Available online at www.sciencedirect.com
SCIENCE DIRECT®
Environmental Modelling & Software 20 (2005) 827–842
Environmental Modelling & Software
www.elsevier.com/locate/envsoft

Modelling and simulation of ecological propagation processes: application to fire spread

Alexandre Muzy[a,*], Eric Innocenti[a], Antoine Aïello[a], Jean-François Santucci[a], Paul-Antoine Santoni[a], David R.C. Hill[b]

[a] *SPE – UMR, CNRS 6134, University of Corsica, B.P. 52, Campus Grossetti, 20250 Corti, France*
[b] *ISIMA/LIMOS, UMR, CNRS 6158, Blaise Pascal University, Campus des Cézeaux, B.P. 10125, 63177 Aubière Cedex, France*

Received 19 September 2003; received in revised form 30 March 2004; accepted 28 April 2004

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 simulate efficiently because of the number of entities. Studies of this kind of phenomena lack genericity and reusability because they are often presented from 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 the 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 help in developing generic and reusable models. From modelling to simulation, the Unified Modelling 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 discrete time models of propagation. The approach is applied to the modelling and simulation of fire spread. Starting from wasteland 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 wasteland fire problem.

Keywords: Fire spread modelling; Fire spread simulation; Propagation process; Object oriented technique; Discrete event simulation; Model Driven Architecture

1. Introduction

Ecosystems show a high degree of heterogeneity in space and time (Jørgensen and Bendoricchio, 2001). Studying the dynamics of such systems requires the development of models, able to consider both time and space parameters. 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 large 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.

The 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 Sæther, 2000; Spanou

* Corresponding author. Fax: +33-4-9545-0162.
E-mail address: a.muzy@univ-corse.fr (A. Muzy).
URL: http://spe.univ-corse.fr/simffweb/debut.htm.

1364-8152/$ - see front matter © 2004 Elsevier Ltd. All rights reserved.
doi:10.1016/j.envsoft.2004.04.011

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.

The simulation of propagation processes is performed through cell space models. In ecosystem propagation, the 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 'neighbour' cells according to some uniform location-independent rules.

Cell space dynamics 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 neighbouring 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-behavioural 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 behaviour so unexpected (Couclelis, 1985) (neighbourhood and rules uniformity of the cells, one discrete state per cell, closure of the system to external events and infinite lattice). According to experimental conditions, cell behaviours and neighbourhoods of a CA often need to be different. One boolean state per cell is also usually not sufficient when dealing with complex deterministic systems. Finally, external events modifying cell states of the CA during the simulation can be necessary. 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). Considering a propagation domain, traditional CA compute at each time step the transition functions of all the cells, even the cells that remain in an inactive state. However, in a fire spreading application, for example, computations only need to be concentrated around the fire front. Discrete events allow to focus the simulation on the active cells of a propagation domain thus optimising the simulation. DES has been used recently 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 carefully 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 efficiently define the problem of interest and its interactive relationships. Nowadays there is no generic method capable of representing and guiding scientists in such a field.

The object technology revolution has allowed the replacement of more than 20-year-old step-wise 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 latest 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 proposes a model engineering approach, not to suppress OO modelling or component-based modelling but to enhance what has proved to be limited.

Using the MDA a new generic methodology can be defined. Non-computer science experts, such as ecologists, benefit directly using such a methodology. First, at a high level of description, fully abstract models can be defined using the expert terminology. Then, cooperation with other experts from different disciplines is also eased. For example, to study forest spreading, one can imagine a *Forest Model*, composed of a *Forest Spreading Model* and a *Forest Decreasing Model*. The *Forest Decreasing Model* can then be composed of a *Fire Spread Model*, defined by physicists, and a *Deforestation Model*, defined by biologists. After that, *the Physical Fire Spread Model* can be decomposed into different *Fire Spread Simulation Models* by computer scientists, and so on. At each level, using a common ontology and standardized graphical

representations to describe the models and their connections, expert communications are enhanced, even if, at a high level of specification, models are simply viewed as black boxes by some experts.

The paper is organised as follows. Firstly, 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 wasteland fire phenomenon, we gradually focus on the study of fire spread through a mathematical model. This is followed by a design framework for cell space simulation. 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 involve 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) and Fishwick (1995).

The analogy between multimodelling and object-oriented programming has been demonstrated by

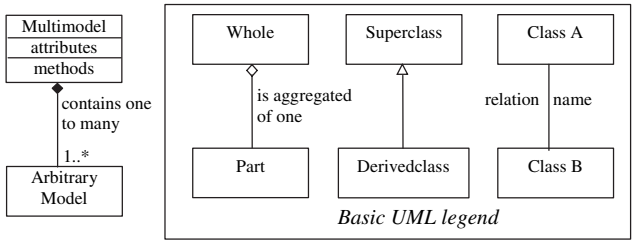

Fig. 1. Simple UML multimodel and basic UML legend.

Cubert et al. (1997), Frick (1997) and Fishwick et al. (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 multi-modelling: they are both founded on theoretic system concepts but they can be distinguished since they rely on different abstraction levels. In order to meet multi-modelling requirements, Fishwick (1996) introduced a new methodology called Object-Oriented Physical Modelling (OOPM) to extend the classical object-oriented analysis and design methods in use in the simulation community (Hill, 1996).

OOPM and DEVS models are well suited to precisely specify simulation models. Incidentally, DEVS concepts and techniques are used hereafter to design a cellular simulation model. However, both methodologies lack well defined tools to describe models at a very high level of abstraction. Studying large complex systems needs more and more to connect and identify abstract models from different disciplines. This can be achieved by using high level interdisciplinary models. The specification of the models can then be performed by different domain experts. Nevertheless, generic tools have to be provided to describe these models at each level of specification. These tools can then be used for communication between different domain experts. Both DEVS and OOPM approaches use the object-oriented reasoning. Recent advances in object-oriented concepts can be used to define a new high level methodology.

The history of object-oriented analysis and design methods led to UML (Unified Modelling Language) in 1997. UML is the subject of a lot of research and UML 2.0 has recently been adopted by the OMG as the new de facto 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 work we retain UML for our multimodel designs since they will be more widely read using this graphical notation. It proposes to various domain experts a 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 provides 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 ready-to-run software for fire spread simulation (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; Albright and Meisner, 1999; 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 (Karafyllidis 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, 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.

By the way, object-oriented programs of cellular models are closer to reality than procedural programs. Considering an object *Cell*, the behaviour of a patch of a land can be designed by the *Cell* methods. The states of the patch can be described by the *Cell* attributes. A *Cell* object corresponding to a class, the class can be reused to implement other cells containing different behaviours or states. Modifications of the behaviour or states can then be easily achieved by designing a new class, inheriting from the previous one. Polymorphism can also be used to specify new behaviours.

Nevertheless, all these recent advances in the field of fire spread simulation do not deal with real-time constraints. As far as we know, even if these approaches improve simulation model reusability and simplify simulation model modifications, only one of them is capable of simulating fire spread models requiring more computer resources under real time deadlines (Muzy et al., 2003). To give real-time advice for fire fighters, execution times of fire spread simulators have to be as small as possible compared to an actual fire propagation. Moreover, small execution times mean increased productivity of researchers experimenting new models. Simulation software have thus to deal efficiently with the large amount of data and operations required by fire spread models. 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 conceptualised the whole or 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 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

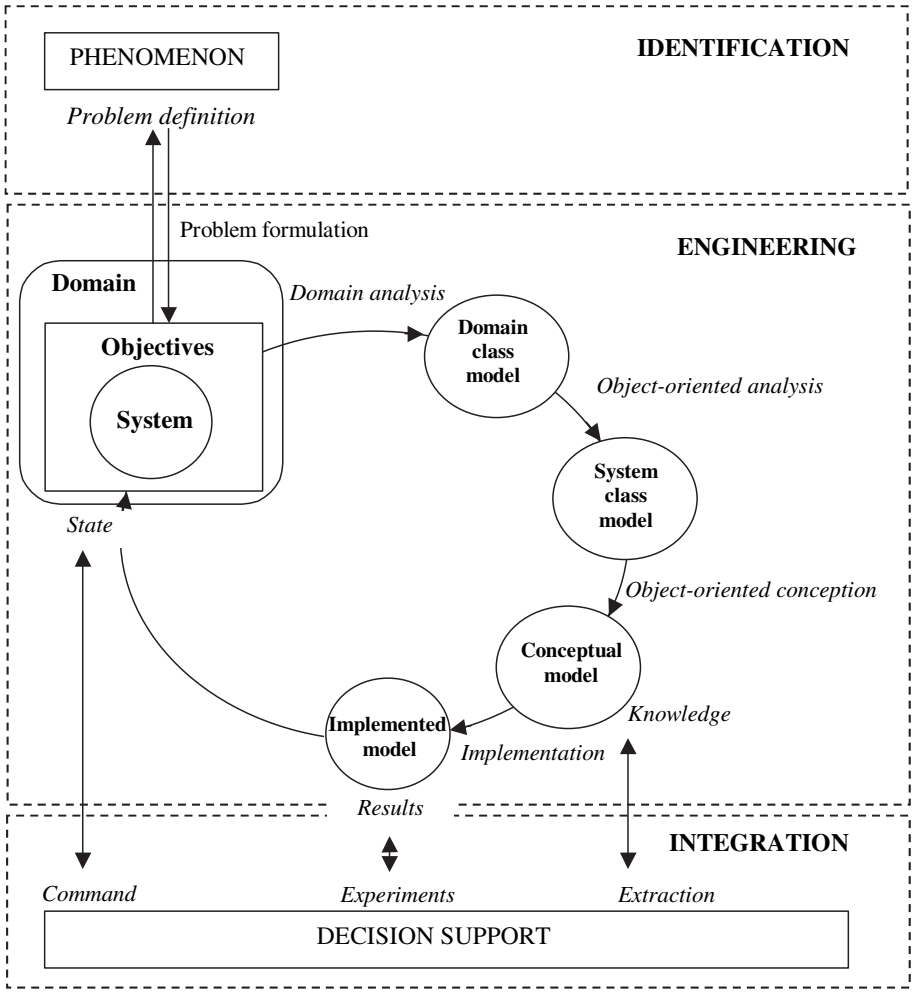

Fig. 2. Modelling and simulation life cycle.

abstraction has to be verified with regard 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 brought 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 and (re)use of the simulation results to solve the problem; (2) improving the system knowledge looking at the conceptual model; and (3) sending commands to the physical system (if it exists) or to the conceptual system (if the system has to be (re)configured).

All the abstraction phases of Fig. 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 and Bendoricchio, 2001). Separating the model description from the experiment limits the changing of experiment description for new experiments (Lorek and Sonnenschein, 1998).

4. Application to fire spread modelling

Considering wasteland fire phenomena, defining a conceptual model consists in gradually specifying a particular domain of interest. Here, starting from a domain class model of wasteland fire we focus on the fire

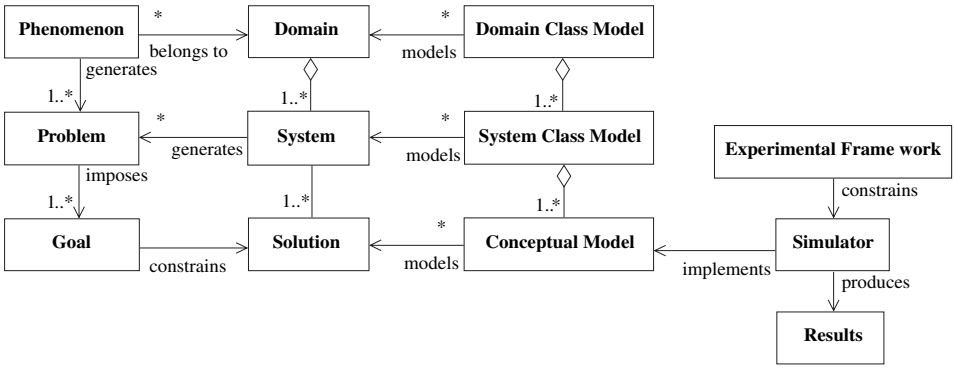

Fig. 3. UML meta-model of the modelling and simulation process.

behaviour dynamics. The system class model then consists 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 wasteland fire phenomenon can be broken down into four packages (Fig. 4):

1. The *Human Action* package encompasses development of wasteland fire prevention strategies including land management and equipment policies as well as the elaboration of technical or regulatory solutions for reducing the number of wasteland fire ignitions and fighting fires.
2. The *Environmental Conditions* package evaluates wasteland fire risks, describes geographic and climatic conditions and provides vegetation models.
3. The *Ecological Impact* package includes knowledge of fire consequences.
4. The *Fire Dynamics* package provides a description of the fire behaviour dynamics.

Fig. 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.

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, simple mathematical models have to be used to predict the main behavioural features of fire.

Based on the classification of Weber (1990), three kinds of mathematical models for fire propagation can

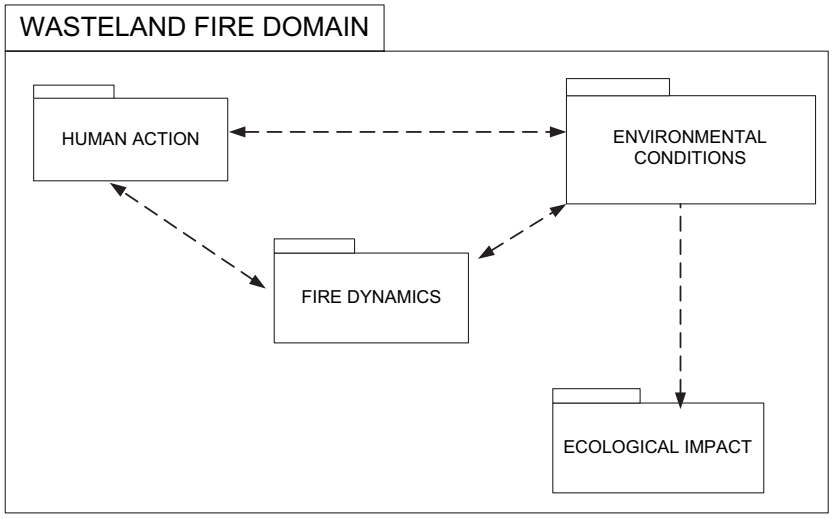

Fig. 4. Package diagram of fire domain breakdown.

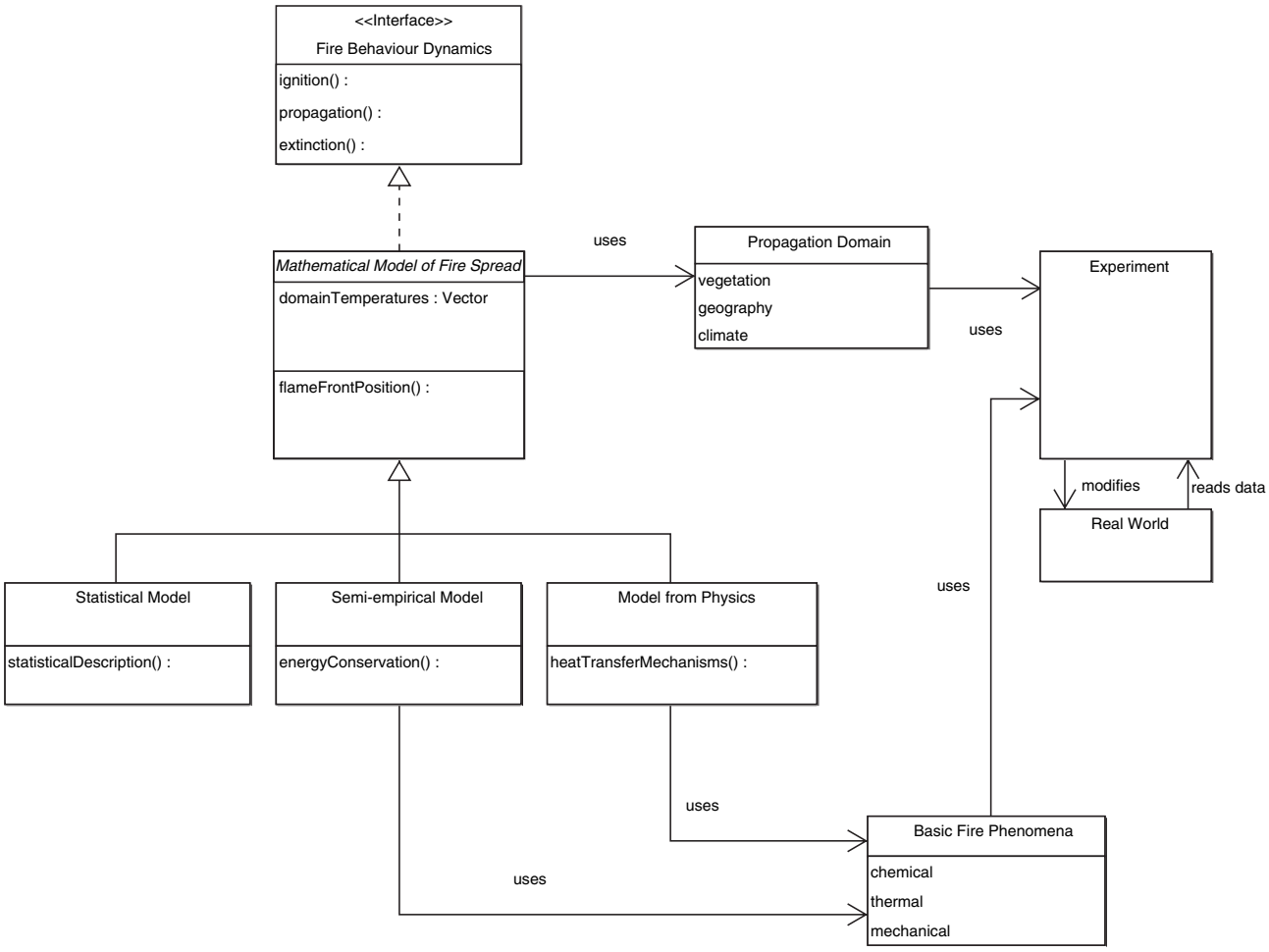

Fig. 5. Excerpt of the fire domain class model: the Fire Dynamics package.

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.

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.

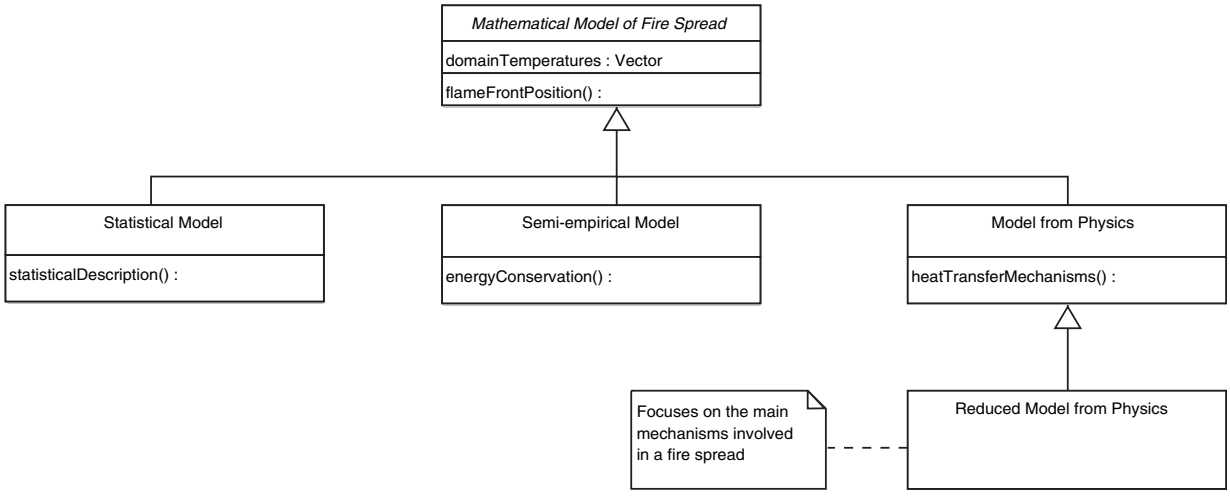

Fig. 6. Mathematical modelling of fire spread.

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 combining different mathematical models.

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 a 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:

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

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

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

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

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

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

where, considering a cell, T_a (27 °C) is the ambient temperature, T_{ig} (300 °C) is the ignition temperature, t_{ig} (s) is the ignition time, T (°C) is the temperature, K ($m^2\ s^{-1}$) is the thermal diffusivity, Q (m^2 °C/kg) is the reduced combustion enthalpy, Δ is the Laplacian in two-dimensional Cartesian coordinates, α (s^{-1}) combustion time constant, σ_v (kg m^{-2}) is the vegetable surface mass, σ_{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 Fig. 7.

Eq. (1a) has to be solved numerically. 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

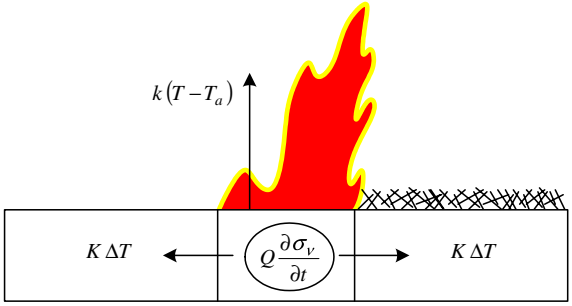

Fig. 7. Heat transfer of the semi-physical model.

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, Eq. (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 stable.

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-cm^2 and a time step of 0.01 s. The following algebraic equation is obtained:

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

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 Eq. (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-cm^2 and a time step of 0.1 s.

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

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

5. Design framework for cell space simulation

Different choices of software architectures can be selected to simulate propagation processes. Developing efficient, reusable and easy to maintain programs is a long and difficult task. However, for a class of systems, different programming techniques and principles can be identified to propose a reusable software architecture. A framework is a guide for 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 between 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 the real world to simulate. Separating computer model and simulator makes it possible to reuse a simulator for many models thus reducing development time (Zeigler et al., 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 and Reynolds, 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). 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, behaviour 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 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 and Bendoricchio, 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.

5.2. Coupling the world model and data landscape

Let us 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

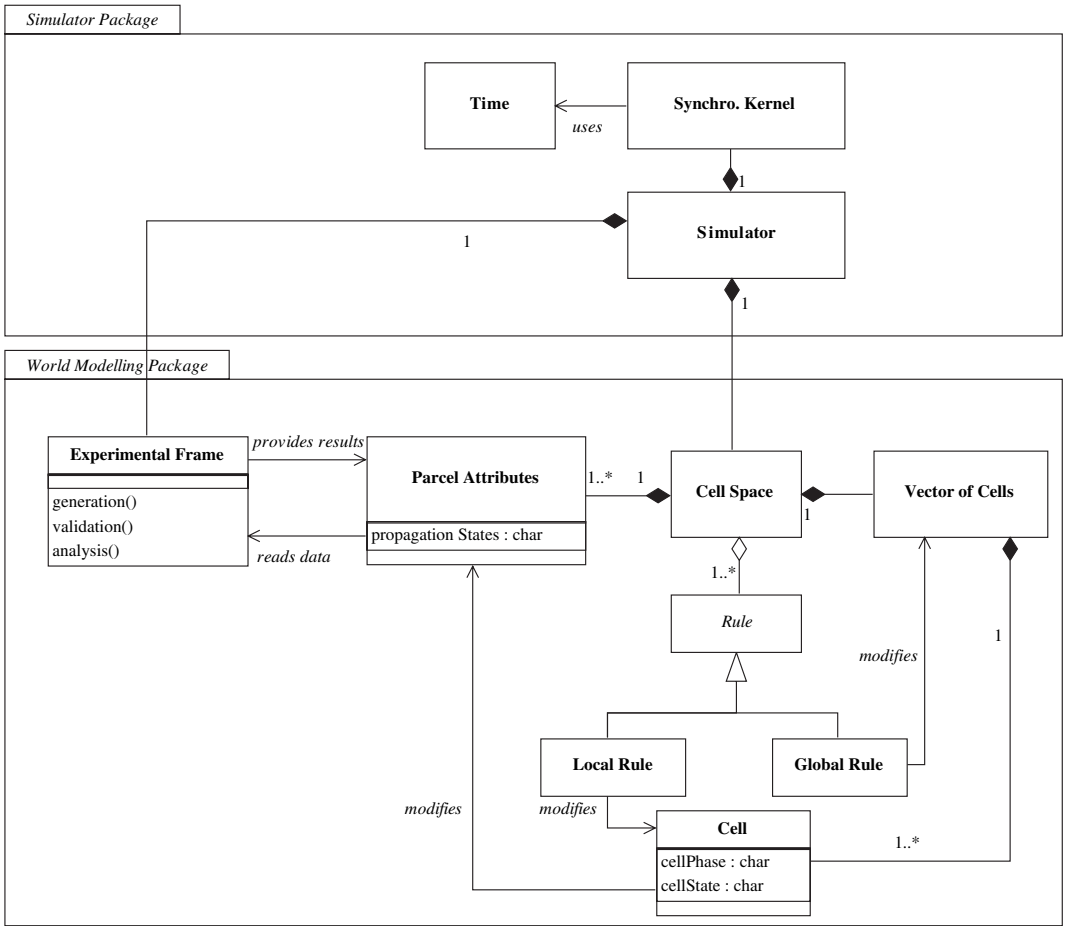

Fig. 8. Proposition of a design framework for cell space simulation.

simulation and being coupled with the data landscape. In large propagation phenomena, a GIS generally provides the latter. 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.

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 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 first two 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

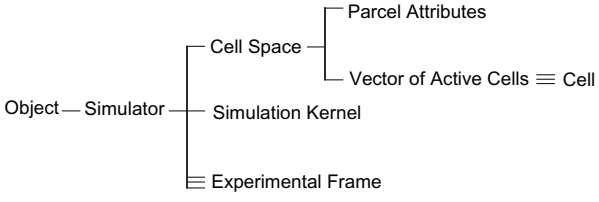

Fig. 9. The system entity structure for the cell space simulation.

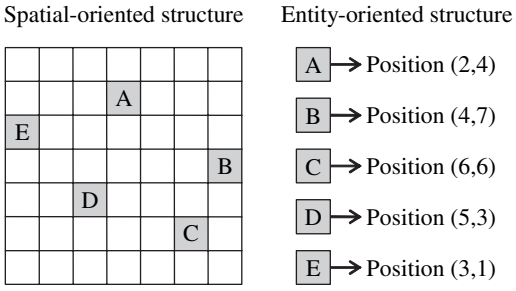

Fig. 10. Choices of world data structure for DES.

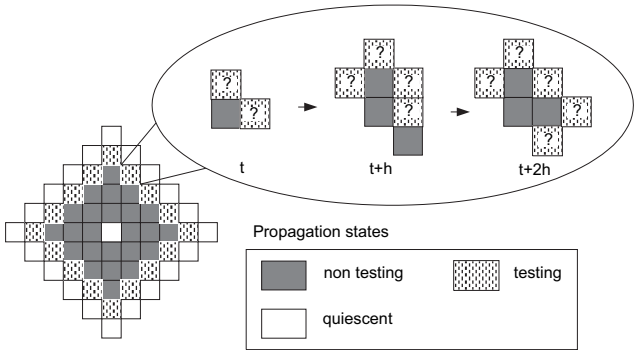

Fig. 11. Evolution of the calculation domain.

and a specialised static allocation has 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 delimit 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 optimised simulation of a neighbour-to-neighbour propagation. In Section 4 we saw that the numerical resolution of the PDE (Eq. (1a)) leads to two algebraic Eqs. (2) and (3). The differences between Eqs. (2) and (3) are (1) the condition used to increment the discrete time base added for Eq. (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 testing 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 Fig. 11 for a cardinal neighbourhood. In our algorithm, only the bordering cells test their neighbourhood, 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 neighbouring cells are added to the set of active cells.

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 bound-to-occur or book-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).

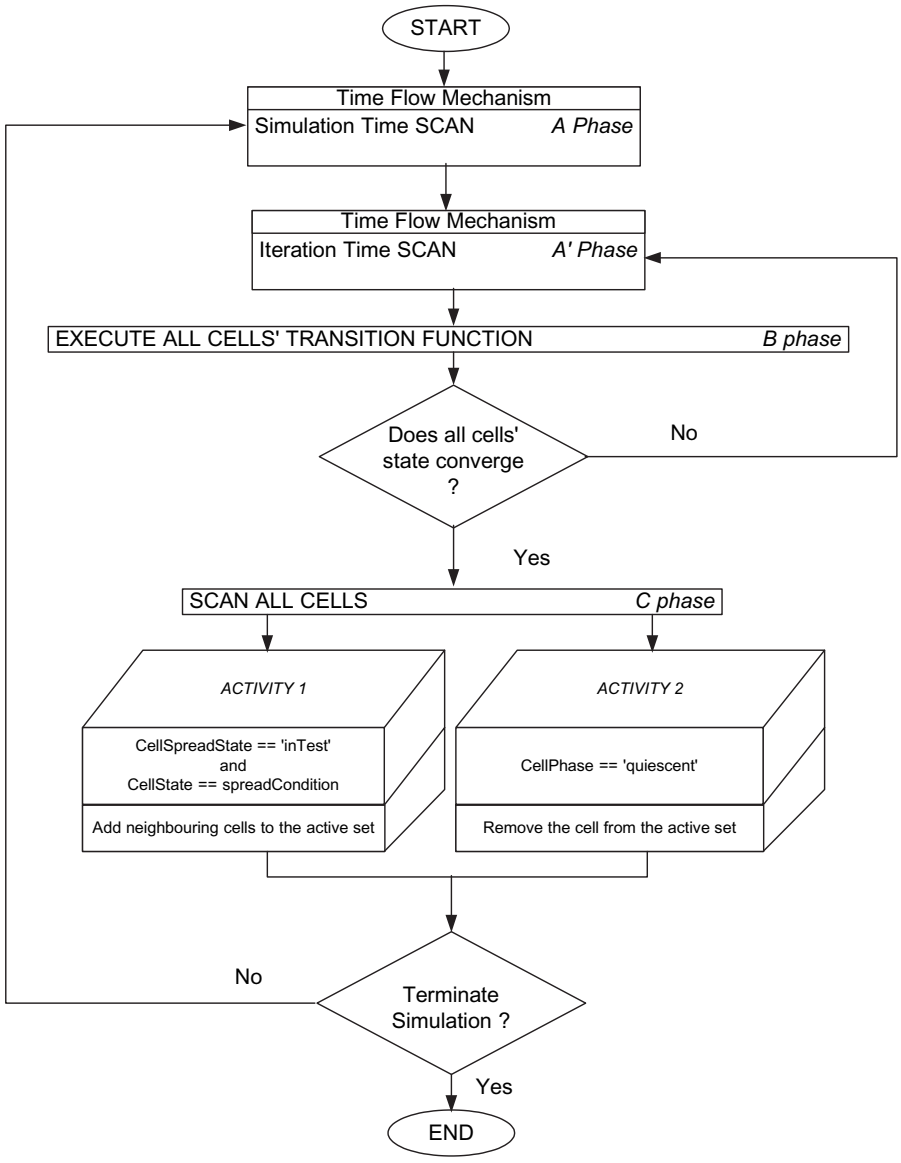

Fig. 12. Three-phase approach modification for implicit model simulation.

Fig. 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 duplicated 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.

Fig. 13 shows a comparison of the simulated and experimental fire fronts obtained for a point-ignition. Black squares represent the experimental fire fronts. Fig. 14 represents the evolution of the active cells around the fire fronts. Finally, Figs. 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. It is noticeable that this gain is more important for a three-phase simulation. Indeed,

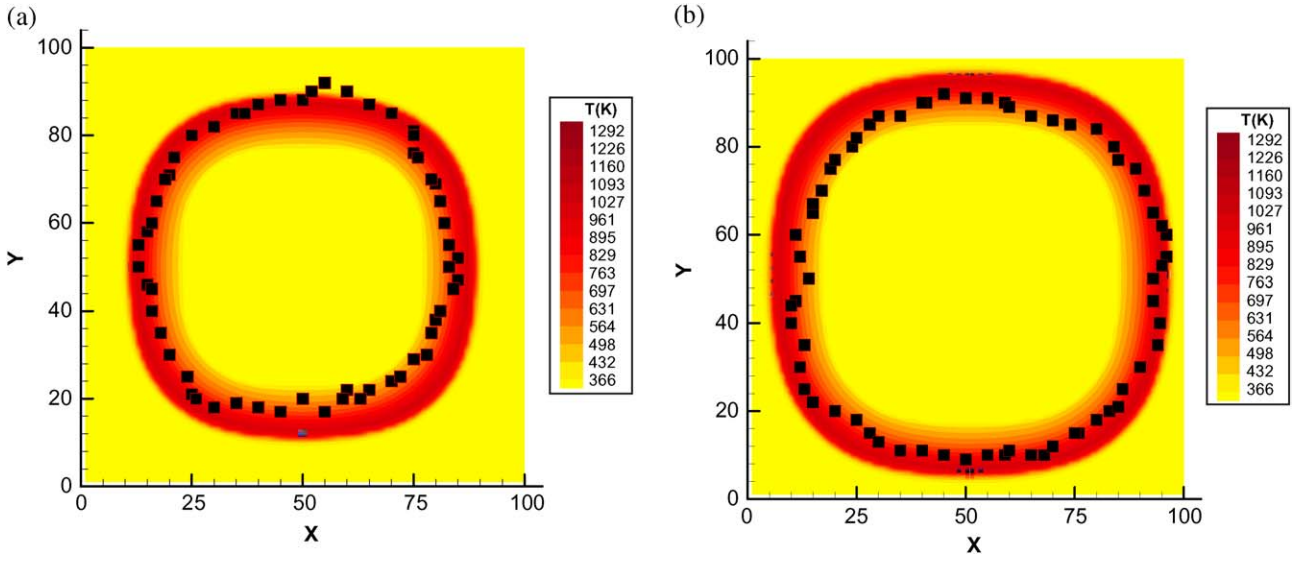

Fig. 13. Simulated and experimental fire fronts at: (a) $t = 122$ s and (b) $t = 144$ s.

for the implicit method if the time step is greater than the explicit one (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 cell transition functions, the pure discrete event one corresponding to the explicit model necessitates 56 million cell transition functions.

8. Conclusion

We have presented a method for modelling wasteland fires and simulating fire spread. This method uses both the 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 wasteland 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), behaviours (rules of the cells or cell space) or time schemes (simulator part of the framework). Optimisation concepts

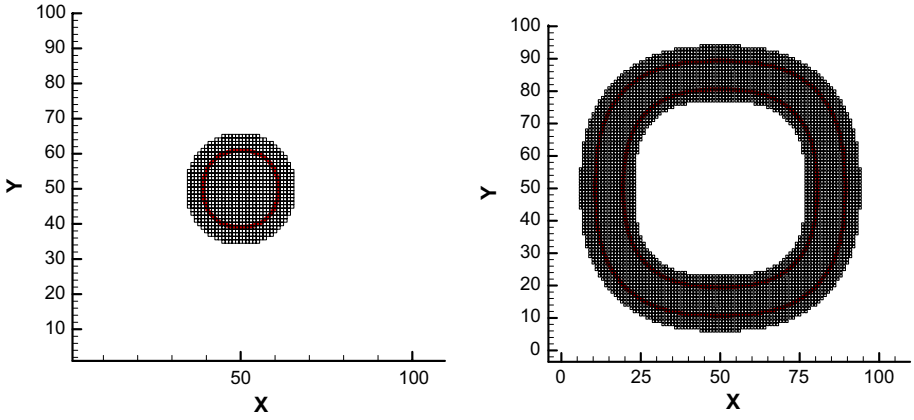

Fig. 14. Evolution of the active cells.

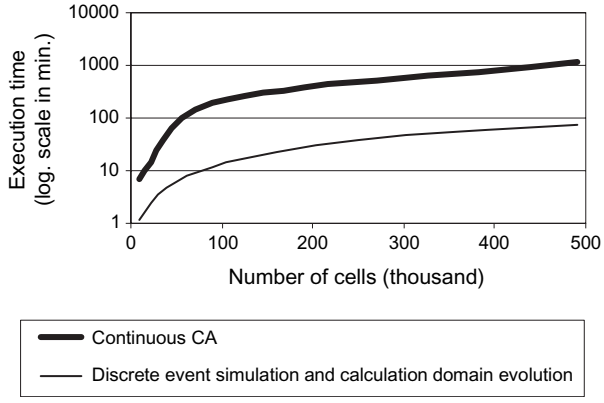

Fig. 15. Execution time gain using a discrete event simulation focusing on active cells of the propagation domain.

concentrating the simulation on active cells can also be reused to significantly reduce simulation time of large cell spaces.

The method now needs to be tested for other ecological systems and thus refined for the encountered problems. In the wasteland 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 has been done in a specific application domain, UML profiles (Fuentes and Vallecillo, 2001) will be able to be designed for these applications. 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).

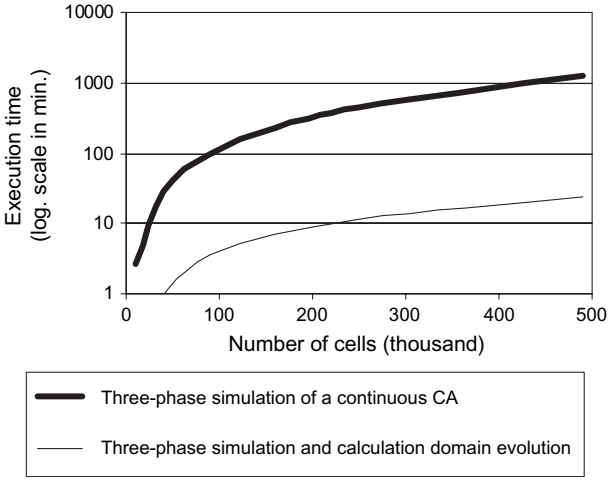

Fig. 16. Execution time gain using a three-phase simulation focusing on active cells of the propagation domain.

Acknowledgements

The authors thank three anonymous reviewers who provided comments and suggestions on the content of this paper that greatly improved its value.

References

Albini, F.A., 1985. A model for fire spread in wasteland fuels by radiation. Combustion Science and Technology 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 & Software 15 (3), 215–224.

Ameghino, T., Tróccoli, A., Wainer, G., 2001. Models of complex physical systems using Cell − DEVS. Proceedings of Annual Simulation Symposium, Seattle, WA, USA.

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

Balbi, J.H., Santoni, P.A., Dupuy, J.L., 1999. Dynamic modelling of fire spread across a fuel bed. International Journal of Wasteland 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 modelling using dynamic structure cellular automata. III International Conference On Forest Fire Research, 14th Conference on Fire and Forest Meteorology, Luso, 16–20 November, vol. I, pp. 879–888.

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

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

Baveco, J.M., Smeulders, A.M.W., 1994. Objects for simulation: smalltalk 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. Ecological Modelling 148, 133–151.

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

Breckling, B., Müller, F., 1994. Current trends in ecological modelling and the 8th ISEM conference on the state-of-the-art. Ecological Modelling 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. Ecological 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. Ecological Modelling 103, 1–17.

Coquillard, P., 1995. Simulation of the cyclical process of heathlands. Induction of mosaics structures, evolution to irreversible states. Ecological Modelling 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 multimodelling simulation system, enabling technology for simulation science. SPIE Aero Sense 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. Ecological Modelling 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. International Journal of Wasteland Fire 9, 155–172.

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.

Finney, M.A., 1995. Fire Area Simulator. User Guide and Technical Documentation, 75 pp.

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

Fishwick, P.A., 1996. Extending object-oriented design for physical modelling. ACM Transactions on Modelling and Computer Simulation.

Fishwick, P.A., Sanderson, J., Wolff, W., 1998. A multimodelling basis for across-trophic-level ecosystem modelling: the Florida Everglades example. SCS Transactions 15 (2), 76–89.

Frick, A., 1997. Multi-Model analysis and design. Available from: preprint: http://www.aifb.uni-karlsruhe.de/Staff/afr.html.

Fuentes, F., Vallecillo, A., 2001. Using UML profiles: a case study. In: Edi, A. (Ed.), Proceedings of the ECOOP'2001 Workshop on Automating Object-Oriented Software Development Methods (AOOM'01), Hungary, June.

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. International Journal of General Systems 5, 31–43.

Grishin, A.M., 1997. Mathematical modelling of forest fires and new methods of fighting them. In: Albini, F. (Ed.), Publishing house of the Tomsk State University, pp. 81–91. (Chapter 1).

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

Hill, D., 1996. Object-oriented Analysis and Simulation. Addison-Wesley Longman, UK, 291 pp.

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

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 1996, October 24–26, Genova, Italy.

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 109, 251–265.

Holst, N., Axelsen, J.A., Olesen, J.E., Ruggle, P., 1997. Object-oriented implementation of the metabolic pool model. Ecological 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. Ecological Modelling 99, 87–97.

Larini, M., Giroux, F., Porterie, B., Loraud, J.C., 1997. A multiphase formulation for fire propagation in heterogeneous combustible media'. International Journal of Heat Mass Transfer 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. Ecological Modelling 88, 113–124.

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

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. Object-oriented support for modelling and simulation of individual-oriented ecological models. Ecological 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. Ecological Modelling 149, 71–83.

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

McArthur, A.G., 1966. Weather and grassland fire behaviour. Australian Forest and Timber Bureau Leaflet, No. 100.

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

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

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

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 & 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. International Journal of Nurner Methods Eng. 30, 1163–1179.
Rothermel, R.C., 1972. A Mathematical Model for Predicting Fire Spread in Wasteland Fuels. USDA Forestry Service Research 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. 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. Ecological Modelling 58, 55–89.
Sibony, M., Mardon, J.C., 1998. Approximations et équations différentielles. Hermann.
Silvert, W., 1993. Object-oriented ecosystem modelling. Ecological 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. Ecological 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. Ecological 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 & 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. Ecological Modelling 94, 299–316.
Stroustrup, B., 2000. The C++ Programming Language (Special Edition). Addison-Wesley, Reading, MA, 1029 pp.
Tobler, W.R., 1979. Cellular geography. In: Gale, S., Olssein, G. (Eds.), Philosophy in Geography, pp. 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-neighbour pollination. Proceedings of the National Academy Science USA 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. Progress in Energy and Combustion Science 17, 67–82.
Wolfram, S., 1994. Cellular Automata and Complexity: Collected Papers. Addison-Wesley.
Wolfram, S., 2002. A New Kind of Science. Wolfram Media, 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. Ecological 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. Johns Hopkins APL Tech. Dig. 16 (2), 197–205.
Zeigler, B.P., 1976. The aggregation problem. In: Patten, B. (Ed.), System Analysis and Simulation in Ecology. Academic Press.
Zeigler, B.P., 1979. Multi-level multiformalism modelling: an ecosystem example. In: Halfton, E. (Ed.), Theoretical Systems Ecology. 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, second ed. Academic Press.