ID2209 - Distributed Artificial Intelligence and Intelligent Agents Final Project

Kishore Kumar, kishorek@kth.se   Abel Valko, valko@kth.se

December 2021

1 Introduction

The objective of this project is to combine all the knowledge we have gained so far and simulate a real world scenario with several agents which communicate with each other in an environment. The nature of communication depends on whom the agents are conversing with and where the agents are conversing in.

1.1 How To Run

Extract the "Project.zip" folder provided to you. Open Gama and import the Project - v1.gaml model found in the extracted folder called "Project". The experiments can be run by clicking the MyExperiment button. The model can be tweaked using the following parameters:

- *partyPersonCount*, *sportsPersonCount*, *historyPersonCount*, *readingPersonCount* and *gamingPersonCount* control the number of created agents of each type
- *typePlaceModifiers* and *placeTraitModifiers* determines the default experience of an agent in a location and how that location amplifies or weakens the influence of certain traits, during a conversation with another agent respectively.
- The visual representation of each agent can be altered by changing the base aspect. Colour, shape and size can be set here.

2 Species

This model employ two species, the Person species and the Place species, both of which have various subtypes which determine the exact behaviour and nature of the agents. The Person species represents the guests, individuals who travel to the Places where they interact with each other. The Place species are simple representations of locations where interactions can occur.

2.1 Person

The Person agents visit the Places and have interactions with other Person agents. These interactions are random, but their outcome is determined by the compatibility of traits of the two agents. The Person agent has the properties, reflexes, and actions as outlined below.

Properties

The Person species implements the fipa skill for communication between agents as well as moving for simulating movement between places. There are a number of critical parameters that define the behaviour of Person agents, which are the following:

1. **type**: This determines the type of person the agent is and can be one of five categories; party, sports, history, reading, and gaming. Naturally, this is a simplification of reality, but is a suitable classification as it is intuitive for a demonstration of a multi agent system with agent-agent interactions.

2. **generosityScore**: This score, which can take values between -1 and 1 and is generated randomly at the beginning of the simulation, models how generous the agent is and influences the outcome of interactions with other agents.

3. **loudnessScore**: Similar to the generosityScore, this trait also takes values between -1 and 1 and is generated randomly. It models how loud an agent is.

4. **adherencyScore**: Similar to the above two traits this takes values between -1 and 1 and models how flexible the agent is with following rules.

5. **happinessScore**: This models how happy the agent is, and is the main parameter being observed. The initial value is 0 and is affected by each interaction the agent has. The above traits, as well as the type of agent and location of the interaction determine how an interaction affects this score, either negatively or positively.

In addition to the above variables there are a number of internal variables that affect functionality but not the overall model. These are omitted from the report.

Also worth noting is that the agents are visualized as small dots in the experiment gui with colours depending on their type, as seen in the table below:

party	blue
sports	red
reading	yellow
history	green
gaming	purple

Reflexes

The Person species has the following reflexes implemented:

- **chooseTarget**: When the simulation starts, the agents select a target location to visit at random.
- **reachTarget**: This reflex is used when the agent has not yet gotten to their target location. They will move (using the moving skill) towards their target location until they have reached it. Once the agent has reached the location they are added to the location's guest list which is used to keep track of possible interactions, and they will begin interacting with other agents.
- **choosePerson**: If the agent is not already engaged in an interaction, they will randomly select a target agent to interact with who is also at the same location. Once a partner (other Person agent to interact with) has been randomly selected the agent will send a fipa cfp message to start a conversation. This message will contain the initiating agent's type, generosityScore, loudnessScore, and adherenceScore.
- **receiveCallForProposes**: Agents receive these conversation initiating cfp messages using this reflex. If they are already in a conversation with an other agent they refuse, sending a refusal message to the initiator. If however they are not already engaged, they will respond to the cfp and send their own type, generosityScore, loudnessScore, and adherenceScore. The agent also calculates the effect of the interaction based on the combination of both agent's scores, types, and the location they are at. This is done through the calculateInteractionEffect action, described later in this section.
- **receiveRefuses**: This reflex handles any refusals received to their cfp interaction initiations.
- **receiveProposes**: This reflex handles the response to a cfp (for the initiating agent). Similar to the successful conversation case in receiveCallForProposes this reflex calculates the conversation's effect by calling calculateInteractionEffect with the provided data.
- **endConversation**: This function ends the conversation after some time has passed. If the conversation had a negative effect, i.e. the agent did not like the other person, a counter is incremented which tracks consecutive negative interactions.

Actions

The following is the only action implemented by the Person species:

- **calculateInteractionEffect**: This action implements the interaction rules as described in the following section. It calculates the outcome of an interaction based on the two agent's types, location, and traits and produces a number. The more the two agent's are well aligned the more positive this value is, while the more their personality clashes the more negative it is.

2.2 Place

The Place species is very simple. It has a type, which may be either "pub" or "museum", a guestList array which has all the Person agents currently visiting it, and an idleGuesList array which keeps track of agents that are also not in a conversation. The pub is represented by a triangle while the museum is a square.

3 Implementation

3.1 Overview

The simulation is implemented using the above two species. An overview of the logic is as follows

1. One agent of each type of Place is initialized at the beginning of the simulation as well as 10 agents of each type of Person.
2. The Person agents select a Place at random to go to. Once there, they begin interacting.
3. A Person selects another Person to interact with. They send a cfp with all of their traits.
4. A Person reacts to a cfp by sending a reject message if they are already in an interaction. Otherwise they reply to it with their own traits and calculate the effect of the interaction.
5. The initiator of an interaction receives the traits from the other agent and calculates the interaction effect.
6. If an agent has had more than a certain number (default 2) of consecutive negative interactions they move to the other location.
7. the global average happiness score is tracked over time.

3.2 Interaction Rules

The effect of an interaction is calculated using the set of rules presented in this section. These rules depend on the location, the types, and all three traits of the agents. These interaction rules are completely arbitrary, however, they are largely intuitive. The exact values and calculations may be unrealistic, but are sufficient for the purposes of this demonstration.

Firstly, two tables are created which determine the importance of a trait based on the location and the added happiness value of an agent due to the location they are in. To make this more concrete, note that an adherent person might have less of a problem with a person that is less adherent to rules at a pub than at a museum. On the other hand, a history loving person may rather have an average conversation in a museum than a slightly better one in a pub. This is accounted for through the following two tables.

Below the modifier for each trait based on the location can be seen, these are the placeTraitModifiers:

	pub	**museum**
generosity	2	0.5
loudness	0.5	2
adherence	0.5	2

In the table below, the additional happiness for each type of person and place is presented, note that some values are negative:

	pub	museum
party	1	-1
sports	1	0
history	0.5	1
reading	-0.5	1
gaming	0.5	0.5

A type modifier is calculated based on how compatible two agents' types, i.e. interests, are. This is done as per the table below:

	Gamer	**Sports**	**Reading**	**History**	**Partying**
Gamer	1	1/2	0	1/2	1/2
Sports	1/2	1	0	0	1/2
Reading	0	0	1	1/2	0
History	1/2	0	1/2	1	0
Partying	1/2	1/2	0	0	1

Next, modifiers for loudness and adherence are calculated according to the following formula (presented for adherence, but equivalent for loudness):

$$aderencehModifier = placeTraitModifier * selfAdherenceScore * partnerAdherenceScore$$

This results in a positive modifier if the agents are similar, and negative if they are different. The more extreme they are, the larger the modifier becomes.

The generosityModifier is calculated somewhat differently. If both agents are generous, then the outcome is good, if one agent is generous and the other not generous then the outcome is still good, but less so. If both agents are not generous the outcome is poor. This can be seen in the table below:

		Agent 1	
		+	-
Agent 2	+	a1.generosityScore * a2.generosityScore	-0.5a1.generosityScore * a2.generosityScore
	-	-0.5a1.generosityScore * a2.generosityScore	-a1.generosityScore * a2.generosityScore

The final score is the sum of these values.

	typePlaceModifier
	loudnessModifier
	adherenceModifier
	generosityModifier
+	typeModifier
=	effect

This is then added to the agent's happiness score.

3.3 Communication

The agents communicate with each other using the FIPA-Contract-Net protocol. The conversation protocol is very much similar to the Dutch Auction task which we performed in lab 2. As you can see in Figure 1, an agent who wants to strike a conversation with another agent in the same location, first picks a random guest (*chosenPerson*) from the *idleGuestsList* array of the place of visit. Then it sends a Call For Proposal (cfp) message to the *chosenPerson (Agent 2)* with its **chosenPerson,type,generosityScore,loudnessScore and adherencyScore**. If the *chosenPerson* agent has another *chosenPerson* of its own, then it sends a *refuse* message to Agent 1. If Agent 2's *chosenPerson* holds a *nil* value, then it sends a *propose* message to Agent 1 with its own chosenPerson, type and traits in addition to a value called *converseTill*, which indicates the time step at which the conversation between the two agents come to an end. After the conversation ends, the agents reset their *chosenPerson* values and repeat the process with other agents.

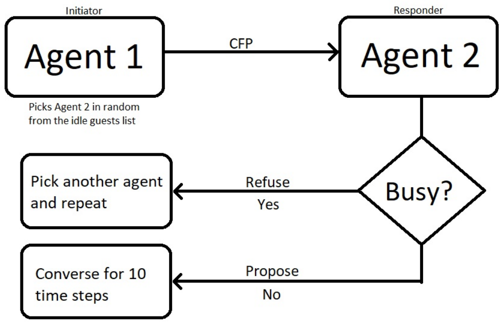


Figure 1: Block diagram of a FIPA-Contract-Net conversation between two agents belonging to the *Person* species.

4 Results

When the simulation is executed, few interesting observations could be noticed. Before discussing the observed data in detail, let us have a look at the simulation visually. Figure 2 shows how the simulation looks when the experiment is run by the user. We could see the pub shaped as a triangle and the museum shaped as a square on the top left and the bottom right corners of the arena. There are 50 *Person* agents belonging to 5 different types, spread all over the arena. Each agent has a colour corresponding to its type.

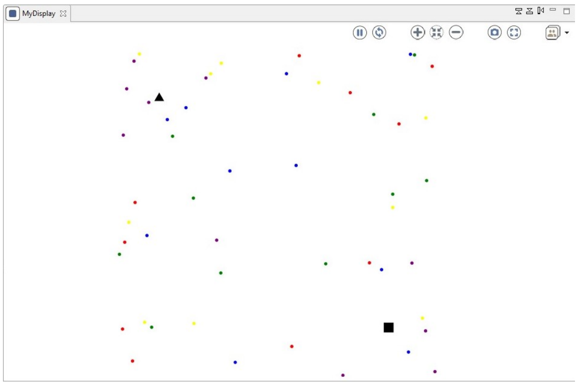


Figure 2: Beginning the Simulation in GAMA.

Figure 3 explains how the simulation looks like when it is run for several time cycles. The agents move to the place which they have chosen in random and starts conversing with their fellow neighbours.

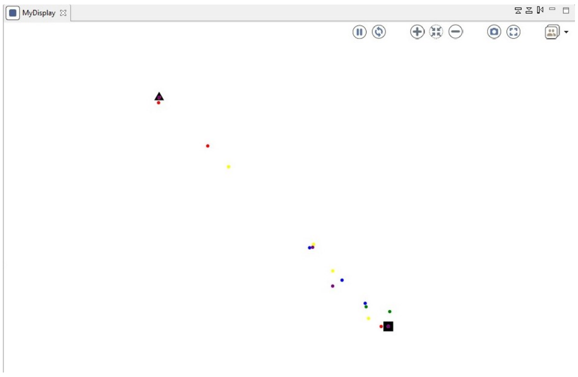
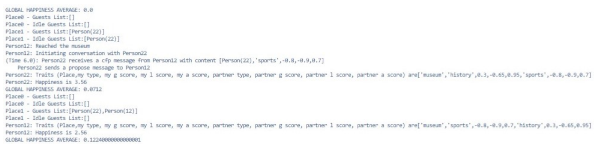


Figure 3: GAMA Simulation after 'n' time cycles.

An interaction taking place between a *Sports* person and a *History* person inside a museum is shown in 4. We could see that the Global average happiness increases as the agents are having a good conversation with each other. The *idleGuestsList* becomes empty once the agents start conversing with each other.



```
GLOBAL HAPPINESS AVERAGE: 0.0
Place0 - Guests List:[]
Place0 - Idle Guests List:[]
Place1 - Guests List:[Person(22)]
Place1 - Idle Guests List:[Person(22)]
Person12: Reached the museum
Person12: Initiating conversation with Person22
(Time 6.0): Person22 receives a cfp message from Person12 with content [Person(22),'sports',-0.8,-0.9,0.7]
    Person22 sends a propose message to Person12
Person22: Traits (Place,my type, my g score, my l score, my a score, partner type, partner g score, partner l score, partner a score) are['museum','history',0.3,-0.65,0.95,'sports',-0.8,-0.9,0.7]
Person22: Happiness is 3.56
GLOBAL HAPPINESS AVERAGE: 0.0712
Place0 - Guests List:[]
Place0 - Idle Guests List:[]
Place1 - Guests List:[Person(22),Person(12)]
Place1 - Idle Guests List:[]
Person12: Traits (Place,my type, my g score, my l score, my a score, partner type, partner g score, partner l score, partner a score) are['museum','sports',-0.8,-0.9,0.7,'history',0.3,-0.65,0.95]
Person12: Happiness is 2.56
GLOBAL HAPPINESS AVERAGE: 0.12240000000000001
```

Figure 4: GAMA console of the simulation

Let us discuss the observed parameters. It is very important to note that the results depend on the conditions specified by the user. We will look into this in detail below. We have chosen to observe several local parameters which are specific to the agents and several global parameters which are measured based on the interaction of every single agents. The first parameter which is being observed is the happiness of individual agents as shown in Figures 5,6 and 7. The result which we observe is totally dependent on the initial conditions and the places of visit. In this case, we can notice that the happiness of the Partying person and Reading person is below 0 while the happiness of the Sports person,History person and the Gaming person is increasing positively. We can notice that the score stays constant for a longer duration for the agents with a negative score compared to the agents with a steadily increasing positive score. This is due to the fact that an agent which is unpleasant experiences in a sequence constantly moves to the other location. This movement takes a lot of time during which there are no interactions taking place with another agent.

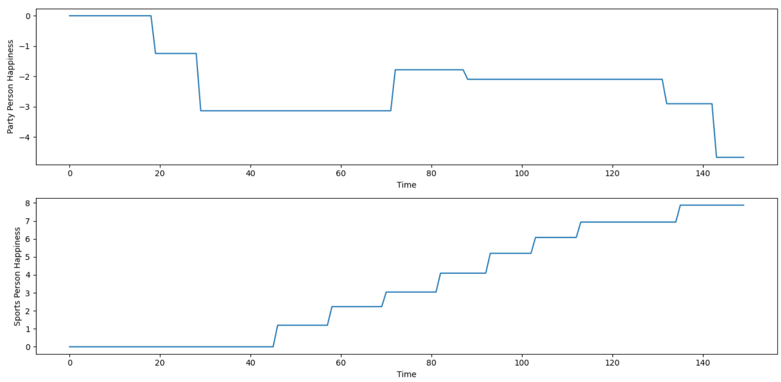


Figure 5: Happiness plot of a Partying agent and a Sports agent for 150 time cycles.

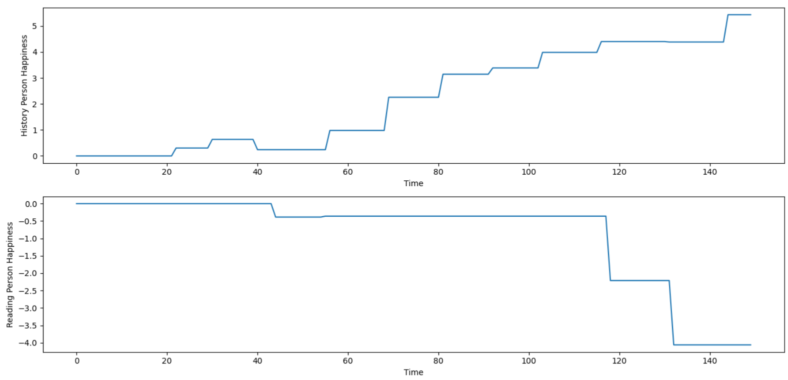


Figure 6: Happiness plot of a History agent and a Reading agent for 150 time cycles.

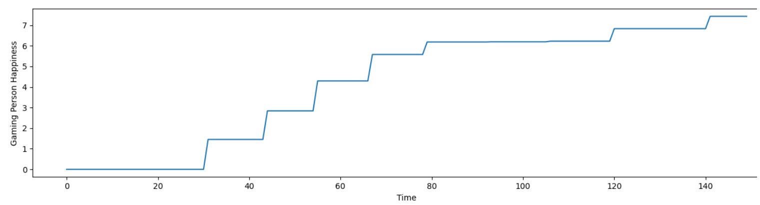


Figure 7: Happiness plot of a Gaming agent for 150 time cycles.

The next parameter which we focused on is the Global Happiness Average. There are two different scenarios which can be observed based on whether the *happinessScore* of each agent is reset after a conversation or added on top of the previous *happinessScore*. In Figure 8, we can see the Global Average happiness of all the 50 guests increases constantly when the *happinessScore* of the agents is added on top of the previous score. This is because each agent inherently tries to maximise their *happinessScore*. Even though the *happinessScore* might have a negative trend for a few unlucky agents, the cumulative score is always on the rise due to this reason. When the *happinessScore* is reset however, we notice that Global Average Happiness is fluctuating between 0 and 1 as no matter how good an agent's conversation was, it won't matter in the next conversation and it starts all over again.

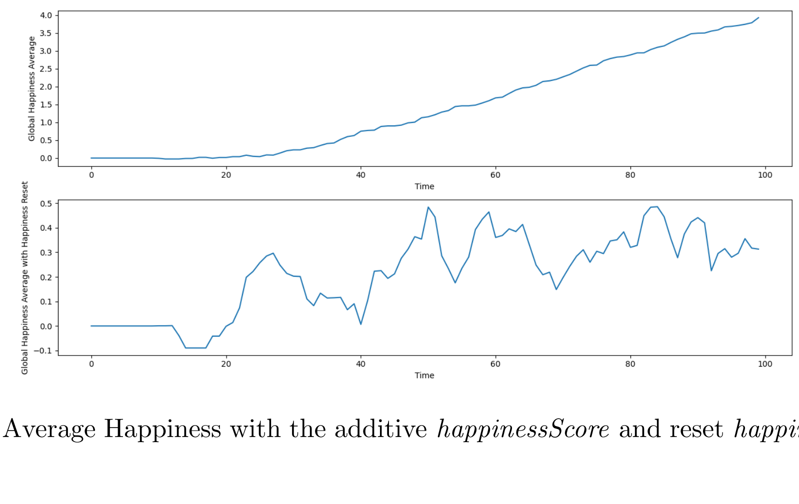


Figure 8: Global Average Happiness with the additive *happinessScore* and reset *happinessScore* over 100 time cycles.

Finally, we are observing another local parameter which is associated with the two places of visit: The pub and the museum. In Figure 9 we could realise the varying trend in the crowd of these places over time. There is a generally increasing trend for a while and later the crowd starts to fluctuate around an equilibrium. The existence of some agents with their inherent traits being incompatible with every location and agent is the cause for this phenomenon. For example, we noticed previously that the partying agent and the reading agent experience unpleasant conversations all the time and keep moving between the two locations. Another interesting observation is that the crowd is unevenly distributed between the two places. This totally depends on the location chosen by the agents in the beginning and varies with each rerun of the simulation.

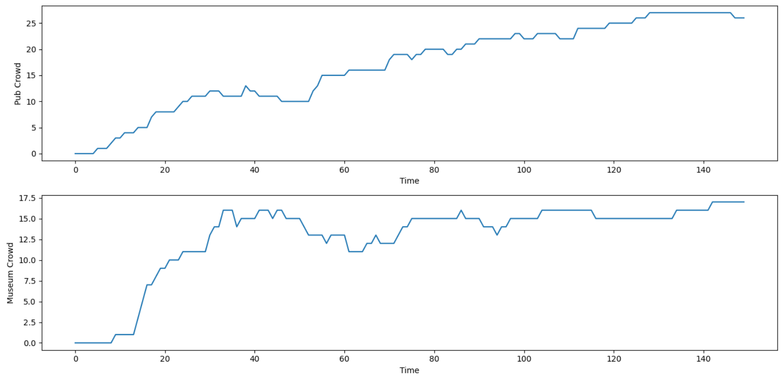


Figure 9: The crowd of the Pub and the Museum varying over 150 time cycles.

5 Discussion

This project is surely one of the most challenging undertakings we have implemented so far in GAMA. It required the knowledge of all that we have learned so far in this course ranging from FIPA communication to using dynamic parameters in the agents. The model which we have implement is nonetheless a very simple approximation and doesn't exactly replicate the intricacies involved in the real world. The interaction taking place between the agents are of a fixed duration which might not be the case in the real world and people might choose to converse continuously with the same person throughout their stay in a particular location instead of looking for new people to communicate with after every conversation. We could say that the travel is highly idealized as well, as the agents have complete knowledge about the locations which they can visit and the people present in those locations. That might not be the case in the real world. The interaction rules which we have described in the objective system is arbitrary. Modelling the psychology of a human is a task is orders of magnitude perplexing and thus, we have once again settled with simpler rules. However, the model does a really good job in showcasing the behaviour of a multi-agent system.