Forest Fire Spread and Suppression in DEVS

Lewis Ntaimo
Department of Industrial Engineering
Texas A&M University
3131 TAMU
College Station, TX 77843
ntaimo@tamu.edu

Bernard P. Zeigler
Arizona Center for Integrative Modeling and Simulation
Department of Electrical and Computer Engineering
University of Arizona
1230 E. Speedway
Tucson, AZ 85721

Maria J. Vasconcelos
Tropical Research Institute
Travessa do Conde da Riberira 9
1300-142 Lisbon
Portugal

Bithika Khargharia
Arizona Center for Integrative Modeling and Simulation
Department of Electrical and Computer Engineering
University of Arizona
1230 E. Speedway
Tucson, AZ 85721

In this article, the authors discuss modeling and simulation of forest fire spread and suppression using the discrete event system specification (DEVS) cell space approach in DEVSJAVA. The event-based modeling approach enables efficient simulation of cell space and allows one to obtain timely simulation-based predictions of forest fire spread and suppression in uniform and nonuniform environmental conditions. This model represents an advance toward developing a real-time decision support simulation system for predicting forest fire spread and the effects of water-based suppression attempts.

Keywords: DEVS, cell space, forest fire spread, fire suppression

1. Introduction

Wildfires have become ever more destructive throughout the world, and the prospects are unfortunately that this trend will continue. Greater attention must be focused on the underlying causes, the effect of land management on fire ecology, wildfire risk, the dynamics of vegetation fuel, and how to reduce the likelihood of catastrophic fires [1]. Forest fires destroy important resources such as plant life, animal life, houses, and other civil infrastructures every year. “Strategic” planning attempts to avoid the large losses and suffering associated with wildfires by trying to prevent fire outbreaks. However, once a forest fire has started, a real-time simulation system for accurately predicting where and how fast the fire will spread would assist at the “tactical” level to effectively bring it under control. Furthermore, such a system would allow for ample warning to be given to the people living in the predicted path of a fire, thus enabling a well-coordinated evacuation plan. Important resources would be saved from destruction by effective fire suppression and would save a lot of money and time involved in fighting forest fires and in evacuating people.

SIMULATION, Vol. 80, Issue 10, October 2004 479-500

DOI: 10.1177/0037549704050918

The long-term goal of this work is to develop a simulation tool that can be used in real time or as-fast-as-can tactical decision making to aid in forest fire control and suppression. Toward that objective, the research reported here is to develop a cellular discrete event system specification or DEVS model [2, 3] of forest fire spread that includes response to control measures. DEVS provides a sound modeling and simulation framework and is derived from mathematical dynamical system theory [2]. It supports hierarchical, modular composition, and reuse and can express discrete time, continuous, and hybrid models. DEVS allows for the construction of a hierarchical simulation model composed of atomic and coupled models. Each stand-alone atomic model is assigned to an atomic simulator, and atomic models as components within coupled models are assigned to a coupled simulator. Coupled models are assigned to coordinators, while coupled models as components within larger models are assigned to coupled coordinator simulators. The simulators keep track of the events and execute the simulation model-defined methods based on the events list.

The key feature of the DEVS modeling approach we seek to exploit is its ability to effectively represent large spatial dynamic phenomena for efficient simulation [4-6]. Unlike simulations that are based on cellular automata, in which all cells perform computations at every time step, the cellular DEVS approach allows computations to occur only in active cells, thus enhancing simulation performance. Our approach differs from the other cellular DEVS approaches (e.g., [3, 5]) mainly in the way we exploit the fact that computations are concentrated on the cells along the fire front and optimize the forest cell space for improved simulation performance [7]. We use DEVSJAVA [8], a Java-based modeling and simulation software implementation of the DEVS formalism. The idea of improving the cell space in cellular-based simulations is found in Muzy et al. [9], who apply the Multicomponent Discrete Time System Specification (MultiDTSS) formalism to a fire spread model. In the DEVS abstract simulation protocol, the number of *imminents* (those components for which there are internal transitions scheduled at the next event time) determines the amount of processing in each simulation cycle. In a DEVS forest fire model, the cells along the fire front (active cells) represent most of the imminents. Therefore, our DEVS simulation engine takes advantage of the small number of imminents to improve simulation time.

Our model predicts fire spread (speed and direction) as the fire propagates based on both static and dynamic conditions. The vegetation and topographical conditions are assumed to be uniform across a cell, while the weather conditions are allowed to be dynamic. The model considers nonuniform fire spread parameters to address the issue of spatial (temporal) variability of the variables. It follows along the line of work of Vasconcelos [10], who introduced and illustrated the conceptual basis for a discrete event hierarchical modular fire spread model and covered a variety of issues in forest fire simulation and modeling in general.

The contribution of this article includes the development and implementation of a dynamic forest fire spread simulation model that represents an advance toward improving simulations of forest fire spread with innovations in the way the process is represented, implemented, and optimized. In addition, we incorporate into our simulation model simple rules for firefighting that are derived from well-accepted rules of thumb for fire suppression. This allows our model to provide important information needed in tactical decision making for forest fire control and suppression.

This article is organized as follows. In section 2, we provide basic concepts and develop our dynamic forest fire model. In particular, we derive and state cell space model rules and fire suppression rules. In section 3, we present the simulation model as represented in DEVSJAVA, discussing the system hierarchical structure through hierarchical diagrams, and give a detailed explanation of the model's operation. The experimental frame and how it serves to achieve the goals of the simulation are discussed in section 4. In section 5, we report our simulation experimental results, and in section 6, we conclude and give some future directions. A more detailed explanation of the cell's transition functions is given in the appendix.

2. Dynamic Forest Fire Model Conceptualization

In this section, we give a general description of the conceptualization of a dynamic forest fire spread model and provide rules for optimizing the cell space. We model a forest as a two-dimensional cell space composed of individual forest cells with uniform vegetation, environmental, and topographical conditions across each cell. We capture the complex behavior of forest fire spread by allowing each forest cell to be represented as a "cell" in the simulation that performs its local fire spread computations based on a given fire spread model and exchange messages with its neighbors. DEVSJAVA allows us to represent forest cells as atomic models with input and output ports between neighboring cells and enables message exchanges to occur on the ports. A forest cell space is composed of a number of forest cells and constitutes a coupled model.

Here we distinguish between two types of *cells*. We have *static* grid cells representing space and stored in the environmental database and *model/simulation* cells. The static grid cells are external to the simulation and represent environmental conditions and fire position, while the dynamic cells are simply an abstraction of the forest cell and are represented in DEVSJAVA. Graphically and database-wise, all static cells of a given space of concern are always present and represented in the database and could be designated by geographical units. However, the cells internal to DEVSJAVA in the software are created at initialization, and this is where the burning process is computed. In particular, we model fire spread in each cell according to Rothermel's [11] stationary model, which is a one-dimension semi-empirical model. We obtain the second dimension by applying a propagation algorithm that uses maximum rate of spread and wind and slope factors. We allow the behavior of a burning cell to be influenced by external inputs from neighboring cells as well as changes in weather conditions. In addition, we incorporate uncertainty in our model by allowing certain critical parameters to be sampled from arbitrary probability distributions during the simulation run if those parameters cannot be determined with certainty. Next we provide the forest fire spread model concepts.

2.1 Basic Model Concepts

2.1.1 Cell Description

We represent a forest as a two-dimensional cell space composed of cells of dimensions $l \times b$, where l and b are the length and breadth of the cell, respectively. For each cell, we define fixed major spread directions (propagation lines) N, NE, E, SE, S, SW, W, and NW as shown in Figure 1. This allows for the computation of fire spread in only the specified major directions instead of all directions, thus significantly reducing fire spread computation time.

These directions correspond to an azimuth (degrees measured clockwise from the north) of 0, 45, 90, 135, 180, 225, 270, and 315 degrees, respectively. The definition of a cell in this way is the same as that of Vasconcelos [10] and Ameghino, Troccoli, and Wainer [5], among others.

2.1.2 Fire Spread

Fire spread is the propagation of a flaming front that constitutes a series of ignitions, with the heat of the fire raising successive stripes of fuel to ignition temperature via a contagion process that is a steady-state process for homogeneous fuels and unsteady for nonhomogeneous fuels [11, 12]. The behavior of the fire spread is controlled by the following factors: the *fuel type* (particle size and array in the fuel bed, chemical composition, and fuel moisture content), *weather conditions* (ambient temperature, air humidity, wind speed, and direction), and *topography* (mainly slope) of the fire location.

In our model, fire spread is computed along the major wind directions based on Rothermel's [11] semi-empirical model for uniform fuels. Given a cell and its forest fuel model type and environmental parameters (slope, wind speed, and wind direction), Rothermel's model computes the maximum fire spread and the corresponding direction of spread. Fire spread is given in meters per second (m/sec), and the direction is given in degrees (0-360 degrees clockwise from uphill). Once the maximum *fire spread* and *direction* are known, we decompose the fire spread in a cell in the major spread directions using a model that defines fire shape as an ellipse and is proposed by Anderson [13] and Alexander [14]. The same idea is implemented in Ameghino, Troccoli, and Wainer [5]. Given the computed rate of spread in each of the eight directions for a given cell, the time it takes for the fire to spread from the center of the cell to the center of the neighboring cell is computed based on the distance between the cell centers. The computed times provide the spread time delays in the major spread directions for each cell. The direction with the minimum fire spread gives the maximum spread time delay for that cell.

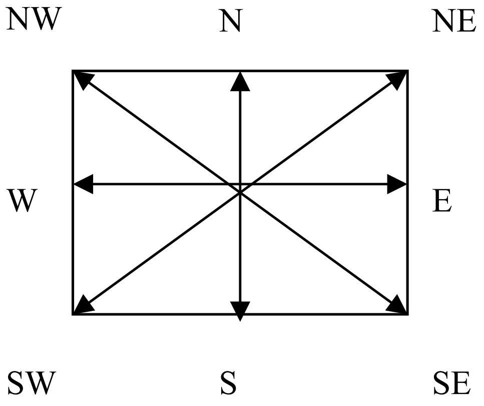


Figure 1. Cell with major spread directions

The wind speed and direction for each cell are assumed to be available from a meteorological station closest to the forest fire that provides the current wind speed and direction as the fire advances. However, the wind flow at a certain site is strongly influenced by the surrounding terrain and topography elements. We therefore assume that a wind flow model is available to capture the variability of the wind speed and direction due to complex terrain and computes the effective wind speed in each cell. An example of a wind flow model is given by Nelson [15]. In our simulation model, we allow for varying wind speed and direction in each cell to approximate reality more closely. The wind speed and direction are updated during the simulation run if a significant change in the mean wind speed and direction is detected.

2.1.3 Fireline Intensity

We use *fireline intensity* (I) to determine if a cell is burnable. Fireline intensity is the product of the available heat of combustion per unit area of ground and the rate of spread of the fire [16] and is given by $I = hwR$, where h is the heat yield in kJ/kg, w is the fuel consumption in kg/m^2, R is the rate of spread of the fire in m/sec, and I is given in kW/m. Fireline intensity is computed based on fuel loading and fuel moisture conditions for surface fire behavior [11, 17]. We allow cells to compute their fireline intensity values, and only cells with intensities greater than 45 kW/m (13 BTU/ft sec) are considered burnable. This follows after Miller and Urban [18], who have determined this value for fire regimes in Sierra Nevada. In the next subsection, we describe the cell states and cell state transitions.

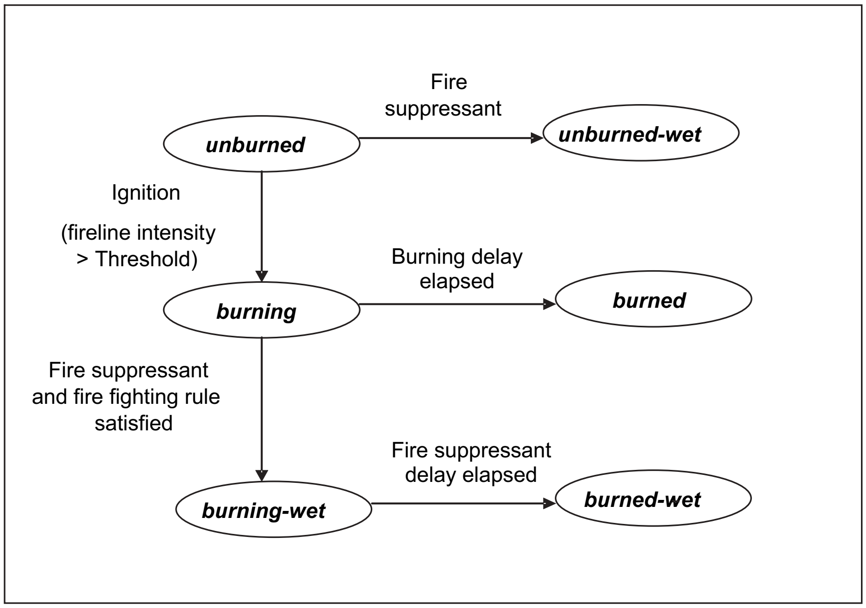

unburned
Fire suppressant
unburned-wet
Ignition
(fireline intensity > Threshold)
burning
Burning delay elapsed
burned
Fire suppressant and fire fighting rule satisfied
burning-wet
Fire suppressant delay elapsed
burned-wet

Figure 2. Forest cell state transitions

2.1.4 Cell States and Cell State Transitions

To model fire spread in DEVSJAVA, we make an abstraction from the actual forest cell to a “forest cell” in DEVSJAVA that can be in only one state at any time. We define the following six “dry” and “wet” states: *unburned*, *burning*, *burned*, *unburned-wet*, *burning-wet*, and *burned-wet*. Initially, all the cells are in the *unburned* state (passive state), with their current intrinsic vegetation and environmental conditions given. A cell that transitions into *unburned-wet*, *burned*, or *burned-wet* stays in this state for the duration of the simulation (absorbing state). The forest cell state transitions occur as shown in Figure 2.

A cell in the *unburned* state will remain in this state forever unless it is either ignited or fire suppressant (or water) is poured on the cell. A cell in the *unburned* state transitions to the *burning* state (active state) if it is ignited and the computed *fireline intensity* value for the cell is above the *threshold* value given in the previous section or one determined experimentally for the fuel model. Otherwise, if a fire suppressant is poured on the cell in the *unburned* state, the cell immediately transitions to the *unburned-wet* state. Once in the *burning* state, a cell transitions to the *burned* state when all the eight time delay components have elapsed. Otherwise, if a fire suppressant is introduced into a cell that is in the *burning* state, it transitions to the *burning-wet* state if some firefighting rule is satisfied, as described in section 2.3. It stays in this state for a delay that is computed based on the firefighting scenario for that cell and then transitions into the *burned-wet* state. A cell in any wet state does not propagate fire.

A cell in a given dry state transitions to the corresponding wet state if fire suppressant is poured on the cell and the flame length (or *fireline intensity*) for that cell is in a given range based on the fire suppression rules defined in section 2.3. Otherwise, the cell stays in the dry state. This allows us to model firefighting influence (or rain) on the model using simple fire suppression rules. Ameghino, Troccoli, and Wainer [5] also use simple rules to study the effect of rain on fire spread. Their model assumes that a fire would take 16 minutes to extinguish in stages of different length. In our model, we implement simple fire suppression rules based on flame length or fireline intensity, as given by the general reliable rules for fire suppression [12, 19].

2.2 Cell Space Model Rules

In the model, each cell has eight neighbor cells—N, NE, E, SE, S, SW, W, and NW neighbors, as shown in Figure 3, except the boundary cells. In addition, there is an *igniter* atomic model that is linked to all the cells and randomly (or deterministically) selects a cell to ignite at some given time (e.g., at the beginning of the simulation run).

We now define the model rules as follows:

1. A cell starts to burn if:

   (a) A cell is ignited by the *igniter* atomic model if its fireline intensity is above the threshold. Otherwise, it remains in the *unburned* state. If it is ignited, the cell starts to burn at the center,

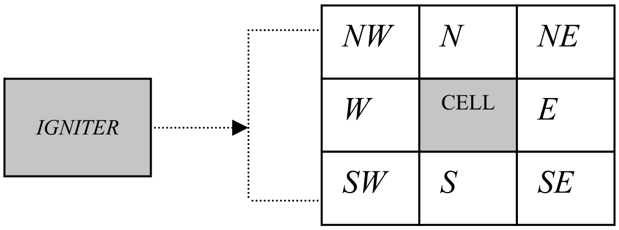


Figure 3. A two-dimensional cell space

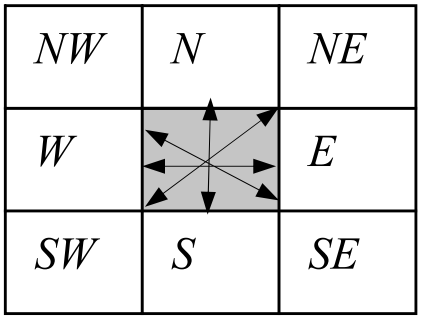


Figure 4. Fire spread directions after ignition by the *igniter* module

and the fire spreads in the major directions toward the center of each neighbor cell, as shown in Figure 4.

(b) A cell is ignited by the fire from the neighbor cell if the cell's *fireline intensity* is above the threshold defined earlier. Similarly, once the cell starts to burn, the fire immediately starts to spread along the eight major spread directions. A neighbor cell has the potential of being ignited by the fire spread component in the direction of the cell (see Fig. 5).

2. A cell burns completely when all the fire spread components associated with the cell reach the neighboring cell centers. In terms of simulation, the state change from *burning* to *burned* will occur after the maximum spread time delay for that cell has elapsed.

3. If a cell that is burning but not completely burned receives another input from a neighbor cell, the input is simply ignored. However, if there is a change in the wind speed and direction, then the cell recomputes the rate of spread and updates its time delays based on the associated remaining distance to reach the neighboring cell center. This facilitates for real-time simulation in which weather conditions are updated periodically during the simulation. The time delay updates apply only to spread components whose time delays are greater than zero. Otherwise, any spread component with time delay equal to zero would have burned out some time in the past.

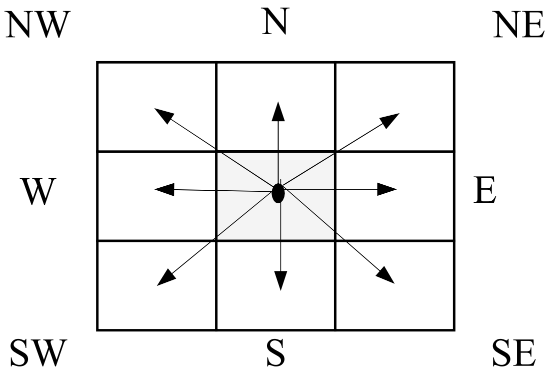


Figure 5. Potential neighbor cells to ignite by fire from center cell

4. When water is poured on the cell, the cell takes a random amount of time to transition to the corresponding wet state according to simple fire suppression rules.

5. The local computing function takes into account the following parameters as required by Rothermel's model:

   (a) Cell forest fuel model

   (b) Cell field topography (mainly slope)

   (c) Wind speed and wind direction

   (d) Cell conditions, wet, dry, humidity, and ambient temperature

The first two properties are assumed to be uniform for each cell, but the wind speed and wind direction are assumed to be determined from a wind flow generation model, given the mean values from the nearest meteorological station. The wetness conditions are also assumed to be available from some water source such as a fire suppression water source or rain.

2.3 Fire Suppression Rules

To allow our forest fire spread model to handle fire suppression scenarios, we implement simple fire suppression rules based on flame length or fireline intensity, as given by the general reliable rules for fire suppression [12, 19]. First, we assume that suppressing fire in a given cell that is in the *burning* state would take, at most, the time remaining for that cell to transition into the *burned* state. Therefore, we can define a nonnegative random variable α_i as the fraction of the remaining time for a cell to transition from the *burning* to *burned* state at the instant fire suppression is initiated for firefighting scenario i (i= 1, 2, 3, 4). We assume that α_i is either given or randomly determined. We now state fire suppression rules for the cases i= 1, 2, 3, and 4 as follows:

1. If the flame length in a given cell is less than 1.2 meters:

   (a) Fires can generally be attacked at the head or flanks of the fire by persons using hand tools.

   (b) Handlines should be adequate to hold the fire.

   In this case, our model assumes that $0 \leq \alpha_1 \leq 1$, and the fire in the cell can be extinguished. A value of $\alpha_1 = 0$ means that the fire is put out instantly when fire suppressant is poured on the cell, while $\alpha_1 = 1$ means that fire is put out at the instant the cell burns out.

2. If the flame length is between 1.2 meters and 2.4 meters:

   (a) Fires are too intense for direct attack at the head of the fire by persons using hand tools.

   (b) Handlines cannot be relied on to hold the fire.

   (c) Equipment such as bulldozers, pumpers, and retardant aircraft may still be effective.

   (d) Fires are potentially dangerous to personnel and equipment.

   In this case, we similarly we assume that $0 \leq \alpha_2 \leq 1$, and the fire in the cell can be extinguished. A value of $\alpha_2 = 0$ means that the fire is put out instantly when fire suppressant is poured on the cell, while $\alpha_2 = 1$ means that fire is put out at the instant the cell burns out. Due to the increased flame length and fireline intensity in this firefighting scenario, we assume that $\alpha_2 \geq \alpha_1$.

3. If the flame length is between 2.4 meters and 3.4 meters:

   (a) Fires may present serious control problems, such as torching out, crowning, and spotting ahead.

   (b) Control efforts at the head of the fire probably will be ineffective. Indirect attack is probably the only means of suppression. Equipment such as bulldozers, pumpers, and retardant aircraft may still be effective.

   (c) Fires are definitely dangerous to personnel and equipment.

   In this case, our model assumes that a fire cannot be extinguished by direct attack except by pouring fire suppressant on the forest cells before they catch fire.

4. If the flame length is greater than 3.4 meters:

   (a) Crowning, spotting, and major fire runs are probable.

   (b) Control efforts at the head of the fire are ineffective by any known means of suppression. Indirect attack and tactical counterfiring may be the only means to slow the spread of the fire in certain directions.

   (c) Fires are extremely dangerous to personnel and equipment in the immediate vicinity of the fire.

   Like in the previous case, our model assumes that a fire would not be extinguished by direct attack except by pouring fire suppressant on the forest cells before they catch fire.

We assume in general that an unburned cell in the wet state will never burn and will stay in this state for the duration of the simulation. The model rules are by no means complete, and it is up to the experimenter to modify these rules based on the availability of knowledge on fire suppression. Here we simply provide some basic rules that one can extend to gain more insight into fire suppression modeling.

In our simulation experiments in section 5, we generate the values of α_i, i = 1, 2, 3, and 4 by sampling from a random distribution due to lack of real experimental fire suppression data. In particular, we arbitrarily sample from the uniform distribution with α_1 = *uniform* (0, 0.5) and α_2 = *uniform* (0.5, 1.0).

3. Simulation Model

In this section, we present and describe the system entity structure and give the hierarchical diagrams of the overall structure of the simulation model. We also explain how the cell space is optimized for better simulation performance and provide implementation details and explanation of operation of the simulation.

The DEVS cell space approach allows us to define a two-dimensional cell space, with individual cells representing the smallest physical forest unit whose vegetation properties remain uniform but the environment properties are allowed to be dynamic. The objectives of the modeling and simulation study are to model and simulate forest fire spread under dynamic and nonuniform conditions using the DEVS cell space approach, with an emphasis on simulation efficient performance. We employ local computing in each cell to determine the fire spread for that cell based on the vegetation and current environmental conditions.

The advantage of using the DEVS cell space approach is that the model allows computation and transmission of messages only in active cells, thereby enhancing the efficiency of the simulation. In addition, we design a conceptual simulation model architecture that allows cells to be dynamically created and deleted as necessary during the

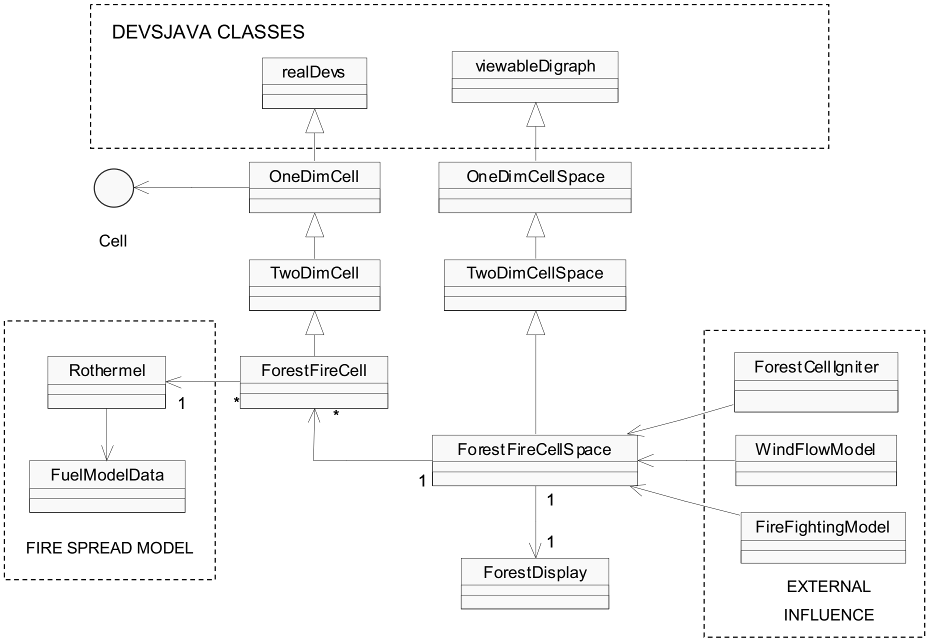


Figure 6. Overall system entity structure

simulation run. By coupling together simple cells with independent local behaviors, we are able to build a complex system that allows dynamism and uncertainty to be added to the model. This also enables us to capture the complex relationships among vegetation, topography, and meteorology that result in highly heterogeneous environmental conditions for the spread of forest fires.

3.1 System Entity Structure

The overall system structure for the forest fire spread simulation model is hierarchical and is given in Figure 6. The *realDevs* and *viewableDigraph* classes are already available in the DEVSJAVA package.

The *Rothermel* class is actually a JAVA package called BEHAVE that implements the Rothermel model [11, 17] for computing fire spread and is freely available on the Web (http://www.geo.unizh.ch/gis/research/edmg/fire/unc.html), together with data for the 13 different forest fuel models as defined by the National Forest Fire Laboratory (NFFL). We used this package in our simulation model for computing fire spread. The BEHAVE model takes in the fuel model, topographical conditions (mainly slope), and weather conditions as input values and computes, among other things, the rate of spread, fireline intensity, and heat released per unit area for a particular homogeneous area (forest cell) in the landscape.

We developed and implemented all the other classes shown in the diagram. The *TwoDimCell* class is implemented by inheriting from the *OneDimCell* class, which implements the *Cell* interface. This class has a method for doing cell neighbor-to-neighbor coupling and has additional ports to the *cell* interface as needed for the forest fire model. Similarly, the *TwoDimCellSpace* class is implemented by inheriting from the *OneDimCellSpace* class, which is inherited from the DEVSJAVA class *realDevs* that handles atomic and coupled models that exchange real values.

The *ForestFireCellSpace* inherits the *TwoDimCellSpace* and is composed of several *ForestFireCells* coupled together according to the *TwoDimCellSpace* neighbor-to-neighbor coupling rules as described below. The *TwoDimCellSpace* is inherited from the DEVSJAVA class *viewableDigraph*. This class allows for the creation of coupled models that can be graphically viewed in SimView, the DEVSJAVA Simulation Viewer described in section 5. The *forestCelligniter* is responsible for the initial ignition of the forest by igniting a cell chosen at random (or deterministically) at the beginning of the simulation run and is, therefore, connected to the *ForestFireCellSpace*.

The *WindFlowModel* computes the local wind speed and direction, given the mean wind speed and direction values from the meteorological station, and is also connected to the *ForestFireCellSpace*. This model can generate different wind speeds and direction at scheduled intervals during the simulation. Here we allow the user to enter a mean speed and direction for the entire cell space, and the model computes the local wind speed and direction for each cell by sampling from the normal distribution with the given mean and a user-selected standard deviation. It

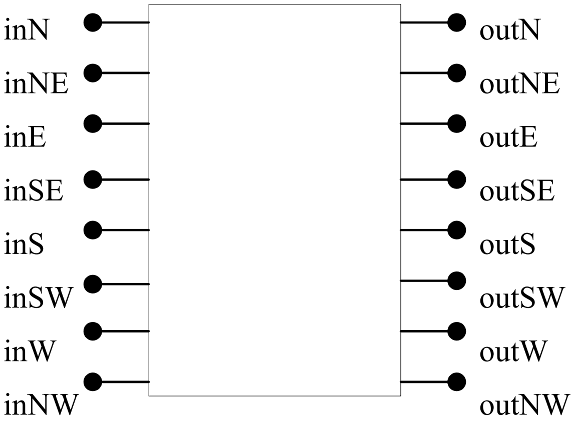


Figure 7. A forest fire cell atomic model

should be pointed out here that we are not currently using any wind flow algorithms that take into account the terrain in each forest cell to compute the effective wind speed. Nonetheless, if an implementation of such an algorithm is readily available, it can be easily incorporated into the *WindFlowModel*.

Similarly, the *FireFightingModel* is an atomic model that is connected to the *ForestFireCellSpace* and allows for rain or pouring of water on cells in the cell space. This model allows firefighting strategies to be modeled in that we can select cells on which to pour water based on the overall predicted maximum rate and direction of fire spread. For example, in our implementation, the model computes a random amount of time it would take for the cell to transition to the corresponding wet state. As mentioned before, this transition delay time is sampled from the exponential distribution, with a mean that is selected by the modeler.

The *ForestDisplay* is a two-dimensional graphical display that shows the spread of the forest fire. The following color code is used to represent the state of the cell:

- Green—*unburned*
- Blue—*unburned and wet*
- Pink—*starting to burn*
- Red—*burning*
- Yellow—*burning and wet*
- Gray—*burned and wet*
- Black—*burned*

A forest fire cell atomic model implemented by the *ForestFireCell* class is defined as shown in Figure 7. Each forest cell has eight inputs and eight corresponding outputs for neighbor-to-neighbor couplings.

In the two-dimensional implementation of the *TwoDimCellSpace*, the x-dimension (*xDim*) and the y-dimension (*yDim*) of the cell space must be given before the construction of the cell space. These dimensions give the number of cells along each of the two axes, and the total number of cells in the cell space is equal to *xDim* × *yDim*. The cells are arranged in a two-dimensional structure from bottom up, and each cell is assigned a unique ID. For example, if (*xDim*, *yDim*) = (5, 4), the cells (their IDs shown) are arranged as in Figure 8.

15	16	17	18	19
10	11	12	13	14
5	6	7	8	9
0	1	2	3	4

Figure 8. Arrangement of cells for a 5 × 4 two-dimensional cell space

Also, the length (l) and breadth (b) of the cell must be given, and we assume that all cells have the same size. Nonetheless, we can have cells of different sizes in the cell space. It should be pointed out here that the overall entity structure of the model is modular and allows for any of the component modules to be replaced by whatever the user has available as long as the appropriate messages are passed on to the input and external ports of the model. For example, the Rothermel's fire spread computation model can be replaced with another appropriate model. Similarly, one can design a wind flow model that uses specialized algorithms for computing the local wind speed and direction, and our model would still work. In the next subsection, we give hierarchical diagrams and provide a detailed explanation of the system operation.

3.2 Hierarchical Diagrams and Explanation of Operation

The system structure hierarchical diagram is given in Figure 9. The cell space is a coupled model composed of several cell atomic models coupled together according to the neighbor-to-neighbor coupling rules, as well as to the input and output ports of the cell space. Except for the boundary cells, each cell is coupled to the adjacent eight neighbors and has access to a fire spread computation model. In our case, we use Rothermel's fire spread model. Outside the cell space, we have the forest cell igniter, wind flow, and firefighting models with their output ports coupled to the input ports of the cell space for receiving external inputs. The forest cell igniter is outside the cell space and dictates

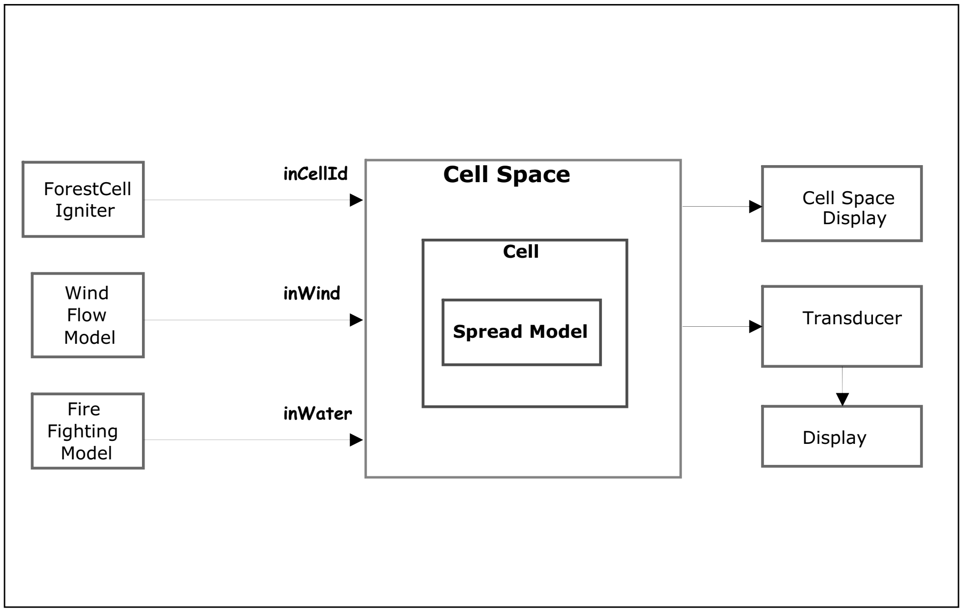


Figure 9. Forest fire model hierarchical diagram

what cell(s) to ignite in the cell space. The user can select the starting location of the fire by specifying a cell ID in the forest cell igniter model or can have it select a cell at random. The cell space display is simply a graphical user interface (GUI) that shows the state of each cell in the cell space following the predefined color code scheme given in section 3.1. The transducer allows the user to define an experimental frame and choose what parameters output by the cell space are of interest for simulation performance analysis and can display the results via the display model. The cell space sends information on the state of the cells to the transducer whenever a cell makes a transition. For example, one may be interested in computing and displaying the current average rate of spread in the major directions of spread based on the currently burning cells.

3.3 Phase Transitions, Inputs, and Outputs of the Forest Cell

The illustration in Figure 2 depicts the states transition for a forest cell. We start the simulation with all the forest cells in the initial state *unburned*. A cell could be ignited either by an *igniter* or by a burning neighbor cell. The transition from *burning* to *burning-wet* represents a cell that is cooling down with the fire not yet completely extinguished. The cell changes state from *burning-wet* to *burned-wet* after some random time determined according to the fire suppression rules. These phase transitions are implemented in DEVSJAVA by using the external, confluent, and internal transition functions. An illustration of the operation of these three DEVSJAVA transition functions is given in the appendix.

When a cell enters the *burning* state, fire spread within the cell is modeled as eight different spread components along the eight directions (Fig. 4). At this point, the components start spreading in all the eight directions at individual spreads given by Rothermel's equations for the given value of wind speed, wind direction, and fuel and slope parameters for the cell. For example, if the southwest (SW) spread component reaches the center of the SW neighbor first, then the SW neighbor will potentially be ignited first if the fireline intensity for that cell is above the threshold for the cell's fuel model. As pointed out before, a cell changes state from *burning* to *burned* when all the eight components have reached the centers of the corresponding neighboring cells. In terms of simulation, this means that all the time delay components have been consumed.

Once a cell enters the *burning* phase, the time t_i for a component in direction i ($i = N, NE, E, SE, S, SW, W,$ and NW) to reach the center of the associated neighbor cell is calculated using the following simple equation:

$$t_i = \frac{d_i}{R_i}, \qquad (1)$$

where d_i is the distance from the cell center to the neighbor cell center in the direction i, and R_i is the rate of spread in the direction i, which is a component of the maximum rate of spread computed by Rothermel's model. Let us now consider the influence of changes in wind speed and direction on cells in the *burning* phase. Here we simply require that the cell update its rates of spread based on the current weather conditions for the spread components that are still active. Otherwise, no update is made. Let us assume that a spread component i has covered a distance $d$out of a total distance d_i when new wind speed and direction values are input to the cell. At this point, the new spread R_i^{new} is computed from the maximum rate of spread and direction given by Rothermel's model with current cell weather conditions (wind speed and direction). This allows the model to be dynamic by adapting to changes in weather conditions as the forest fire propagates. The remaining time, t_i^{new}, for the component to reach the center of the neighbor cell is now computed over the remaining distance, $d_i - d$, as given in equation (2).

$$t_i^{new} = \frac{d_i - d}{R_i^{new}}. \tag{2}$$

The remaining time, t_i^{new}, depends on the new rate of fire spread computed based on the current prevailing weather conditions.

4. The Forest Fire Experimental Frame

To meet the basic objective of simulating forest fire spread dynamically in an optimized cell space, we implemented a forest fire experimental frame as shown in Figure 10. The experimental frame allows us to specify the type of experimentation that would enable us to obtain answers to the questions of interest [20, 21]. The forest fire experimental frame is composed of the transducer, the cell space display and parameter displays, the *ForestCellIgniter*, the *WindFlowModel*, and the *FireFightingModel*. The experimental frame is coupled to the cell space. The external influences on the cell space are initiated by the experimental frame via the forest cell igniter model, wind flow model, and firefighting model. The output parameters of interest from the forest cell space are fed to the experimental frame.

In the experimental frame, the transducer computes the average rate of spread based on the burning forest cells, simulation performance measures, and any other measures that may be of interest. The transducer also computes the ratio of active cells to the total number of cells in the cell space and the ratio of active cells to the total number of cells created to analyze cell space optimization performance. Throughout the simulation, the cell space display shows the status of the entire cell space in terms of states and provides a tool for analyzing fire spread behavior due to changes in wind speed and direction and the effect of pouring coolant on the forest cells.

The computation of the average rate of spread and direction is done as follows. Whenever a cell transitions from *unburned* to *burning* or is in the *burning* state and receives a new wind speed and direction value from the wind flow model, it transmits its cell ID, the maximum rate of spread, and direction to the transducer as an entity implemented as *spreadEnt*. Also, when a cell changes state, it sends a notification message to the transducer. Therefore, the transducer can compute the average rate of spread and direction based on the cells in the burning state. Let n be the number of cells currently in the burning state, $s_{\max}^j$, and let θ_j be the maximum rate of spread and direction for cell j. Then, the average rate of spread (R_{aver}) and direction (dir_{aver}) are computed according to equation (3) and equation (4), respectively.

$$R_{aver} = \sum_{j=1}^{n} s_{\max}^j, \tag{3}$$

$$dir_{aver} = \sum_{j=1}^{n} \theta_j. \tag{4}$$

The average rate of spread and direction are updated according to the cell transition during the simulation run and give a general rate of spread and direction for the entire cell space.

5. Simulation Experiments and Results

To test and validate the operation of our forest fire spread simulation model, we conducted several experiments. These experiments were aimed at meeting our initial objective of simulating forest fire spread and suppression under dynamic and nonuniform conditions using the DEVS cell space approach. The first experiment tests the effect of constant wind speed and direction on fire spreading in homogeneous fuel and slope. The second experiment simulates fire spread in a homogeneous forest under varying wind speed and direction. The third experiment simulates fire spread in a forest with varying fuel models and nonuniform topography. Finally, the fourth and fifth experiments consider suppression by simulating the influence of water or rain (firefighting) on fire spread behavior. In all the experiments, we use a total of 900 (15 × 15 meters) forest cells with a square forest cell space of 30 × 30 cells. To initiate the forest fire, we arbitrarily ignite one cell in the cell space in all the experiments. Table 1 gives the name and brief description of each of the 13 NFFL fuel models.

All the experiments were conducted on a 1.8-GHz PC with 256 MB RAM running DEVSJAVA with SimView DEVSJAVA Simulation Viewer Version 1.0.4. SimView allows the experimenter to view all the simulation models with input and output ports together with the corresponding couplings among the various models. Furthermore, SimView has a number of convenient functionalities such as allowing the user to stop and start the simulation at any time during a simulation run, fast-forwarding or slowing down the simulation, and being able to input the appropriate parameters into any model by simply clicking on a model's input port and selecting the desired option from a pop-up menu. Also, the inherent parameters of a model can

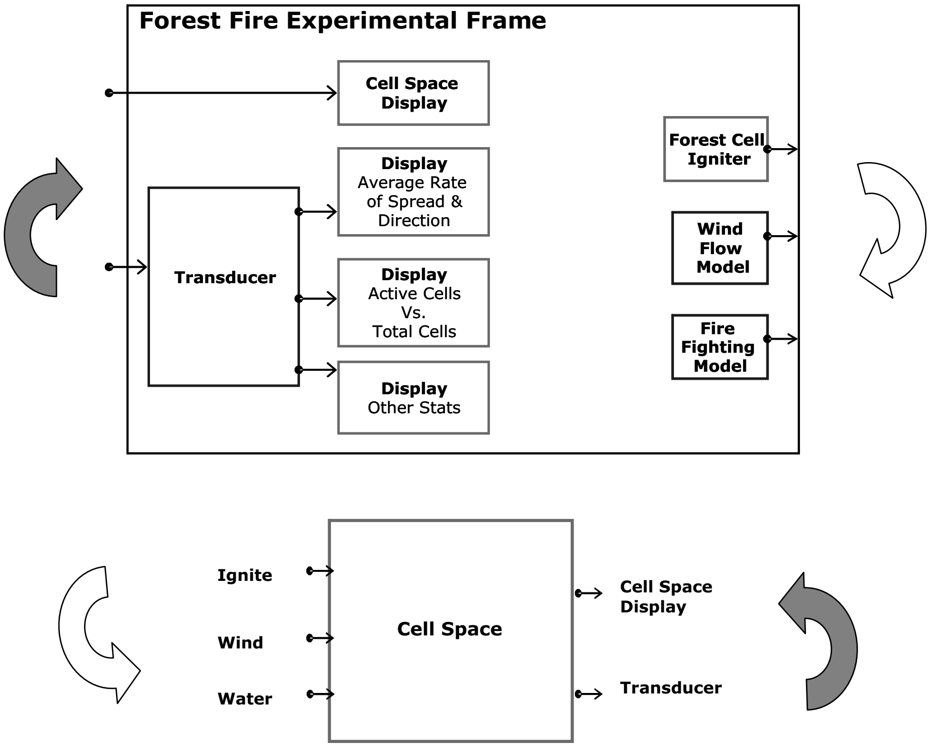


Figure 10. The forest fire experimental frame

Table 1. The 13 NFFL fuel models

NFFL-Fuel-Model 1: Short grass (1 ft)
NFFL-Fuel-Model 2: Timber (grass and understory)
NFFL-Fuel-Model 3: Tall grass (2.5 ft)
NFFL-Fuel-Model 4: Chaparral
NFFL-Fuel-Model 5: Brush (2 ft)
NFFL-Fuel-Model 6: Dormant brush, hardwood slash
NFFL-Fuel-Model 7: Southern rough
NFFL-Fuel-Model 8: Closed timber litter
NFFL-Fuel-Model 9: Hardwood litter
NFFL-Fuel-Model 10: Timber (litter and understory)
NFFL-Fuel-Model 11: Light logging slash
NFFL-Fuel-Model 12: Medium logging slash
NFFL-Fuel-Model 13: Heavy logging slash

be viewed on a pop-up menu by simply positioning the cursor on the model. Figure 11 gives a picture of the SimView window with some of the forest fire simulation model components shown. To run a given simulation model, the user simply needs to select the appropriate model from a menu bar and then click run. The simulation clock is also shown on the SimView window.

5.1 Experiments

Our experiments were conducted on the standard fuel models NFFL-4 (chaparral), NFFL-7 (southern rough), and NFFL-11 (light logging slash). These fuel models were arbitrarily chosen for their differences in fuel loadings, and they exhibit varied fire spread behaviors. We first conducted preliminary experiments to study the effect of extreme wind speed conditions and extreme terrain conditions, respectively, on the rate of spread. The results of these experiments are given in Figures 12 and 13. To study the effect of extreme wind speed conditions on fire spread, we set the slope to zero and had the wind blow northward and then varied wind speed from 0 to 10 m/sec. As shown in Figure 12, the rate of spread increases as expected, with an increase in wind speed for the three fuel models. The rate of spread in NFFL-4 is highest, followed by NFFL-7 and NFFL-11, in that order. Indeed, the rates of spread we obtained for NFFL-4 fall within 10% of the rates of spread reported by Morais [22, 23], who has extensively studied fire spread in NFFL-4 and *Ceanothus* chaparral and uses a continuous-based simulation software called HFire (Highly Optimized Tolerance Fire Spread Model). HFire is a new raster-based spatially explicit model of surface fire spread through Southern California chaparral, written in the C programming language. The results obtained with HFire are compatible with field measurements and observations made for the validation of HFire and are thus usable for the validation of our model.

To study the effect of extreme terrain conditions on fire spread, we set the wind speed to zero and varied the slope

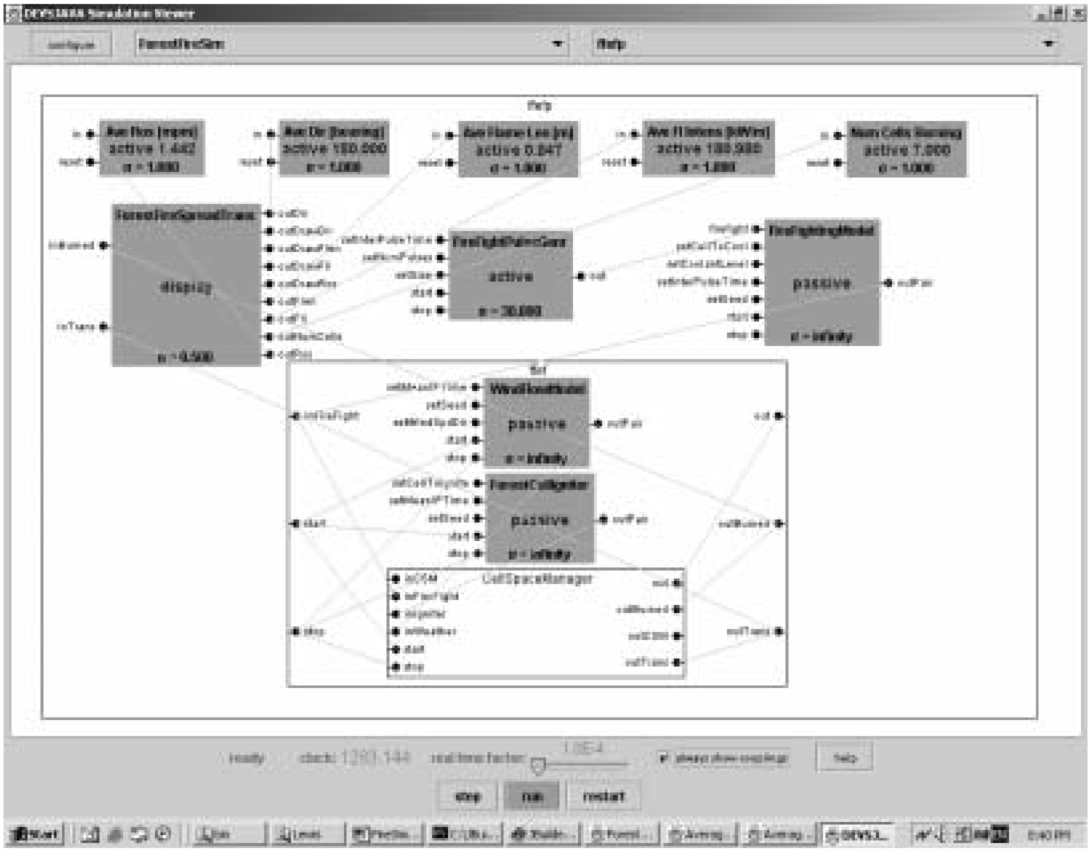


Figure 11. An example appearance of the SimView window

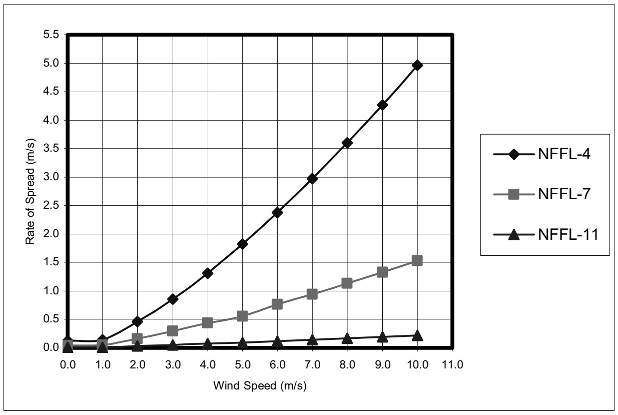


Figure 12. Rate of spread for varying wind speed conditions

(rise/run) from 0 to 1. The results (Fig. 13) show that the rates of spread in all the three fuel models increase with a rise in slope. The rate of increase is highest in NFFL-4, followed by NFFL-7 and NFFL-11, in that order. In this case also, the results obtained for NFFL-4 are within 10% of those reported by Morais [22, 23] for all the points on the graph. The rate-of-spread results for NFFL-4 under extreme wind and slope conditions, respectively, are in close agreement with those observed, simulated, and reported in Morais [22, 23]. We were unable to find rate-of-spread results in the literature for the other two NFFL models under the described or similar experimental conditions. The discrepancy between our results and those obtained by HFire may be attributed to the fundamental differences between the two approaches. However, it is possible to fine-tune or parameterize our model so that it mimics HFire. Next we report on the results obtained from five different fire spread experiments to test the operation of our simulation model.

5.1.1 Experiment 1: Fire Spread under Constant Wind Speed and Direction

In this experiment, the wind speed was arbitrarily fixed at 8.048 kmh or 2.2352 m/sec, and the corresponding wind direction was fixed at 0 degrees or north. In all the experiments, all the parameters were taken as deterministic, and no sampling from any probability distribution was done for

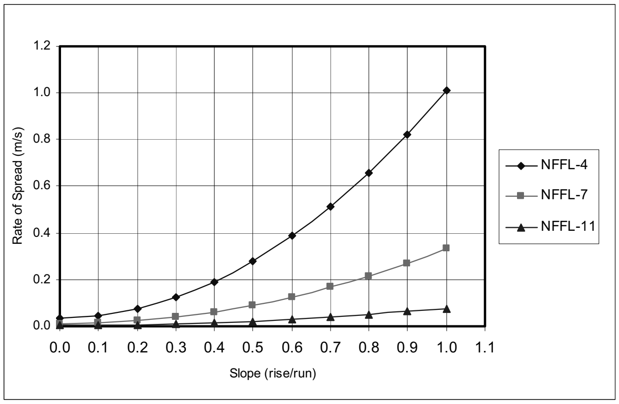


Figure 13. Rate of spread for varying terrain conditions

Table 2. Average spread parameter values for experiment 1

Rate of Spread (m/sec)	Direction (Bearing)	Flame Length (m)	Fireline Intensity (kW/m)
0.5459	0.000	7.5811	21233.7218

lack of information on the appropriate probability distributions to use. In this experiment, the terrain slope was fixed at zero, while the forest fuel model NFFL-4 (chaparral) was used for all the forest cells. We did not adopt the standard meteorological definition for wind direction, which has the convention that wind direction is the direction where the wind is blowing "from." Instead, we state wind direction as the direction where the wind is blowing "to." Figure 14 gives the fire propagation results of the simulation, with the simulation clock given for each diagram. The first diagram shows the initial fire ignition point.

In the diagrams, unburned cells are shown in green, burning cells in red, and burned cells in black. The forest cells that are just starting to burn are shown in pink or light red. As can be seen in the diagrams, the fire spreads faster northward, along the direction of wind flow, and slower southward. The average parameters for this experiment, as computed by the simulation transducer model, are given in Table 2. The rate of spread, flame length, and fireline intensity are fairly high for this fuel model.

These values remain constant throughout the simulation due to the uniformity of the cell vegetation and environmental conditions across all cells. Again, the rate of spread under these conditions is comparable to that obtained by Morais [22] using the HFire simulation.

5.1.2 Experiment 2: Varying Wind Speed and Direction

In this experiment, the terrain slope was also fixed at zero while the forest fuel model NFFL-4 (chaparral) was used for all the cells. The wind speed and direction were varied two times during the experiment. The experiment was initiated with a wind speed of 4.0233 kmh (1.1176 m/sec) in the direction 0 degrees (N) and then changed to a wind speed of 8.048 kmh (2.2352 m/sec) in the direction 45 degrees (NE) at simulation clock 1100 seconds. The evolution of the fire is given in Figure 15.

Table 3. Average spread parameter values for experiment 2

Clock (sec)	Rate of Spread (m/sec)	Direction (Bearing)	Flame Length (m)	Fireline Intensity (kW/m)
729	0.1715	0	4.4506	6670.4770
1009	0.1715	0	4.4506	6670.4770
1157	0.3819	26.9009	6.1789	14856.7053
1462	0.5376	44.7916	7.4568	20909.6636
2003	0.5363	45.5222	7.4418	20816.7741

The fire initially spreads northward at a slower pace and picks up speed after 1100 seconds and changes course heading toward the NE, as influenced by the dynamic change in wind direction. With an increase in wind speed, we see more cells catching fire in the new wind direction. The average spread parameters for this experiment are given in Table 3.

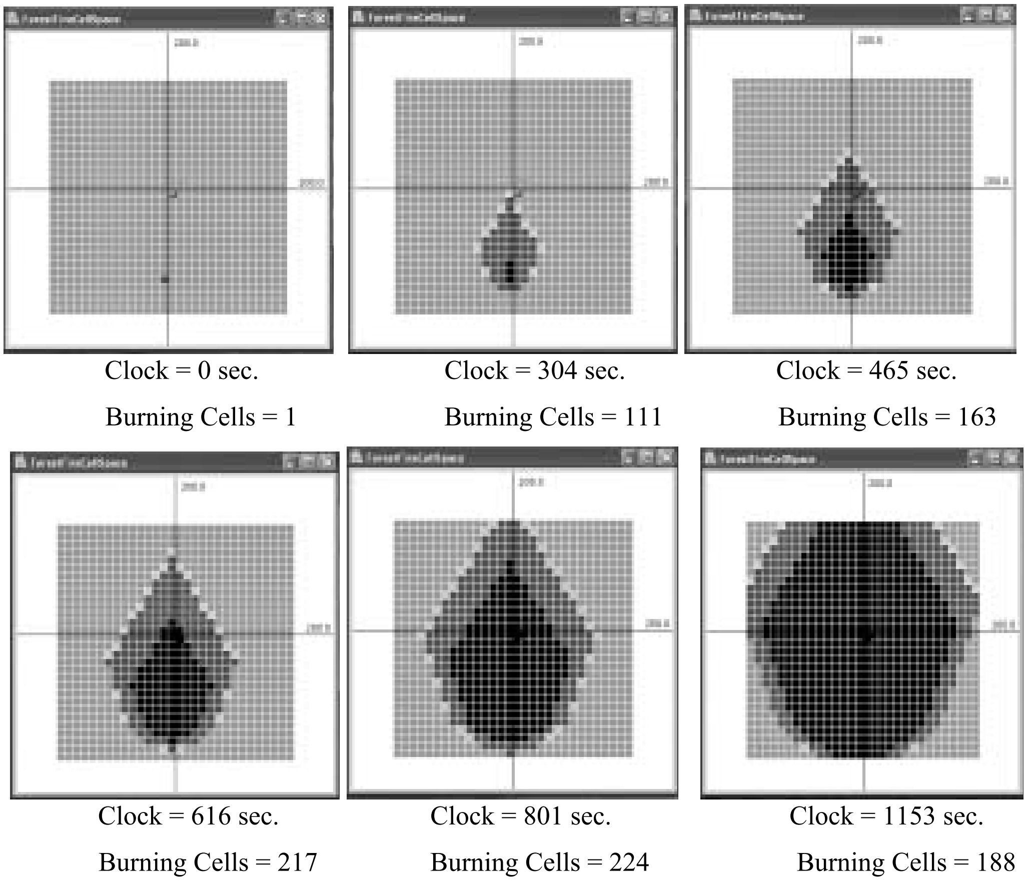
There is an increase in the rate of spread, flame length, and fireline intensity corresponding to the increase in wind speed. Similarly, we see a change in the spread direction from 0 degrees going about 45 degrees (NE). These results demonstrate how the simulation model adapts to a dynamic change in weather conditions (wind speed and direction).

5.1.3 Experiment 3: Fire Spread in a Nonuniform Forest (Across a Valley)

In this experiment, we simulate fire behavior across a valley in a forest composed of three different fuel models and varying topography, as shown in Figures 16 and 17. Each block in the forest is a 10 × 30 cell space composed of 300 forest cells, and the NFFL fuel model number and slope are shown in each block. In the graphical display (Fig. 18), the three fuel models are represented by the different shades of green.

Here our goal is to simulate a fire that crosses a valley that lies E-W from the south toward the north. The wind blows northward down the declining side of the valley, then the flat portion of the valley, and finally blows uphill. The wind speed and direction for this experiment were kept as in the first experiment at 8.048 kmh (2.2352 m/sec) in the north direction (0 degrees). The evolution of the fire is shown in Figure 18.

First, the fire propagates down the slope through NFFL-11 (light logging slash) to the bottom section of the forest with NFFL-4 (chaparral), where it starts spreading sideways. Meanwhile, the fire steadily climbs up the top sec-



Clock = 0 sec. Burning Cells = 1

Clock = 304 sec. Burning Cells = 111

Clock = 465 sec. Burning Cells = 163

Clock = 616 sec. Burning Cells = 217

Clock = 801 sec. Burning Cells = 224

Clock = 1153 sec. Burning Cells = 188

Figure 14. Fire propagation results for experiment 1: Constant wind speed and direction

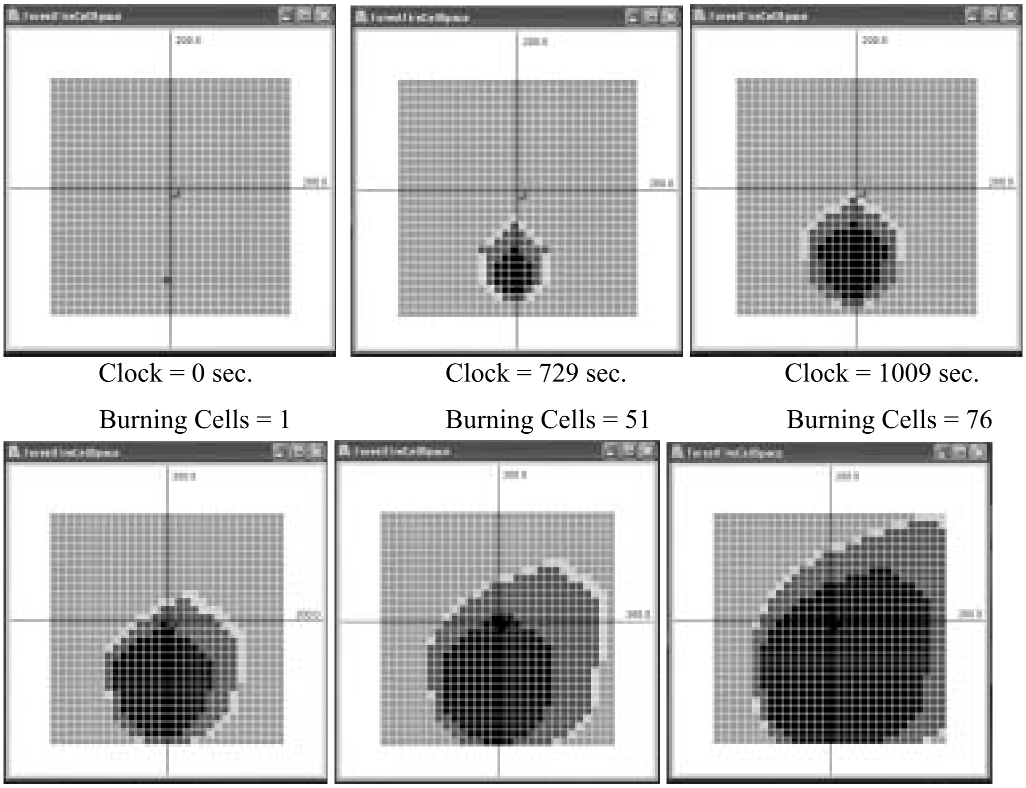
tion of the forest through NFFL-7 (southern rough), where it picks up the pace and spreads across the entire section. Here we see the fire spreading quickly in the second section to catch fire and then the last. This is because the middle section has the fuel model with the highest fuel load, while the last section has a high fuel load model coupled with wind in the uphill direction. Fire spread is slowest in the first section with the intermediate load fuel type and where the wind blows downhill. Indeed, this is what should be expected—a slow backing fire in an intermediate fuel type, a faster fire spread in the heavier fuel load model, and an intermediate situation. The average spread parameters for this experiment are given in Table 4.

These results show an increase in the values and then a decrease toward the end. This corresponds to the fire spreading across the three different fuel models under different terrain conditions coupled with limited burnable space. Again, our simulation model demonstrates the capability of simulating fire spread under nonuniform terrain conditions.

5.1.4 Experiment 4: Fire Spread under the Influence of Water in a Uniform Forest

The fourth experiment tests the influence of water or rain on fire spread behavior. In this experiment, the wind speed and direction were again set at 8.048 kmh (2.2352 m/sec) in the north direction (0 degrees). The forest fuel model NFFL-4 (chaparral) was used. We considered a firefighting scenario in which water is poured on the forest cells from north to south against the major spread direction at 0.2679 m/sec. Here we allow cells to transition from a dry state to a wet state according to the firefighting rules defined in section 2.3. We arbitrarily sample from the uniform distribution with α_1 = *uniform* (0, 0.5) and α_2 = *uniform* (0.5, 1.0) for the first and second cases of the fire suppression rules, respectively. Figure 19 shows the experimental results.

Here we see how a fire is attacked head-on by water (or rain), and the unburned cells transition to blue (*unburned and wet*) and never catch fire. In this scenario, the fire



Clock = 0 sec. Burning Cells = 1

Clock = 729 sec. Burning Cells = 51

Clock = 1009 sec. Burning Cells = 76

Clock = 1157 sec. Burning Cells = 142

Clock = 1462 sec. Burning Cells = 258

Clock = 2003 sec. Burning Cells = 226

Figure 15. Fire propagation results for experiment 2: Varying wind speed and direction

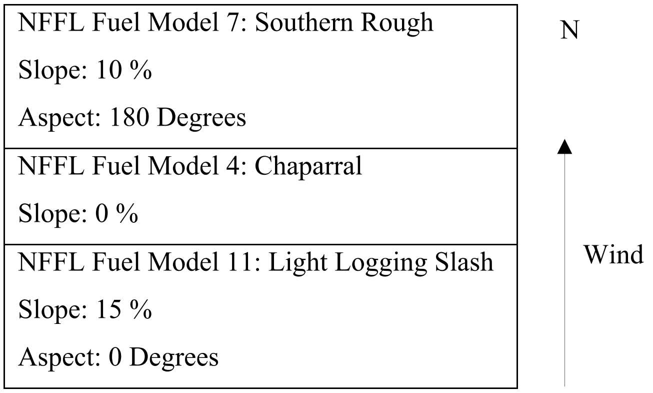

NFFL Fuel Model 7: Southern Rough
Slope: 10 %
Aspect: 180 Degrees
NFFL Fuel Model 4: Chaparral
Slope: 0 %
NFFL Fuel Model 11: Light Logging Slash
Slope: 15 %
Aspect: 0 Degrees
N
Wind

Figure 16. Forest cell space with nonuniform parameters

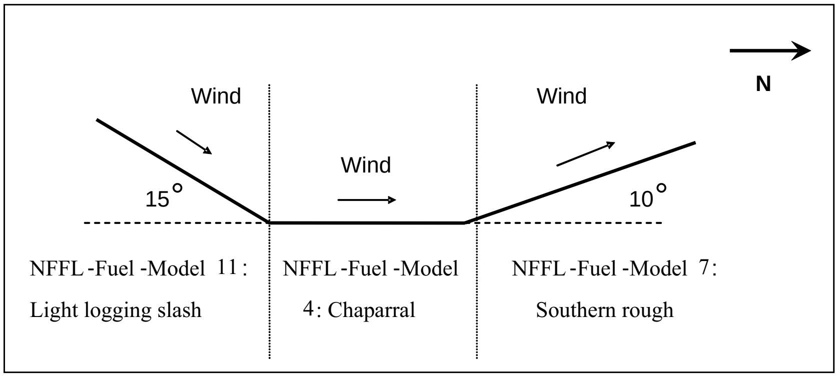
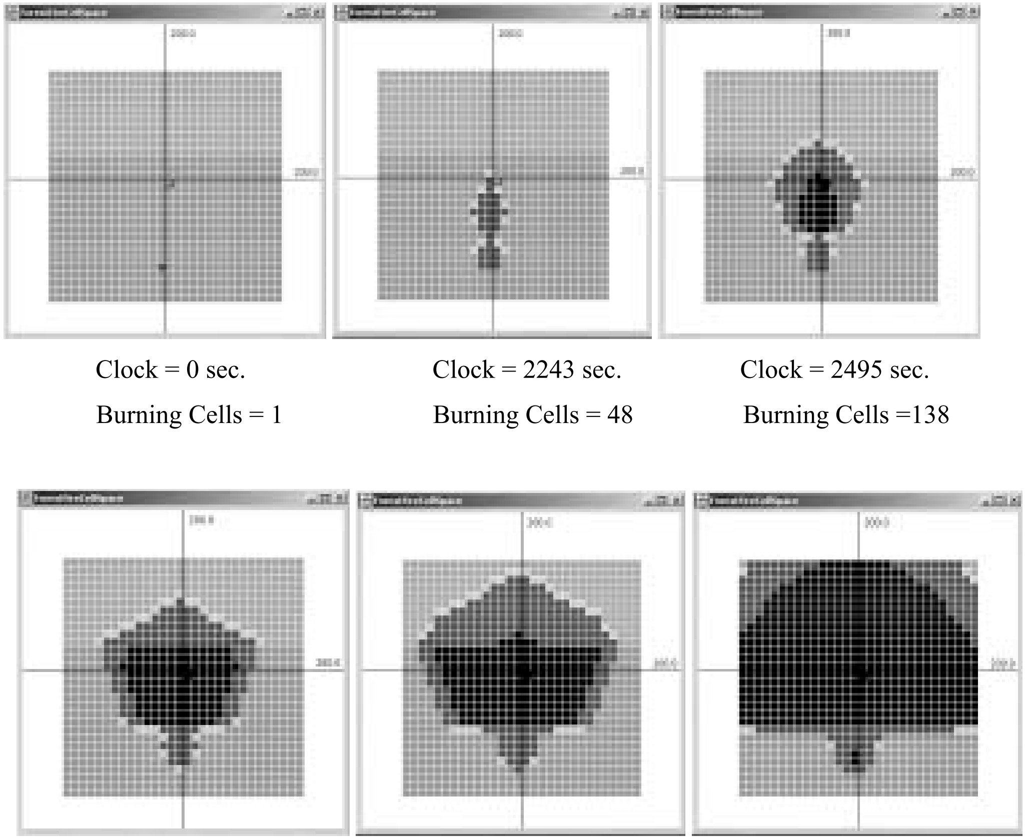


Figure 17. Forest terrain with nonuniform parameters



Clock = 0 sec.
Burning Cells = 1

Clock = 2243 sec.
Burning Cells = 48

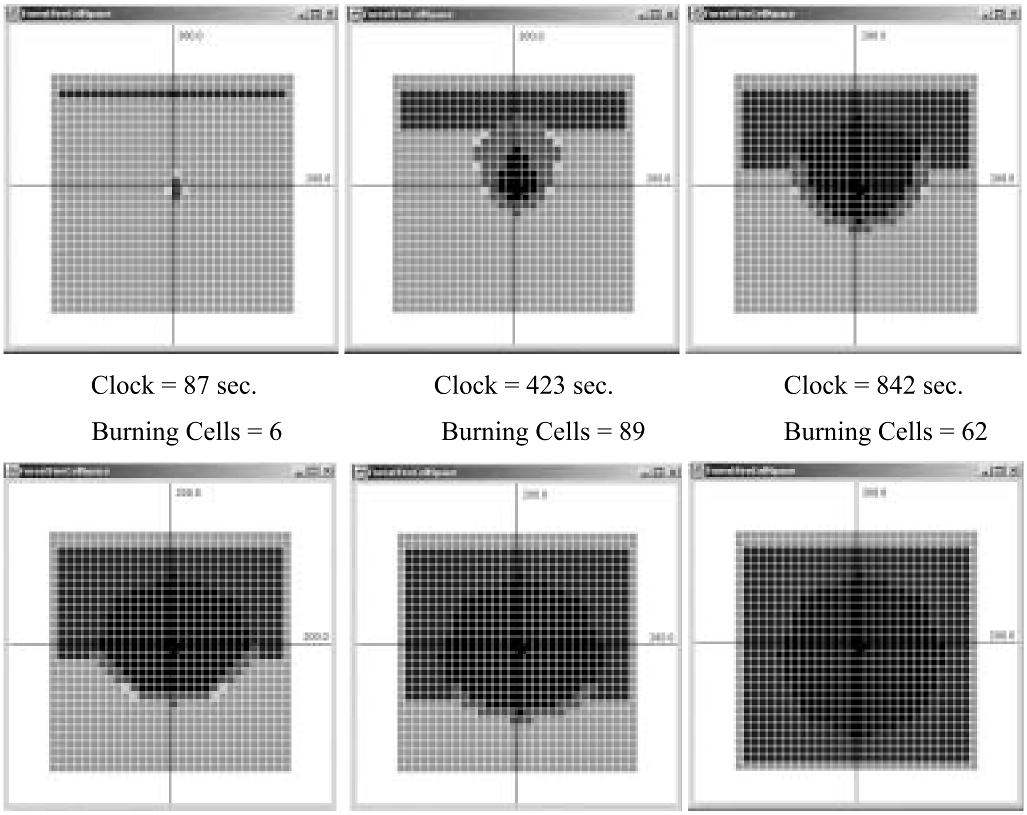
Clock = 2495 sec.
Burning Cells =138

Clock = 2811 sec.
Burning Cells = 203

Clock = 3094 sec.
Burning Cells = 285

Clock = 3972 sec.
Burning Cells = 199

Figure 18. Fire propagation results for experiment 3: A nonuniform forest cell space



Clock = 87 sec.
Burning Cells = 6

Clock = 423 sec.
Burning Cells = 89

Clock = 842 sec.
Burning Cells = 62

Clock = 1178 sec.
Burning Cells = 33

Clock = 1599 sec.
Burning Cells = 26

Clock = 2271 sec.
Burning Cells = 0

Figure 19. Fire spread under the influence of water

Table 4. Average spread parameter values for experiment 3

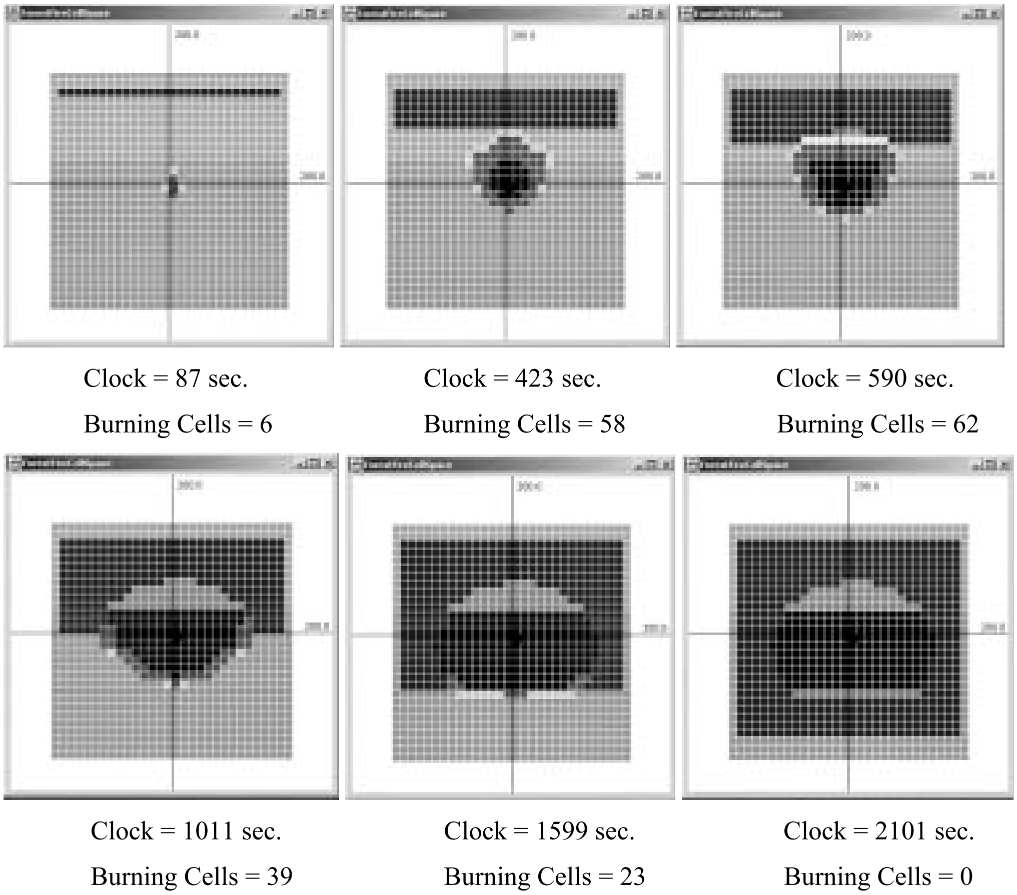
Clock (sec)	Rate of Spread (m/sec)	Direction (Bearing)	Flame Length (m)	Fireline Intensity (kW/m)
729	0.1715	0	4.4506	6670.4770
1009	0.1715	0	4.4506	6670.4770
1157	0.3819	26.9009	6.1789	14856.7053
1462	0.5376	44.7916	7.4568	20909.6636
2003	0.5363	45.5222	7.4418	20816.7741

cannot be put off by pouring water on the burning cells, and the burning cells change color from red to black after burning. This is because the fuel model NFFL-4 is high in fuel, and the flame lengths are well above 2.4 meters and are as high as 7.5811 meters (see Table 2). According to our fire suppression rules, such a fire cannot be contained by any direct attack means. Thus, by pouring water on cells before they catch fire, we are able to contain the fire.

5.1.5 Experiment 5: Fire Spread under the Influence of Water in a Nonuniform Forest

The final experiment tests the influence of water or rain on fire spread behavior in a nonuniform forest described in experiment 3. As in the previous experiment, the wind speed and direction were set at 8.048 kmh (2.2352 m/sec) in the north direction (0 degrees), and we consider a firefighting scenario in which water is poured on the forest from north to south against the major spread direction at 0.2679 m/sec. Again, we set α_1 = *uniform* (0, 0.5) and α_2 = *uniform* (0.5, 1.0). The experimental results are shown Figure 20.

Unlike the previous experiment with the same fuel model with flame lengths well over 2.4 meters, here we see fire being put out in upper and lower section fuel models but not the middle one. The middle fuel model (NFFL-4) has a flame length greater than 2.4 meters, as shown in the previous experiment, and the fire cannot be put out by control efforts at the head of the fire. However, indirect attack (cooling the forest cells before they catch fire) seems



Clock = 87 sec.
Burning Cells = 6

Clock = 423 sec.
Burning Cells = 58

Clock = 590 sec.
Burning Cells = 62

Clock = 1011 sec.
Burning Cells = 39

Clock = 1599 sec.
Burning Cells = 23

Clock = 2101 sec.
Burning Cells = 0

Figure 20. Fire propagation results for experiment 5: Influence of water or rain on fire spread in a nonuniform forest

to be the only means of suppression. The upper and lower section fuel models, NFFL-11 and NFFL-7, have flame lengths well below 2.4 meters, and thus the fire is contained by pouring water on the cells. In this experiment, we see how the simulation model can provide a platform for experimenting with different fire suppression rules.

5.1.6 Simulation Efficiency Parameters

Finally, in Table 5, we give the ratio of total active cells (burning cells) to the total number of cells in the cell space for all the experiments. The first column of Table 5 shows the plate number corresponding to the figure shown in the fire evolution results in Figures 14 through 20. The rest of the columns show the ratios for each experiment. The ratios are generally very small, an indication that very few cells are active (actually burning) at any point in time during the simulation. In fact, the results show that only less than 32% of the cells in the cell space are actually active at any point in time. The importance of these ratios in discrete event simulation is that they are a factor in determining simulation efficiency since they are a strong indication of the average number of *imminents*. The simulation engine used in our simulation model takes advantage of the small number of *imminents* to decrease simulation time significantly.

We should also mention that simulating larger cell spaces (with more than 1000 cells) resulted in memory problems for the computer platform we used. This is because we create and instantiate all the cells in the cell space all at once before starting the simulation. To avoid memory problems, one can consider dynamic creation and removal of cells from the cell space since only a small fraction of cells are needed at any point in time during the simulation. However, this may have an impact on the simulation speed due to the time needed to instantiate and create cell neighbor-to-neighbor couplings. Also, one can resort to a distributed simulation implementation. We leave these avenues for our future research due to the implementation issues associated with these approaches.

6. Discussion and Conclusions

In this article, we have presented the conceptualization, modeling, and simulation of forest fire spread and suppression using the DEVS cell space approach. The event-based

Table 5. Ratio of number of burning cells to total cells in the cell space

Plate Number	Experiment 1	Experiment 2	Experiment 3	Experiment 4	Experiment 5
1	0.0011	0.0011	0.0011	0.0067	0.0067
2	0.1233	0.0567	0.0533	0.0989	0.0644
3	0.1811	0.0844	0.1533	0.0689	0.0689
4	0.2411	0.1578	0.2256	0.0367	0.0433
5	0.2489	0.2867	0.3167	0.0289	0.0256
6	0.2089	0.2511	0.2211	0.0000	0.0000

modeling approach enables efficient simulation, especially for the cell space application considered in this study. Realizing the fact that in fire spread simulation, computations are concentrated along the fire front, we adopt an approach in which, at a given instant, only cells that correspond to burning zones are active in the DEVS cell space. Also, our model considers nonuniform fire spread parameters to address the issue of spatial/temporal variability of the forest fire spread variables and allows computation and message exchanges to occur only in active cells. The model dynamically predicts average fire spread as the fire propagates and adapts to changes in wind speed and direction.

The domain of validity of the Rothermel model is determined by its assumptions and is described in the literature [11, 12]. The domain is, strictly speaking, for steady-state surface fires under homogeneous conditions in space and in time (fuel, weather, and topography)—conditions for which the model has been validated [24, 25]. By letting each cellular region in space have its specific parameters, the conditions of spatial homogeneity are met within each cell. Moreover, since the calculations are performed based on a discrete event systems specification, there is homogeneity of parameters between events. Consequently, the calculations performed within each piecewise constant portion of our calculation space are valid. Furthermore, we assume that once a cell is ignited, it reaches the steady-state conditions of the Rothermel model in a manner that employs external events to account for the effects of its neighbors.

To validate a model based on this piecewise constant space/time approach, we must compare its predictions with reality or with the results of validated process-based models. Comparisons with real fire data first appeared in Vasconcelos, Zeigler, and Pereira [26], where predictions were compared with the behavior of one real forest fire, and an agreement within 10% was found. In the work presented here, we conducted several more experiments, and the results matched our expectations as well as those in the literature. In particular, the rate-of-spread results for NFFL-4 under increasingly severe wind and slope conditions, respectively, are very similar to what is reported in the literature for field observations. Some limitations of the approach, however, are to be expected. Rothermel's model is designed for surface fires; thus when a transition to large fire behavior occurs, the assumptions of the Rothermel model are no longer met, and the rates of spread calculated within each cell may no longer be valid.

The work presented indicates that the model has potential for providing essential information needed in tactical decision making for effective forest fire suppression and control. Our future work includes the extension of the current forest fire spread simulation model to support real-time or as-fast-as-can tactical decision making for fire control and suppression. In particular, we plan on extending the current forest cell atomic model into a coupled model that will allow for the addition of "agents" to the simulation model. By *agents*, we mean something that can act on the fire spread so as to control it, such as firefighting equipment or, indeed, firefighters. Furthermore, we plan on incorporating fire spread models other than the Rothermel model that consider fire spread under nonuniform conditions and account for back-burning, spotting, and crowning.

Finally, we plan on implementing the simulation model in a distributed setting. This would allow for simulations with a large number of cells or simulations with different geographically spaced forest cells to be implemented much more efficiently. Simulations that allow for a large number of cells mean that the forest cell space can be represented with much smaller size cells and would result in more accuracy in the representation of the forest fire model input parameters. Nevertheless, the forest cell size also depends on the fire spread model being used. Finally, validation of the models must be performed for particular forest applications by testing its predictions with those of observed or prescribed forest fire events.

7. Appendix

7.1 Transition Functions and Illustrative DEVSJAVA Code

In this appendix, we give an illustration of DEVSJAVA code by highlighting the operation of the external, confluent, and internal transition functions.

7.2 External Transition Function

Figure 21 gives an illustration of the external transition function with the DEVSJAVA illustrative code given

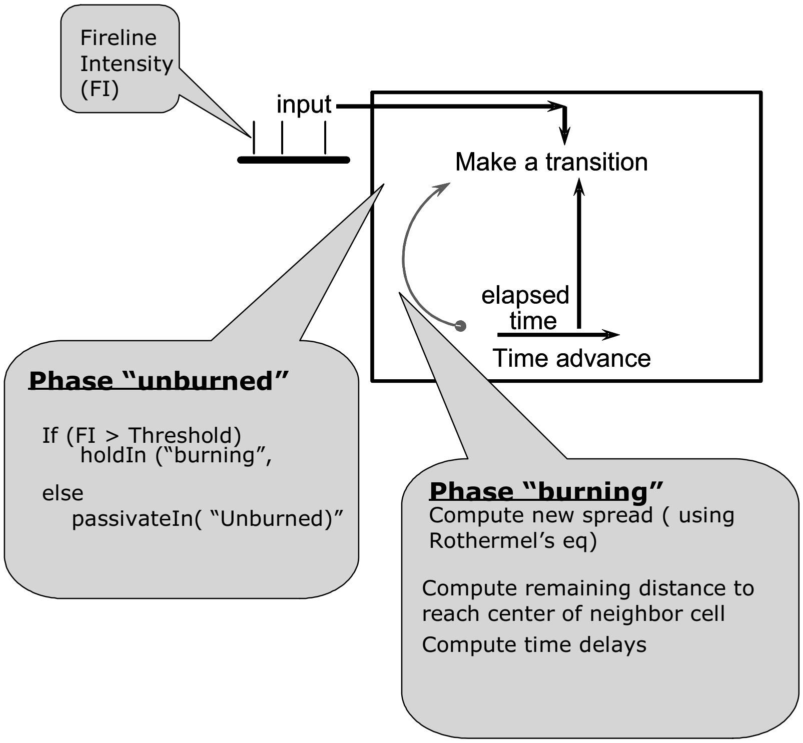


Figure 21. External transition function

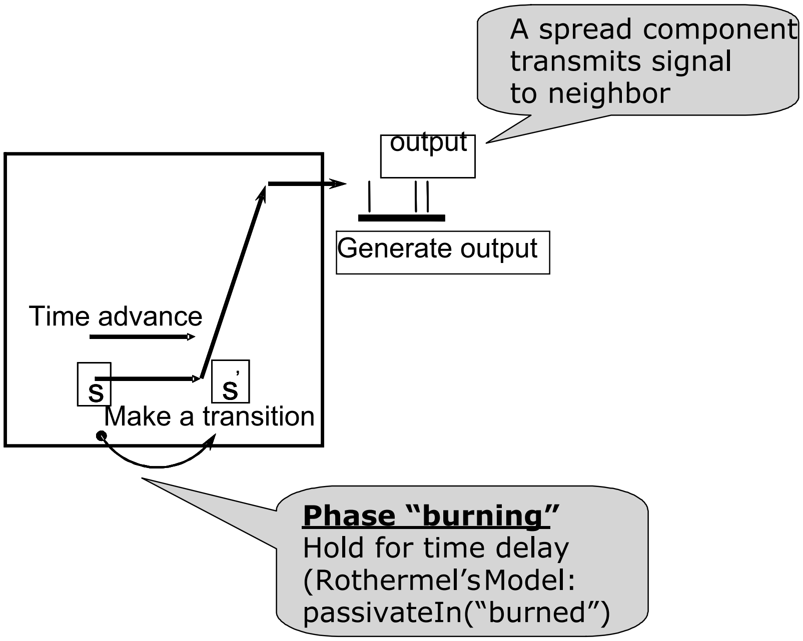


Figure 22. Internal transition function

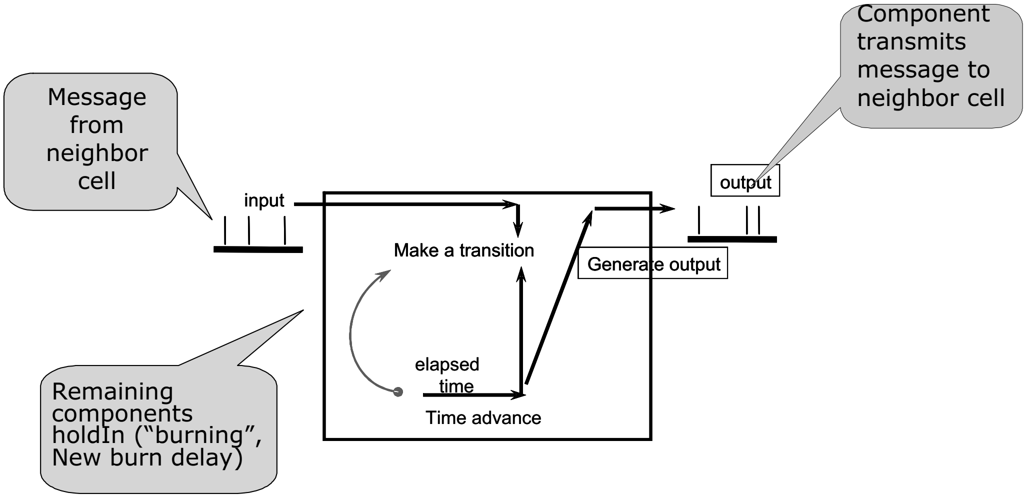

Message from neighbor cell
input
Make a transition
elapsed time
Time advance
Generate output
output
Component transmits message to neighbor cell
Remaining components holdIn ("burning", New burn delay)

Figure 23. Confluent transition function

describing the operation of the external transition function. A cell transitions into another state if the external input message satisfies a given condition.

7.3 Internal Transition Function

A diagrammatic description of the internal transition function is given in Figure 22. A cell outputs a message to the neighbor cell if the corresponding spread component reaches the center of the neighbor cell.

7.4 Confluent Transition Function

Figure 23 gives an illustration of the confluent transition function. The confluent transition function decides the function to execute when an input arrives at the port of the cell atomic model when a spread component is imminent. In our simulation model, the confluent function gives preference to the internal transition function over the external transition function. That is, we first output the message and then process the input message.

As shown in Figure 23, an input arrives when a spread component is imminent, that is, when its time delay is just becoming zero. Thus, the component will transmit a signal to the neighbor cell associated with the component before the cell atomic model processes the input message: recomputing the rate of spread in all the remaining spread directions, and thus the new time delays are set as described in the external transition function.

8. References

[1] Andrews, P. L., and L. P. Queen. 2001. Fire modeling and information system technology. *International Journal of Wildland Fire* 10:343-52.

[2] Zeigler B. P., H. Praehofer, and T. G. Kim. 2000. *Theory of modeling and simulation*. 2nd ed. New York: Academic Press.

[3] Wainer, G., and N. Giambiasi. 2001. Timed cell-DEVS: Modeling and simulation of cell spaces. In *Discrete event modeling & simulation: Enabling future technologies,* edited by H. S. Sarjoughian, 187-213. New York: Springer-Verlag.

[4] Zeigler, B. P. 2003. Discrete event abstraction: An emerging paradigm for modeling complex adaptive systems. In *Advances in adaptive complex systems,* edited by L. Booker. Oxford, UK: Sante Fe Institute/Oxford Press.

[5] Ameghino, J., A. Troccoli, and G. Wainer. 2001. Models of complex physical systems using Cell-DEVS. In *Proceedings of the Annual Simulation Symposium,* Seattle, WA.

[6] Muzy, A., G. Wainer, E. Innocenti, A. Aiello, and J. F. Santucci. 2002. Comparing simulation methods for fire spreading across a fuel bed. In *Proceedings of AIS'2002,* Lisbon, Portugal.

[7] Hu, X., and B. P. Zeigler. 2004. A high performance simulation engine for large-scale cellular DEVS models. In *High Performance Computing Symposium (HPC'04), Advanced Simulation Technologies Conference*.

[8] Zeigler, B. P., and H. Sarjoughian. 2002. *Introduction to DEVS modeling and simulation with JAVA: A simplified approach to HLA-compliant distributed simulations*. Tucson: University of Arizona. Accessed from http://www.acims.arizona.edu

[9] Muzy, A., E. Innocenti, J. F. Santucci, and D. R. C. Hill. 2003. Optimization of cell spaces simulation for the modeling of fire spreading. In *Proceedings of the 36th Annual Simulation Symposium*, Orlando, FL, pp. 289-96.

[10] Vasconcelos, J. M. 1993. Modeling spatial dynamic ecological processes with DEVS-scheme and geographical information systems. Ph.D. diss., Department of Renewable and Natural Resources, University of Arizona, Tucson.

[11] Rothermel, R. 1972. A mathematical model for predicting fire spread in wildland fuels. Research Paper INT-115, U.S. Department of Agriculture, Forest Service, Intermountain Forest and Range Experiment Station, Ogden, UT.

[12] Andrews, P. L. 1986. BEHAVE: Fire behavior predictions and fuel modeling system-BURN subsystem part 1. General Tech. Rep. INT-194, U.S. Department of Agriculture, Forest Service, Intermountain Forest and Range Experiment Station, Ogden, UT.

[13] Anderson, H. E. 1983. Predicting wind-driven fire size and shape. Research Paper INT-305, U.S. Department of Agriculture,

Forest Service, Intermountain Forest and Range Experiment Station, Ogden, UT.

[14] Alexander, M. E. 1985. Estimating the length-to-breadth ratio of elliptical forest fire patterns. In *Proceedings of the Eighth Conference on Fire and Forest Meteorology*, Bethesda, MD, pp. 287-304.

[15] Nelson, R. M., Jr. 2002. An effective wind speed for models of fire spread. *International Journal of Wildland Fire* 11:153-61.

[16] Byram, G. M. 1959. Combustion of forest fuels. In *Forest fire: Control and use*, edited by K. P. Davis, 113-26. New York: McGraw-Hill.

[17] Albini, F. A. 1976. Estimating wildfire behavior and effects. General Tech. Rep. INT-30, U.S. Department of Agriculture, Forest Service, Intermountain Forest and Range Experiment Station, Ogden, UT.

[18] Miller, C., and D. L. Urban. 1999. Forest pattern, fire, and climatic change in the Sierra Nevada. *Ecosystems* 2:76-87.

[19] Rothermel, R. C., and G. C. Rinehard. 1983. Field procedures for verification and adjustment of fire behavior predictions. General Tech. Rep. INT-142, U.S. Department of Agriculture, Forest Service, Intermountain Forest and Range Experiment Station, Ogden, UT.

[20] Zeigler, B. P. 1984. *Multifacetted modelling and discrete event simulation*. London: Academic Press.

[21] Zeigler, B. P. 1990. *Object-oriented simulation with hierarchical, modular models: Intelligent analysis and endomorphic systems*. Boston: Academic Press.

[22] Morais, M. 2001. Modeling fire behavior in chaparral with HFire. Accessed from http://www.physics.ucsb.edu/~complex/research/hfire/fbehave/refs/morais_2001_hfire_pres.pdf

[23] Morais, M. 2001. Comparing spatially explicit models of fire spread through chaparral fuels: A new model based upon the Rothermel fire spread equation. M.A. thesis, University of California, Santa Barbara.

[24] Sneeuwjagt, R. J., and W. H. Frandsen. 1977. Behavior of experimental grass fires vs. predictions based on Rothermel's fire model. *Canadian Journal of Forest Research* 7:357-67.

[25] Wilgen, B. W. V., D. C. L. Maitre, and F. J. Kruger. 1985. Fire behavior in South African Fynbos (Macchia) vegetation and prediction from Rothermel's fire model. *Journal of Applied Ecology* 22:207-16.

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

Lewis Ntaimo *is Assistant Professor in the Department of Industrial Engineering, Texas A&M University, College Station, Texas.*

Bernard P. Zeigler *is Professor at the Arizona Center for Integrative Modeling and Simulation, Department of Electrical and Computer Engineering, University of Arizona, Tucson.*

Maria J. Vasconcelos *is Professor at the Tropical Research Institute, Lisbon, Portugal.*

Bithika Khargharia *is a PhD Student at the Arizona Center for Integrative Modeling and Simulation, Department of Electrical and Computer Engineering, University of Arizona, Tucson.*