Advanced Cell-DEVS Model to Demonstrate Vegetable Ecosystem

Babak Simaie
The Ottawa-Carleton Institute for Electrical and Computer engineering
bsimaie@connect.carleon.ca

ABSTRACT
Discrete EVent Systems Specification (DEVS) is an object-oriented computational environment based on DEVS formalism and has been used in modeling and simulation of various real world systems. In this paper report, I proposed modeling and simulation of the vegetable ecosystem environment based on the Cell-DEVS specification using CD++ toolkit. The propose system is based on coupled Cell-DEVS model described as a composition of atomic and coupled components. The model is implemented thoroughly debugged and tested and the results are presented.

Keywords
DEVS formalism, Atomic and Coupled DEVS, Coupled Cell-DEVS, CD++

1. INTRODUCTION
Traditionally, the formal modelling, or modeling, of systems has been via a mathematical model, while computer simulations might use some algorithms from purely mathematical models, computers can combine simulations with reality of actual events, such as generating input responses, to simulate test subjects who are no longer present. In other words, a computer modeling and simulation is a computer program, or network of computers, that attempts to model and simulate an abstract model of a particular system. Computer simulations have become a useful part of mathematical modelling of many natural systems in physics (computational physics), chemistry and biology.

In a simulation modeling of a system the events occurrence (or trigger) and their duration (time) is defines the type of the system simulation that is either continues or discrete i.e. this report make use of a discrete event simulation (DES) or DEVS formalism that manages events in time. Most computer, logic-test and fault-tree simulations are of this type. In this type of simulation, the simulator maintains a queue of events sorted by the simulated time they should occur. The simulator reads the queue and triggers new events as each event is processed.

More specifically a special type of DEVS called Cell-DEVS simulation that is based on the Cellular Automata used in our system.

The system aim to model and simulate a comprehensive vegetable ecosystem using Coupled Cell-DEVS simulation. We begin in Section 1 with a background overview of DEVS simulation and formalism and CD++ toolkit. In section 2 the components of the system are described and a overall conceptual description of the system given. Section 3 is designed to explain simulation scheme details used in population & growth evolution and Plant shape & growth models. Section 4 offers a formal Cell-DEVS description and elaborates on the simulation implementation and the relations between components and the complete set of rules used in our model. Section 5 illustrates the results of the simulation and some test cases and compare the result and test analysis of the system. Finally in section 6 and 7 the conclusion and future direction are discussed. Some test patterns and compare the result

1.1 Background to DEVS M&S
Modeling techniques for discrete-event systems only appeared recently and simulation of these applications was related to the creation of the computer and rapid development in the field of computing. Modeling techniques with a solid mathematical background are more recent; where the first Discrete-Event M&S approaches were tightly coupled to the computer hardware and (formal) languages.

DEVS Modeling and Simulation [4] theory is one of the new techniques, which was based on Systems Theory concepts [1][2][3]. In DEVS theory, a real system is seen as a source of behavioral data for the study within a given experimental frame (EF) as depicted in Figure 1. In this model, experiment frame is a set of components under observation with a given condition. It contains the source system under study and its behavior data. The model or abstract representation of the system is created exploiting the data from EF. In fact, the model contains a simplified version of reality and its structure. Then the model is used to build a simulator, i.e., a device capable of executing the model's instructions and generating the model's behavior. Eventually, the M&S model can be formalized as a Mathematical Dynamic System in which the mathematical entity simulator is able to correctly execute the behavior described by the mathematical entity model.

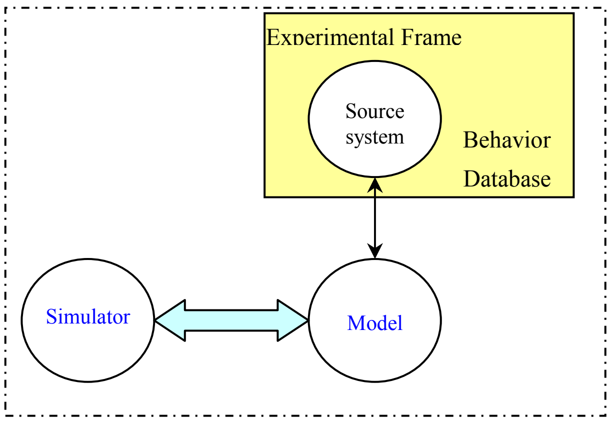


Figure 1. M&S System Structure

1.2 Introduction to DEVS

Modeling techniques are classified according to the system dynamics based on discreteness and continuousness of the time and state event. DEVS was created for modeling and simulating of Discrete-Event Dynamic Systems (DEDS), thus, it defines a way to specify systems whose states change either upon the reception of an input event or due to the expiration of a time delay.

1.2.1 Atomic and Coupled DEVS formalism

Structure of DEVS model can be defined as a set of inputs, outputs, and internal states expressed in a language, such as the mathematical DEVS "Formalism". And "What output is produced upon a given input?" explains Behavior of Model. DEVS can be described as a composition of atomic and coupled components. Atomic model formalism is specified as:

$$M = < X, Y, S, \delta_{\text{int}}, \delta_{ext}, \lambda, ta >$$

Where:

X: set of input values.

S: set of states.

Y: set of output values.

δint: Internal transition function.

δext : External transition function.

λ: Output Function.

ta : Time advance function

In any given moment, a DEVS model is in a state $s \in S$. In the absence of external events, it remains in that state for a lifetime defined by ta(s). When ta(s) expires, the model outputs the value λ(s) and then changes to a new state given by δint(s). A transition that occurs due to the consumption of time indicated by ta(s) is called an internal transition. On the other hand, an external transition occurs due to the reception of an external event. In this case, the external transition function determines the new state, given by δext(s, e, x) where s is the current state, e is the time elapsed since the last transition and $x \in X$ is the external event that has been received. The time advance function can take any real value between 0 and infinity. If ta(s) = ∞ then s is said to be a passive state, in which the system will remain perpetually unless an external event is received (a condition that can be used for termination of the simulation).In Figure 1.2 the DEVS formalism interpretation is illustrated.

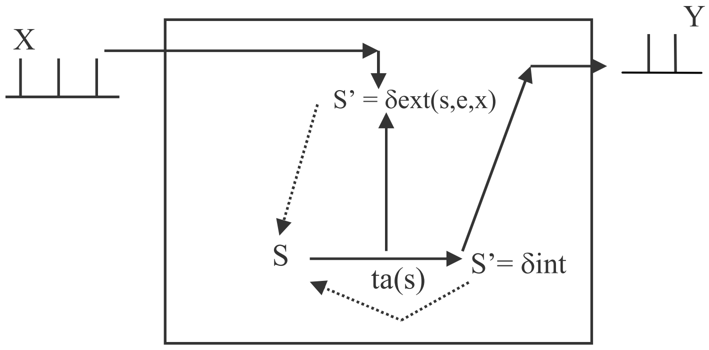


Figure 2. DEVS Model

DEVS coupled model compose of number of above atomic model and its formalism is specified as:

$$CM = < X, Y, D, \{Mi\}, \{Ii\}, \{Zij\}, select >$$

Where:

Mi is an atomic or coupled DEVS model as previously described.

Ii is the set of influencers.

Zij is the *i* to *j* translation function.

D is the index for the components

1.2.2 Cell DEVS formalism

If the system under study organized as a grid of cells that geometrically distributed a well-known formalism named cellular automata (CA) [5] can be used to describe it. It consists of a regular n-dimensional lattice grid of cells, each in one of a finite number of states. It is a discrete model and Time is also discrete, and the state of a cell at time t is a function of the states of a finite number of cells (called its neighborhood) at time $t - 1$. Every cell has the same rule for updating, based on the values in this neighborhood. Each time the rules are applied to the whole grid a new generation is created. A formal definition of CA called Cell-DEVS formalism [6] was considered to allow the defining the cell spaces based on DEVS and CA models.

Cell-DEVS can also be described as a composition of atomic and coupled components. Cell-DEVS atomic models can be described as in Figure 3. Each cell uses N inputs (from its neighborhood) to compute its next state. These inputs, which are received through the model's interface, activate a local computing function (τ). A delay (d) can be associated with each cell. The state (s) changes can be transmitted to other models, but only after the consumption of this delay.

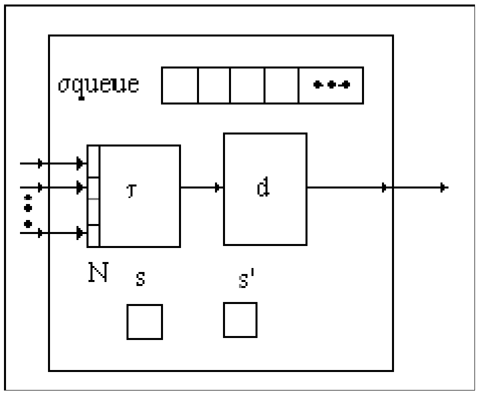


Figure 3. DEVS Informal Description of Cell-DEVS (transport)

Two kinds of delays can be defined: *transport* delays model a variable commuting time, and *inertial* delays, which have preemptive semantics (scheduled events can be discarded if the computed value is different than the future state).

The formal description of the Cell DEVS is as follows:
Cell-DEVS Atomic Model:

$$TDC = < X, Y, I, S, \theta, N, d, \tau, \delta_{int}, \delta_{ext}, \lambda, D>$$

Where:

$X \subset R$ is the set of input external events;

$Y \subset R$ is the set of output external events;

I :is the model's modular interface;

S :is the set of possible states for a given cell;

θ :is the definition of the cell's state variables;

N :is the set of the input events;

d :is the delay of the cells;

τ :local computing function;

δ_{int} :is the internal transition function;

δ_{ext} :is the external transition function;

λ :is the output function;

D :is the duration function;

Cell-DEVS Coupled Model:

$$GCC = < X_{list}, Y_{list}, I, X, Y, \eta, \{t_1,...,t_n\}, N, C, B, Z, select >$$

Where:

X_{list}, Y_{list} : are the list of input, output coupling

η :is the neighborhood size;

N :is the neighborhood ;

C :is the cell space set;

B :is the boarder cells set;

Z :is the transition function;

Select :is the tie-breaking selector function;

There are several tools that are implemented based on the DEVS M&S theory. The CD++ tool [7] is one of them. The simulation engine tool is built as a class hierarchy of models in C++. Coupled and Cell-DEVS models are created using the language built in the engine.

In CD++, Cell DEVS models are a special case of coupled models. Therefore, when defining a cellular model, all the coupled model parameters are available in addition to parameters used for cellular models.

In this section a three-dimension structure of the heat diffusion model including the three components: a 3D space reproducing the behaviour of a room, and two generators (one source of heat and one source of cold) is described. A cell can be connected to heat generator, a cold generator or none. A cell's temperature is measured as the average of its neighborhood. It's possible to see the cell's temperature in each step.

The temperature is calculated as the average of the temperature in the cell and its eight next near neighbors that is shown in Figure 4(a). Also, two generators (one source of heat and one source of cold) generates a flow of temperatures with uniform distribution are connected to two cells in the model as depicted in Figure 4(b).

Room Cell-DEVS model is defined as 3D dimensional model including the grid size, kind of delay and border. It is composed by a 4*X*4*X*4. The heat-rule local computing function calculates the present value as an average between all the neighbouring cells. Two temperature value in the ranges of [24, 80] and [-45, 10] are set respectively, using a uniform probabilistic distribution (setHeat, setCold) and received through the In port of the cells (3,3,0), (2,2,1), (3,3,2) and (1,3,3).

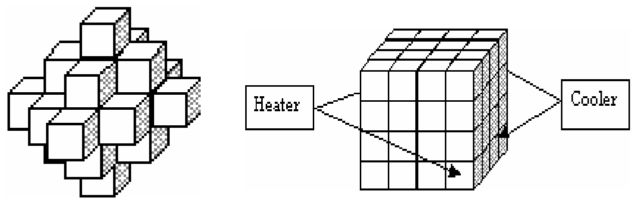


Figure 4. (a) Neighborhood shape (b) Coupling scheme

The model of heat diffusion is a Cell-DEVS Coupled model that has other DEVS Coupled model and atomic Cell-DEVS model as its components. For the heat diffusion a implementation in CD++ is presented in Figure 5. The first part of the code represents the top level model that lists the components of the coupled model and also the internal links and external 'in' , 'out' ports (if any). The Heater and Cooler are two Coupled DEVS where the heater and cooler are connected via *out@Heater* and *Out@Cooler* respectively to the *inputHeat* and *inputCold* of the cell space external port. Then the in ports are connected to a specific port through *portInTransition*. The *[room]* symbol introduces the room sub-model that is a Cell-DEVS component of 4X4X4 dimension. For each of its cells we have a specific number of neighbour cells that are defined in the specification of the *neighbours*. A local computing function (τ) and a delay (d) can be associated with each cell and express in the *[heat-rule]*.

Upon start of the simulation, the heater/cold sources produce changes in the cells in each time step (or delay=d) define by 1000msec where they are connected. Consequently, the state of the neighbours of these cells will change in time as it shown in Figure 6.

As one can see it contains 4 of 4X4 tables in a row that represent the heat 4X4X4 heat Cell-DEVS simulation results. It is primarily a draw file that is produced from LOG file in CD++. Each row shows the result of the local computing function after the delay time. Consequently the number of rows depends on the delay time and the duration of the simulation.

```
[top]
components:room Heater@Generator
Cooler@Generator
link : out@Heater inputHeat@room
link : out@Cooler  inputCold@room

[room]
type : cell
dim : (4, 4, 4)
delay : transport
defaultDelayTime: 100
border:wrapped
neighbors : room(-1,0,-1) room(0,-1,-1)
room(0,0,-1) room(0,1,-1)
...
in : HeatInput ColdInput
link : HeatInput in@room(3,3,0)
link : HeatInput in@room(2,2,1)
link : ColdInput in@room(3,3,2)
link : ColdInput in@room(1,3,3)

localtransition : heat-rule

portInTransition : in@room(3,3,0)
in@room(2,2,1) setHeat
portInTransition : in@room(3,3,2)
in@room(1,3,3) setCold

[heat-rule]
Rule:  {  (  (-1,0,-1)+(0,-1,-1)+(0,0,-
1)+(0,1,-1)+(1,0,-1)+(-1,-1,0)+(-
1,0,0)+(-1,1,0)+(0,-
1,0)+(0,0,0)+(0,1,0)+(1,-
1,0)+(1,0,0)+(1,1,0)+(-1,0,1)+(0,-
1,1)+(0,0,1)+(0,1,1)+(1,0,1)+(0,0,-2)+
(0,0,2)+(0,2,0)+(0,-2,0)+(2,0,0)+(-
2,0,0) ) / 25 } 1000 { t }

[setHeat]
rule: { uniform(24,80) } 1000 { t }

[setCold]
rule: { uniform(-45,10) } 1000 { t }

[Heater]
distribution: exponential
mean : 10
initial : 1


[Cooler]
distribution : exponential
mean : 10
initial : 1
```

Figure 5. Model of heat diffusion

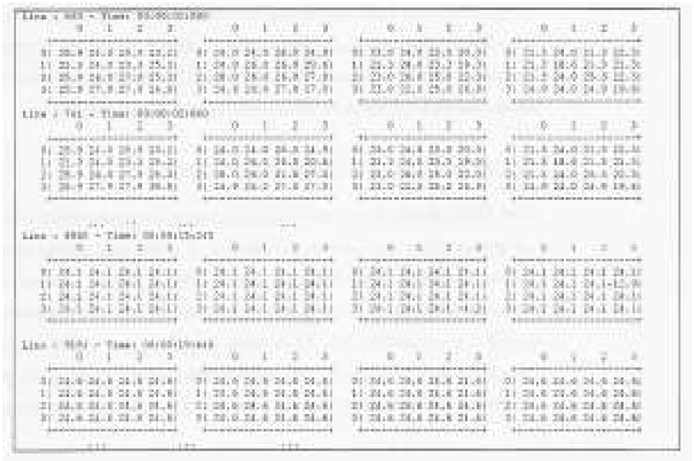


Figure 6. Some of Simulation Execution Results

2. Conceptual Model

Modelling the dynamics of vegetable ecosystem is an extremely challenging problem in which we have very complex system behaviour. It is an interdisciplinary research that involves computer science, biology, and ecosystem management.

Traditional models for the vegetable ecosystem study are continuous and based on differential equations and usually model the evolution of the system with global parameters such as the total number of trees and their overall biomass [10][11]. In addition, most of the data needed to provide reliable parameters for these models are usually scarce and difficult to obtain.

More recently, the cellular automata has been used in [8][9] to model and simulate the vegetable ecosystem, however just the evolution of a single species is studied in these models. Moreover some key parameters are missing in the study such as shape evolution in the growth process of the vegetable.

In this report I proposed a comprehensive ecosystem analyses primarily inspired by the Bandini and Pavesi Work [12] and modified in [13]. Different rules of L-system apply to construct various types of vegetables (Patterns) and their growth model [14]. The model aims to evaluate the influences of the ecosystem on the vegetable existence, population, growth and shape in a single and consolidated model and simulator.

Proposed comprehensive vegetable ecosystem modelling and simulation can be comprised of the following models as it shown in Figure 7; Model of the variety of vegetable shapes and patterns that is, of all the weeds, plants and trees in a given area, model of the horizontal and vertical growth as well as shape evolution of the vegetables, model of the vegetable existence and population and model of Growth, shape and population influenced by the resources available on the ecosystem of the vegetable (i.e. sunlight, water, weather and substances present in the soil) and the interactions among single individuals and their competition for the resources available on the ecosystem.

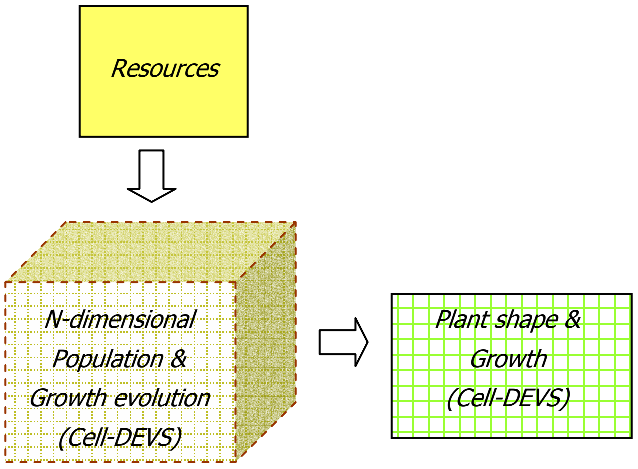

Resources
N-dimensional Population & Growth evolution (Cell-DEVS)
Plant shape & Growth (Cell-DEVS)

Figure 7. Conceptual Model Structural model

In the proposed structure, a discrete model based on 3–dimensional Cellular Automata has been used with different layers that allow to model and simulate the evolution of heterogeneous vegetable populations composed by different perennial species as in real woods and forests. Also a 2-dimensional Cellular-DEVS model simulates the growth processes of tree development using in tree morphology. The resource block produce seeds and plant them in N-dimensional Population & Growth evolution (Cell-DEVS) part. The sells containing the seed and ready to grow are mapped to the Plant shape &Growth (Cell-DEVS) part in order to demonstrate different shapes of the vegetables used in the ecosystem.

3. Simulation Scheme Details

3.1 Population & Growth evolution

As indicated in the pervious section a Cell-DEVS model is used for representing the model. In this section the model used in the implementation details of each section are discussed and its implication on the model will be studied.

Each cell in the cellular automata representing vegetable population dynamic simulation shows a square portion of model that contains some resources (i.e. water, light, nitrogen, and potassium) and can host a tree.

The finite set of cell states values can be defined by Q as:

$$Q = \{R, M, P, T, Z_T, N_T, U_T^G, U_T^S, R_T, M_T, G_T, S, A_T\}$$

Where:

R :the vector referring to the amount of resources in the cell

M :maximum amount of each resource

P :amount of each resource produced by the cell at each update step

T :flag indicating whether a tree is present in the cell

Z_T : vector defining the size of the different parts of the tree

N_T :amount of each resource the tree takes from the cell

U_T^G :amount of each resource needed at each update step by tree to grow

U_T^S :amount of each resource needed at each update step by tree to survive

R_T :amount of each resource stored by the tree at previous update steps

M_T :vector of threshold values for different parameters defining the tree, such as maximum size, maximum age, minimum age for reproduction, maximum number of seeds produced for each mass unity of fruits, and so on

G_T :vector defining the growth rate of each of the parts of the tree when enough resources are available.

S :vector defining the number of seeds present in the cell for each of the species growing in the territory

A_T : vector defining the age of the tree in the cell

The Cellular Automata for vegetable Ecosystems model is based on two–dimensional Cellular Automata, whose cells, arranged on a square grid, represent portions of a given area. Some resources are present on the area, divided among the cells. A cell can host a tree, represented in the model by a set of parameters defining its species, its size, and the amount of each resource it needs to survive, grow, and/or reproduce itself. Some of the parameters of Q can be future define as:

$R = \{(i, j) | 1 \le i \le N, 1 \le j \le M\}$ is a two-dimensional NXM lattice that defines the dimension of the area under study.

IF we define H as the Moore neighborhood,

$f : Q \times Q^{|H|} \rightarrow Q$ is the state transition function and

$I : R \rightarrow Q$ is the initialization function

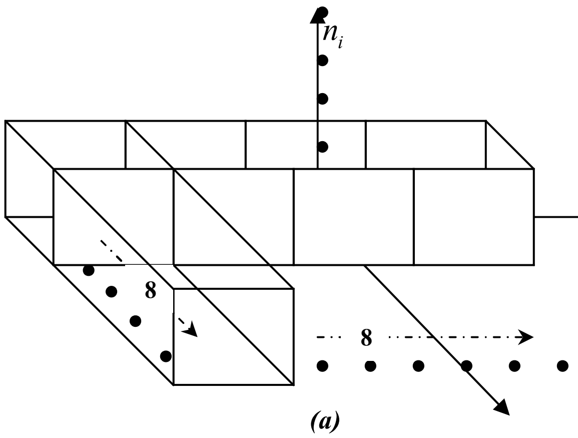
Now based on the [13] we define update rules as follow: The initial configuration of the CA can be defined by setting some trees or seeds of a specific type in several cells and appropriate resource parameters (or resources) for each cell.

At the initial step three cases might happen first we have seeds, second already planed trees and third combination of both in the cells.

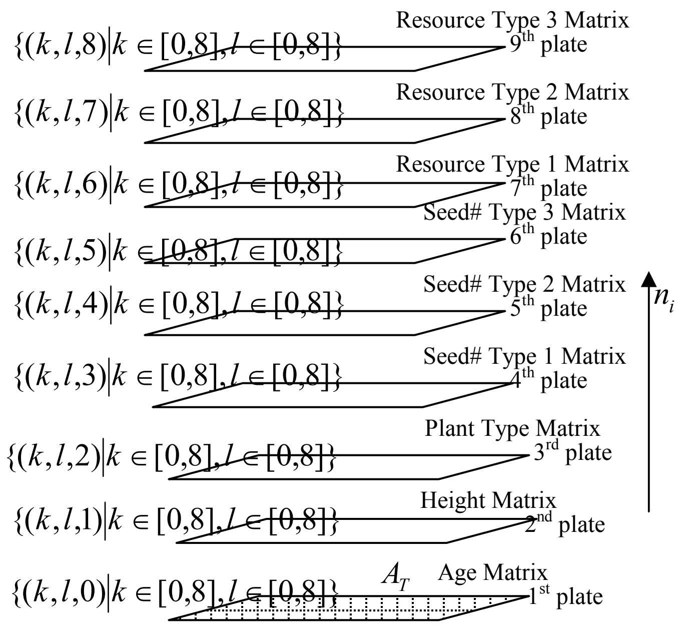
In the least former case, at each update step the seed(s) present(s) in the cell start to grow if enough resources are available, otherwise the tree dies and seed disposed. In the second case the already grown trees are planted in each cell (if any) takes the resources it needs from the cell itself and uses them to survive, grow (if enough resources are available), and produce seeds(if it has reached minimum age for reproduction and enough resources are available). A cell can host more than one type of the seed and several seeds that define with the S vector in the model. However the seed do not sprout in a cell unless the cell is free without a tree then the seed with more number of seed defines the type of tree to grow. If a tree is present in cell C (i, j) [C (i, j) is the cell located at position (i, j) in the model lattice], A_T vector defining the age of the tree in the cell indicating the already grown tree age.

It is worth noticing that there is a limit for the resources allowed to be in each individual cell. Also the trees can reach a certain age and height and can not over grow.

To implement the initiation part we consider an 8X8Xn_i space in which the model under study is presented by 8X8 plate and n_i represent various parameters layers used in the simulation. The Figure 8(a) indicates the cell space used for the model, and 8(b) plate parameters.



(a)


Resource Type 3 Matrix
9th plate
Resource Type 2 Matrix
8th plate
Resource Type 1 Matrix
7th plate
Seed# Type 3 Matrix
6th plate
Seed# Type 2 Matrix
5th plate
Seed# Type 1 Matrix
4th plate
Plant Type Matrix
3rd plate
Height Matrix
2nd plate
Age Matrix
1st plate
A_T
n_i
$\{(k,l,8)|k \in [0,8], l \in [0,8]\}$
$\{(k,l,7)|k \in [0,8], l \in [0,8]\}$
$\{(k,l,6)|k \in [0,8], l \in [0,8]\}$
$\{(k,l,5)|k \in [0,8], l \in [0,8]\}$
$\{(k,l,4)|k \in [0,8], l \in [0,8]\}$
$\{(k,l,3)|k \in [0,8], l \in [0,8]\}$
$\{(k,l,2)|k \in [0,8], l \in [0,8]\}$
$\{(k,l,1)|k \in [0,8], l \in [0,8]\}$
$\{(k,l,0)|k \in [0,8], l \in [0,8]\}$

(b)

Figure 8. (a)Cell-DEVS Model Shape (b) Parameters in n_i layer

The initial value for the whole model space cells is "0" that means the presences of the seed in the initial stage. In the other words, it implies the zero in the age matrix whose age is zero. It is worth noticing that unless the type of the vegetable assigns in the correspondent cell in other plate, the zero has no meaning the program logic.

As indicated above a specific plate is designated to assign the type of vegetable for the model at is input to the simulation at the beginning. Basically these two plates are responsible for the initial stage of the program.

The initial values for the model can be apply at the beginning and further values can be apply during simulation by the "generateSource@Generator" using

"link : out@generateSource inputSource@plant".

Upon start of the simulation at each update step of the simulation, the tree present in each cell (If a tree is present in cell $C(i,j)$) starts the growth evolution and takes the resources (defined by $N_T(i,j)$ of each available resource $R(i,j)$) it needs from the cell that already assign in the initial stage and consumes them to survive, grow (if enough resources are available), and produce seeds. The amount of resources taken depends on the size of the tree $Z_T(i,j)$. If enough resources (those taken at this step, plus the resources stored at previous steps), are available, as defined by vector U_T^G , the tree grows. Otherwise, the resources might be just sufficient for the tree to survive (as defined by vector U_T^S).In both cases, the tree "burns" an amount of each resource, as defined by vector $U_T^G(i,j)$ or $U_T^S(i,j)$ .If none of above cases occurs, the overall amount (stored plus collected) of at least one resource is under the "survival threshold" of the tree, the plant dies. The tree also dies when it reaches its maximum age defined in vector $M_T(i,j)$.

In this model, the resources are balanced among neighboring cells in each step. It means the remaining resources re-distributed among the cell and its neighbours. The 9 neighbours (Moore model) are listed below:

1. $P_{ij}{}^{Y}{}_{1} \rightarrow P_{ij-1}{}^{X}{}_{1}$	$P_{ij+1}{}^{Y}{}_{1} \leftarrow P_{ij+1}{}^{X}{}_{1}$
2. $P_{ij}{}^{Y}{}_{2} \rightarrow P_{i+1j}{}^{X}{}_{2}$	$P_{ij}{}^{Y}{}_{2} \leftarrow P_{i-1j}{}^{X}{}_{2}$
3. $P_{ij}{}^{Y}{}_{3} \rightarrow P_{ij+1}{}^{X}{}_{3}$	$P_{ij}{}^{Y}{}_{3} \leftarrow P_{ij-1}{}^{X}{}_{3}$
4. $P_{ij}{}^{Y}{}_{4} \rightarrow P_{i-1j}{}^{X}{}_{4}$	$P_{ij}{}^{Y}{}_{4} \leftarrow P_{i+1j}{}^{X}{}_{4}$
5. $P_{ij}{}^{Y}{}_{5} \rightarrow P_{ij}{}^{X}{}_{5}$	$P_{ij}{}^{Y}{}_{5} \leftarrow P_{ij}{}^{X}{}_{5}$
6. $P_{ij}{}^{Y}{}_{6} \rightarrow P_{i-1j-1}{}^{X}{}_{6}$	$P_{ij}{}^{Y}{}_{6} \leftarrow P_{i-1j-1}{}^{X}{}_{6}$
7. $P_{ij}{}^{Y}{}_{7} \rightarrow P_{i-1j+1}{}^{X}{}_{7}$	$P_{ij}{}^{Y}{}_{7} \leftarrow P_{i-1j+1}{}^{X}{}_{7}$
8. $P_{ij}{}^{Y}{}_{8} \rightarrow P_{i+1j-1}{}^{X}{}_{8}$	$P_{ij}{}^{Y}{}_{8} \leftarrow P_{i+1j-1}{}^{X}{}_{8}$
9. $P_{ij}{}^{Y}{}_{9} \rightarrow P_{i+1j+1}{}^{X}{}_{9}$	$P_{ij}{}^{Y}{}_{9} \leftarrow P_{i+1j+1}{}^{X}{}_{9}$

Now if we consider $R'(i,j)$ the amount of each cell after the simulation step and $R(i,j)$ before that we have:

$$r'(i,j)=\frac{r(i,j)+\frac{r(i+1,j)+r(i,j+1)+r(i-1,j)+r(i,j-1)+r(i+1,j+1)+r(i+1,j-1)+r(i-1,j-1)+r(i-1,j+1)}{8}}{2}$$

In other words, we can see each cell as divided in eight parts, each one containing the amount $r(i,j)/8$ of resource h , and corresponding to one of the neighbours. The amount of resource h contained in each part is balanced with the corresponding part of the neighbours.

We have to consider in above equation $r'(i,j)$ cannot exceed the corresponding maximum value defined for the cell ($M(i,j)$).In this case, we set $M(i,j) = r'(i,j)$.

We have two cases to consider: a tree is present in the cell, or the cell is empty. In the former case,

The reproduction of a tree may happen (production of some seeds) if it is old enough. Then the model updates the corresponding variable in the seed vector $S(i,j)$ .The new trees cannot sprout from the seeds contained in the cell if a tree is already present. Instead, when the cell is vacant and contains some seeds. If the resources present in the cell are sufficient a new tree is born.

If seeds from different species are present in the cell, the seed type with higher number has more probability number and can be chosen.

3.2 Plant shape & Growth

Another component of the system is Plant shape simulation. I proposed to use a Cellular-DEVS model to implement a model and simulate the growth processes of tree development using in tree morphology. The aim of this simulation is to demonstrate the growth toward sunlight and also using Cell DEVS simulation.

In my proposed model as soon as the seed will be planted into a cell the tree starts seedling and the trunk comes out of the earth and growth towards the light that is above the seed.To simulate the tree growth I exploit Lindenmayer system that is a formal grammar (a set of rules and symbols) most famously used to model the growth processes of plant development.

Lindenmayer systems, or L-systems for short, are a particular type of symbolic dynamical system with the added feature of a geometrical interpretation of the evolution of the system. They were invented in 1968 by Aristid Lindenmayer to model biological growth.

Formal description of L-System is a tuple G = < V, w, P > consisting of:

V : an alphabet is a set of symbols containing elements that can be replaced (variables)

w : a non-empty starting word (or axiom) defining the initial state of the system

P : is a set of production rules or productions defining the way variables can be replaced with combinations of constants and other variables.

Depending on the context, letters of the alphabet could represent cells or modules, e.g., V= {Seed, Trunk, Branch}

w = Seed

P = rules of tree growth

Now we exploit the L-system for modeling and visualization of the growth of a tree as below in the Cell DEVS model:

Variables: X F

Constants: + −

Start: X

Rules: (X → F[+F]F[-F]F) , (F → FF)

Angle: 45°

Here, F means "draw forward", - means "turn left 45º", and + means "turn right 45º". X does not correspond to any drawing action and is used to control the evolution of the curve. [Corresponds to saving the current values for position and angle, which are restored when the corresponding] is executed. In fact, Square brackets, are defined that mark the beginning and end of a branch.

4. Implementation Details

This part aims to explain thoroughly the details of the implementation of our Coupled Cell-DEVS system based on the scheme described in the pervious section.

First we introduce the formal Couple Cell-DEVS model of our implementation, then the top level of the system and the relations between components. At the end each individual block of the system is discussed.

4.1 Formal Description

This section presents the formal specification of each of the Cell-DEVS models:

For the shape, Cell-DEVS model the formal description is as follow:

GCTD = < *X*, *Y*, Xlist, Ylist, I, η, N, {i, j, k}, C, B, Z, select >

where for #T < ∞ ∧ T ∈{*N*, *Z*, ***R***, *{0,1}* } ∪ {ϕ}

- ***X*** ⊆ T is the set of external input events; T = {seed}
- ***Y*** ⊆ T is the set of external output events; T = {growth}
- $\mathbf{Y_{list}}$ = { (i, j, k,) / i ∈ [0,m], j ∈ [0,n], k ∈ [0,o]} is the list of output coupling. Where the i, j, k represent the index values of the cells (that couple with its neighbors) which are bound by m, n, o dimension.
- **Xlist** = { (i, j, k,) / i ∈ [0,m], j ∈ [0,n], k ∈ [0,o]} is the list of input coupling. Where the i, j, k represent the index values of the cells (that couple with its neighbors) which are bound by m, n, o dimension.
- **I** = < Px, Py, Pz > represents the definition of the modular model interface. Here,
- for *i* = X | Y|Z, P*i* is a port definition (input or output respectively). For example, the resource exchanges and the confirmation of rules will be established by communication through these ports (which enable identification of the values of resources in the cell).
- η = *9* and the neighbor list set is given as follows:

N = {(-1,-1), (-1,0), (-1,1), (0,-1), (0,0), (0,1), (1,-1), (1,0), (1,1)}

These illustrated scenarios show the neighborhood list for the immediate neighbors in the same plane (nine of them) and their inverse list. The same applies for the other neighbors.

- C = {Cijk / i ∈ [0,19], j ∈ [0,30]}
- B = {∅} if the cell space is wrapped; or
- **Z** is the translation function, which determines the dynamics of the vegetable population (refer to the rules section).

And for the Population model the Coupled Cell-DEVS can be defined as the following

GCTD = < *X*, *Y*, Xlist, Ylist, I, η, N, {i, j, k}, C, B, Z, select >

where for $\#T < \infty \wedge T \in \{N, Z, \boldsymbol{R}, \{0,1\}\} \cup \{\phi\}$

- $\boldsymbol{X} \subseteq T$ is the set of external input events; T = {water, minerals, sunlight}
- $\boldsymbol{Y} \subseteq T$ is the set of external output events; T = {growth}
- $\mathbf{Y}_{\mathbf{list}}$ = { (i, j, k,) / i ∈ [0,m], j ∈ [0,n], k ∈ [0,o]} is the list of output coupling. Where the i, j, k represent the index values of the cells (that couple with its neighbors) which are bound by m, n, o dimension.
- **Xlist** = { (i, j, k,) / i ∈ [0,m], j ∈ [0,n], k ∈ [0,o]} is the list of input coupling. Where the i, j, k represent the index values of the cells (that couple with its neighbors) which are bound by m, n, o dimension.
- **I** = < Px, Py, Pz > represents the definition of the modular model interface. Here,
- for *i* = X | Y|Z, P*i* is a port definition (input or output respectively). For example, the resource exchanges and the confirmation of rules will be established by communication through these ports (which enable identification of the values of resources in the cell).
- η = *61* and the neighbor list set is given as follows:

N = {(-1,-1,0) , (-1,0,0), (-1,1,0), (0,-1,0), (0,0,0), (0,1,0) , (1,-1,0), (1,0,0), (1,1,0), (-1,-1,1) , (-1,0,1), (-1,1,1), (0,-1,1), (0,0,1), (0,1,1) , (1,-1,1), (1,0,1), (1,1,1), (-1,-1,2) , (-1,0,2), (-1,1,2), (0,-1,2), (0,0,2), (0,1,2) , (1,-1,2), (1,0,2), (1,1,2),(-1,-1,3) , (-1,0,3), (-1,1,3),(0,-1,3), (0,0,3), (0,1,3), (1,-1,3),(1,0,3), (1,1,3), (-1,-1,-2) (-1,0,-2), (-1,1,-2), (0,-1,-2), (0,0,-2), (0,1,-2) , (1,-1,-2), (1,0,-2), (1,1,-2), (0,0,-1), (0,0,-2), (0,0,-3), (0,0,-4), (0,0,-5), (0,0,-6),(0,0,-7),(0,0,-8),(0,0,1),(0,0,2),(0,0,3),(0,0,4),(0,0,5), (0,0,6), (0,0,7), (0,0,8)}

Now,	The corresponding inverse neighborhood would be,
1. $P_{ij}{}^{Y}{}_{1} \rightarrow P_{ij-1}{}^{X}{}_{1}$	$P_{ij+1}{}^{Y}{}_{1} \leftarrow P_{ij+1}{}^{X}{}_{1}$
2. $P_{ij}{}^{Y}{}_{2} \rightarrow P_{i+1j}{}^{X}{}_{2}$	$P_{ij}{}^{Y}{}_{2} \leftarrow P_{i-1j}{}^{X}{}_{2}$
3. $P_{ij}{}^{Y}{}_{3} \rightarrow P_{ij+1}{}^{X}{}_{3}$	$P_{ij}{}^{Y}{}_{3} \leftarrow P_{ij-1}{}^{X}{}_{3}$
4. $P_{ij}{}^{Y}{}_{4} \rightarrow P_{i-1j}{}^{X}{}_{4}$	$P_{ij}{}^{Y}{}_{4} \leftarrow P_{i+1j}{}^{X}{}_{4}$
5. $P_{ij}{}^{Y}{}_{5} \rightarrow P_{ij}{}^{X}{}_{5}$	$P_{ij}{}^{Y}{}_{5} \leftarrow P_{ij}{}^{X}{}_{5}$
6. $P_{ij}{}^{Y}{}_{6} \rightarrow P_{i-1j-1}{}^{X}{}_{6}$	$P_{ij}{}^{Y}{}_{6} \leftarrow P_{i-1j-1}{}^{X}{}_{6}$
7. $P_{ij}{}^{Y}{}_{7} \rightarrow P_{i-1j+1}{}^{X}{}_{7}$	$P_{ij}{}^{Y}{}_{7} \leftarrow P_{i-1j+1}{}^{X}{}_{7}$
8. $P_{ij}{}^{Y}{}_{8} \rightarrow P_{i+1j-1}{}^{X}{}_{8}$	$P_{ij}{}^{Y}{}_{8} \leftarrow P_{i+1j-1}{}^{X}{}_{8}$
9. $P_{ij}{}^{Y}{}_{9} \rightarrow P_{i+1j+1}{}^{X}{}_{9}$	$P_{ij}{}^{Y}{}_{9} \leftarrow P_{i+1j+1}{}^{X}{}_{9}$

These illustrated scenarios show the neighborhood list for the immediate neighbors in the same plane (nine of them) and their inverse list. The same applies for the other neighbors.

- C = {Cijk / i ∈ [0,7], j ∈ [0,7], k ∈ [0,8]}
- B = {∅} if the cell space is wrapped; or
- **Z** is the translation function, which determines the dynamics of the vegetable population (refer to the rules section).

4.2 Coupled Cell-DEVS implementation

Figure 9 demonstrate the behaviour of the proposed Cell-DEVS Coupled model:

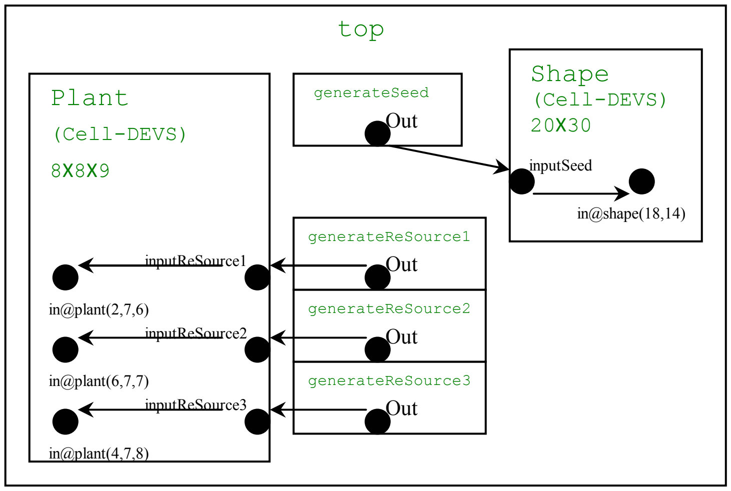


Figure 9. Coupled Cell-DEVS Behavioral model

The file "*ecosystem.ma*" defines our Coupled Cell-DEVS model that includes the components and the relationship between them.

in@shape(18,14)

As you may notice in Figure 9 and code description below the *'[top]'*describes the components and the relations among them. In total there are 6 components that 2 of them defines the population evolution and plants tree named *'[plant]'* and *'[shape]'* respectively. These model are Cell-DEVS that connected to sources output ports: *out@generateReSource1*, *out@generateReSource2*, *out@generateReSource3* and *out@generateSeed* via input ports: *inputReSource1*, *inputReSource2*, *inputReSource3* and *inputSeed*

```
[top]
components : shape plant
generateSeed@Generator
components : generateReSource1@Generator
generateReSource2@Generator
generateReSource3@Generator
link : out@generateSeed inputSeed@shape
link : out@generateReSource1
inputReSource1@plant
link : out@generateReSource2
inputReSource2@plant
link : out@generateReSource3
inputReSource3@plant
```

After definition of the top model we define our components as describe and implemented in "*ecosystem.ma*".

In this section the details of each model is describe and the rules and parameters that are used in the model to produce the simulation results are discussed. Also Comments are provided throughout of the source code to describe various parts functionality.

4.2.1 Population & Growth model Implementation

As described in pervious section we use an 8X8X9 space dimension to represent the population model *([plant])*. the population component of the model comprised of 9 plane presenting different parameters as shown in Figure 10

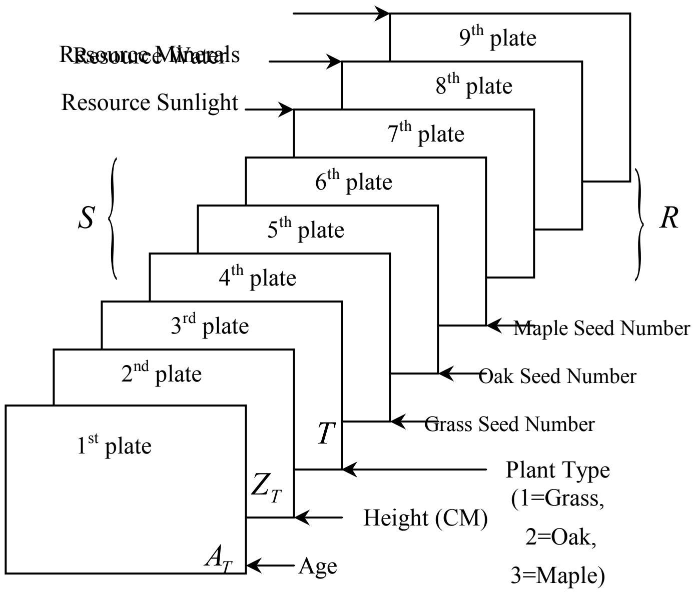


Figure 10. Planes for the Parameters of the population model

To make use of this planes I defined 12 different zones and 61 neighbours as shown in the code below in which the zone name represent the rule for that particular parameter;

```
zone : sunamount { (0,0,6)..(7,6,6) }
zone : mineralamount {(0,0,7)..(7,6,7) }
zone : wateramount { (0,0,8)..(7,6,8) }

zone : addsun { (0,7,6)..(7,7,6) }
zone : addmineral { (0,7,7)..(7,7,7) }
zone : addwater { (0,7,8)..(7,7,8) }

zone : plantage { (0,0,0)..(7,7,0) }
zone : plantheight { (0,0,1)..(7,7,1) }
zone : planttypeproduce{(0,0,2)..(7,7,2)
}
zone : graseed { (0,0,3)..(7,7,3) }
zone : oakseed { (0,0,4)..(7,7,4) }
zone : mapleseed { (0,0,5)..(7,7,5) }
```

The planes dynamically interact with each other and pass the values and compare their parameters and conditions to output results of the simulation. In this part of the report, the set of rules and parameters used in the model population simulation are introduced as follows:

4.2.1.1 [Planttypeproduce]

The Third plate((0,0,2) to (7,7,2)) is designed to represent the vegetable type, I assign three different values for three different type of vegetable as grass=1, Oak=2 and Maple=3.

When the enough resources are available in a cell, The plants start to reproduce seed and spread them to adjunct cells when they reach certain age to produce the fruit. The age for grass is 2 for oak it is 5 and for maple 7.The enough resources

It is important to notice that this plate simulates the reproduction of the seeds and the type seed with the higher value (number of seed) in a cell defines the type of the plant in that cell ready to grow in case of tree absence in that cell. It shows the competition among the seeds to grow in a cell.

In the case that we get equal number of seeds, a priority scheme are implemented in the model that gives grass higher priority than oak and maple and superior priority to oak over maple.

4.2.1.2 [Plantage]

This occupies the cells from (0,0,0) to (7,7,0) and responsible for determining the age of the plant. Each simulation step increases the plants age equally by one unit. The aging process does not affected by the resources however the plant will die if they reach their maximum age and the resources are not adequate to survive. The maximum age is defined as one of the parameters of M_T discussed in section 3.1;

Maximum age = {Grass=4, Oak=55, Maple=60}

This rule also presents dying process which reflected in *Planttypeproduce, Plantheight* and *itself* with presenting zero value when the plant dies.

4.2.1.3 [Plantheight]

Unlike the age, the speed of plant height growth is different with respect to the plant type. Plant height is model in the system where different values are given to different plants to show the different rate of tree height growth. We assume for each year 1cm is added to the grass height, 20 cm to oak and 25cm to maple height. The height growth is not infinite and when the plant reaches certain height stop and dies. This parameter is also defines in M_T Where

Maximum height = {Grass=5cm, Oak=20m, Maple=25m}

The height of the tree is affected by environment (resources) where the resource shortage slows the height grows in my propose model.

4.2.1.4 [graseed] [oakseed] [mapleseed]

Three unique plates are assigning to present the number of the seed (S vector in the 3.1 section) in for our plants. This plate's parameters are used for the reproduction process explained in *planttypeproduce* reproduction process. The rule starts to add the value for incrementally for each year after the plant reaches productivity age as defines in 4.2.1.1

4.2.1.5 [sunamount][mineralamount] [wateramount]

Each plant requires resources such as sunlight, minerals and water to grow or survive or reproduce. The initial resources values are applied as an input file (plant.val) to the system through the 7^{th}, 8^{th} and 9^{th} plates. Also a zone describe later to constantly supply resources to our model to carry on the simulation execution.

Based on the distribution formula discussed in section 3.1 the amount of resources is calculated by subtraction of the resource of the current cell and the plant use to grow. Then the new remaining resource redistribute to the cell and adjacent cells.

As an example, The rules to subtract the resources from current cell resource value and the way the sun light resource value redistribution for the plant type grass is shown in this part of the code as below:

```
%sun amount is calculated by subtraction
of sun light resource of the current
cell and the grass use to grow (2
calories)
%new sun light resource (remaining)
redistribution to the cell and adjacent
cells
rule : { ((0,0,0) - 2) + ((-
1,1,0)+(0,1,0)+(1,1,0)+(-
1,0,0)+(0,0,0)+(1,0,0)+(-1,-1,0)+(0,-
1,0)+(1,-1,0))/9 } 1000 { (0,0,0) >= 2
and (0,0,-4) = 1 and (0,0,1) >= 7 and
(0,0,2) >= 4 }

%sun amount is calculated by subtraction
of sun light resource of the current
cell and the grass use to survive(1
calories)
%new sun light resource(remaining)
redistribution to the cell and adjacent
cells
rule : { ((0,0,0) - 1) + ((-
1,1,0)+(0,1,0)+(1,1,0)+(-
1,0,0)+(0,0,0)+(1,0,0)+(-1,-1,0)+(0,-
1,0)+(1,-1,0))/9 } 1000 { (0,0,0) >= 1
and (0,0,0) < 2 and (0,0,-4) = 1 and
(0,0,1) >= 6 and (0,0,1) < 7 and (0,0,2)
>= 2 and (0,0,2) < 4 }
```

As one may notice there are two separate set of rules for a particular plant type that determined by the vectors U_T^S and U_T^G .These two vectors define amount of each resource needed at each update step by tree to survive and grow respectively. The values for this vector for different resources (*Sunlight, Minerals, and Water*) for the plant types in our model are given in table below:

Resource / Vector	Sunlight (Calorie)	Minerals (Gram)	Water (Liter)
$U_{T\ grass}^S$	1	6	2
$U_{T\ grass}^G$	2	7	4
$U_{T\ oak}^S$	2	8	6
$U_{T\ oak}^G$	3	10	7
$U_{T\ maple}^S$	2	8	5
$U_{T\ maple}^G$	3	10	6

According to the values above, in the first rule if the available sun light resource (plate 7^{th}) within the cell is equal or greater than 2 calories and mineral amount (plate 8^{th}) is equal or more than 7 grams and also water amount (plate 9^{th}) is equal or greater than 4 then grass will be able to grow where subtract the sun light amount required for grow (2 calories) from its current value and add it to neighbours average. In fact by doing this for every cell we redistribute it to the cells and adjacent cells in our model.

In case that the resource amounts are less than threshold to grow the second rule will be applied. If the sun light amount value is equal or greater than 1 calories (sun light threshold required for the grass to survive) but less than 2 calories (sun light threshold required for the grass to grow) will be able to grow then grass will be able to survive and not to die where subtract the sun light amount required to survive (1 calories) from its current value and add it to neighbours average. In fact by doing this for every cell we redistribute it to the cells and adjacent cells in our model.

Also this condition must be true for mineral and water.

4.2.1.6 [addsun][addmineral] [addwater]

In real situation in a vegetable ecosystem the resources are supplied to the environment from outside in order for the vegetable to survive, grow and produce.

Likewise the real world our simulation also needs the continuous supply of the resources from outside to carry on the simulation and produce the results.

To model this system behaviour I defined three separate add resource zone and their correspondent rules named *[addsun], [addmineral],* and *[addwater]* in which constant value is added to the current value of the cells and subtract the value for grow and redistribute among the neighbours again. In addition some values are added to some specific cells (not region) via:

```
portInTransition : in@plant(2,7,6)
setReSource1
portInTransition : in@plant(6,7,7)
setReSource2
portInTransition : in@plant(4,7,8)
setReSource3
```

This rule allows supplying values to some desirable cells anywhere in the source plate in more manageable fashion. Finally if no tree exists or not enough resource available the extra resources add to the *addsun* zone or add via the input port to a cell. Then the resources simply redistribute into adjacent cells.

4.2.2 Shape model Implementation

The propose model has cell-space size of 20X30 consist of two Zone:

1) `zone : trunk-rule { (15,14)..(18,14) }`
2) `zone : branch-rule { (1,2)..(14,27) }`

The Seed represented by "1" state that is located in the trunk-rule zone by using the following code that insert a “1” value as a seed into cell shape (18, 14)

```
portInTransition : in@shape(18,14)
setseed
[setseed]
rule : { 1 } 100 {portValue(thisPort)=2}
rule : {(0,0)} 100 {t}
```

Upon start of the simulation the trunk start to grow “2" with applied trunk-rules within truck-rules zone as below:

```
rule : { 2 } 100 { (0,0)=0 and (1,0)=1 }
rule : { 2 } 100 { (0,0)=0 and (1,0)=2 }
rule : {(0,0)} 100 {t}
```

The trunk growth and reaches the border of branch-rules zone then the model start executing the branch rules within branch-rules zone. The model makes use of macros located in tree1.inc to define the location of the seed and also the location of the branches by using macros. By inputting the location of the inserted seed (18, 14) we can calculate the branch1 and branch2 sprout location using the propose formula for the simulation as below:

#macro(XseedPos) = 18
#macro(YseedPos) = 14

#macro(branch1) = [#macro(YseedPos)- β]-
[#macro(XseedPos)- α]

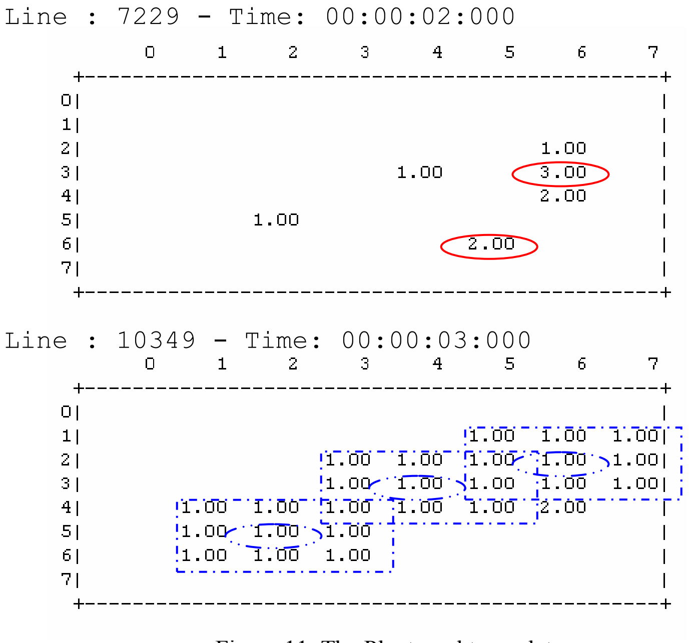
Where α, β is the distance for first branch division with respect to seed
#macro(branch1) = [14-1]- [18- 8] = 3
&
#macro(branch2) = [#macro(YseedPos)+ δ]- [20-
(#macro(XseedPos)- λ)]
Where λ, δ is the distance from second branch division with respect to seed
#macro(branch2) = [14+1]- [20-(18- 12)]=1

And replace them in the shape.inc file and run the simulation. From the above we can see that by applying my proposed rules the simulator draw the same pattern for tree type growth independent from seed location (shape2.ma).
We also demonstrate the evolution of plant branch from bud to branch and branch to old branch by different values in different parts (shape3.ma)

5. Test and Model behavioral analysis

The purpose of this section is to define several test case scenarios in order to first verify the simulation behaviour as it expected and secondly observe and analyze different results obtain from simulation outcome.
Initially the “plant.val”or “plant2.val” is applied to the Coupled Cell-DEVS model applied where it supplies the plant Cell-DEVS component. It includes input data to initial the 3rd plane (plant type) and the three resource plates.
To demonstrate the model behaviour, I drew log different plates of the plant Cell-DEVS separately to study individual parameter behaviour or in groups to observe the parameters interaction and their relationship with respect to each other.
The “.drw” files then apply to the CD++ modeller to animate the Cell-DEVS simulation with the appropriate plate file. Since my model is a 3-D having 9 different slice or plate I created one drw file for each parameter with the parameter name as a suffix append to “drw” file name. In addition, a palette file is created for the better value visualization.
Some case instances are introduced in this section as follows:

Figure 11 shows the third plate in our Cell-DEVS model that defines the plant seed type. The top part of the figure shows the initial type of plant that are assume in plant.val.

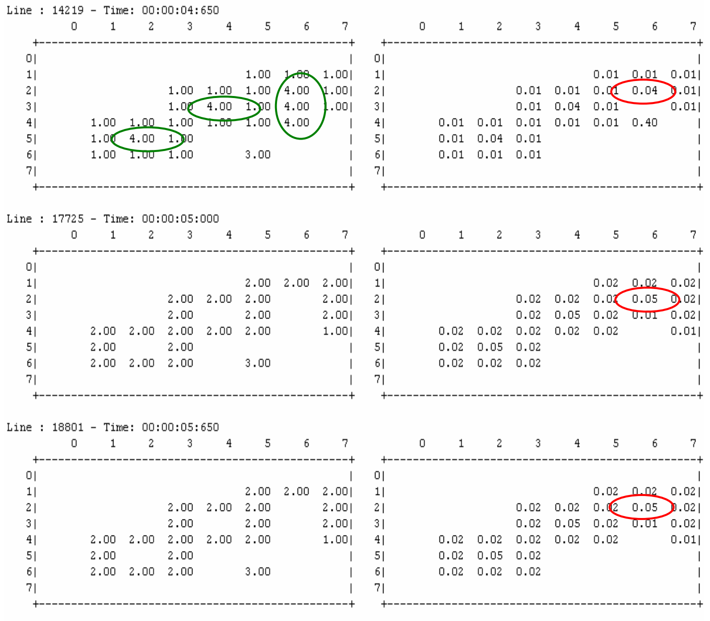


```
Line : 7229 - Time: 00:00:02:000
        0        1        2        3        4        5        6        7
   +------------------------------------------------------------------------+
  0|                                                                        |
  1|                                                                        |
  2|                                                           1.00         |
  3|                                         1.00              3.00         |
  4|                                                           2.00         |
  5|                       1.00                                             |
  6|                                                  2.00                  |
  7|                                                                        |
   +------------------------------------------------------------------------+

Line : 10349 - Time: 00:00:03:000
        0        1        2        3        4        5        6        7
   +------------------------------------------------------------------------+
  0|                                                                        |
  1|                                                  1.00     1.00     1.00|
  2|                                1.00     1.00     1.00     1.00     1.00|
  3|                                1.00     1.00     1.00     1.00     1.00|
  4|              1.00     1.00     1.00     1.00     1.00     2.00         |
  5|              1.00     1.00     1.00                                    |
  6|              1.00     1.00     1.00                                    |
  7|                                                                        |
   +------------------------------------------------------------------------+
```

Figure 11. The Plant seed type plate

In this plate the number represent the seed type that wins competition from other type of the seed since there might be more than one seed from each type, present in a given cell at any time. Therefore the seed with higher number survives and stays in that cell till the cell become empty (or if it is empty) then it starts to grow.
As one can observe the top section of the picture presents the second simulation step in which the grass is ready (>2 years old) to reproduce new seeds and spread to adjacent cells as it is the case in the lower part of the picture.
The seed type=3 will be discarded (3 inside the red circle in the time = 2:000) and replace by one since it will lose the competition to the seed type=1 where type one has grater priority in equal number of seed situation.
Another observation is the seed type 2 at time=2:000, below one in red circle that is disappear in the time=3:000 that causes by the shortage of resources for seed survivability (the cell available water resource =5.63 threshold =6 that is less than the survive amount for the seed in this case).Also this figure shows that the seed of the grass is spread fast and cover the whole area since the grass reaches productivity age faster than oak or maple and has priority advantage when it comes to the choice for the plant in a cell to grow with multiple seeds type and equal seed number.

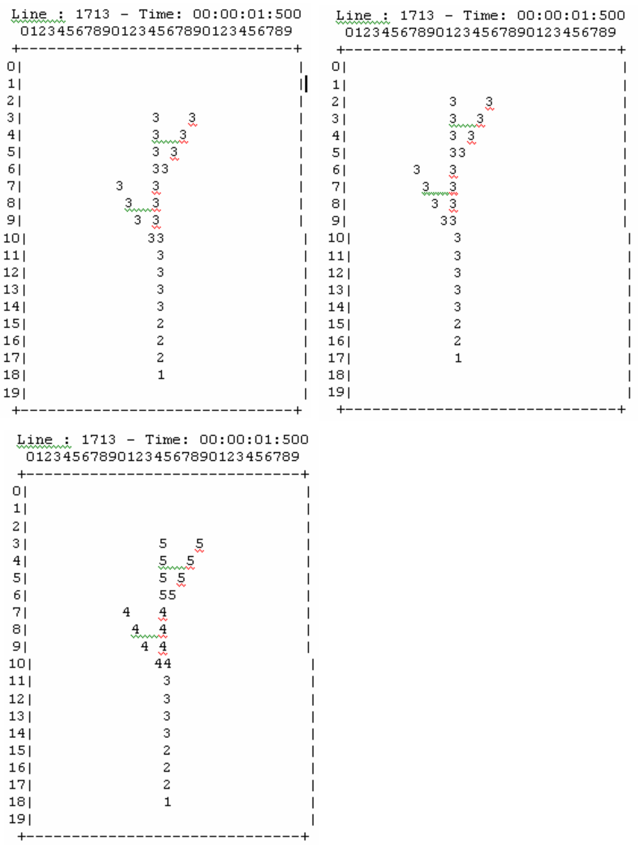
The other validate observation is shown in Figure 12



```
Line : 14219 - Time: 00:00:04:650
        0        1        2        3        4        5        6        7
   +------------------------------------------------------------------------+
  0|                                                                        |
  1|                                                  1.00     1.00     1.00|
  2|                                1.00     1.00     1.00     4.00     1.00|
  3|                                1.00     4.00     1.00     4.00     1.00|
  4|              1.00     1.00     1.00     1.00     1.00     4.00         |
  5|              1.00     4.00     1.00                                    |
  6|              1.00     1.00     1.00              3.00                  |
  7|                                                                        |
   +------------------------------------------------------------------------+

        0        1        2        3        4        5        6        7
   +------------------------------------------------------------------------+
  0|                                                                        |
  1|                                                  0.01     0.01     0.01|
  2|                                0.01     0.01     0.01     0.04     0.01|
  3|                                0.01     0.04     0.01              0.01|
  4|              0.01     0.01     0.01     0.01     0.01     0.40         |
  5|              0.01     0.04     0.01                                    |
  6|              0.01     0.01     0.01                                    |
  7|                                                                        |
   +------------------------------------------------------------------------+

Line : 17725 - Time: 00:00:05:000
        0        1        2        3        4        5        6        7
   +------------------------------------------------------------------------+
  0|                                                                        |
  1|                                                  2.00     2.00     2.00|
  2|                                2.00     2.00     2.00              2.00|
  3|                                2.00              2.00              2.00|
  4|              2.00     2.00     2.00     2.00     2.00              1.00|
  5|              2.00              2.00                                    |
  6|              2.00     2.00     2.00              3.00                  |
  7|                                                                        |
   +------------------------------------------------------------------------+

        0        1        2        3        4        5        6        7
   +------------------------------------------------------------------------+
  0|                                                                        |
  1|                                                  0.02     0.02     0.02|
  2|                                0.02     0.02     0.02     0.05     0.02|
  3|                                0.02     0.05     0.02     0.01     0.02|
  4|              0.02     0.02     0.02     0.02     0.02              0.01|
  5|              0.02     0.05     0.02                                    |
  6|              0.02     0.02     0.02                                    |
  7|                                                                        |
   +------------------------------------------------------------------------+

Line : 18801 - Time: 00:00:05:650
        0        1        2        3        4        5        6        7
   +------------------------------------------------------------------------+
  0|                                                                        |
  1|                                                  2.00     2.00     2.00|
  2|                                2.00     2.00     2.00              2.00|
  3|                                2.00              2.00              2.00|
  4|              2.00     2.00     2.00     2.00     2.00              1.00|
  5|              2.00              2.00                                    |
  6|              2.00     2.00     2.00              3.00                  |
  7|                                                                        |
   +------------------------------------------------------------------------+

        0        1        2        3        4        5        6        7
   +------------------------------------------------------------------------+
  0|                                                                        |
  1|                                                  0.02     0.02     0.02|
  2|                                0.02     0.02     0.02     0.05     0.02|
  3|                                0.02     0.05     0.02     0.01     0.02|
  4|              0.02     0.02     0.02     0.02     0.02              0.01|
  5|              0.02     0.05     0.02                                    |
  6|              0.02     0.02     0.02                                    |
  7|                                                                        |
   +------------------------------------------------------------------------+
```

Figure 12. The age and height grow in the first 3 steps

These two columns demonstrate the tree age and height evolution throughout the growth time.
Here, the second columns represent the height grow that is 0.01 meter for each year for the plant type=1. Having said that, inside the red circle at the second column of the figure inconsistency in the plant height grows exists where the height shows growth from $0.04 \to 0.05 \to 0.05$ instead of expected $0.04 \to 0.05 \to 0.06$.
The reason for that is the shortage of resources that affects the growth of the height in the grass.
Yet another event to watch is the dying processes of the plant that when reaches either its maximum ages or height grow it will dies. As shown with green circles in figure 12.
Furthermore in this section some shape examples is presented to show the dependence of the tree shape from its seed location in the shape Cell-DEVS in shape1.drw and shape2.drw that is the feathers of the L_system in drawing various shapes of the plants. I also represent the different tree parts in shape3.draw to represent various parts of the tree and its evolution from bud to branch and branch to old branch respectively.
The Figure 13 demonstrates the 3 different shapes for the shape Cell-DEVS model that is use for this model. The two top figures shows different position for the seed insertion in soil and the below one is for showing different part of a tree (trunks-old branch-branch).



```
Line : 1713 - Time: 00:00:01:500
   012345678901234567890123456789
  +------------------------------+
 0|                              |
 1|                              |
 2|                              |
 3|               3   3          |
 4|               3  3           |
 5|               3 3            |
 6|               33             |
 7|           3   3              |
 8|            3  3              |
 9|             3 3              |
10|              33              |
11|               3              |
12|               3              |
13|               3              |
14|               3              |
15|               2              |
16|               2              |
17|               2              |
18|               1              |
19|                              |
  +------------------------------+
```

```
Line : 1713 - Time: 00:00:01:500
   012345678901234567890123456789
  +------------------------------+
 0|                              |
 1|                              |
 2|            3   3             |
 3|            3  3              |
 4|            3 3               |
 5|            33                |
 6|        3   3                 |
 7|         3  3                 |
 8|          3 3                 |
 9|           33                 |
10|            3                 |
11|            3                 |
12|            3                 |
13|            3                 |
14|            3                 |
15|            2                 |
16|            2                 |
17|            1                 |
18|                              |
19|                              |
  +------------------------------+
```

```
Line : 1713 - Time: 00:00:01:500
   012345678901234567890123456789
  +------------------------------+
 0|                              |
 1|                              |
 2|                              |
 3|               5   5          |
 4|               5  5           |
 5|               5 5            |
 6|               55             |
 7|           4   4              |
 8|            4  4              |
 9|             4 4              |
10|              44              |
11|               3              |
12|               3              |
13|               3              |
14|               3              |
15|               2              |
16|               2              |
17|               2              |
18|               1              |
19|                              |
  +------------------------------+
```

Figure 13. The tree shapes

6. Conclusion

In This project I demonstrate a complete ecosystem environment in which the components interact with each other in a Coupled Cell-DEVS specification to model and simulate the vegetable population and shape with respect to the resources available and the competition for the resources in the environment.

I Presented 'Introduction to modelling a simulation' in chapter one and a background to DEVS models including atomic, coupled and Coupled Cell-DEVS in the next chapter along with the example of DEVS model implementation in CD++. Next chapter I described the conceptual model of my proposed system and explanation of the system logic and rules base on the papers that define the vegetable ecosystem environment. Then, I introduced the formal Cell-DEVS description of the population model and shape in chapter 4 and explained the implementation of my project.

The core of my project implementation exploits different slices or plates for the model parameters that interact with each other to simulate the dynamics of the system.

The model tested and debugged successfully in CD++ and some test verifications are presented to demonstrate the system functionality and the simulation result we expected based on the vegetable ecosystem behaviour in real world.

7. Future Direction

This model can be further improved by inclusion of more complicated ecosystem parameters and advance plants shapes.

Moreover the system can introduce animal species and their interactivity with the system in a global aspect.

Some of the algorithms and rules primarily introduced in the main paper by Bandini at el can be redefine and improve to draw more realistic picture of reality for the ecosystem environment.

8. REFERENCES

[1] Zeigler, B. P. "DEVS Theory of Quantization". *DARPA Contract N6133997K-0007*: ECE Dept., UA, Tucson, AZ. 1998.

[2] L. A. Zadeh and C. A. Desoer, "Linear System Theory, The State Space Approach," New York, McGraw-Hill, 1963.

[3] Zeigler, B.P.. Belogus, D., and Belshoi, "A. ESP: An interactive tool for system structuring." *In Proceedings European Meeting Cybernetics and Systems Research*, (Vienna), Hemisphere, New York, 1980.

[4] Zeigler, B.; PRAEHOFER, H.; KIM, T. "Theory of Modeling and Simulation: Integrating Discrete Event and Continuous Complex Dynamic Systems," *Academic Press*. 2000.

[5] S. Wolfram, Theory and Applications of Cellular Automata, (World Scientific, 1986); Cellular Automata and Complexity (Addison-Wesley, 1994).

[6] Neumann, J. V. 1966 Theory of Self-Reproducing Automata. University of Illinois Press.

[7] WAINER. 2002 CD++: a toolkit to develop DEVS models. Software Practice and Experience.

[8] H. Baltzer, W.P. Braun, W. K¨ohler. Cellular Automata Models for Vegetation Dynamics. Ecological Modelling, 107(1998), pp. 113–125.

[9] 6. R.L. Colasanti, J.P. Grime. Resource dynamics and vegetation processes: a deterministic model using two–dimensional cellular automata. Functional Ecology, 7(1993), pp. 169–176.

[10] H. Baltzer, W.P. Braun, W. K¨ohler. Cellular Automata Models for Vegetation Dynamics. Ecological Modelling, 107(1998), pp. 113–125.

[11] 6. R.L. Colasanti, J.P. Grime. Resource dynamics and vegetation processes: a deterministic model using two–dimensional cellular automata. Functional Ecology, 7(1993), pp. 169–176.

[12] D.G.Green, “Modelling plants in landscape”, in Plants to Ecosystem – Harek T. Michalewicz, ed. CSIRO, Lollingwood Ans., 1997.

[13] J.L. Uso–Domenech, Y. Villacampa–Esteve, G. St¨ubing–Martinez, T. Karjalainen, M.P. Ramo. MARIOLA: A Model *for Calculating the Response of Mediterranean Bush Ecosystem to Climatic Variations. Ecological Modeling*, 80(1995), pp. 113–129.

[14] Bandini, S., Pavesi, G.: “Simulation of vegetable populations dynamics based on cellular automata” In Bandini, S., Chopard, B., Tomassini, M., eds.: *Cellular Automata,* Volume 2493 of LNCS, Berlin, Springer-Verlag (2002)

[15] S Bandini, S Manzoni, S Redaelli, L Vanneschi, “Emergent Spatial Patterns in Vegetable Population Dynamics: Towards Pattern Detection and Interpretation” LECTURE NOTES IN COMPUTER SCIENCE, 2006 - Springe

[16] Prusinkiewicz P., Hammel M., Hanan J., Mech R., Lsystem: from the theory to visual models of plants, in: Michalewicz M.T. (Ed.) Plants to Ecosystems. Advances in Computational Life Sciences, I, CSIRO publishing, Melbourne, 1997, pp. 1-27.