCondition
Add neighbouring cells to the active set
ACTIVITY 2
CellPhase == 'quiescent'
Remove the cell from the active set
Terminate Simulation ?
No
Yes
END

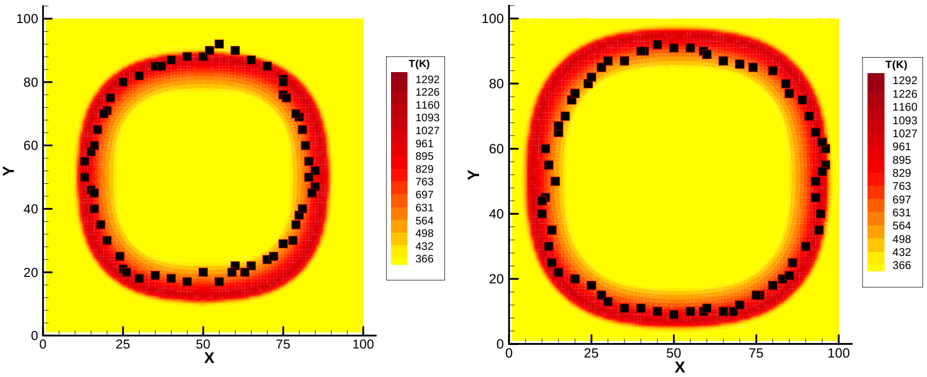
Figure 12: Three-phase approach modification for implicit model simulation

Figure 12 depicts an adaptation of the three-phase approach for the simulation of implicit models. A principal time scanning loop is in charge of the simulation time (A phase). However, the A phase has been doubled by a smaller loop in charge of the iteration time base (A’ phase). As long as the states of all cells do not converge, the transition function of a cell is executed. Once the state of a cell converges it is added or removed from the calculation domain (C phase).

7 RESULTS OF FIRE SPREAD SIMULATION

For the PSM we chose to program our implemented model in C++ for efficiency reasons. We used three techniques for the simulation phase (Balci and Sargent, 1989; Youngblood and Pace, 1995; Hill et al., 1996) (1) comparison validation: comparison of simulation results with an experiment; (2) confrontation validation: asking physicists if the results and behaviour of the simulation model were consistent; (3) graphic validation: using visualisation and animation to make use of the human ability to apprehend spatial relationships.

Figure 13 shows a comparison of the simulated and experimental fire fronts obtained for a point-ignition. Black squares represent the experimental fire fronts. Figure 14 represents the evolution of the active cells around the fire fronts. Finally, Figures 15 and 16 depict the execution time gain (on a *500 MHz Pentium III* processor) obtained focusing on the active cells of the propagation domain. We can remark that this gain is more important for a three-phase simulation. Indeed, for the implicit method if the time step is greater than the explicit one (of ten times), the number of iteration per time step is lower than ten. Moreover, the number of active cells is approximately equivalent at each time step. Hence, if at the end of the simulation the three-phase method will execute 10 million of cell transition functions, the pure discrete event one corresponding to the explicit model necessitates 56 millions of cell transition functions.



(a) (b)

Figure 13: Simulated and experimental fire fronts at: (a) *t=122s* and (b) *t=144s*

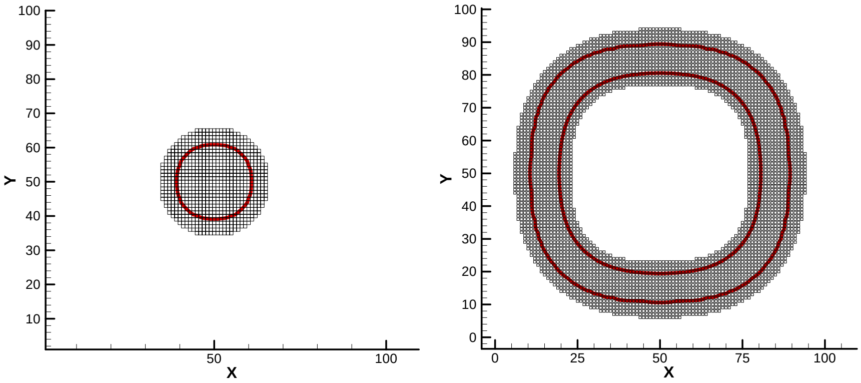


Figure 14: Evolution of the active cells

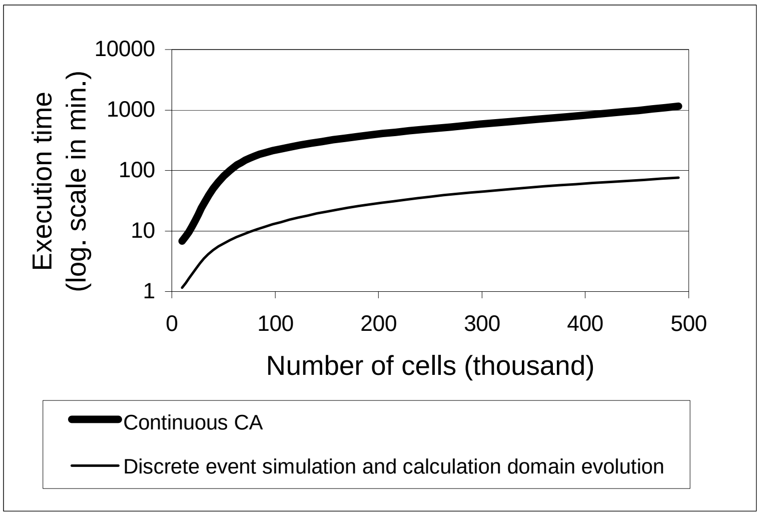


Figure 15: Execution time gain using a discrete event simulation focusing on active cells of the propagation domain

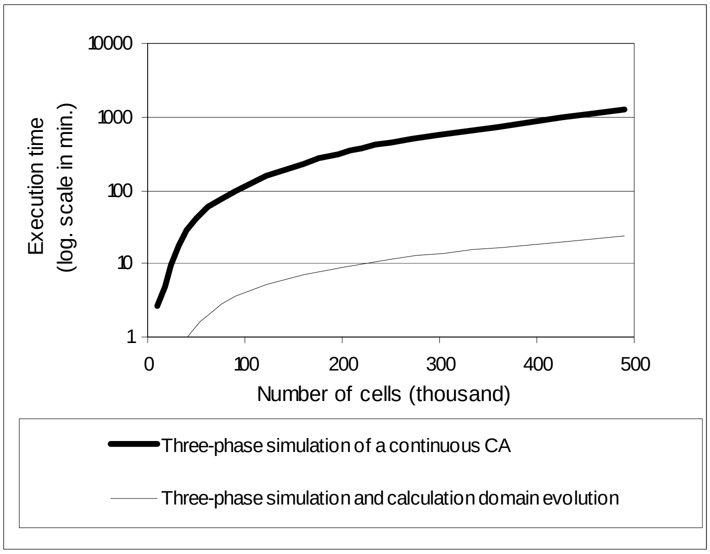


Figure 16: Execution time gain using a three-phase simulation focusing on active cells of the propagation domain

8 CONCLUSION

We presented a method for modelling wildland fires and simulating fire spread. This method uses both latest advances in the field of software engineering and new developed simulation and modelling tools. Except for the PSM all the PIMs can be reused in another propagation problem or by another research domain on wildland fires. The final PSM developed using the MDA architecture proved to be efficient in terms of execution time and has been qualitatively validated against experimental data.

Model composition seems to be well suited to ecological modelling. Model representation allows aggregating different ecological features providing a problem overview. Description of model interactions facilitates the communication between specialists of different research domains.

Structure or concepts of this new framework can be used by other object-oriented approaches designed for cell space simulations which generally lack genericity. Modules of the framework can be adapted for new problems integrating specific data (experiments and parcel attributes), behaviors (rules of the cells or cell space) or time schemes (simulator part of the framework). Optimization concepts concentrating the simulation on active cells can also be reused to significantly reduce simulation time of large cell spaces.

The method needs now to be tested to other ecological systems and thus refined for the encountered problems. In the wildland fire field other subdomains than fire spread can be studied in more detail and connections to the fire spread models can be developed. Other PSM can also be tested using our first architecture. Once sufficient work will be done in a specific application domain, UML profiles (Fuentes and Vallecillo, 2001) will be able to be designed for these application. These profiles focus on the common properties of systems of the same application domain. Domain experts can then use these macro-architectures as the software designers used the well known design pattern micro-architectures (Gamma et al., 1995, Holst et al., 1997).

ACKNOWLEDGMENTS

The authors gratefully acknowledge the Assistant Professor Paul Antoine Santoni of the University of Corsica for his help in applying our methodology to the modelling and simulation of fire spread.

REFERENCES

Albini, F.A., 1985. A model for fire spread in wildland fuels by radiation. Comb. Sci. and Tech., 42: 229-258.

Albright, D., Meisner, B.N., 1999. Classification of fire simulation systems. Fire Management Notes, 59(2): 5-12.

Alfredsen, K., Sæther, B., 2000. An object-oriented application framework for building water resource information and planning tools applied to the design of a flood analysis system, Environmental Modelling and Software, 15(3): 215-224.

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

Andrews, P.L., Bevins, C.D., 1998. Update and Expansion of the BEHAVE Fire Behavior Prediction System. Technical report.

Auger, P., Faivre, B., 1993. Propagation of extinction waves in spatial models of interspecific competition and selective predation. Acta Oecol. 14(6): 781–805.

Balbi, J. H., Santoni, P. A., Dupuy, J.L., 1999. Dynamic modelling of fire spread across a fuel bed. Int. J. of Wildland Fire, 9: 275-284.

Balci, O., Sargent, R.E., 1989. Guidelines for selecting and using simulation model verification techniques. Proceedings of the Winter Simulation Conference, pp. 559–568.

Barros, F.; Ball, G.L., 1998. Fire modeling using dynamic structure cellular automata. III International Confer. On Forest Fire Research, 14th Conference on Fire and Forest Meteorology, VOL I, PP. Luso, 879-888. 16/20 November 1998.

Baskent, E.Z., Wightman, R.A., Glenwood, A.J., Zhai, Y., 2001. Object-oriented abstraction of contemporary forest management design. Ecol. Modelling, 143: 147-164.

Baveco, J.M., Lingeman, R., 1992. An object-oriented tool for individual-oriented simulation: host parasitoid system application. Ecol. Modelling, 61: 267-286.

Baveco, J.M., Smeulders, A.M.W., 1994. Objects for simulation: smaltalk and ecology. Simulation, 62 (1): 42–57.

Berjak, S.G., Hearne, J.W., 2002. An improved cellular automaton model for simulating fire in a spatially heterogeneous Savanna system. Ecol. Modelling, 148: 133-151

Bézivin, J., 2001. From object composition to model transformation with the MDA. Proceedings of the TOOLS' USA August 2001, Santa Barbara, USA.

Breckling, B., Muller, F., 1994. Current trends in ecological modelling and the 8th ISEM conference on the state-of-theart. Ecol. Model., 75: 667–675.

Breton, E., Bézivin, J., 2002. Weaving definition and execution aspects of process meta-models Proceedings of the 35th Annual Hawaii International Conference on System Sciences (HICSS'02), Big Island, Hawaii.

Buxton, J.N., Laski, J.G., 1962. Control and simulation language. The Computer Journal, 5: 194-199.

Campos, A., Hill, D.R.C., 1998. An Agent-based Framework for Visual-Interactive Ecosystem Simulations. SCS Transactions, 15: 139-152.

Chen, J., Reynolds, J.F., 1997. GePSi: A generic plant simulator based on object-oriented principles. Ecol. Modelling, 94: 53-66.

Coleman, J. R.; Sullivan, A.L., 1996. A Real-time Computer Application for the Prediction of Fire Spread across the Australian Landscape. Simulation, 67(4): 230-240.

Congleton, R.C., Pearce, B.R., Beal, B.F., 1997. A C++ implementation of an individual/landscape model. Ecol. Modelling, 103: 1-17.

Coquillard, P., 1995. Simulation of the cyclical process of heathlands. Induction of mosaics structures, evolution to irreversible states. Ecol. Model., 80: 97–111.

Couclelis, H., 1985. Cellular worlds: A framework for modelling micro-macro dynamics. Environment and Planning A, 17: 585-596.

Cubert, R.M., Goktekin, T., Fishwick, P.A., 1997. MOOSE : Architecture for an Object-Oriented Multimodeling Simulation System, Enabling Technology for Simulation Science, SPIE, AeroSense 1997, Conference, Orlando, FL.

De la Fuente, L.F., Causon, D.M., Ingram, D.M., Mingham, C.G., Raper, D.W., 2003. Towards simulating urban canyon circulations with a 2D lattice Boltzmann model, Environmental Modelling & Software, 18(1) 71-79.

Dumont, B., Hill, D., 2001. Multi-agent simulation of group foraging in sheep: effects of spatial memory, conspecific attraction and plot size.", Ecological Modelling, 141: 201-215.

Dunkerley, D.L., 1999. Banded chenopod shrublands of arid Australia: modelling responses to interannual rainfall variability with cellular automata. Ecol. Model., 121: 127-138.

Dupuy, J.L., Larini, M., 2001. Fire spread through a porous forest fuel bed: a radiative and convective model including fire-induced flow effects. Int. J. of Wildland Fire, 9: 155-172.

Finney, M.A., 1995. Fire area simulator. User guide and technical documentation, 75 pp.

Fishwick P.A.; Sanderson J.; and Wolff W., 1998. A Multimodeling Basis for Across-Trophic-Level Ecosystem Modeling : The Florida Everglades Example, SCS Transactions, 15(2): 76-89.

Fishwick, P.A., 1995. Simulation Model Design and Execution: Building Digital Worlds. Prentice Hall.

Fishwick, P.A., 1996. Extending Object-Oriented Design for Physical Modeling. ACM Transactions on Modeling and Computer Simulation.

Frick, A., 1997. Multi-Model Analysis and Design (pre-print: http://www.aifb.uni-karlsruhe.de/Staff/afr.html).

Fuentes, F., Vallecillo, A., 2001. Using UML Profiles: A Case Study. In Proc. of the ECOOP'2001 Workshop on Automating Object-Oriented Software Development Methods (AOOM'01). Hungary, June 2001.

Gamma, E., Helm, R., Johnson, R., Vlissides, J., 1995. Design Patterns. Addison-Wesley, New York.

Giroux, F., 1997 Contribution to fire spreading modelling: multiphase approach of forest fires, development of a propellant fire in a semi-confined medium" [In French], Thesis (Doctorate of University), University of Provence, IUSTI, Aix-Marseille-I, France.

Goguen, J.A., Varela, F.J., 1979. Systems and distinctions; duality and complementarity. Int. J. General Systems, 5: 31-43.

Grishin, A.M., 1997. Mathematical modeling of forest fires and new methods of fighting them. Publishing house of the Tomsk state university. Albini Ed. Chap. 1, pp. 81-91.

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

Hill, D., 1996. Object-oriented analysis and simulation. Addison-Wisley Longman, UK, 291 pp.

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. Ecol. Model., 109: 251-265.

Hill, D., Mazel, C., Coquillard, P., 1996. Integrating V&V in the object-oriented life cycle of ecological modelling simulation projects. Proceedings of the European Simulation Symposium 96, October 24–26, Genova, Italy.

Hill, D., Pastre, J., Coquillard, P., Gueugnot, J., 1994. Design of an ecosystem modeling environment application to forest growth simulation. Proceedings of the First Joint Conference of International Simulation Societies, August 22–25, Zurich, pp. 538–544.

Holst, N., Axelsen, J.A., Olesen, J.E., Ruggle, P., 1997. Object-oriented implementation of the metabolic pool model. Ecol. Modelling, 104: 175-187.

Huston, M., DeAngelis, D., Post, W., 1988. New computer models unify ecological theory. Computer simulation shows that many ecological patterns can be explained by interactions among individual organisms. BioScience, 38 (10): 682–691.

Jen, E., 1990. A periodicity in one-dimensional cellular automata. Physica D 45, 3-18.

Jørgensen, S.E., Bendoricchio, G., 2001. Fundamentals of ecological modelling. Elsevier, Amsterdam, 530 pp.

Karafyllidis, I., Thanailakis, A., 1997. A model for predicting fire spreading using cellular automata. Ecol. Modelling, 99: 87-97.

Larini, M., Giroux, F., Porterie, B., Loraud, J.C., 1997. A multiphase formulation for fire propagation in heterogeneous combustible media". Int. J. of Heat Mass Trans., 41: 881-897.

Laughlin, G.P., Zuo, H., Walcott, J., Bugg, A.L., 2003. The rainfall reliability wizard—a new tool to rapidly analyse spatial rainfall reliability with examples, Environmental Modelling & Software, 18(1): 49-57.

Laval, P., 1996. The representation of space in an object-oriented computational pelagic ecosystem. Ecol. Modelling, 88: 113-124.

Loibl, W., Toetzer, T., 2003.Modeling growth and densification processes in suburban regions—simulation of landscape transition with spatial agents, Environmental Modelling & Software, 18(6): 553-563.

El Yacoubi, S., El Jai, A., Jacewicz, P., Pausas, J. G., 2003. LUCAS: an original tool for landscape modelling, Environmental Modelling & Software, 18(5): 429-437.

Lopes, A.M.G., Cruz, M.G., Viegas, D.X., 2002. FireStation — an integrated software system for the numerical simulation of fire spread on complex topography, Environmental Modelling & Software, 17(3): 269-285.

Lorek, H., Sonnenschein, M., 1998. Oject-oriented support for modelling and simulation of individual-oriented ecological models. Ecol. Modelling, 108: 77-96.

Matsinos, Y.G., Troumbis, A.Y., 2002. Modelling competition, dispersal and effects of disturbance in the dynamics of a grassland community using a cellular automaton model. Ecol. Modelling, 149: 71-83.

Maxwell, T., Costanza, R., 1994. Distributed object-oriented spatial ecosystem modeling. Object-Oriented Simulation Conference, pp. 85–90.

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

Miller, J.D., Yool, S.R., 2002. Modeling fire in semi-desert grassland/oak wooldland: the spatial implications. Ecol. Modelling, 153: 229-245.

Muzy, A., Innocenti, E., Hill, D., Santucci, J.F., 2003. Optimization of cell spaces simulation for the modelling of fire spreading. In Proceedings of the 36th Annual Simulation Symposium, IEEE/SCS/ACM, March 30-April 2003, Orlando, pp. 289-296.

Muzy, A., Wainer, G., Innocenti, E., Aiello, A., Santucci, J. F., 2002. Comparing simulation methods for fire spreading across a fuel bed. In Proceedings of AI, Simulation and Planning in High Autonomy Systems, AIS'2002, ed. F.J. Barros and N. Giambiasi, Lisbon, Portugal, pp. 219-224.

Neil, P.G., Sherlock, R.A., Bright, K.P., 1999. Integration of legacy sub-system components into an object-oriented simulation model of a complete pastoral dairy farm, Environmental Modelling and Software, 14(6): 495-502.

Pidd, M., 1984. Computer simulation in management science. John Wiley & Sons, Chichester.

Pukkala, T., 1988. Effect of spatial distribution of trees on the volume increment of a young Scots pine stand. Silva Fenn. 22(1): 1–17.

Richards, G.D., 1990. An elliptical growth model of forest fire fronts and its numerical solution. Int. J. Nurner. Meth. Eng., 30: 1163-1179.

Rothermel, R.C., 1972. A mathematical model for predicting fire spread in wildland fuels. USDA, For. Serv. Res. Paper INT-115.

Santoni, P.A., 1997. Propagation de feux de forêt, modélisation dynamique et résolution numérique, validation sur des feux de litière. PhD thesis of the University of Corsica.

Sequeira, R.A., Sharpe, P.J.H., Stone, N.D., El-Zik, K.M., Makel, M.E., 1991. Object-oriented simulation: plant growth and discrete organ to organ interactions. Ecol. Modelling, 58: 55-89.

Sibony, M. ; Mardon, J.C., 1998. Approximations et équations différentielles. Hermann.

Silvert, W., 1993. Object-oriented ecosystem modelling. Ecol. Modelling, 68: 91-118.

Sirakoulis, G.C., Karafyllidis, I., Thanailakis, A., 2000. A cellular automaton model for the effects of population movement and vaccination on epidemic propagation. Ecol. Modelling, 133: 209-222.

Soares-Filho, B.S., Cerqueira, G.C., Pennachin, C.L., 2002. DINAMICA a stochastic cellular automata model designed to simulate the landscape dynamics in an Amazonian colonization frontier. Ecol. Modelling, 154: 217-235.

Spanou, M., Daoyi, C., 2000. An object-oriented tool for the control of point-source pollution in river systems, Environmental Modelling and Software, 15(1): 35-54.

Spencer, M., 1997. The effects of habitat size and energy on food web structure: An individual-based cellular automata model. Ecol. Modelling, 94: 299-316.

Stroustrup, B., 2000. The C++ Programming Language (Special Edition). Addison Wesley, Reading Mass, 1029 pp.

Tobler, W.R., 1979. Cellular geography, S. Gale, G. Olssein (Editors). Philosophy in geography, 279-386.

Tocher, K.D., 1963. The art of simulation. English Universities Press, London.

Traore, M.K., Hill, D., 2001. Réflexion sur la Théorie de la Simulation, Le statut épistémologique de la simulation. Rencontre interdisciplinaires sur les systèmes complexes naturels et artificiels, 26 janvier au 3 février, 10ème journée de Rochebrune, ENST 2001 S 001, pp. 279-297.

Turner, M.E., Claiborne, J., Anderson, W.W., 1982. Homozygosity and patch structure in plant population as a result of nearest-neighbor pollination. Proc. Natl. Acad. Sci., 79: 203–207.

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

Veach, M.S., Coddington, P.D., Fox, G.C., 1994. BURN: A Simulation of Firest Fire Propagation. Project Report for the Northeast Parallel Architectures Center. Research Experiences for Undergraduates Programme.

Von Neuman, J., 1966. Theory of Self-reproducing Automata, University of Illinois Press. Edited and completed by A. Burks.

Weber, R.O., 1989. Analytical models for fire spread due to radiation. Combustion and flame, 78: 398-408.

Weber, R.O., 1990. Modelling fire spread through fuel beds. Prog. in En. and Comb. Sci., 17: 67-82.

Wolfram, S., 1994. Cellular automata and complexity: Collected papers. Addison-Wesly.

Wolfram, S., 2002. A new kind of science. Wolfram Media, Inc, 1192 pp.

Worsch, T., 1999. Simulation of cellular automata. Future generation computer systems, 16: 157-170.

Wu, Y., Sklar, F.H., Gopu, K., Rutchey, K., 1996. Fire simulations in the Everglades Landscape using parallel programming. Ecol. Modelling, 93: 113-124.

Yang, L., Lin, B., Kashefipour, S.M., Falconer, R.A., 2002. Integration of a 1-D river model with object-oriented methodology, Environmental Modelling & Software, 17(8): 693-701.

Youngblood, S.M., Pace, D.K., 1995. An overview of model and simulation verification, validation and accreditation. John Hopkins APL Tech. Dig. 16 (2), 197–205.

Zeigler, B.P., 1976. The aggregation problem, System Analysis and Simulation in Ecology, B. Patten Editor, Academic Press.

Zeigler, B.P., 1979. Multi-level Multiformalism Modelling : An Ecosystem Example, Theoretical Systems Ecology, E. Halfton editor, Academic Press.

Zeigler, B.P., 1984. Multifacetted Modelling and Discrete Event Simulation, Academic Press.

Zeigler, B.P., Praehofer, H., Kim, T.G., 2000. Theory of modelling and simulation, 2nd Edition, Academic Press.