Distributed computing GIK2NX

Version 1.2

Innehåll

Prerequisite

You need to have SPADE up and running before the lab and checked that the Hello Dummy Agent program works. You need to have the Netlogo program downloaded and usable.

Purpose

This assignment will link the course elements so far and give you a chance to show that you have achieved an understanding of using your knowledge. It will be focused on SPADE -software agents and Agent Based Modelling.

Deadline

See Learn

How to study

This lab is solved within the groups.

Aid

Everything, but watch out for plagiarism. Plagiarism is the shortness to hand in on someone else's work as their own. Obviously, if it is a groups work, then you will have the same solution, but all your names need to be on the solution. In many cases, plagiarism, this can happen unconsciously, and that is why you have to be careful and conscious about how to use external sources.

Tools

Use Intelij IDEA for the coding, python as programing language and SPADEs framework. Use Netlogo for the ABM part.

Submitting the assignment

When submitting :

You choose whether you want to write text/Code in either Swedish or English).

Submit you documentation (make sure to have all names and group number in the doc), recorded links and zip the Spade project code and the ABM project.

When submitting add a comment on what grade you are aiming for: G or VG.

Name the zip with your group number followed by lab number, for example, group1_lab1.zip in Learn (Blackboard). Your submission time is automatically registered when you submit.

Grades

U – Fail

G- Pass

VG – Pass with merit. To achieve VG the work, need to pass G and the extra parts for VG, two parts! It needs to be handed in in time (not after deadline) and it is not in need of complementary work. You need to submit the parts, at VG level, for both SPADE and ABM to be able to get VG as a grade for the assignment.

Assignment

For grade G [SPADE]

You are going to create two Agents that can communicate to each other. The sender Agent is going to read a csv dataset file. And is going to send it to the second Agent. The second Agent is going to do calculations based on the CSV files data. And present its results.

Documentation:

Create a documentation where, based on the data set you choose, you justify and explain why it is needed and address a need for the program and the agents you use. For example: Is it a pharmacy company that sells vaccines and needs to know how much to buy for the next flu season? Does the agent make a prediction? A summation of last year's flu season??

Remember to use references for your statements, include a reference list on the last page.

Create a use case UML diagram and include it in the documentation. (https://www.youtube.com/watch?v=Omp4RbHbB0s) here's a link to a YouTube explanation if you don't know what it is. See Figure 1 of an example of a use case for another course where R is connected to Agents. Make sure that it is transparent with the "made up" company and its chosen calculations.

Coding:

Choose a dataset file of your own choosing but it should be at a minimum of 100 lines of input. There is a lot of free data at Kaggle (https://www.kaggle.com/datasets) . You can use other datasets too but make sure to reference where you got it. Make sure to include a copy of the dataset in your zip file.

Based on the data, the receiving agent (calculation agent) should make at least three calculations and present it to the user. These three calculations need to be motivated and relevant to the usage of your program.

Comment the code on a level as you would explain it to a fellow student make it the way you would have wanted it to be if you were to take over it from another group. Make sure all group members names are included as comments in the beginning of the code.

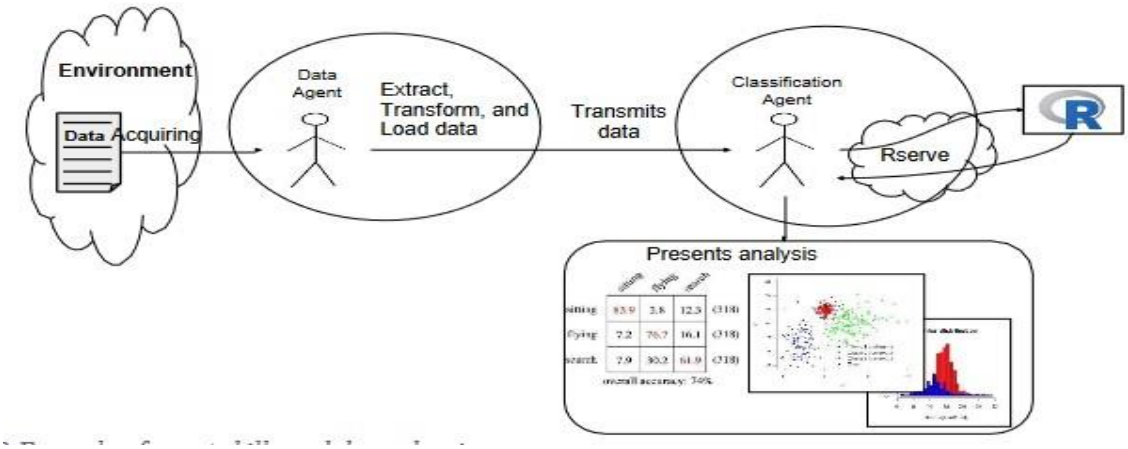

Figure 1 Use Case Uml diagram example

For grade VG [SPADE and ABM]

Part 1 [SPADE] VG-level

Coding:

The sender agent cannot have the CSV file name hard coded; it needs to check in a folder if there is a csv file. It should check the folder with an interval, when it finds a csv file it sends its information to the other agent. When the csv file is used it is removed from this folder.

The receiving/calculation agent uses the csv file and does **five** motivated calculations. It presents these to the user in the console window and saves a file in a destinated folder where the user later can reach it and use it for the companies favor. The new file with calculations should be named with the date so the company can have this done repeatedly and the files don't overwrite each other. See documentation demands for G – it's the same just this added features mentioned above.

Part 2 [ABM] VG-level

You are going to create a model for a virus infection in Netlogo.

Coding:

Create a Netlogo model for a virus infection.

Humans can have different states: Healthy and infected.

The start population should be determined by a slider and one slider for how many who has the infection from start. The infection spreads if a infected human collides with a uninfected human. If an infected meets an infected the infection becomes more severe. The infections get worse for each time a human who is infected meets another human infected or by the days not treated. Humans should die of either old age or infection.

Healthy humans can reproduce with 100% success rate. Infected humans can reproduce with lesser chance of success. They need a certain level of energy/health to reproduce. If the human reproducing the child is born infected.

Hospital: A patch area should be a hospital, infected humans need to try to get there, one move per tick. If they get there in time they get treated and becomes healthy. If they come to late the hospital can't save them but lets them die, there and not infecting more people. Make a slider for the value of when the infection is treatable or not.

Use monitors and labels where needed to make the model easier to overlook and understand.

Documentation:

In the documentation write about the different chooses and values you used. How you planed the model and its execution. Is there a turnover point? A number where the humans all get healthy? All get infected?

EXTRAS below – Does *not need to be completed and/or submitted*! Only according to your own preferences and if there is time.

Additions to the part described above. Add these rules and features in the model and in your documentation reflections:

Human Age, fear– An old person gets more rapidly sick by the infection.

Fear - Unvaccinated humans are afraid of infected and when getting close they will choose to move away from an infected human. (That could mean that they jump into another infected, who knows?

But they will react to getting to close to a infected with a "moving reaction" but only one reaction per human/tick.)

New State: Immunity – being able to collide with infectious humans without being infected. (see the vaccination part below)

Hospital Resources - used when healing some infected back to health.

Make a limit for how many the hospital can treat until it has used up all its resources. When a hospital has no more vaccine another hospital patch opens randomly somewhere else. All infected will try to get to the new location. Humans healed after infection gets partial immunity.

Vaccination patch – Humans can get vaccinated.

Infected humans can't not enter the vaccinations patch. To young can not be vaccinated.

Add a vaccinations patch when healthy humans step on it they get a partial immunity. Which protected them for a while from collisions with infected. Add a slider to change the value of the vaccine effectiveness. Create a slider of how many of this vaccination patches are available.

If a healthy person is vaccinated several times its totally immune. Make a slider for this number of times. Immune humans produce children will be born with partial immunity.

Make a limit for how many the hospital can treat until it has used up all its resources. When a hospital has no more vaccine another hospital patch opens somewhere else. All infected will try to get there when infected moving one step each time.

Recording

Finish this assignment with doing a recording where all the group members participate and start with saying your names! When recording explain the code (you can keep the comments in the code but try not to read them as text, talk freely) and make a demonstration of the code and run it. Do this for all parts. It can be two recording or one, it does not matter. Include the link to the recordings in the documentation. If you want to add them in the hand in section in learn so I can easily find them.

Good luck!