

A novel model to predict a slab avalanche configuration using $m:n\text{-CA}^k$ cellular automata

Pau Fonseca *, Màxim Colls, Josep Casanovas

Computing Laboratory of Barcelona Informatics School, Universitat Politècnica de Catalunya, Campus Nord, Jordi Girona 31, 08035 Barcelona, Catalunya, Spain

ARTICLE INFO

Article history:

Received 19 January 2010

Received in revised form 8 July 2010

Accepted 8 July 2010

Keywords:

Cellular automata

$m:n\text{-CA}^k$

GIS

Simulation

Avalanche

ABSTRACT

The paper introduces a novel method to represent slab avalanche phenomena. The model is based on a cellular automaton generalization named $m:n\text{-CA}^k$. This generalization allows the use of different layers in a single cellular automaton. Each one of these layers represents the GIS data needed to perform the calculations. The model is represented using Specification and Description Language, and is implemented in C++. The clear separation between the model representation and implementation simplifies the understanding of the behavior that rules the model. We determine experimentally two parameters that generate more accurate results in all the tested scenarios, namely the Penalization Coefficient (K_p) and the Inertial Coefficient (K_i). All the models have been validated using data from the Catalan Pyrenees.

© 2010 Elsevier Ltd. All rights reserved.

1. Introduction

An avalanche is a massive slide of snow, ice, rock or debris down a mountainside. It can be originated by different factors, like an earth tremor, an extreme precipitation or man-made disturbances (slab avalanches are often caused by a mountain skier). The impact of the falling material and the winds related to the avalanche's flow can cause extensive damage to anything in its path. In this paper we are focused in the snow avalanche phenomenon.

Minimal requirements for the occurrence of an avalanche are snow and an inclined surface, usually a mountainside. Most avalanches occur on slopes between 30° and 45°, see Fig. 1.

According to McGraw-Hill Encyclopedia of Science and Technology, two basic main types of avalanches are recognized according to snow cover conditions at the point of origin. A loose-snow avalanche originates at a point and propagates downhill by successively dislodging increasing numbers of poorly cohering snow grains, typically gaining thickness as movement continues down slope. This type of avalanche commonly involves only those snow layers near the surface. The mechanism is analogous to dry sand.

The second type, the slab avalanche, occurs when a distinct cohesive snow layer breaks away as a unit and slides because it is poorly anchored to the snow or ground below. A clearly defined

gliding surface as well as a lubricating layer may be identifiable at the base of the slab. However the meteorological conditions which create these layers are complex to be analyzed.

In this paper the slab avalanche is the only kind of avalanche considered.

Different works consider this phenomenon. As an example (Barpi, Borri-Brunetto, & Veneri, 2007; Calidonna, Napoli, Giordano, Furnari, & Gregorio, 2001; Di Gregorio, Rongo, & Siciliano, 1999; Kronholm & Birkeland, 2005) can be reviewed. In Barpi et al. (2007) a similar methodology (using cellular automata) is presented, hexagonal cells are used for the cellular automaton space representation. In our approach we try to go a little further, trying to validate our model (going further the verification/calibration process) and trying to be able to model the phenomenon accurately although the data we have is not complete. Our main interest is that the results obtained from the simulator are no more than a 10% difference than those obtained from the empirical data for the two variable responses analyzed (the height and the weight of the snow). To allow this the model proposes the use of two parameters to deal with the problem of the lack of data. As is stated in Barpi et al. (2007), is quite difficult to define all the parameters in the model needed to perform a good calibration. As we see next, using K_i and K_c parameters the problem of lack of data could be simplified. This, finally, was confirmed for the models we analyze in the experimental section of the paper.

Other goal is to be capable of represent the model graphically using an ISO language, specifically we are using Specification and Description Language (Telecommunication standardization sector of ITU) (SDL). SDL is an object-oriented, formal language defined

* Corresponding author. Tel.: +34 934017732.

E-mail addresses: pau@fib.upc.edu (P. Fonseca), maxim.colls@fib.upc.edu (M. Colls), josepk@fib.upc.edu (J. Casanovas).

URL: <http://www-eio.upc.es/~pau/> (P. Fonseca).

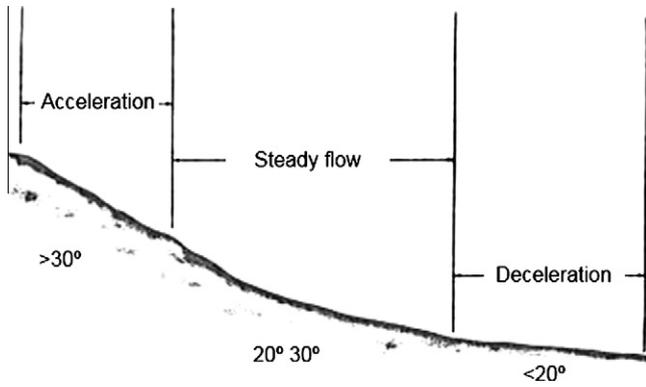


Fig. 1. Speed avalanche behavior, adapted from Mears (2002).

by The International Telecommunications Union as recommendation Z.100. It is intended for the specification of complex, event-driven, real-time, and interactive applications involving many concurrent activities that communicate using discrete signals. To know more about SDL (Reed R. SDL, 2000; Telecommunication standardization sector of ITU; SDL Tutorial) can be consulted. To see applications of SDL in environmental areas or social areas (Fonseca i Casas, 2004a,b, 2008) can be reviewed. Thanks this, the avalanche experts do not need to understand the implementation of the model, and can focus its efforts in the model definition. Also, this implies that the model is independent of the implementation tool used.

And last, and in a more theoretical aspect, the model is based in an extension of the common cellular automaton, the $m:n-CA^k$ (Fonseca i Casas, 2005), which simplifies the use of different layers and enables the possibility of use vectorial files in the model. This extension of the common cellular automaton is useful to describe this kind of simulation models (mainly simulation models that use SIG data).

This paper is organized as follows. First we describe some theoretical aspects related with the model, the cellular automaton structures and our generalization (in Sections 2 and 3). In Section 4 we present our model. In Section 5 we present the preliminary results of the experimentation. In Sections 6 and 7 we discuss the model verification and the model validation. Finally in Sections 8 and 9 we present the discussion and the concluding remarks.

2. Cellular automata

Cellular automata are discrete dynamical systems whose behavior is completely specified in terms of a local relation (Emmeche & Barcelona, 1998). Cells represent automaton space; time advances in discrete steps following “the rules”, the laws of “automaton universe”, usually expressed in a small look-up table. At each step every cell computes its new state in function of its closer neighbors. Thus, system’s laws are local and uniform. Fig. 2 shows one-dimensional cellular automaton initial state and successive three states after rules application. In that case the rule is defined as follows:

1. If one of the two cells of the neighborhood is filled, then the cell becomes filled.
2. If the cell is filled, then the cell becomes empty.

Initial state																			
1 iteration																			
2 iteration																			
3 iteration																			

Fig. 2. One-dimension cellular automata.

As we can see in a cellular automata is important to define the rules, the neighborhood and what mean cell.

For the integration of the simulation model with GIS data is useful to use cellular automata element, due his ability to effectively represent large-scale spatial dynamic phenomena (Benenson and Torrens, 2004; Ntaimo and Zeigler, 2004).

The cellular automata stores in his matrixes the GIS data used in the transitions, and implements a specific logic that enables the transitions calculations. Since we need to work with different layers we propose the use of an extension of the common cellular automaton to simplify this, and other issues, as we can see in the next section.

Similar approaches to use cellular automaton to represent an avalanche can be reviewed on (Barpi et al., 2007; Di Gregorio et al., 1999; Kronholm & Birkeland, 2005). On these works the representation of the model is based on the description of the equations that rules the evolution of the cellular automata. In Webb and Wainer (2008), a description of the model of Di Gregorio et al. (1999) using DEVS formalism is done.

Cellular automaton structure lacks in how to work with different layers or how to deal with continuous space. CAN cellular automaton presented in Calidonna et al. (2001) allows to use different layers, however are limited to use a discrete space. To allow and simplify this management we propose the use of a multi:n-dimensional cellular automata (Fonseca i Casas, 2005), an extension that defines the cellular automaton over a mathematical definition of a topological space and allows to use different layers. Using this extension CAN networks can be viewed as a particular case of a $m:n-CA^k$ cellular automaton with a specific topological space over discrete space.

Also we propose to use a graphical language like Specification and Description Language (Reed R. SDL, 2000; Telecommunication standardization sector of ITU; SDL Tutorial) (ISO IEC language) to represent the model. This implies that the cellular automaton evolution functions can be represented graphically, and the dependencies between all the different elements that compose the model can be reviewed graphically too.

3. Multi:n-dimensional cellular automata ($m:n-CA^k$)

A multi:n-dimensional cellular automaton ($m:n-CA^k$) is a generalization of a cellular automata defined as follows (Fonseca i Casas, 2005):

Definition 1. $m:n-CA^k$

A multi n dimensional cellular automaton is a cellular automaton generalization composed by m layers with n dimensions each one.

The representation is:

$$m : n - CA^k \tag{1}$$

where m : is the automaton number of layers. n : is the different layers dimension. k : is the number of main layers (1 by default).

A layer in a $m:n-CA^k$ is a main layer if a transition function Λ is defined in order to modify its state. A $m:n-CA^k$ automaton only presents one main layer, while $m:n-CA^k$ automaton presents k main layers.

Since multiple layers belong to a single automaton, its state is defined as follows.

Definition 2. $E_m[x_1, \dots, x_n]$, layer m state in x_1, \dots, x_n position
 E_m is a function describing cell state in position x_1, \dots, x_n of layer m .

E_m function allows state representation for each cell in the different layers of the automata, but this is not the global state of the automata. This state is represented by the EG function.

Definition 3. $EG[x_1, \dots, x_n]$, automata status in x_1, \dots, x_n position.

EG returns automata global state in position georeferenced by coordinates x_1, \dots, x_n .

The global state of cellular automata depends on EG function in all automata positions.

Combination functions Ψ is represented by equation:

$$\Psi(E_1[x_1 \dots x_n]^{m-2}, E_m[x_1 \dots x_n]) = EG[x_1 \dots x_n] \tag{2}$$

In a common cellular automaton, evolution function allows global automata state change through cells value modification.

In a $m:n\text{-CA}^k$ vectorial layers use makes necessary to generalize the neighborhood and later define a new function that defines something similar to cell size (nucleus function).

Definition 4. Evolution function $\mathcal{A}m$

Function defined for the layer m to modify its state through the state of others layers using combination function Ψ , and vicinity and nucleus functions.

Intuitively evolution function allows the representation of the modifications in this layer (modifications in nucleus area of a point $x_1 \dots x_n$), using the state of other layers with combination function Ψ , and the vicinity area.

For more information about the vicinity and nucleus area and the vicinity and nucleus function, (Fonseca i Casas, 2005) can be reviewed.

4. Our slab avalanche model: $6 + N:2\text{-CA}^{4+N}$ on Z^2

From this definition ($6 + N:2\text{-CA}^{4+N}$ on Z^2), $N \in [0, 5]$, represents the maximum number of obstacles typologies (trees, houses, rocks, etc.) that can be added to the model. Since all the parameters of the simulation are represented in these layers, is easy to perform different simulations that represent different alternatives or situations. The problem to perform a new simulation scenario is reduced to find the layers that represent the area that we want to simulate. It is not needed to modify the simulation engine. These data is stored following the IDRISI32 file format. The data can be acquire on the Catalonia Cartographic Institute (ICC) **Institut Cartogràfic de Catalunya**, on the Center for Ecological Research and Forestry Applications (CREAF) **REAF** and on the Meteorological Institute of Catalonia (METEOCAT) **METEOCAT**. Table 1 shows the different layers of the automaton, and the source.

Table 1
Model layers.

Name	Type	Description	Qty	Source	Main layer (change its data)
Height	Raster	Layer representing the height of the environment.	1	ICC	No
Thickness of the snow	Raster	Represents the thickness of the “slab snow”	1	METEOCAT	Yes
Floor features	Raster	Represents the kind of surface (rocks, sand, snow, ice...). Each surface has his own specific rough parameter.	1	METEOCAT CREAM	No
Snow that causes the slab features	Raster	Density, compactness of the snow.	1	METEOCAT	Yes
Obstacles	Raster	Represents the obstacles that have the environment (small rocks, big rocks, houses, trees...)	N	CREAF	Yes
Crack	Vectorial	Line representing the breakdown of the ice.	1	Input data	Yes, at beginning
State of the snow	Raster	Shows the state of the terrain, <i>empty</i> (without snow that causes the windslab), <i>static</i> (contains snow that is stable) and <i>dynamic</i> (contains snow that in the next iteration is moving)	1	METEOCAT	Yes

4.1. Vicinity, nucleus and evolution functions for the slab avalanche model

The Moore neighborhood is used over Z^2 . The vicinity and nucleus functions allowing the definition of the evolution function are based in the discrete topology over Z^2 . The vicinity function represents, over Z , whose cells that must be taken in consideration to perform a calculation for the cell defined by (x_1, x_2) , while the nucleus function represents the cell that is modified due to the calculus.

$$\text{Vicinity function : } \nu n(x_1, x_2) = \{(x_{1-1}, x_{2-1}), (x_{1-1}, x_2), (x_{1-1}, x_{2+1}), (x_1, x_{2-1}), (x_1, x_2), (x_1, x_{2+1}), (x_{1+1}, x_{2-1}), (x_{1+1}, x_2), (x_{1+1}, x_{2+1})\}$$

$$\text{Nucleus function : } nc(x_1, x_2) = \{(x_1, x_2)\}$$

The *evolution function* (\mathcal{A}) is based in the analysis of the global state ($EG(x_1, x_2)$) for a selected cell, through the *combination function* that merges the data contained in each one of the different layers of the automaton ($\Psi(E_1[x_1, x_2], E_2[x_1, x_2], \dots, E_4 + N[x_1, x_2])$).

In our model, we need to define the next *evolution functions*:

- $E_2[i]$: Thickness of the snow. The “Modify information (p)” function defines its behavior.
- $E_4[i]$: Density, compactness of the snow, in our case is 0.5 (Mears, 1976).
- $E_6[i]$: State of the snow. The function is defined in the next diagrams.
- $E_N[i]$: Obstacles. The function that defines the obstacles we use in the model.

Focusing on the definition of the layer state of the snow ($E_6[i]$), Fig. 3 shows the combination function, Fig. 4 shows a state machine representing the evolution function, Fig. 5 shows the process diagram for the *Empty* and *Dynamic* states. Is not a complete description of the evolution function, but represents the three main states in which each cell can be, and how the model behaves when receives an event in the *Empty* state.

As stated before, the language used to represent the behavior of the layer is SDL.

The different procedures represented in the process diagram, completely define the evolution function. As an example, propagation procedure defines the speed and the quantity of snow that must be propagated to the neighborhood cells. The increment in the force is used in the (3) expression to determine if the snow continues its movement to other cell, or stops its movement if the force is equal to zero.

$$F_{i,t} = \max(IF_{i,t} + \Delta F_{i,t}, 0) \tag{3}$$

Here, we only define the relation between the different main layers of the mn:CAk automaton. To obtain the information of the neighborhood cells of all other layers, mn:CAk uses a procedure defined named **vicinity**. The **evolution function** is defined in each one of the main layers through its processes.

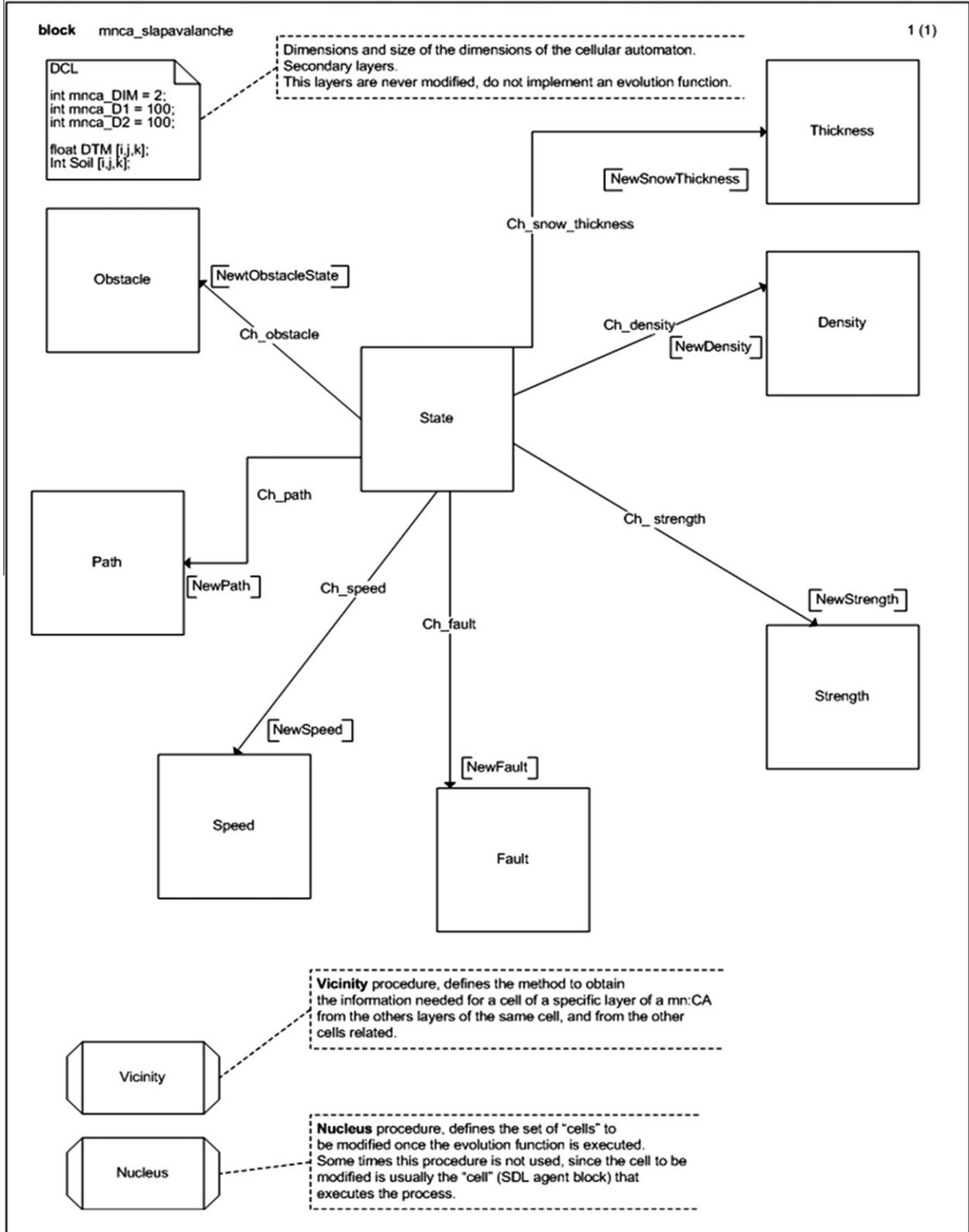


Fig. 3. Combination functions for the cellular automaton. This diagram represents a single cell and how combines with the different layers of the automaton.

To calculate the increment of the force we use the (4) expression.

$$\Delta F_{i,t} = IF_{i,t} - (SFF_{i,t} + IFF_{i,t} + ASFF_{i,t} + AFF_{i,t} + FFF_{i,t} + OFF_{i,t}) \quad (4)$$

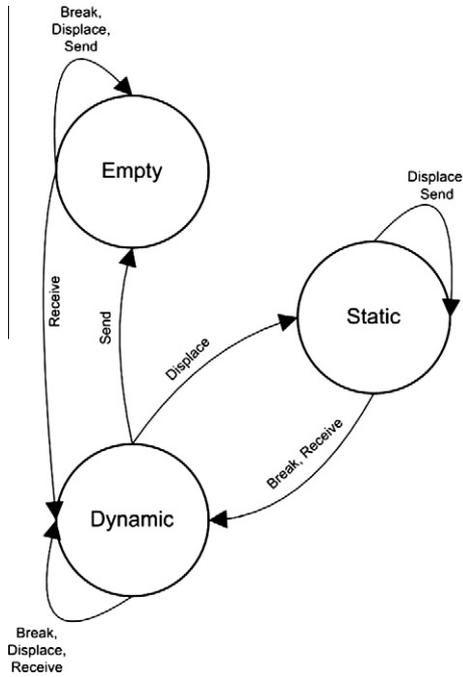


Fig. 4. Evolution function (Λ). This represents the behavior of each cell of the state layer.

where $IF_{i,t}$ = impulse force, depends on the quantity and quality of the snow, and the slope. $SFF_{i,t}$ = sliding friction force between the avalanche and the underlying snow or ground. $IFF_{i,t}$ = internal dynamic shear resistance due to collisions and momentum exchange between particles and blocks of snow, (internal friction force). $ASFF_{i,t}$ = turbulent friction within the snow/air suspension, (air suspension friction force). $AFF_{i,t}$ = shear between the avalanche and the surrounding air, (air friction force). $FFF_{i,t}$ = fluid-dynamic drag at the front of the avalanche (front friction force). $OFF_{i,t}$ = obstacle friction force.

To define the friction in the model (needed to calculate $SFF_{i,t}$) is needed a layer (soil) with the different values showed in the Table 2. This layer is defined as a fixed parameter for the scenarios as we can see next.

In the scenarios studied the ground lies in the category of snow. A configuration for the ground with ice has not been reported in

Table 2
Soil layers values.

Category	Terrain type	Value in the layer
Snow	Powder snow, non-compacted snow.	0
	Dry snow, that causes the slab avalanche	1
	Ice	2
Ground	Big gravel	3
	Thin gravel	4
Vegetation	Clear vegetation	5
	Normal vegetation	6
	Dense vegetation	7
Water	Water	8

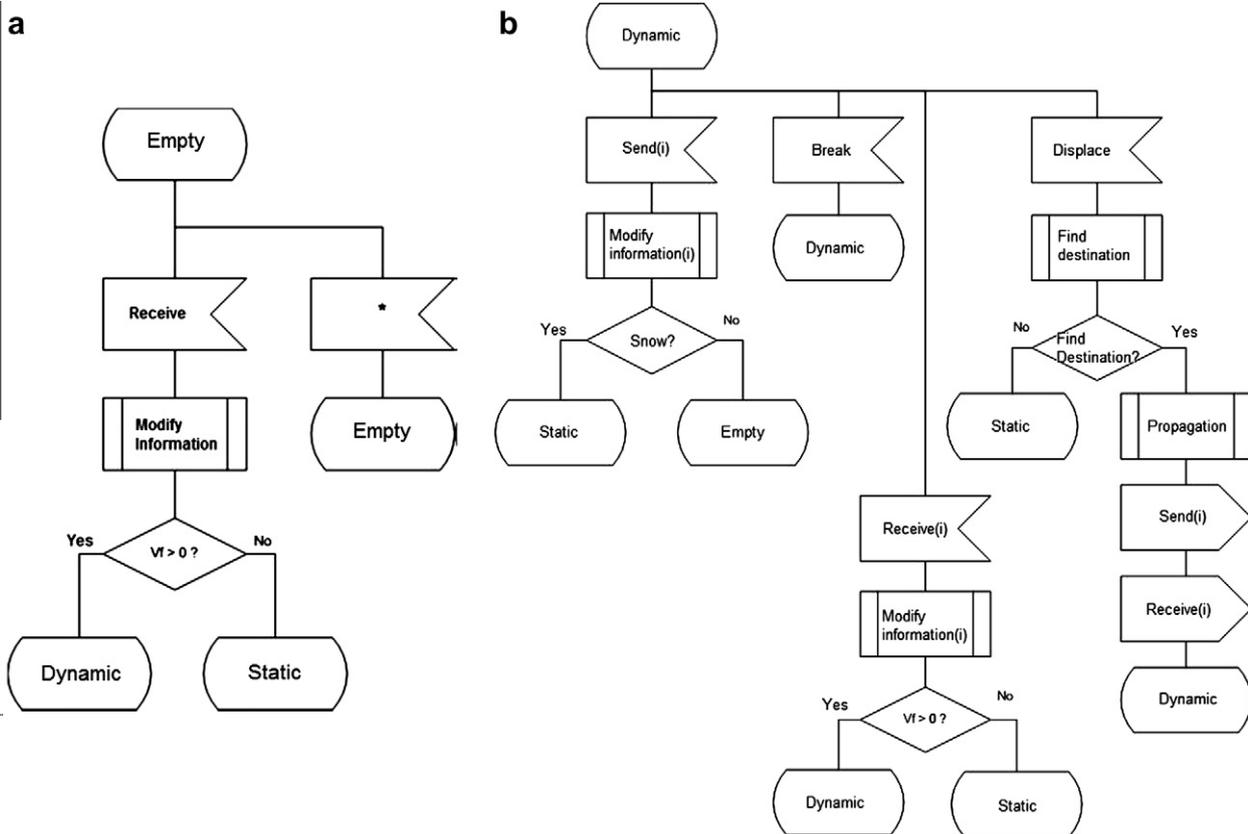


Fig. 5. SDL process diagram for the Empty and Dynamic states for the evolution function of the layer state (state of the snow). V_f is the speed of the snow in the cell.

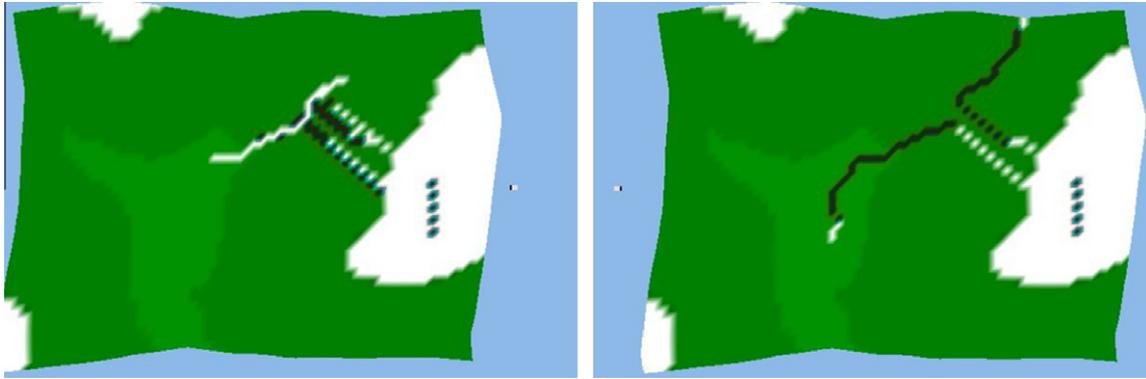


Fig. 6. The same avalanche with $K_i = 0$ (right) and $K_i = 0.1$ (left). The behavior of the right figure (see the line representing the snow movement) is quite similar to the water behavior.

the scenarios. That means that the layers only have two possible values, 0 or 1. For the definition of these values we assume that the high zones of the mountain are composed by the slab, which means dry snow suitable to provoke an avalanche. That defines the amount of snow that belongs to the initial crack (“soft snow starting height” variable as we can see next). This value is defined using the mean value of the observational parameters of the slab avalanches in the Pyrenees area (Oller & Institut Geològic de Catalunya, 2009), which is 201 m.

5. Experimentation

To define the experiments, we first need to obtain information from the scenarios that we want to represent. The main problem is that not all the information needed can be obtained from the system. This implies the use of some assumptions in order to perform the simulations. In our approach the information that we cannot obtain from the system is represented using two variables, the Penalization Coefficient (K_p) and the Inertial Coefficient (K_i).

K_p allows to represent the opposing forces to the movement of the snow. This value is not only depending on the mountain structure, but also on the snow configuration and the friction with the air. This factor is obtained empirically from the different experiments performed in the different scenarios as we can see next.

The K_i coefficient is needed in order to give to the snow the behavior expected; it represents the inertial forces inherent to the snow downhill movement. As an example we can see what happens if $K_i = 0$ (no K_i coefficient is used). In Fig. 6 a surface (a mountain) is represented, white areas represent the highest elevations. Dark lines represent the snow movement, while white lines represent the accumulation of the snow after its movement. The dark line on the white area represents the initial fracture of the slab avalanche.

In the left the Inertial Coefficient have a correct value; in the right the behavior of the snow is like water, the Inertial Coefficient is set to 0. To see a similar approach to simulate the water behavior you can see (López, Casas, Casanovas, & Modelling, 2008).

The parameters used in the model are represented in the cellular automaton structure. This data, that represents the main and secondary layers (Table 1), are represented using IDRISI (Clark labs) files.

The variables that we want to analyze are (response variables):

1. Total distance of the snow movement.
2. Height of the snow at the beginning and at the end of the movement.

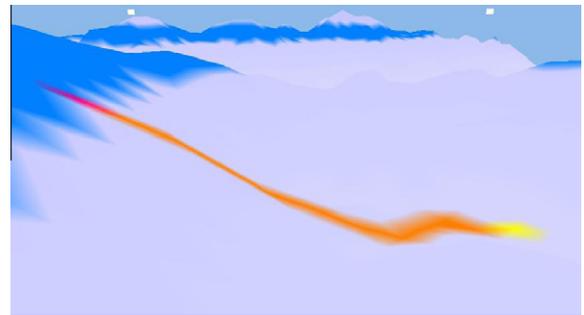


Fig. 7. Avalanche representation.

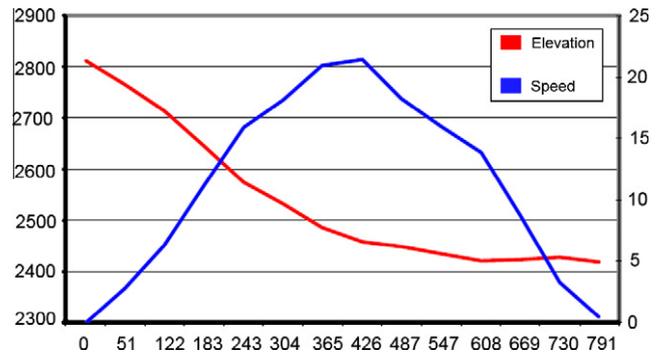


Fig. 8. Model results for the height, and speed.

We also perform a visual validation of the model, through the virtual reality output of the simulator.

6. Preliminary results, model calibration

In the preliminary stages of the model development, we define different experiments to perform the verification of the implemented solution. The different scenarios we tested represent hypothetical slab avalanches and have the main objective to test if the implementation of the model was done correctly. In addition we developed a graphical tool, useful to understand the avalanche evolution. This tool represents the movement of the snow in a 3D world using Virtual Reality Modeling Language (VRML). The

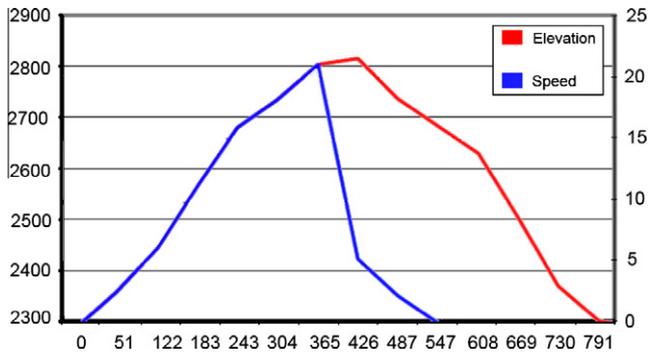


Fig. 9. Model results. Speed of the model without obstacles, and speed of the model with obstacles.

picture shows a representation of an avalanche (orange¹ line represents the movement of the snow and yellow area the accumulation zone for the snow.

Fig. 7 shows a virtual reality representation of an avalanche model. Fig. 8 represents the snow speed in meters per second and the height in meters of the scenario of Fig. 7. Fig. 9 shows the comparison of this model with other where the obstacles are defined. Fig. 10 shows a virtual reality representation of an avalanche using obstacles in the model.

From each scenario we obtain, in addition to the graphical VRML representation, the description of the dynamic behavior of the avalanche in a summary as we can see next.



Fig. 10. Avalanche and obstacles (trees).

Also we compare the simulation speed with Table 3, obtaining similar results. In four different models, the maximum speed is about 21 m/s (small or medium avalanche). In a test for a big avalanche, the maximum speed in the model is about 67 m/s (corresponding to the suggested values of Leaf and Martinelli, 1977 (1977), Mears et al. (1976)).

Regarding the obstacles, we compare our results with the expected results defined in Mears, 2002 (see Fig. 11), and again the results we obtained were similar. In our model, see Fig. 9, the snow traverses the obstacles, and continues its movement, obviously reducing its speed.

Since this preliminary version of the model behaves similar to a theoretical slab avalanche, we go further to represent specific real

Parameters

Inertial Coefficient:

0.300000

Internal Penalization

Coefficient: 0.150000

Time: 0.00 s

Coord: 1839->1772

H: 2410.9 m

Speed 0:

A: 1.40

M:

Thickness:

0.000 m/s

m/s^2

220.50 kg

0.490 m

Time: 0.00 s

Coord: 1773->1706

H: 2405.5 m

Speed 0:

A: 0.90

M:

Thickness:

0.000 m/s

m/s^2

220.50 kg

0.490 m

(...)

Time: 41.23 s

Coord: 1352->1351

H: 1990.0 m

Speed 0:

A:

M:

Thickness:

13.821 m/s

-4.02 m/s^2

135.23 kg

0.285 m

Statistical data:

Maximum distance: 807.77 m

Maximum height: 424.00 m

Transported snow: 8550.00 kg

Snow in deposit: 6015.31 kg

Snow loss: 2534.69 kg

Maximum strength: 5594.48 N

Maximum speed: 51.27 m/s

Maximum acceleration: 4.72 m/s^2

Maximum slope: 56.07°

Minimum slope: -32.35°

Duration: 41.23 seg

In this phase of the model construction, we are focused in representing correctly the phenomenon. We compare the behavior of the model with the expected behavior described in Mears, 2002, see Fig. 1. Fig. 8 (model behavior) can be compared with Fig. 1 (avalanche behavior). As we can see the model responds to the observed behavior.

¹ For interpretation of color in Figs. 7–11, the reader is referred to the web version of this article.

scenarios. To do that we focus in five different real slab avalanche and we try to emulate they behavior.

7. Model validation, final results

During the validation process, we have been conducted the simulation of much more than 30 experiments for each one of five different proposed avalanche scenarios. The obtained data have been compared with real avalanches data. From all the different results

Table 3
Avalanche speed Mears (2002).

Type	Small	Medium	Large
Wet flowing	<10 m/s	10–20 m/s	20–35 m/s
Dry flowing	<10 m/s	10–35 m/s	35–60 m/s
Powder	<25 m/s	25–60 m/s	60–90 m/s

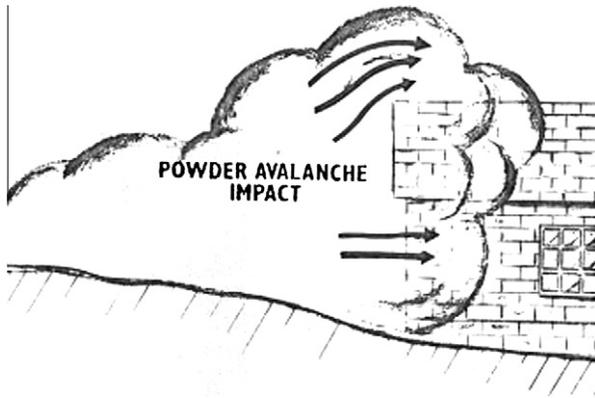


Fig. 11. Obstacles collision avalanche behavior, adapted from Mears (2002).

Table 4
Data for the NUR029199801 scenario.

Crack deep	0.5 m
High elevation	2380 m
Low elevation	2030 m
Distance	635 m
Height difference	350 m
Soft snow starting height	2245 m
Crack line	60 m (two cells)
Crack UTM coordinates	(431425.375000, 4694534.500000) (431425.375000, 4694504.000000)

Table 5
Experimental values for height and distance variables in NUR029199801 scenario.

K_p	$K_i = 0.30$		$K_i = 0.45$		$K_i = 0.60$		$K_i = 0.75$		$K_i = 0.90$	
	Distance	Height	Distance	Height	Distance	Height	Distance	Height	Distance	Height
0.05	894.99	434.10	836.78	427.00	684.76	385.00	670.82	389.60	697.78	399.00
0.10	836.78	427.00	836.78	427.00	684.76	385.00	684.76	385.00	670.82	389.60
0.15	778.85	420.90	807.77	424.00	658.63	385.40	615.55	353.60	615.55	353.60

Table 6
NTR184200301 avalanche configuration.

Crack deep	1 m
High elevation	2960 m
Low elevation	2210 m
Distance	1350 m
Height difference	750 m
Soft snow starting height	2750 m
Crack line	180 m (6 cells)
Crack UTM coordinates	(326238.906250, 4717457.000000) (326238.906250, 4717427.000000) (326238.906250, 4717397.000000) (326178.468750, 4717457.000000) (326178.468750, 4717427.000000) (326178.468750, 4717397.000000)

obtained from the model, (speed, duration of the avalanche, strength, etc.), we are focused in the distance and the height. Obviously the simulation model does not give an exact value for the response variables (height and distance). This is due not only to the inherent complexity of the model, but also to the inherent complexity of obtaining real data from the avalanche. To deal with this problem we assume a variability of 0.1 on the response variables.

For each one of these five scenarios we define a set of experiments with different values for the factors K_i and K_p . As we stated before K_i and K_p are factors that define different elements that cannot be obtained from the real avalanche. We use a set of K_i and K_p values in each one of the different scenarios, since we are looking for good K_i and K_p candidates for each individual scenario. As we can see next, we achieve to obtain a K_i and K_p values that are good for all the scenarios. For each one of the different experiment we obtain a virtual representation of the slab avalanche, in the paper we show only one for each scenario. In this representation the movement of the snow is represented with a dark (or blue) line, while the accumulations of snow are represented using white color. Green color (light grey on black and white image) represents the area with soft snow, while the white area on the top of the mountains represents the elevations where the snow is suitable to become part of the initial crack of a slab avalanche. Also we

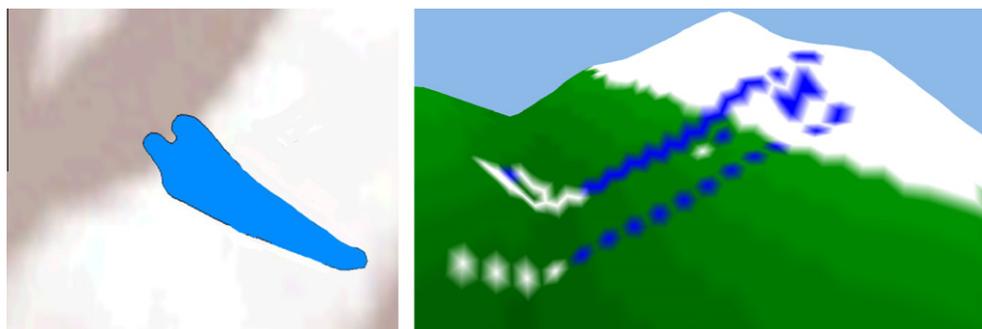


Fig. 12. An execution representation for the NUR029199801 scenario.

obtain a table with the data. In the tables, values in bold are representing those who have a variability of less than a 10% with the real values (observed from the real avalanche). The experiments are defined by the combination of the possible values for K_i and K_p represented next:

$$K_p = \{0.05, 0.10, 0.15, 0.20, 0.25, 0.30, 0.35, 0.40, 0.45, 0.50\}$$

$$K_i = \{0.30, 0.60, 0.90\}$$

The snow height is 0.5 m for all the models and the “soft snow starting height” depends on the 201 m of distance from the crack as we see previously (Oller, 2009).

In the next section we present the data obtained from the different simulated avalanches.

7.1. Slab avalanche NUR029199801

This slab avalanche takes place near the Núria Sanctuary, in the Catalan Pyrenees.

The data used in this scenario is shown in Table 4.

For height variable, the valid interval for the values obtained from the simulator is [315, 385] and for distance variable the interval is [571.5, 698.5]. Table 5 shows the simulation results.

Table 7
Experimental values for height and distance variables in NTR184200301 scenario.

K_p	$K_i = 0.30$		$K_i = 0.45$		$K_i = 0.60$		$K_i = 0.75$		$K_i = 0.90$	
	Distance	Height	Distance	Height	Distance	Height	Distance	Height	Distance	Height
0.05	990.45	487.20	2100.21	1082.40	2100.21	1082.40	1209.34	622.40	1103.90	460.70
0.10	732.39	391.10	2072.39	1082.40	976.73	430.60	976.73	430.60	976.73	430.60
0.15	684.11	363.50	684.11	363.50	708.03	380.90	722.50	343.80	722.50	343.80

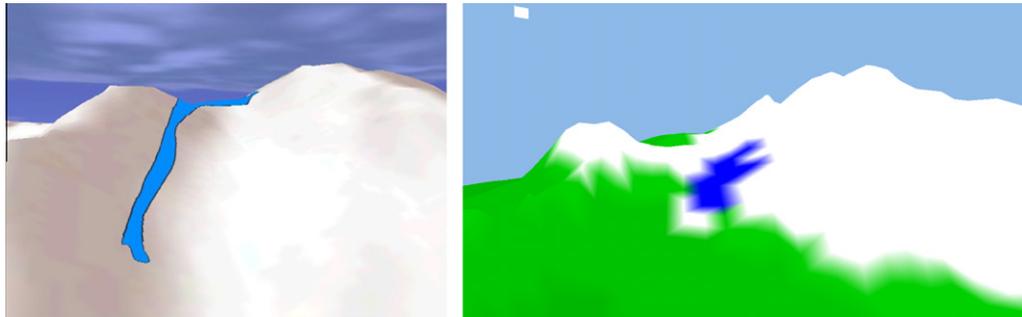


Fig. 13. An execution representation for the NTR184200301 scenario.



Fig. 14. Corrected georeferenzation for the NTR184200301 avalanche.

In Fig. 12 we can see the representation of the real avalanche and the simulated avalanche. In all the scenarios, the figure showed is for the pair $K_i = 0.6$ and $K_p = 0.1$. As we can see next these values are always good candidates in all the scenarios.

7.2. Slab avalanche NTR184200301

This slab avalanche took place in the Noguera de Tor area (in Pyrenees).

Table 6 presents the data used for this scenario.

For height variable the interval is [675, 825] and for distance variable the interval is [1215, 1485]. Table 7 shows the simulation results.

This scenario allows us to detect some mistakes in the process of represent the path of the real avalanche (the left picture that

represents the path obtained from the empirical data). As you can see in the representation of the avalanche (Fig. 13, left, real data), the avalanche is not going down following the usual mountain canal. This is due to an incorrect representation of the UTM points defining the start position of the avalanche (displaced to the left from our point of view, see Fig. 14 to see the correct location of the avalanche). However the height and the distance data of the slab avalanche were correctly taken. Because that the simulated avalanche starts the movement from an incorrect origin, the behavior was completely different. When we detected this incongruence, and after a detailed review of the model, we suppose that maybe the mistake was in the data. We confirm this hypothesis with the IGC who have a corrected version of this scenario when we contact with them. It is remarkable the huge work that the geological institutions do to store and catalogue this phenomenon, a phenomenon that often implies to work in hazardous envi-

Table 8
RaspesRoies96 avalanche configuration.

Crack deep	0.5 m
High elevation	2710 m
Low elevation	2210 m
Distance	840 m
Height difference	500 m
Soft snow starting height	2400 m
Crack line	140 m (5 cells)
Crack UTM coordinates	(326387.375000, 4705156.000000) (326387.375000, 4705125.500000) (326387.375000, 4705095.000000) (326387.375000, 4705034.000000) (326387.375000, 4705003.500000)

Table 10
SMR170200604 avalanche configuration.

Crack deep	0.5 m
High elevation	2495 m
Low elevation	2264 m
Distance	467 m
Height difference	233 m
Soft snow starting height	2400 m
Crack line	90 m (2 cells)
Crack UTM coordinates	UTM (326003.031250, 4704019.000000) (326003.031250, 4703988.000000) (326003.031250, 4703957.500000)

Table 9
Experimental values for height and distance variables in RaspesRoies96 scenario.

K_p	$K_i = 0.30$		$K_i = 0.45$		$K_i = 0.60$		$K_i = 0.75$		$K_i = 0.90$	
	Distance	Height								
0.05	1164.99	570.10	942.02	498.30	942.02	498.30	942.02	498.30	942.02	498.30
0.10	1006.23	474.50	844.81	457.60	823.77	457.20	823.77	457.20	823.77	457.20
0.15	768.96	430.90	787.46	441.90	787.46	441.90	768.96	430.90	735.46	422.60

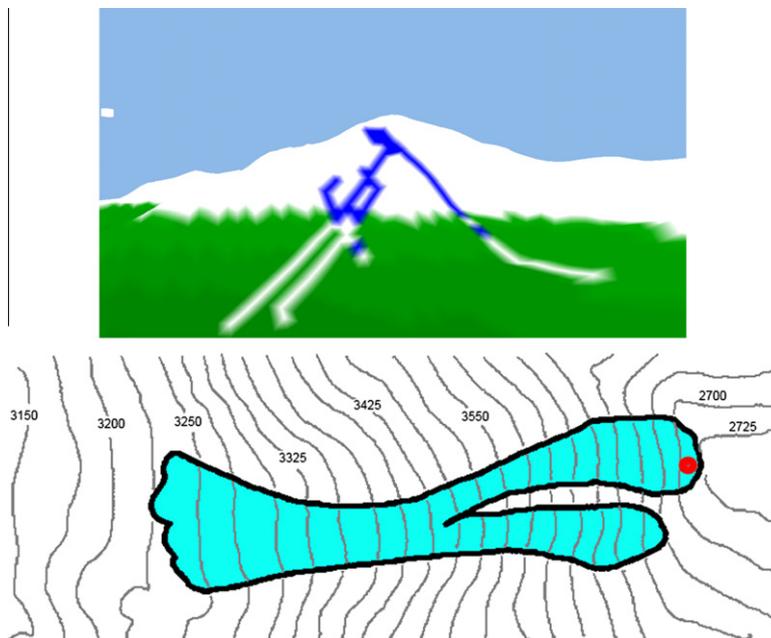


Fig. 15. An execution representation for the RaspesRoies96 scenario.

Table 11
Experimental values for height and distance variables in SMR170200604 scenario.

K_p	$K_i = 0.30$		$K_i = 0.45$		$K_i = 0.60$		$K_i = 0.75$		$K_i = 0.90$	
	Distance	Height	Distance	Height	Distance	Height	Distance	Height	Distance	Height
0.05	488.36	254.00	540.83	264.90	827.59	323.60	827.59	323.60	827.59	323.60
0.10	465.73	248.10	474.34	249.10	474.34	249.10	700.36	288.70	700.36	288.70
0.15	436.81	240.20	445.98	241.30	445.98	241.30	371.08	221.50	371.08	221.50

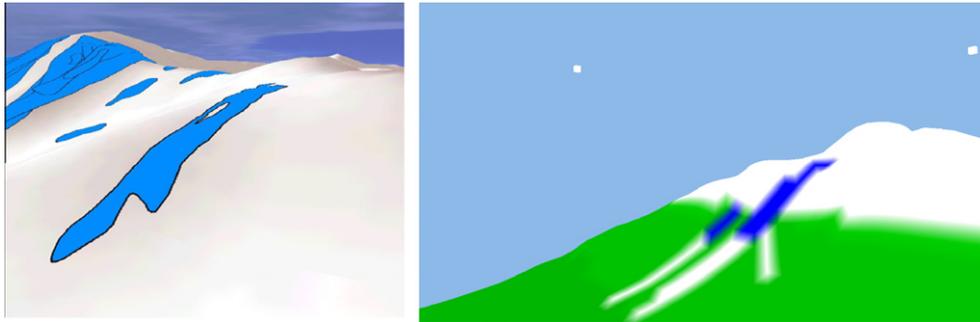


Fig. 16. An execution representation for the SMR170200604 scenario.

Table 12
RUD006200408 avalanche configuration.

Crack deep	1.8 m
High elevation	1920 m
Low elevation	1580 m
Distance	562 m
Height difference	360 m
Soft snow starting height	1800 m
Crack line	12 m (4 cells)
Crack UTM coordinates	UTM (332152.500000, 4729137.500000) (332152.500000, 4729106.500000) (332152.500000, 4729075.500000) (332152.500000, 4729045.000000)

ronment and fighting against different factors, weather, environment, timing, etc. This makes that sometimes some information can introduce little errors. This lead us to think that in the future we can use the avalanche model to validate the accuracy of the data obtained from the real system, or to detect some errors in the data acquisition.

7.3. Slab avalanche RaspesRoies96

This slab avalanche took place in the Boí Taüll sky station on 1996 (Catalan Pyrenees). This slab avalanche is caused by an explosion. This makes that its behavior is slightly different as we can see next. Table 8 presents the data used for this scenario.

Table 13
Experimental values for height and distance variables in RUD006200408 scenario.

K_p	$K_i = 0.30$		$K_i = 0.45$		$K_i = 0.60$		$K_i = 0.75$		$K_i = 0.90$	
	Distance	Height								
0.05	445.98	243.40	785.18	422.00	785.18	422.00	785.18	422.00	785.18	422.00
0.10	530.75	310.00								
0.15	445.98	243.40	445.98	243.40	445.98	243.40	445.98	243.40	445.98	243.40

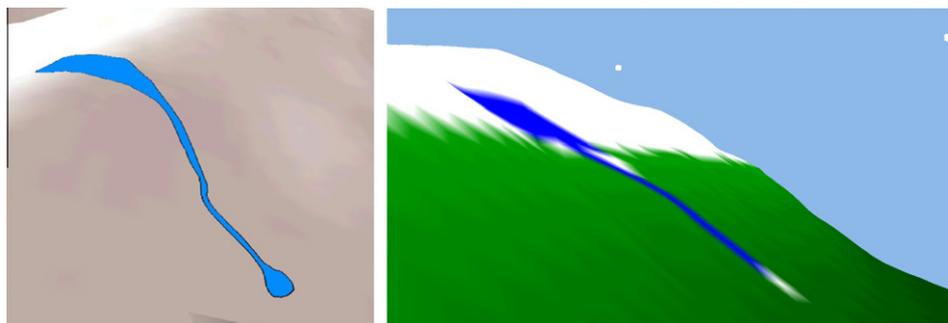


Fig. 17. An execution representation for the RUD006200408 scenario.

The values must be between [756, 924] for distance and [450, 550] for height. Table 9 shows the simulation results.

As we can see (Fig. 15), the simulated avalanche presents a different path compared with the empirical data. However the values for distance and height are good enough. This maybe is due to the structure of the terrain or the disruptive start of the avalanche caused by the explosion. More research is needed in this direction to understand this. However we think that the twin avalanche of the real avalanche was caused by the expansive wave of the explosion. In the simulated model, all the divisions start on the initial crack. This makes that this first division, represented in the simulated model, follows a secondary canal and not follows the same canal of the main avalanche.

7.4. Slab avalanche SMR170200604

This slab avalanche took place in the Sant Martí area, near the Boi Taüll sky station in the south of Puig Falcó (Catalan Pyrenees). Table 10 presents the data used for this scenario.

For height variable the interval is [209.7, 256.3] and for distance variable the interval is [420.3, 513.7]. Table 11 shows the simulation results, we can see the representation of the real avalanche and the simulated avalanche in Fig. 16.

7.5. Slab avalanche RUD006200408

This slab avalanche took place near the Vall de Ruda, on the Vaqueira sky station area (Catalan Pyrenees). Table 12 presents the data used for this scenario.

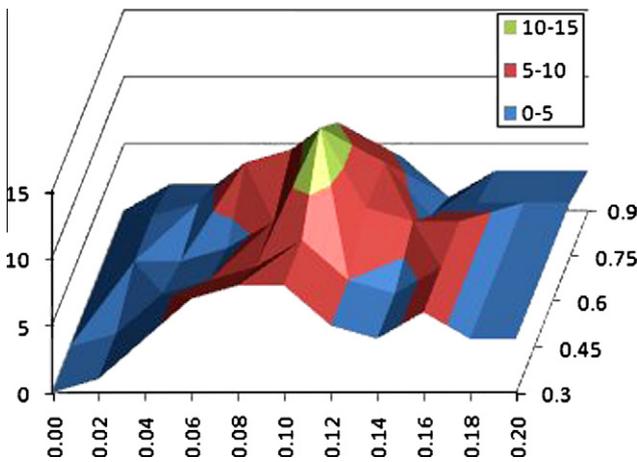


Fig. 18. The best values for K_i and K_p are those who are good enough for all the models.

Table 14 Experimental values for the models with $K_i = 0.6$ and $K_p = 0.1$.

Experiment	Distance		Height	
	Real value	Simulated value	Real value	Simulated value
NUR029199801	635	684.76	350	385.00
NTR184200301	1350	976.73 ^a	750	430.60 ^a
RaspeRoies96_2	840	823.77	500	457.20
SMR170200604	467	474.34	233	249.10
RUD006200408	2	530.75	360	310.00

^a As we see previously the NTR184200301 avalanche model was defined with incorrect data. The obvious difference between the simulated data and the empirical data, helped to detect a mistake in the empirical avalanche data.

For height variable the interval is [306, 374] and for distance variable the interval is [506, 618]. Table 13 shows the simulation results, we can see the representation of the real avalanche and the simulated avalanche in Fig. 17.

8. Discussion, experimental values for K_i and K_p

With the different experiments conducted in each scenario we can obtain the best values for K_i and K_p parameters. Those best values are represented in Fig. 18. The summit represents those values that are good for all the models.

As we see, good values for K_i and K_p are $K_i = 0.6$ and for $K_p = 0.1$. With this values we obtain for the two interest variables to be analyzed a variability of approximately 10%. Table 14 summarized this.

Regarding input data errors simulation results are sensible to data quality and precision. In this initial scenario, we can assume that current information obtained from the avalanche has a sufficient degree of accuracy, at least for the more important parameters of the models. As we see in the paper, these parameters are snow heights, the initial crack, and the trajectory of the slab avalanche. As is described in Oller, 2009, it is necessary to travel to the affected areas immediately after the avalanche takes place to obtain this information. To collect preliminary data, a flight with a helicopter is done to obtain the first visual information regarding the dimensions and the shape of the avalanche. The complete scene is filmed and post processed. Next a team of experts inspects the avalanche location to take information regarding the height of the snow, the crack and other parameters, like the stauwall. As is stated in Oller, 2009, this data has to be collected in situ, since observations obtained and recorded from the helicopter can introduce errors in the measurement. This two steps procedure, along with the final global evaluation of an experts committee, can validate at a first level the data accuracy. Under these conditions and certainly taking in account their present limitations, we assume that input data are suitable for our models. After this experience, new protocols for data collection and validation are in discussion with meteorological and geological authorities to improve the reduction of input data errors for this family of intensive simulation models.

9. Concluding remarks

The model presented shows good results for the slab avalanche phenomena. In a preliminary model we compare the simulation output with (Leaf and Martinelli, 1977; Mears et al., 1976) obtaining similar results. In four different models, the maximum speed is about 21 m/s (small or medium avalanche). In a test for a big avalanche, the maximum speed in the model is about 67 m/s (little over the typical value for a big dimensions avalanche) (Leaf and Martinelli, 1977; Mears et al., 1976). This allows us to go further in the process of the validation of the model.

Up to five different scenario based on real slab avalanche have been analyzed successfully. Since the layers represent all the model variables and are obtained from GIS data, the construction of a new simulation model is based on the definition of these layers, work that can be performed in a GIS. This simplifies the modeling task.

We estimate the values for K_i and K_p factors, showing that the good values are 0.6 for K_i and 0.1 for K_p . With these values we obtain accurate results in the tested models regarding the two response variables analyzed.

In a more theoretical aspect is presented an application to represent the avalanche phenomena using GIS data using the $m:n$ -CAK cellular automaton. The structure in layers of all the elements that

can interact with the evolution function of the avalanche, simplify the calculus of the evolution functions, allowing an easier implementation of the model. This also simplifies the model formalization process. Also, the specification of the model was done using Specification and Description Language. The use of this graphical language simplifies the understanding of the model for personnel that are not used with the common programming languages (C++ in our case). In addition, this methodology clearly separates the model from its implementation, and the tools used to perform this implementation.

References

- Barpi, F., Borri-Brunetto, M., & Veneri, L. (2007). Cellular-automata model for dense-snow avalanches. *Journal of Cold Regions Engineering*, 21(4), 121–140.
- Benenson, I., & Torrens, P. (2004). *Geosimulation, automata-based modeling of urban phenomena*. West Sussex PO19 8SQ, England: John Wiley & Sons Ltd.
- Calidonna, C., Napoli, C., Giordano, M., Furnari, M., & Gregorio, S. (2001). A network of cellular automata for a landslide simulation. In *International conference on supercomputing* (pp. 419–426). Sorrento, Italy.
- Clark labs: IDRISI. In *IDRISI*. <<http://www.idrisi.com/>> (Accessed 2009).
- Di Gregorio, S., Rongo, R., & Siciliano, C. (1999). Mount ontake landslide simulation by the cellular automata model SCIDDICA-3. *Physics and Chemistry of the Earth, Part A – Solid Earth and Geodesy*, 24(4), 31–137.
- Emmeche, C. (1998) *Vida Simulada en el ordenador*. Gedisa, Barcelona.
- Fonseca i Casas, P., Casanovas, J., & Montero, J. (2004a). A cellular automata and intelligent agents use to model natural disasters with discrete simulation. In *Proceedings EMS 2004*.
- Fonseca i Casas, P., & Casanovas, J. (2005). Simplifying GIS data use inside discrete event simulation model through m:n-ac cellular automaton. In C. Brianco, C. Frydman, A. Guasch, P. Angel, (Eds.), *Environmental modeling and simulation symposium* (pp. 7–15). Marsella, France.
- Fonseca i Casas, P., SDL, A graphical language useful to describe social simulation models. In F. Quesada, (Ed.), *2nd Workshop on social simulation and artificial societies analysis (SSASA'08)*, Barcelona.
- Fonseca i Casas, P., Casanovas, J., & Montero, J. (2004b) GIS and simulation system integration in a virtual reality environment. In *Proceedings GISRUK 2004* (pp. 403–408).
- Institut Cartogràfic de Catalunya: Institut Cartogràfic de Catalunya. In *Institut Cartogràfic de Catalunya*. <<http://www.icc.cat>> (Accessed 2009).
- Kronholm, K., & Birkeland, K. (2005). Integrating spatial patterns into a snow avalanche cellular automata model. *Geophysical Research Letters*, 32(19).
- Leaf, C., & Martinelli, J. (1977). *Avalanche dynamics: engineering applications for land use planning*. Port Collins, Colo. 8052 1.
- López, J., Casas, P., & Casanovas, J. (2008). SDL formalization of a hydrologic model. In *Applied simulation and modelling*, Corfú (Grece).
- McGraw-Hill: *Encyclopedia of Science and Technology* (5th ed.). The McGraw-Hill Companies, Inc.
- Mears, A. (1976). *Guidelines and methods for detailed snow avalanche hazard investigations in Colorado*, Denver, Colorado.
- Mears, A. (2002). Avalanche dynamics. In *Avalanche.org*. <<http://www.avalanche.org/~moonstone/zoning/avalanche%20dynamics.htm>>.
- METEOCAT: Muntanya. In *Servei Meteorològic de Catalunya*. <<http://www.meteocat.com>> (Accessed 2009).
- Ntaimo, L., & Zeigler, B.P. (2004). Expressing a forest cell model in parallel DEVS and timed Cell-DEVS formalisms. In *Proceedings of SCSC*, San Jose, CA.
- Oller, P. (2009). Caracterització de les allaus majors registrades al Pirineu de Catalunya. Institut Geològic de Catalunya.
- REAF: Mapa de cobertes del sòl de Catalunya. In *Creaf*. <<http://www.creaf.uab.es/mcsc/mms/index.htm>> (Accessed 2009).
- Reed R. *SDL-2000 form New Millenium Systems* (pp. 20–35). Telektronikk 4.2000.
- SDL Tutorial. In *IEC International engineering consortium*. <<http://www.iec.org/online/tutorials/sdl/>>.
- Telecommunication standardization sector of ITU: Specification and description language (SDL). In *Series Z: Languages and general software aspects for telecommunication systems*. <<http://www.itu.int/ITU-T/studygroups/com17/languages/index.html>> (Accessed 1999).
- Webb, B., & Wainer, G. (2008). Implementing the SCIDDICA landslide model in Cell-DEVS. In *Spring simulation multiconference*.

Pau Fonseca obtained his master degree in Computer Engineering (1999) from the Universitat Politècnica de Catalunya and his Ph.D. on Operations Research area in 2007. He works in the LCFIB (Barcelona Informatics School Computing Laboratory) developing Simulation research projects since 1998. Member of LogiSim and LIAM groups, dedicated to the research and simulation tools development.

Màxim Colls obtained his degree in Computer Engineering (2009) from Universitat Politècnica de Catalunya. Now he works in LCFIB developing simulation models and tools.

Josep Casanovas obtained a PhD in Computer Science (1983), degrees in Industrial Engineering and in Economics and he is professor at Statistics and Operations Research Department of the Universitat Politècnica de Catalunya (UPC) in Barcelona. His main research areas are Simulation, Internet and Information Systems and he has participated in many projects for the EU and different companies and institutions.