Behavior of different agents
Project

Emil Ståhl and Peyman Peirovifar

December 15, 2020

1 Introduction

This report covers an experiment that demonstrates and analyzes the behaviors and interactions between different types of agents by simulating a festival event. The agents have their own personal characteristics, preferences and intentions of visiting the festival. Each agent has a preset level that determines the intensity of three personality traits along with one of five personality attributes. These will together determine how the agent interacts with other agents. The experiment is set to investigate what happens when these agents meet and interact with each other. How will the agents change each others behaviors? The experiment is conducted as a simulation using the GAMA platform which is an open-source environment for spatially explicit multi-agent simulations.

2 Approach

This sections describes the method and approach used during the project work.

2.1 Method

We started this work by deciding what functionality our simulation would have, what values to monitor, the setup of the agents used as well as what rules the agents should follow. Additionally, we studied our previous work in order to decide if we could reuse some parts. When the simulation was running correctly we began to perform experiments by changing some rules, more about this in chapter 4. Results. Finally, the results was gathered and displayed in this report as well as discussed upon on in chapter 5.

2.2 Simulation

To simulate the festival we used some parts from our previous work, such as "GAMA and Agents", "Negotiation and Communication (FIPA)" and "Coordination and Utility". As an example, the dance floors from the third project is reused in this experiment. Similar to the first work in "GAMA and Agents" there are places for the agents to visit when desired. These places are dance floor, store, and pub. Furthermore, each agent has a predetermined personality trait which determines how they will interact with other agents, accept proposals and move around the festival area. Lastly, the simulation has the following structure.

- 5 types of agents totaling a number of 50
- Rules of how the agents interact with other agents
- Each guest has 3 personal traits of a varying intensity
- Three types of places where the agents can meet
- Continuously running
- FIPA protocol for long distance messaging

3 Experiment

This section describes the implementation of the simulation.

3.1 Locations

The simulation is created with three types of locations where the festival guests can meet and interact. These places are:

- Pub
- DanceFloor
- Store

The simulation is by default set to have 9 of these places but can easily be changed by altering the place_num variable.

3.2 Guests

A species called "Guest" is created to handle the logic of the agents moving around and interacting with each other. The number of guests in the simulation is by default set to 50 and can be changed by altering the guest_num variable.

3.2.1 Personality traits

Every guest presented in the simulation are given three types of personality traits with a value [0, 10] that determines how strong each personality trait are. The value is initialized once the simulation is started. The different types of traits are:

- Kindness
- Sympathetic
- Charitable

The degree of each personality trait determines how the guest will interact with other guests and to what extent they accept or reject potential proposals from other guests. Furthermore, this acceptance rate of each guest is accumulated in a global counter and presented in the form of a chart in order to monitor how it changes during the simulation.

3.2.2 Attributes

In addition to the guests personality traits, each guest is also given one of five personality attributes that decides how the guest will interact with other types of guests with other attributes as well as where the guest prefer to hangout within the festival area. These attributes are:

- Extrovert
- Introvert
- Juicehead
- Journalist
- Salesman

The preference of each guest is displayed in the table below:

Attribute	**Preffered Place**
Extrovert	DanceFloor
Introvert	Store
Juicehead	Pub
Journalist	None
Salesman	None

3.3 Interactions and proposals

The goal of this experiment is to investigate what happens when different types of agents meet and interact with each other. These interactions are in the form of making proposals to other agents regarding if they are interested in, for example, join them for a drink. These proposals can be made either on the way towards a place or at the place itself. However, the guest needs to be within at least 5 meters from another guest in order to initiate the proposal. The interaction and proposal are performed thru the FIPA request protocol:

```
    reflex startconversation when: (self.limit_interactions=false) and !self.participant
guestList <- guest at_distance 5;

if !empty(guestList) and !self.participant{
write  name + ’ I would like to know your character’;
self.limit_interactions <- true;

do start_conversation with: [to :: guestList, protocol :: ’fipa-request’, performative ::

contents ::  [self.name,self.type] ];
initiater <- true;
list guestList2 <- guestList;
int M <- length(guestList2);
loop i from:0 to:M-1 {
participant_name<- guestList[i];

participant_name.participant<- true;
participant_name.limit_interactions<- true;
add self to: participant_name.initiater_name;
}
self.initiater <- true;
self.participant <- false;
}
if !empty(guestList) and self.participant{
self.dialoge <-true;
}
}
```

3.3.1 Time limit for proposals

In order to avoid that a guest makes multiple proposals to the same counterpart over and over again an attribute **time_limit** is given to each guest. This attribute will be reset every 200 - 400 simulation cycles and when that happens the guest will be given a new target point to move to. This prohibits guests from being stuck and ensures a smooth running simulation.

3.4 Agents and rules

Every guest is given a set of rules that determines how they will interact with other agents and if they will accept proposals from others. The rules is in the form of a floating point between [0,1] that is predetermined at creation. The rules are defined as an attribute:

Agent Rules
Extrovert_attribute
Introvert_attribute
Juicehead_attribute
Journalist_attribute
Salesman_attribute

A proposal from one agent to another is determined by the corresponding value of the guest making the proposal. For example, if a guest receives a proposal from a guest with the type of a introvert and the receiving guest has an attribute rule of 1.0 as Introvert_attribute the proposal will always be accepted. Similarly, an attribute of 0.0 will result in that the proposal being rejected at all times.

The attributes are initialized like this in the code:

```
aspect base {
if self.initialized = false {
if self.type = 'Extrovert' {
self.color <- #lawngreen;
self.Extrovert_attribute <- 1.0;
self.Introvert_attribute <- 0.5;
self.Juicehead_attribute <- with_precision(rnd(0.0,1.0),1);
self.Journalist_attribute <- 0.2;
self.Salesman_attribute <- with_precision(rnd(0.0,1.0),1);
} else if self.type = 'Introvert' {
self.color <- #cyan;
self.Extrovert_attribute <- 0.0;
self.Introvert_attribute <- 1.0;
self.Juicehead_attribute <- 0.0;
self.Journalist_attribute <- 0.5;
self.Salesman_attribute <- 0.0;
} else if self.type = 'Juicehead' {
self.color <- #orange;
self.Extrovert_attribute <- 0.4;
self.Introvert_attribute <- 0.6;
self.Juicehead_attribute <- 1.0;
self.Journalist_attribute <- 0.2;
self.Salesman_attribute <- 0.8;
} else if self.type = 'Journalist' {
self.color <- #magenta;
self.Extrovert_attribute <- 0.8;
self.Introvert_attribute <- 0.4;
self.Juicehead_attribute <- 0.6;
self.Journalist_attribute <- 1.0;
self.Salesman_attribute <- 0.2;
} else if self.type = 'Salesman' {
self.color <- #black;
self.Extrovert_attribute <- 1.0;
self.Introvert_attribute <- 0.6;
self.Juicehead_attribute <- 0.2;
self.Journalist_attribute <- 0.4;
self.Salesman_attribute <- 0.0;
}
}
```

3.4.1 Priority of personality traits and agent rules

The personality traits given to each agent has absolute priority over the agent rules. If a personality trait is <5.0 the proposal from another agent will always be rejected regardless if the agent has a high attribute for interacting with that type of agent.

4 Results

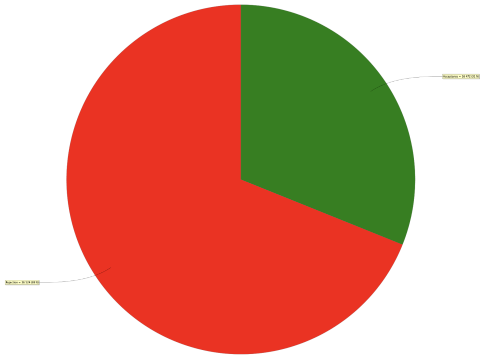
This section presents the results from running the experiment.

4.1 Acceptance and rejection rate

The goal of this experiment is to measure the acceptance and rejection rate of the proposals made between all the agents. This is done by accumulating the global counters accept_rate and reject_rate, these variables are updated in the if-clauses of the interact reflex.

```
    if self.sympathetic>=5 {
decider <- flip(self.interact_Extrovert);
if decider=true {
accept_rate <- accept_rate + 1;
do agree with: [ message :: m, contents :: ['interested'] ];
} else if decider=false {
reject_rate <- reject_rate + 1;
do refuse with: [ message :: m, contents :: ['not-interested'] ];
}
```

In the experiment a chart is displayed with these values and is continuously updated throughout the entirety of the simulation. The results are shown in the chart below:



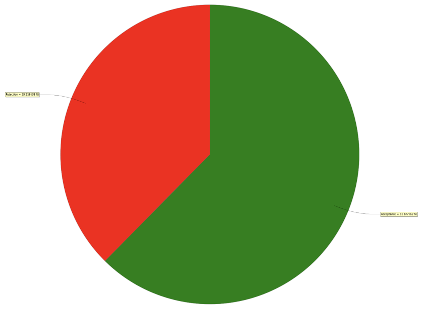
As seen in the chart the rejection rate is ˜69% and the acceptance rate ˜31%. Interestingly, this distribution is the same every instance of the simulation. This is most likely due to the way the agent rule attributes and personality traits are implemented, the simulation will therefore by probabilistic distribution always converge to these values shown in the chart above.

4.2 Altering the rules

This sections shows the results of altering the original rules described in the previous sections. Every alteration is compared to the original set of rules.

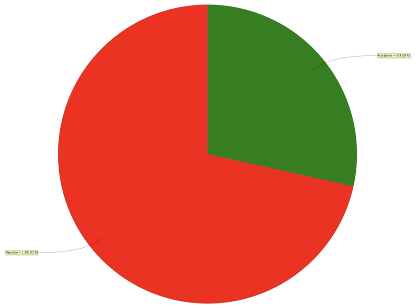
4.2.1 Lowering threshold level

In the following chart, the threshold level for the personality traits was lowered from 5.0 to 3.0. The acceptance rate, self-evidently, increases to ˜68%.



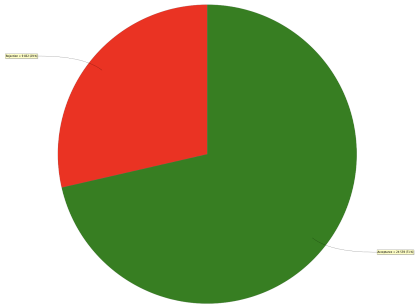
4.2.2 Removing the time limit

By removing the time limit attribute of each agent that prohibits them from making multiple proposals to the same agent it can be seen that the acceptance rate drops significantly. This is because agents are now making new proposals to the same agent even after they have been rejected, which deteriorates their ability to move around and meet new agents that better match their preferences.



4.2.3 Giving priority to the agent rules attributes

As described in an earlier sections it was stated that personality trait have absolute priority over the agent rule attributes. By changing the personality traits to 10 which makes the receiving guest always accept the proposal based on personality trait, not including agent rules, the acceptance rate increases to 71%.



5 Discussion

The results shows that the agents present in the simulation interact with each other according to the predetermined rules. When these rules are altered it gives a different set of results, showing that the agents are in fact interacting with each other based on the rules given to them. Furthermore, the chart of acceptance / rejection rate shows that this rate is depending on who the agent is proposing to interact with. By changing the preference of each agent this rate can be altered. However, what prohibits the rate from going higher is also depending on that the agent is asking agents that do not prefer to interact with it. The solution to this could be to extend our implementation with algorithms that enables the agent to learn which type of agent it should prioritize. This can potentially be achieved with algorithms such as Belief-Desire-Interaction and Reinforcement learning.

6 Conclusion

The GAMA platform can provide a valuable introduction for those who want to simulate a model with a multi-agent environment. The high-level language and multi-library support enable users to experiment with advanced techniques, such as reinforcement learning, and the intuitive user interface can attract beginners as well. However, the platform has its limitations and other alternatives provide more targeted and modern solutions for specific problems.

This work has focused on investigating what happens when intelligent agents are told to interact with each other based on a set of rules. Each agent present in the simulation has a preset level for three different personality traits. This determines how it will react to proposals from other agents. In addition to this, each agent also have one of five possible personality attributes. This, together with the personality trait will determine how the agent will interact with other agents. Each interaction between agents will result in either an acceptance or a rejection which can be monitored in a chart during the simulation. For future work Belief-Desire-Interaction and Reinforcement learning could be implemented on top of this implementation in order to investigate if the agents are capable of learning what types of agents it prefer to interact with.