Definition of an Advanced Cell-DEVS Model for the Mountain Pine Beetle Epidemic

Daanish Khan
Carleton University
dkhan@sce.carleton.ca

Abstract

The DEVS formalism has been used to model various types of phenomena and is a useful tool for simulation. DEVS has been extended with Cell-DEVS to accommodate cellular automata. Cellular automata can be used to model various physical phenomena. The Mountain Pine Beetle epidemic is an outbreak of beetles that kill Lodgepole Pine and their spread can be effectively modeled using cellular automata techniques and specifically the Cell-DEVS framework. Once the model has been created, it is possible to extend the simulation using a new version of Cell-DEVS that supports more advanced functionality. The performance of these two simulations can be compared and the advantages of the newer Cell-DEVS version can be determined.

1. INTRODUCTION

Modeling and Simulation is a popular tool for studying many different types of complex systems. The need for such a tool has arisen with the advent of systems that may be too big, small, costly, dangerous, or time consuming to observe directly. Discrete Event Systems (DEVS) is a popular formalism for describing simulations and can be used to construct simulations of real-world phenomena. Fundamentally, the DEVS formalism is implemented by subdividing a system into atomic models, each with their own state and behavior, and modeling the interaction between these models. This type of modeling allows for hierarchical construction of complex models and is a powerful simulation tool for many different types of simulation. To further extend the simulation capabilities of the system, extensions were specified to accommodate different simulation techniques. This study will specifically utilize the Cell-DEVS specification which allows for the implementation of cellular automata in DEVS. This study aims to demonstrate how Cell-DEVS can be effectively used to create a working model of a biological system. Specifically, a cellular automata model for the spread of the mountain pine beetle will be define and tested using the CD++ toolkit. Additionally, new techniques for simulation will be explored and the performance advantages will be discussed.

2. BACKGROUND

Discrete Event Simulation is a formalism that allows for the modeling and simulation of many different types of systems. One of the goals of DEVS is to be able to separate the modeling and simulation aspects of a given problem. Using hierarchies of atomic components, DEVS is able to abstract these aspects and increases usability and interoperability of simulations.

Atomic models form the basis of any DEVS model[2]. Atomic models consist of input and output ports along with internal states and behaviors. Using time advance functions, a state transition can be triggered and output is generated. Additionally, external events can be received on the input ports and cause external transitions to be executed within the model. The overarching idea is that models only interact through ports and never know anything about other models other than what is provided on the input ports.

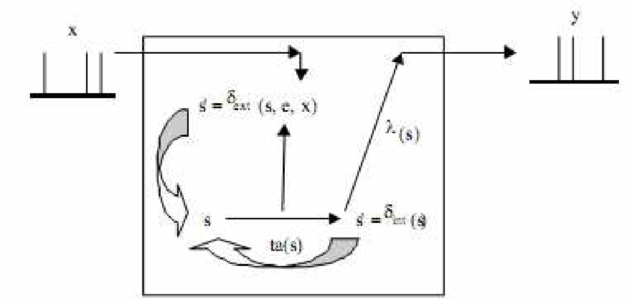

Figure 1 – Atomic DEVS Model

Coupled models are DEVS models that consist of multiple atomic models that are connected through their input and output ports in a specific way. Using the model specification (.ma) file, the specifics of the interconnections between the models and how these models interact with the output of the simulation.

To extend DEVS to include cellular automata phenomena, the Cell-DEVS[3] framework was created. This framework allows for the implementation of cellular automata models and includes functionality for timing delays. It is worth noting that each cell in the cellular automata created in Cell-DEVS is implemented as a DEVS atomic model and the cellular automata can be thought of as a large interconnection of DEVS atomic models.

Each cell in a Cell-DEVS model has inputs coming from each of its neighboring cells. It uses these inputs to compute its next state through the local computing function. The state of the cell is transmitted to other cells using the output port of the model. Output can appear on the output port after a specified delay.

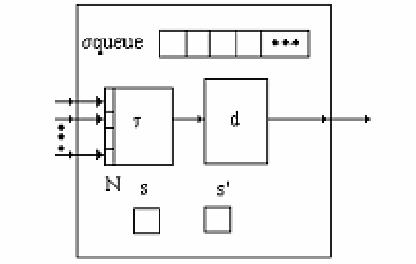

Figure 2 – Atomic Cell-DEVS Model

A coupled Cell-DEVS model is created by interconnecting these cells and specifying the rules with which these cells interact. Generally, the Cell-DEVS models contains many interconnected atomic DEVS models and each cell is connected through input and output ports to cells in its neighborhood. For the purposes of this study, it is useful to note that certain cells can have behaviors that are non-general, such as for border cells, it is possible and useful to specify custom rules.

To create these models, the CD++ modeling environment is used. This framework, developed in C++ implements the DEVS formalism and has been extended to include an implementation of Cell-DEVS. Once models have been written, they can be executed using the CD++ environment. DEVS atomic models are generally written in C++ and included as part of a class hierarchy in a CD++ executable, whereas Cell-DEVS models could function this way or use standard cells with specified rules. There have been many extensions to the DEVS formalism that are worth noting for the purposes of this study. Specifically, the ParallelDEVS framework will be used to implement state variables and specific ports for the later part of the study. Additionally, real-time, embedded and traffic simulation engines have been implemented in DEVS. Since the study presented mainly focuses on the Cell-DEVS models, this framework will be covered in greater detail.

3. IMPLEMENTATION IN CELL-DEVS

DEVS and Cell-DEVS can be used to model complex biological phenomena. In assignment 2, a simplified model for the spread of the Mountain Pine Beetle was created in Cell-DEVS and modeled for various types of forest features including a river, a lake, and an incomplete clear-cut. In this study a more advanced Cell-DEVS model will be created with the intention of more completely and accurately describing the effect of the Mountain Pine Beetle.

Cellular Automata can be effectively used to model the effect of the Mountain Pine Beetle[1]. The Mountain Pine Beetle (MPB) Epidemic has resulted from the explosion in the population of MPB in Canada. These beetles primarily infect Lodgepole pine trees that are typically harvested for use in the commercial forestry industry. It is estimated that up to 80 percent of all harvestable Lodgepole pines could be killed by the MPB infestation by 2013 if nothing is done to stem their progression. Currently, the industry best practices are to clear cut any forest infected with MPB to slow or stop infestation in the surrounding area. The major effect of this clear cutting practice is lost revenue as pines that are not fully matured must be harvested during a clear cut operation.

The CA model for MPB progression involves looking at the total susceptibility (TS) value for each unit of forest. The TS value is obtained by simple multiplication of four factors: the size of the trees in the unit, the proportion of Lodgepole pine trees in the unit, the density of trees in the unit, and the proximity of the unit to a known infestation. The size of the trees (S) parameter reflects the fact that larger trees are more susceptible to infestation. The proportion (P) and density (D) values reflect the fact that MPB will only infect pine trees and tend to do better in dense forest (no matter the type of forest). The location factor (L) represents the beetle's ability to move, and that places closer to infestations are more likely to become infected.

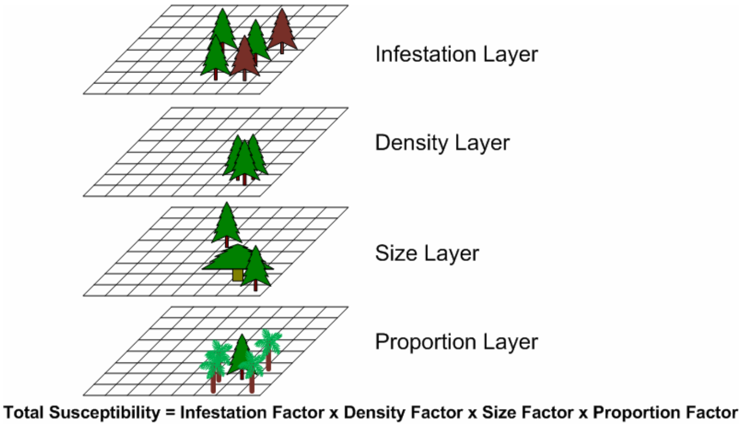
Infestation Layer
Density Layer
Size Layer
Proportion Layer
Total Susceptibility = Infestation Factor x Density Factor x Size Factor x Proportion Factor

Figure 3 – System Layers as Proposed

Once these parameters have been developed, an allometric function is used to determine if the total susceptibility will result in an infestation of the forest unit. The allometric function used is $y=0.05x^{-a}$ with a being empirically set to -1.3. Once a cell has been infected, there is a 100% certainty that all the lodgepole pine in the area will eventually be infected. Additionally, empirical evidence suggests that up to 80% of mountain pine beetle in a given tree will die during the winter (each generation has an 80% mortality).

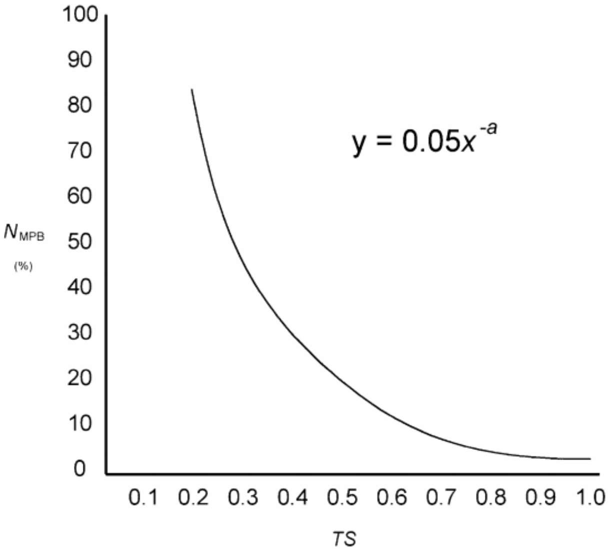

Figure 4 – The Allometric Function Used

The system can be summarized as follows:

- A two dimensional forest that could include rivers, lakes, clear-cuts, thinned trees, etc…
- Each timestep represents a year of real-time
- Each forest is in one of two states: dead or alive
- Each tree is only affected by trees in its neighborhood
- If a tree stand is alive, it monitors the amount of mountain pine beetle in the stand
- If the total concentration of mountain pine beetle is greater than the allometric function for tree mortality specified at the succeptibility level of the stand, then the stand will become infected and die before the next season
- Mountain pine beetle progress through the forest and are computed as the average concentration for the surrounding cells plus the current number of beetles less the number that died during the winter of that year.
- Trees cannot recover from a mountain pine beetle infestation

To accomplish this simulation, a model that consists of several layers needed to be created. The following three layers are of the most importance:

Layer 1 : The mortality layer – This layer represents the trees that have been killed by mountain pine beetle

Layer 2 : The Mountain Pine Beetle Layer – This layer represents the movement of the mountain pine beetles and gives their relative concentration for any given cell in the cell space

Layer 3 : Forest factors – This layer represents the size, density and proportion factors for given stand. Modifying the value in this layer would alter how the mountain pine beetle spread in that stand

3.1. Model Specifications

3.1.1. Simplifications

This model is an advanced version of the model presented in Assignment 2. Therefore, there has been a reduction in the simplification of the model. This model does not assume that the spread of the mountain pine beetle is constant. Additionally, the allometric function governing tree mortality has been implemented.

Since size, proportion and density are constants for the purposes of this simulation, they were not explicitly included for clarity, however by modifying the values in Layer 3 (mentioned above) it is possible to accommodate size, proportion and density values for susceptibility calculations.

The location factor determination was kept in the simplified form from Assignment 2 as it represents a case with a 2km by 2km cell size, which is adequate for large scale simulations. Micro-scale or tree level simulations however would require that the location factor be more accurately determined.

3.1.2. Model Specifics

Formal Specification

The formal specification of the Cell-DEVS model for the neighbor cells is given by:

$M=<I,X,Y,Xlist,Ylist,\eta, N,\{n1,n2,n3\}, C, B, Z, select>$

Where:

$Xlist=\Phi$

$Ylist=\Phi$

$\eta=6$

$I=<PX,Py>$,with $PX=\{\Phi\},Py=\{\Phi\}$;

N={(-1,0,0),(0,-1,0),(0,0,0),(0,1,0),(1,0,0),(0,0,1)}
X={0,1};
Y= {-1,0, 0.4, 0.6, 0.8,1};
{n1, n2, n3} = {16, 16, 3}
B= {Cij / (i = 1v i = 16) ∧ (j = 1v i = 16)} nowrapped;
C={Cijz / iε[1,16], jε[1,16], z ε[1,2]}
Z:

Neighbor List	Inverse Neighbor List
Pijz Y1 → Pi,i-1z X1	Pi,i-1z X1← Pijz Y1
Pijz Y2 → Pj-1,jz X2	Pj-1,jz X2 ←Pijz Y2
Pijz Y3 → Pijz X3	Pijz X3 ←Pijz Y3
Pijz Y4 → Pj+1,jz X4	Pj+1,jz X4← Pijz Y4
Pijz Y5 → Pi+1jz X5	Pi+1jz X5 ←Pijz Y5

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

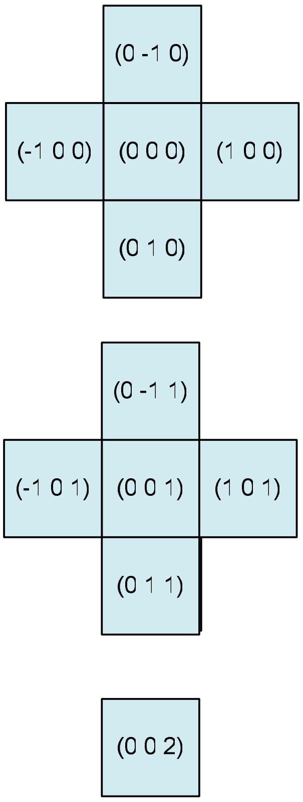

Figure 5 – Neighborhood Overview

4. CD++ IMPLEMENTATION

This model was implemented using the CD++ toolkit provided. The model created contained two planes with 16x16 cells. It is worth mentioning that a simulation with 64x64 cells was created and verified but was not used for the purposes of the report due to the length of time the model takes to simulate (Around 5 minutes). The 16x16 model has non wrapped borders with a default delay time of a millisecond. The infection plane (plane 0) has 1 cell initially infected at (1,1). All the other cells are set to zero. Plane 1 is the density layer and contains all zeros except for places where it is desired that no trees grow (those locations are set to 1). The rules for each plane were created as follows:

```
[mpb-rule]
rule : {1} 100 {cellpos(2)=1 and (0,0,0) >= 1}

%NPB Travel + NBP Annual mortality

rule : {(((-1,-1,0)+(-1,0,0)+(-1,1,0)+(0,-
1,0)+(0,1,0)+(1,0,0)+(1,-
1,0)+(1,1,0))/8)+((0,0,0)*0.2)} 100 {cellpos(2)=1
and (0,0,0) < 1}

%TS Calculation
rule : {(0,0,0)} 100 {cellpos(2)=0 and (0,0,0)>=1}

rule : {1} 100 {cellpos(2)=0 and
#Macro(surround)=1 and 0.05*power(0.4*(1 -
(0,0,2)*(1 - (0,0,3))),-1) < (0,0,1)}
rule : {1} 100 {cellpos(2)=0 and
#Macro(surround)>1 and #Macro(surround) <= 4 and
0.05*power(0.6*(1 - (0,0,2)),-1) < (0,0,1)}
rule : {1} 100 {cellpos(2)=0 and
#Macro(surround)>4 and 0.05*power(0.8*(1 -
(0,0,2)),-1) < (0,0,1)}
rule : {(0,0,0)} 100 {t}

[sides]
rule : {(0,0,0)} 100 {t}

[sides2]
rule : {(0,0,0)} 100 {t}
```

5. SIMULATION RESULTS

In all the below samples, the three separate simulation areas show each different layer of the simulation. The first square represents the tree mortality layer showing only dead trees. The second square shows the number of mountain pine beetle in the given square as a concentration. The last square shows the areas where forest factors such as density, size or proportion of lodgepole pine have been specified. A color code was specified to visualize the results as follows:

Green: The forest is in good health, with no infection (No infected cells around current cell)

Yellow: The forest has a low risk of infection
There exists an infected cell around the current cell

Orange: The forest has a moderate risk of infection
There exists infected cells around the current cell but it is less than half its neighbors

Red: The forest has a high risk of infection
More than half of the neighbors are infected cells

Gray: The forest is currently infected or dead

5.1. Graphical Results

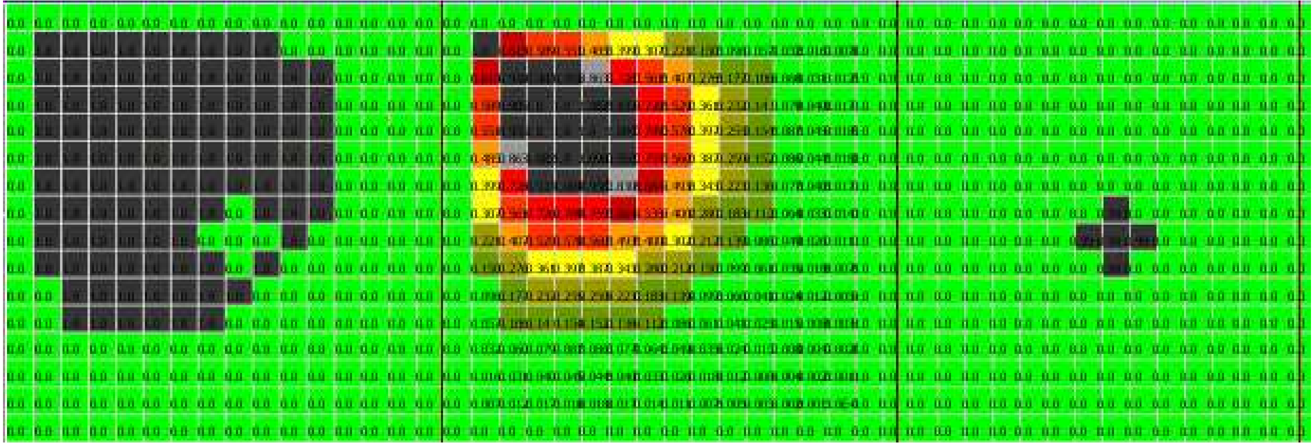

Figure 6 – The Basic Forst Model

Basic Model

This model consisted of a constant infection source at (1,1) and a small pond centered at (8,8). The intention of this simulation is to show that the pond will not be affected by the spread of the mountain pine beetle.

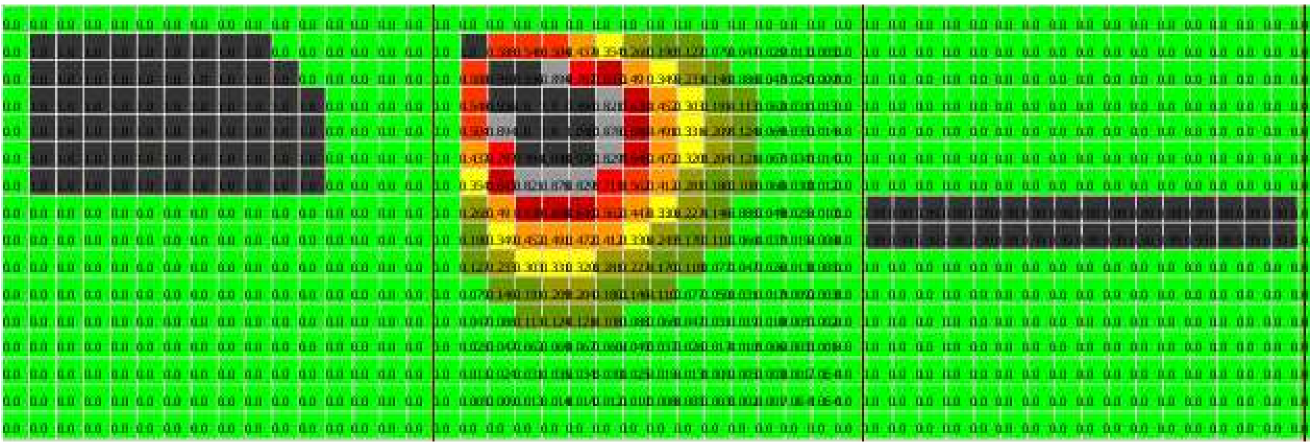

Figure 7 – The River and Forest Model

River Model

This test case presented the case where the forest is bisected by a river, and therefore the mountain pine beetle are unable to cross it. The above figure shows an intermediate step with the progression of the epidemic impeded by the river.

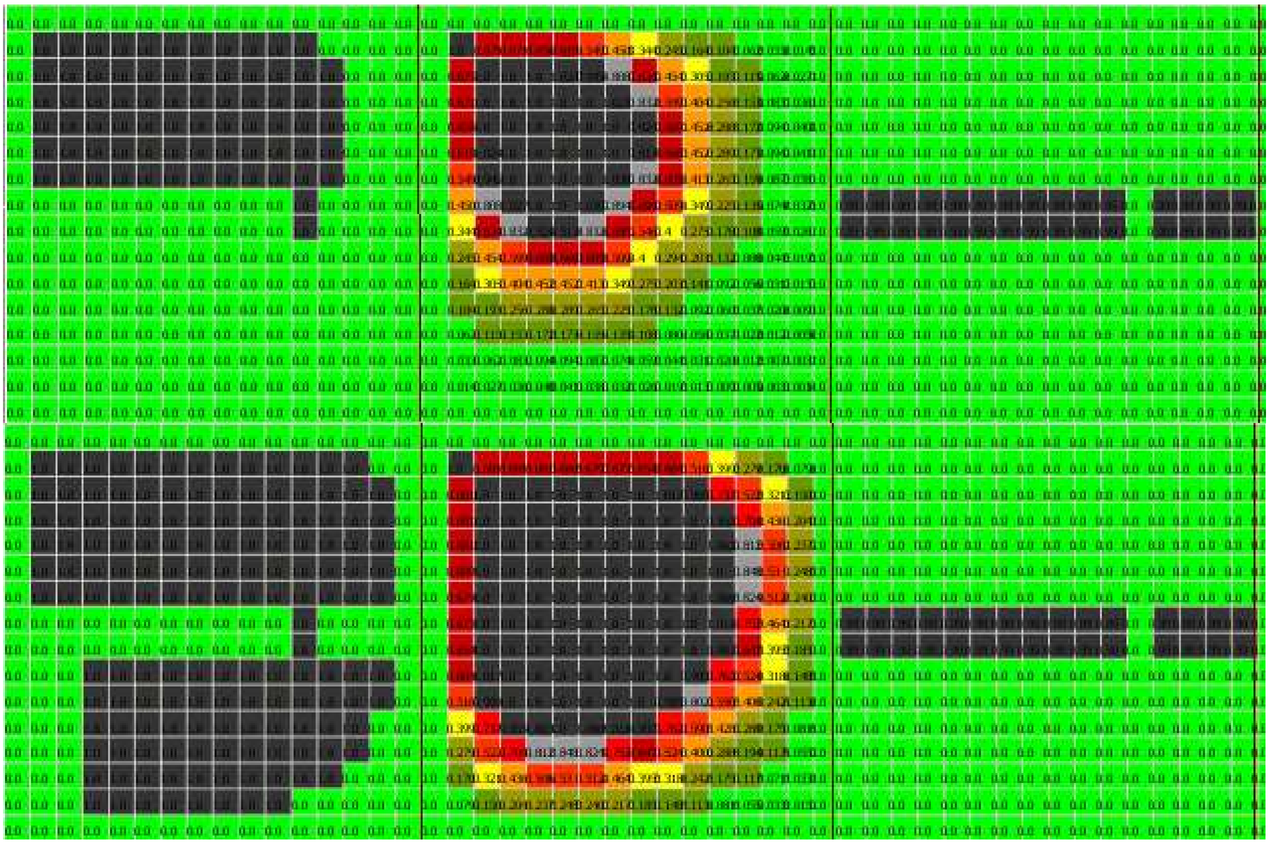

Figure 8 – The incomplete clear cut model

Incomplete clear-cut Model

The above example presents the results when the simulation is presented with an incomplete clearcut. In this case it is shown that the beetles are capable of traveling across the unclear-cut land and affecting the trees on the other side. This example is of critical importance to the simulation of the effects of clear cutting forests for prevention of Mountain Pine Beetle outbreaks.

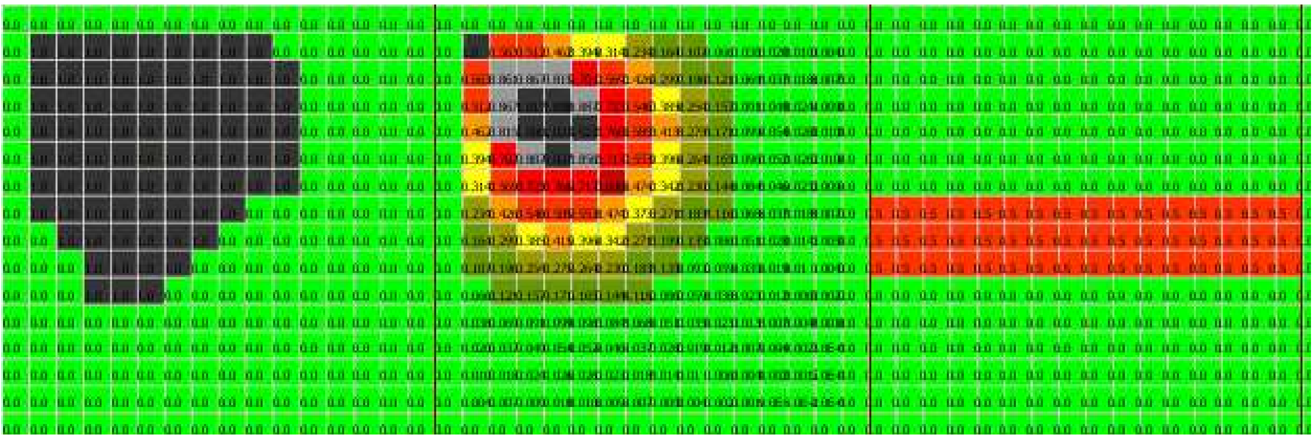

Figure 9 – Forest Thinning Model

Forest Thinning Model

The above example shows how forest thinning can slow the progression of the Mountain Pine Beetle. It is worth noting that the 5 squares in the last square represent a moderate density of Lodgepole Pine. Above it is possible to see that the progression over the lower density area is slowing the progression of the mountain pine beetle. This has physical significance as thinning is used to control the mountain pine beetle and has shown promise for reducing the spread of the beetle.

6. IMPROVING THE MODEL

Once the improved model had been constructed in regular Cell-DEVS, the model was translated to a newer version of Cell-DEVS that supported state variables and ports for the purposes of comparing the performance of the model to the newer version. Additionally, since the state variable and ports functionality will be available, it will be possible to simplify structure of the model while still providing the same level of detail.

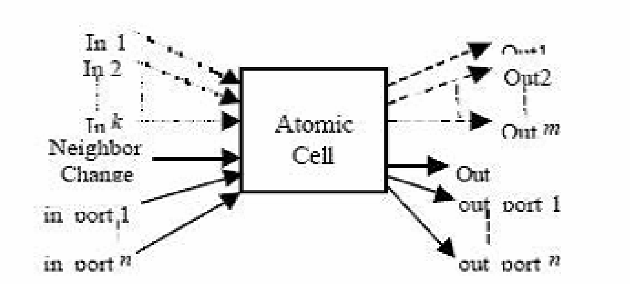

Figure 10 – Multiport Cell

To begin to simplify the model is should be shown that the cells exchange the total susceptibility to MPB infection as well as the concentration of MPB in the cell. Therefore, the improved Cell-DEVS model relies on two ports: the *totalSuccep* and the *nmpb* which represent these two exchanges of data.

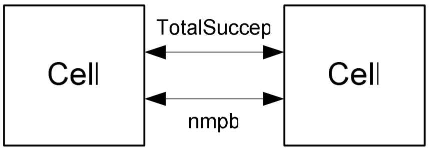

Figure 11 – Cell Ports Specification

To model the characteristic of the stand (as used on layer 3 of the classic model) a state variable was implemented. This state variable allowed for the cell to store their factors rather than be forced to refer to another cell for information.

The modified rules were implemented as follows:

```
[mpb-rule]
rule : {~nmpb :=1;} 100 {(0,0)~nmpb >= 1}
%NPB Travel + NBP Annual mortality
rule : {~nmpb :=#Macro(mpbAvg);~totalSuccep := 1;}
100 {#Macro(surround) = 1 and 0.05*power(0.4*(1 -
$density), -1) < (0,0)~nmpb}
rule : {~nmpb :=#Macro(mpbAvg);~totalSuccep := 1;}
100 {#Macro(surround) > 1 and #Macro(surround) <=
4 and 0.05*power(0.6*(1 - $density), -1) <
(0,0)~nmpb}
rule : {~nmpb :=#Macro(mpbAvg);~totalSuccep := 1;}
100 {#Macro(surround) > 4 and 0.05*power(0.8*(1 -
$density), -1) < (0,0)~nmpb}

%TS Calculation
rule : {~totalSuccep :=(0,0)~totalSuccep;~nmpb
:=(0,0)~nmpb;} 100 {t}
```

Two macros had to be defined for this model as follows:

```
#BeginMacro(surround)
((-1,-1)~totalSuccep+(-1,0)~totalSuccep+(-
1,1)~totalSuccep+(0,-
1)~totalSuccep+(0,1)~totalSuccep+(1,0)~totalSuccep
+(1,-1)~totalSuccep+(1,1)~totalSuccep)
#EndMacro
#BeginMacro(mpbAvg)
((-1,-1)~nmpb + (-1,0)~nmpb + (-1,1)~nmpb + (0,-
1)~nmpb + (0,1)~nmpb + (1,0)~nmpb + (1,-1)~nmpb +
(1,1)~nmpb)/8 + ((0,0)~nmpb*0.2)
#EndMacro
```

6.1. Performance Comparison

Once the two models had been successfully simulated using a 1.5 Ghz Pentium computer with 640MB of RAM, data could be analyzed to determine what performance gains were realized by switching to the newer version of CD++. The scenario used in this comparison is the simple case with one infection point.

It was noticed that the memory usage of both simulators was approximately the same but since the classic Cell-DEVS model was running within Cygwin to work under windows, the RAM usage was slightly more due to the cygwin executable also loaded into memory. To determine performance gains, execution time and number of messages will be compared for both types of CD++ implementations.

The following charts show how the simulations compare.

Execution Time

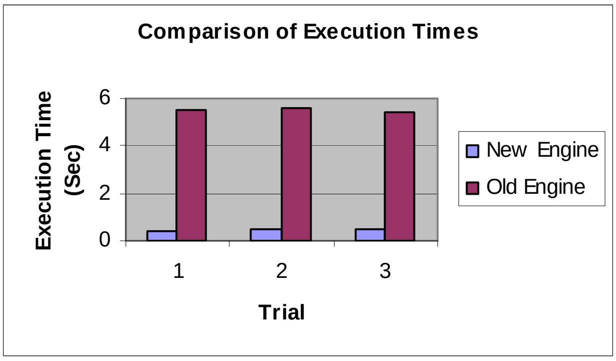

Figure 12 – Execution Time Comparison

Messages Passed

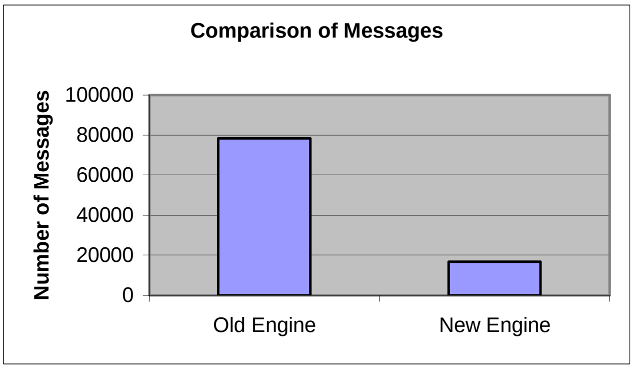

Figure 13 – Messages Passed Comparison

From the above charts it is plainly obvious that the new engine for simulation Cell-DEVS models not only includes more functionality, but also improved the simulation times and number of messages drastically.

7. CONCLUSIONS

As was presented, a Cell-DEVS model was created to simulate the effect of the mountain pine beetle using predefined rules as specified in the paper that formed the basis for this study. Additionally, rules for spread of mountain pine beetle were created and tested. Once the model for this simulation was specified, it could be tested with well known land formations that would be typical in a forest setting. The testing allows for visualizations using the CD++ Modeler tool. Lastly, an advanced Cell-DEVS model was constructed using the latest version of CD++, which was capable of accommodating state variables and ports, through which more advanced communication could take place. This more advanced model was tested and compared against the original model using predefined performance metrics.

8. FUTURE WORK

This project is open to a lot more development and would be greatly aided with a 3D simulation such as in Blender or in DEVSView. The 3D realization would allow for users to see the effects of the epidemic of mountain pine beetle more concretely.

Additionally, more advanced rules for how the beetles propagate along different land forms could be analyzed. Specifically there has been interest to develop a simulation that could accommodate for changes in altitude and the effect of altitude on the Mountain Pine Beetle.

9. REFERENCES

[1] Bone, C. et. Al. “Evaluating forest management practices using a GIS-based cellular automata modeling approach with multispectral imagery”, Environ Model Assess Volume 12 Issue 105, January 10th 2007.

[2] B. Zeigler; T. Kim; H. Praehofer: “Theory of Modeling“ New Model New Imp. 11796 and Simulation: Integrating Discrete Event and

[3] G. Wainer; N. Giambiasi: "Application of the Cell-DEVS Paradigm for Cell Spaces Modeling and Simulation", Simulation, Vol. 71, No. 1, pp. 22-39, January 2001.