Self-recovery of a distributed system after a large disruption

New Mexico Supercomputing Challenge
Final Report

April 1, 2013
Team 61

Team Members:
Victor Popa-Simil 5th Grade
Aspen Elementary School, Los Alamos, NM

Andrei Popa-Simil 7th Grade
Los Alamos Middle School, Los Alamos, NM

Sponsoring Teachers:
Mrs. Stevens Pauline

Project Mentor:
Liviu Popa-Simil

Table of Contents

Executive Summary 3
Statement Of The Problem That Has Been Investigated 4
Method Used 4
Model 4
Stages 5
Stage 1 5
Stage 2 5
Stage 3 5
Stage 5 6
Code Result 6
Conclusions 7
Acknowledgments 7
Annexes: 8
1 - Tables, and Excel simulation of a possible result 8
The most significant and original achievements on the project 9

Executive Summary

Our goal is to identify and understand those interaction conditions and their dependence of the initial state that are driving to recovery of the systems hit by calamities, the self repairing of materials and tissues hit by radiation and how the structures have to be designed in order to increase their resilience. To solve the problem on computer we reduced the problem to a group of animals living together in the same land. We created a group of animals such as rabbits and squirrels. Soon the rabbits and squirrels take control over territory by grouping into clusters. Then a large disruption scares all the animals and they start to scatter in every direction possible out of disruption area. Once the disruption [example: lightning] ends, the animals will try to regroup on their territory. We will apply properties to rabbits, as rabbit likes rabbit but no more than 4, and squirrels like squirrels but no more than 8 and the same properties for patches.

We used Net logo code to simulate what might happen if a population such as rabbits and squirrels is disturbed and trends to recover. In order to better understand the process we used a chessboard simulation with checker pieces, and also Excel based simulations. On Excel we have tried various configurations obtained with random number generators in order to better understand how to interpret the final results and how to make the Net logo programming. Our task is not finished, remaining that next year we learn more programming and mathematics in order to better analyze the recovery dependence on agent features in order to better understand the problem.

Statement Of The Problem That Has Been Investigated

Many natural phenomena like calamities wars or other damaging events produce dislocation of populations of various elements as animals, bacteria from their habitat or may take place even inside materials exposed to damage that all trend to recover.
From the many recovery paths possible, the main interest is toward those processes that drive to the similar structure as before the incidents, and towards those conditions of interaction between them and environment that have to be met.
The process is called self-recovering, self repairing or healing.

Explicative Note
Disruption and recovery happen frequently in nature at all the levels, from atomic scale to macro-systems, from bacteria to humans, with time duration of less than 1 micro second for radiation damage in materials, to more than years in the case of hurricanes, to many thousand years to millions in the case of killer asteroids.
Recovery is a multi-entity process done to fix one or many individuals such as humans, animals or bacteria, structures. When there are many, they form a group that performs many activities inside a complex interaction.
We consider that a large disruption will affect more than 50% of the group but less than 90%, because over 90% such a disruption may drive to extinction, the system losing its memory or vitality and no recovery may be possible.

Method Used

Model

The environment, in which this simulation would be conducted, is drafted from the chess game. We will establish the total number and patch maximum admittance for types of populations and their interaction. The computer would be responsible for generating a disruption, and making all the species move outside the zone, using random directions until they are out, than, they will start repopulating back using rules of movement keeping in account the occupancy limits. It will define the differences between the initial distribution and the current distribution and will present a graph of the recovery in time. We will change the population allocation parameters and observe the process.
Being a complex problem, with many agents and intricate dependences in order to understand the process and to solve the problem on computer, we reduced the problem to a group of animals living together in the same land.
We created a group of animals such as rabbits and squirrels. Soon the rabbits and squirrels take control over territory by grouping into clusters, with quasi-stable population distribution, we called equilibrium.
Then a large disruption scares all the animals and they start to scatter in every direction possible out of disruption area. For simplicity we do not eliminate the agents during disruption, as is the case in many natural phenomena. Once the disruption [example: lightning] ends, the animals will try to regroup on their territory. We will apply properties to rabbits, as rabbit likes rabbit but no more than 4, and squirrels like squirrels but no more than 8 and the same properties for patches. We even take one step downwards, and we tried to understand the relationship of a single population with respect to distributed resources, as rabbits and patches, because we had difficulties understanding when we

may call that the recovery is complete, in order to estimate the recovery time and recovery action magnitude.

Stages

Stage 1

Create patches (input number of patches and color)

Note:

We may create several colors of patches with the properties:

Green supports more rabbits than squirrels (10 rabbits one squirrel) because green could be grass

Brown supports more squirrels than rabbits (1 rabbit 7 squirrels) because brown might be oak forest or nuts

Yellow supports less rabbits and squirrels (2 rabbit 2 squirrels) because it might represent a farmers land and so on.

We have eliminated the squirrels setting their no on zero and leaving in play just rabbits and patches.

Create animals

We create rabbits (input number of rabbits and properties), and we create squirrels (input number of squirrels)

Note: net logo created everything in the central patch coordinate (0,0)

We have 2 options:

1. - Leave them as they are and start the program
2. - Move them randomly and start the program, and this is the solution we have picked.

Stage 2

Move animals

We went patch by patch and see how many are on each patch vs. patch properties.

If there are too many we will decide witch one will move, first in first out or last in first out. We had to decide in what direction and how long it will go until it stops and how much time it will have until it has to leave. If they are not enough animals on a patch to trigger mandatory move we introduce timing for each animal and then move.

This will bring an equilibrium distribution of the animals on patches.

Stage 3

Disturbance

We created a disturbance in order to simulate a calamity like a large noise or a natural disaster (we need to select the patch the calamity starts on and the dimension of the effected zone) we ignore any damages brought to patches and their recovery time or the eventuality of transforming patches into black patches where rabbit nor squirrel is able to go.

In the calamity area, all the animals will evacuate immediately, each animal will go on a random direction until it is out of the area

Stage 4

<u>Move animals back</u>

The calamity is over and the animals will move back to the original land and start to regroup after the same pattern as in stage 2.
From time to time we will see the differences between the distribution before the calamity and present distribution and present like maps or dislocation number

Stage 5

<u>Final analyzes</u>

We identified two possible options:
1. - Stop and analysis data and draw conclusions.
2. - Apply a new calamity and collect more data and then stop.

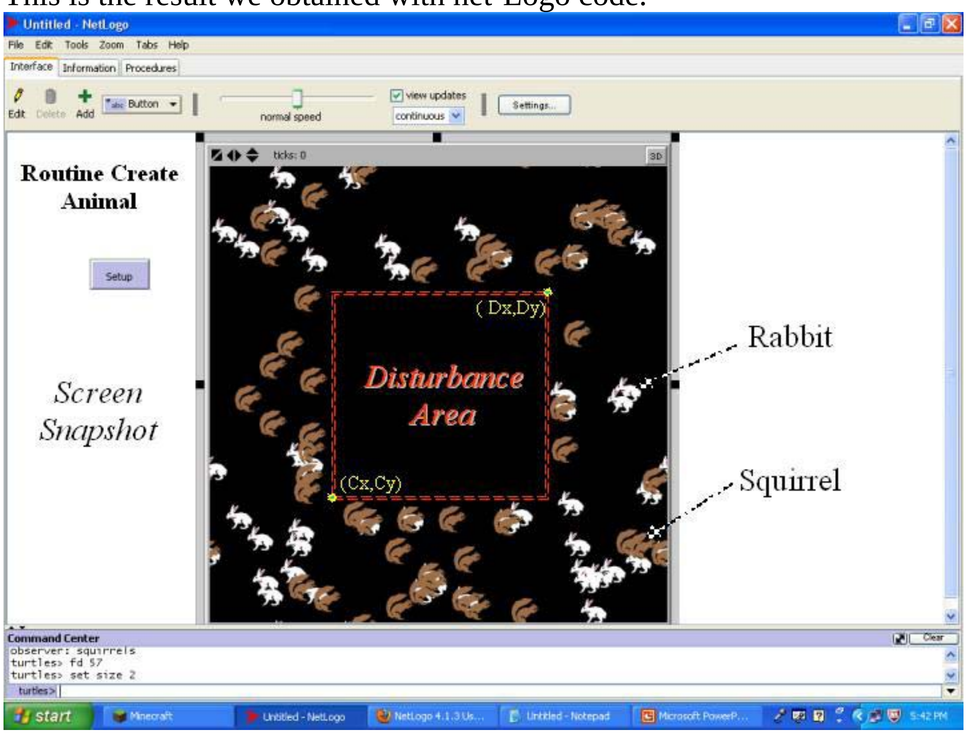
Note: this scenario resembles Los Alamos mandatory evacuation during Los Conches fire.

The Software,

Presently, a simulation has been constructed in which two populations as rabbits and squirrels are generated and distributed on patches, and the way to move them outside after the disruption. We are studying various rules of movement for the agents in order to match real world description, and we try to understand what recovery means to natural systems in order to program this on computer, we are facing difficulties due to the lack of coherent simple data even for human systems in the case of hurricanes, as Katrina or Ivan, and we did not find any studies for animal populations yet.

Code Result

This is the result we obtained with net-Logo code.



The figure represents a screen shot after the calamity was generated were intentionally we took very few animals in order to look like the chess board simulation showed in the previous figure.

After programming, testing, and refining of the algorithms we will define our own way of understanding the problem, and to learn from computer's simulations.
In the case of less population than the patches may allow, the movement freezes at a number of occupancy, call it initial distribution, and after recovery another distribution is obtained that is simply different than the initial one so the recovery never comes.
When the patches are overpopulated there is a continuous variation of the number of rabbits in patches, therefore it is difficult to consider an instance as equilibrium and to compare with the result after disruption. The recovery as initially thought does not happen, and we still have to work at conditions and rules of interaction.
The hope is to get complex enough simulations that to future apply to various practical examples from real life, as environment ecologic equilibrium, or materials under radiation damage.

Results

We defined our problem and using models and simulations we got to understand it a little bit better.
The project is not yet ready because we need to see how we can make in Net logo the analysis of the results after the recovery and how we can interpret the recovery based on initial conditions and the properties imposed to the agents and patches.

Conclusions

Net Logo is a programming environment and easy to use and we still need to learn more about it in order to master it and completely solve our problem.
The disruption and recovery of systems is a very complex process that depends of many conditions and properties of the actors involved in the process.

Acknowledgments

We want to thank to our mentor
Liviu Popa-Simil for :
- guiding us in the supercomputing initiative and
- teaching us valuable lessons

to
Mrs. Stevens Pauline our sponsoring teacher, and to
Supercomputing challenge organizers for helping us learn more on supercomputers and programming

References,

http://ccl.northwestern.edu/netlogo/docs/ Net Logo website
http://en.wikipedia.org/wiki/Healing Wikipedia website/healing
http://www.thefreedictionary.com/healing the free dictionary website
http://jmvidal.cse.sc.edu/netlogomas/ Multiagent NetLogo Models
http://www.challenge.nm.org/interims/interim_guidelines.shtml

Annexes:

1 - Tables, and Excel simulation of a possible result

Rabits before disruption

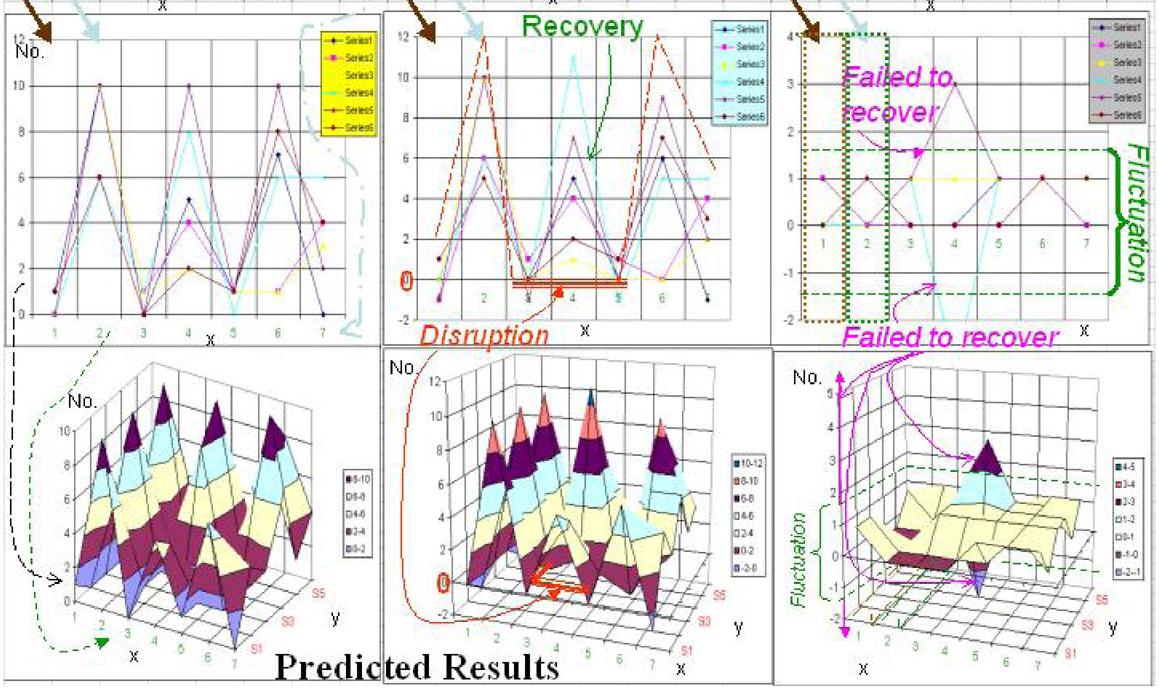
1	10	0	5	1	7	0	S1
0	6	1	4	1	1	4	S2
0	10	1	2	1	1	3	S3
0	6	1	8	0	6	6	S4
0	10	0	10	1	10	2	S5
1	6	0	2	1	8	4	S6
1	2	3	4	5	6	7	

Rabits after recovery

0	10	0	5	0	6	-1	S1
-1	6	1	4	1	0	4	S2
0	10	0	1	0	0	2	S3
0	6	0	11	-1	5	5	S4
-1	10	-1	7	0	9	2	S5
1	5	0	2	1	7	3	S6
1	2	3	4	5	6	7	

Differences before and after disruption

1	0	0	0	1	1	1	S1
1	0	0	0	0	1	0	S2
0	0	1	1	1	1	1	S3
0	0	1	-3	1	1	1	S4
1	0	1	3	1	1	0	S5
0	1	0	0	0	1	1	S6
1	2	3	4	5	6	7	



We encountered difficulties in understanding and interpreting the results therefore a Excel simulation was created using random number generators that simulated the chess board example with a little bit more animals but only one type and less patches in order to be more visible.

The left side represents a simulation of the rabbits only, distribution on a stripe patches brown and green, generated randomly after the routine “create animals”.

The table represents the values of rabbit number on each patch on a 6 x 7 surface, with brown and green strips.

The charts represents the same data in two types – one in line by line, also called “series”, and the bottom one in a 3D view where the colors are showing various levels of the number of rabbits.

In the center by the red-triple-continuous line is represented the effect of the “disruption”, that emptied the central space and brought the number to “zero” in center and increased the number on the borders – in the “no causality” version we plant to develop.

The arrows show this effect, not shown in the table.

After the disruption ends, recovery starts, and the central patches are again repopulated and after a time frame or number of moves, it reaches the numbers presented in the central table, and represented in the two charts.

In order to evaluate how good was the recovery we subtracted the two matrixes – element by element and obtained the right table.

We understand that in life systems the recovery may not be perfect and the difference may not be zero, but we have some small values we will attribute to fluctuation.
In the points the differences will be significant, say greater than 20% of the initial value, we may attribute that to the failures in recovery, inside the timeframe selected.
That feature will help us learn the recovery time needed for each system, to recover completely and further reach our project goals.

The most significant and original achievements on the project

Our original achievement is that we succeed to model a complex problem like calamity and recovery, using a chess board-checkers distribution simulating the behavior of 2 types of animals "rabbits" and "squirrels" distributed on a territory. We have felt in this approach, and even in one more simpler, as shown in the excel tables; because the agent based code we developed is not complete enough to simulate a reality that looked obvious to us.

A significant result for us was that we started to learn net logo and Excel and to use the computers in order to solve our problem and see how well it may match the reality, and we learned that it is not as simple as we previously anticipated when we knew less about the problem, and a fine tuning of parameters that makes our model match the reality is needed.