Believable Synthetic Characters with Social Identity

Bruno Antunes
Instituto Superior Técnico
bruno.antunes@ist.utl.pt

ABSTRACT

In today's games we can see a demand of "believable" AI players. We know that creating a believable synthetic player is not an easy job, but we accept that challenge and in this document we will create an agent[1] conceptual model based in social identity theory. This work was incorporated in the INVITE project. We have created a 3D game to prove the agent's model. The goal of the INVITE project is "explore the role of social identity in partnerships and social dilemmas in mixed motive tasks"[2]. That said, in this work we studied how social identity plays a role in the relations between real Human players and agents[2]. We studied related work from psychology and computer science and in this document we discuss all the work we have studied in order to explain what we think is the best possible solution (right now) for our problem. With all the knowledge learned from others' work we created and implemented a conceptual model to solve our problem and give a Human ability to a computer.

Keywords

Agents, Artificial Intelligence, Believable Synthetic Characters, Social Identity, Intergroup Relations, Social Change

1. INTRODUCTION

When we play a game against synthetic characters we want them to act in a believable way. We want believable synthetic characters. By this we mean that even if the human player realizes that he or she is playing against a synthetic character, the synthetic character should take believable and coherent decisions during the game and should also recognise different social groups.

In this paper we will study how to create believable synthetic characters based on social identity. To achieve that believability we will study how social identity plays a role in intergroup relations. First we should say what it is and why will we use it in our approach. Social identity theory was originally developed and studied by Tajfel and Turner in the 1970s and 1980s, in order to explain intergroup behaviour [3][4]. Tajfel and his colleagues proved that social identity plays an important role in relationships between humans. It can influence our satisfaction and the way we collaborate with other people. Therefore, it has an important role in our everyday life. Social identity states that people do not have only one personal self but also a repertoire of social identities for each social group that he or she feels to belong. Each of these social identities is then going to influence the individual's thoughts, feelings and behaviours. Social identity is central to every aspect of social behaviour [5][6]. It anchors us in the social world by connecting us to other people, people whom we otherwise might have little reason to trust, to like, or even to know at all, and just like Tajfel has shown, it can even increase cooperative behaviour between people who are totally strangers [5][6].

1.1 Motivation

Taking as example the video games interaction, if we can make artificial intelligence players with social identity perception and social identity reactions, and we can give them the ability to maintain relations, we can increase a lot the believability of our synthetic characters. This is our biggest motivation to study this theme and implement a possible solution to this problem in the INVITE project. In the future somebody can pick this work as a first step and build better and more realistic games than we saw today, based on human interactions, where the immersion of the players will be much better (compared to current games).

1.2 Objectives

This work addresses the problem of how the social identity influences a intergroup relation.

> *Goals:* With this work we will create believable synthetic players and integrate them in a 3D game called INVITE. In the end, we hope that human players feel that the synthetic characters with our social identity module are more believable than the same synthetic characters that do not have our social identity module. We will also explore the cooperation between our agents human players and how cooperation and satisfaction are related with each other.

Expected results: The work will produce:

1. a specification of our model.

[1] Synthetic characters, i.e., computational systems that are life-like entities and that will allow the user to interact in a natural way like if they where interacting in a real world with real people. [1]

[2] Player ↔ Player, Agent ↔ Agent, Player ↔ Agent

Appears in: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, Conitzer, Winikoff, Padgham, and van der Hoek (eds.), June, 4–8, 2012, Valencia, Spain.

2. an implementation for our model.
3. an integration of our implementation with:
    (a) ION framework. A framework to help us build our agents without dependencies of a graphic engine.
    (b) Unity 3D. It's our 3D graphic engine.
    (c) INVITE project. Invite is the project that leads this thesis. For more information about INVITE please check this url - http://project-invite.eu/
4. experimental evaluation with users.

2. RELATED WORK

The social identity theory says that, when we are part of a group, we discover the good values and all the good things about the group we belong to, but we try to find negative aspects in the groups we do not belong to, sometimes only to enhance our own self-esteem [3][7].

People do not act only as individuals but also as group members, sharing the same perceptions, goals and identity, the social identity [8][9]. It confers a shared or collective representation of who one is and involves self-categorization (cognitive), self-esteem (evaluative), and commitment (psychological) components [7].

Self-Categorization causes people to think of themselves less as unique individuals and more as relatively typical members of a group, and they act accordingly[10]. They see themselves as having the characteristics associated with group memberships, and they act as they believe group members should act [10].

We categorize people in general, including ourselves, in order to understand the social environment we are in. We do this everyday because if we put people in categories we know what type of person he or she is. We try to do the same exercise in ourselves and observe in what categories we fit in, to really understand ourselves, and what type of person we are.

We categorize ourselves in groups, and this is what social identification is all about. Sometimes we adopt attitudes (and identity) based on the groups we think we belong to. When our identification with a particular group is salient we start to talk, to act, and to do a lot of things like the group [11].

As persons we can have multiple social identities depending on the context or group. We can say that salience of an entity is a product of accessibility and fit [12].

Social comparison happens when we have categorized ourselves as part of a group and have identified with that group, then we tend to compare that group with other groups. If our self-esteem is to be maintained our group need to compare itself with other groups. This is critical to understand the prejudice, because once two groups identify themselves as rivals they are forced to compete in order for the members to maintain their self-esteem. Competition and hostility between groups is thus not only a matter of competing for resources like jobs, but also the result of competing identities [8].

Status is the outcome of intergroup comparison, it reflects a group's relative position on some evaluative dimensions [11], this is when we see our group as a superior group or inferior group, that is also called social stratification.

If we are in a low-status group (inferior group), and we compare it with another out-group (not the high-status out-group), the relevant inferiority should decrease in salience and self-esteem should recover [11]. Competition between subordinate groups is sometimes more intense than between subordinate and dominant groups [11].

The Tajfel experiments showed that individuals achieve positive self-esteem by positively differentiating their in-group from a comparison out-group on some valued dimension [8][5][9].

3. IMPLEMENTATION - INVITE GAME

In this chapter we present the demonstrator application, which we implemented in order to integrate and evaluate our agents. These demonstrator application is a game and is part of the INVITE Project (http://project-invite.eu/).

Before explain the implementation and the tools we used, first we need to clarify what type of game we will implement.

3.1 The Game

In this section we will describe the game and the objectives of it. The scenario of the game is an Island with a volcano that will erupt in a certain amount of days. The game is played by two to five teams with two to five members in each team.

The main objective of the game is to have the highest score at the end, but to complete the game we need to escape from the island. To escape from the Island every team should build a raft with wood. Every team member have an fixed amount of time to spend in everyday and he/she can decide to gather wood or gold. That resources (gold and wood) are placed in the Island.

To obtain points for the final score and win the game individually our team should build the raft, but we also need to have more gold than everybody else. The wood at the end of the game is divided in an egalitarian rule by every team member, but the gold that one player have gathered is only for its score.

Now we will give an example of a single game. When the game starts every player is spawned in his/her team camp site. Then the user should go to the resource site where he can use his/her available hours to obtain wood or gold. If he/she choose the wood, that wood is to share with his/her team in order to build a raft to escape from the island. If the player chooses gold, that gold is for his/her personal benefit.

The game has a lot of parametrizable values, for example, hours in a day, how much wood/gold is earned by one hour of "work and so on. After the mini game is played the player should return to his/her team camp. When all players are in their team camps the day advances and the players have again available hours to spend in gathering resources. The user interaction is only with the mouse. This game is a point-and-click game.

This game has a special requirement, it needs to be multiplayer, on-line multi-player.

3.1.1 *Game Parametrization*

In our game we have some configurable parameters. In this section we will present all the parameters that we can configure. This configurations are in a file called "invite.xml". In this file we can configure:

- PlayerConfiguration, is where we set up the name of all players in the game and the number of players.

- TeamConfiguration, is where we configure a team. A team have some PlayerConfigurations (that are players) and a name.
- PlayerTimePerDay, This is the number of hours that one player has to spend in a single day.
- WoodPerTimeUnit, The player can achieve X units of wood, for each hour.
- GoldPerTimeUnit, The player can achieve X units of gold, for each hour.
- WoodRequiredForRaftCompletion is the number of wood units required to build the raft.
- NumberDaysUntilEruption is the umber of days before the volcano explodes the whole island.
- DistributionRule, this is the distribution rule. When a team ends a raft, the amount of wood will be divided equally by all team elements. Imagine a team of 3 elements: when the team ends the raft, all the three players in their final score will have 85 units of wood.
- GoldPerWood is where we can configure how much one unit of wood represents in gold units. The first team finishing the game can have a higher value. All of this is configurable here.

In the game we can also have a lot of other parametrizations like our own Logger and our own agents adding just a simple "dll" to the Agents folder.

3.2 Technologies and Tools

As we stated in the last section we implemented a multiplayer game to test our agents, so we used a game development tool. We have many options, but for several reasons, including simplicity, good online support and a good community helping a lot with some details, we have chosen Unity3D[3].

We will try from now on to explain our implementation in a Top-Down approach (from a simple and abstract view to a detailed view). The simplest way to look at our system is by having Unity3D running only for graphics and game logic, and our agents will be running in a sandbox abstracted from the Unity3D code. To communicate between our agents and Unity3D we will use ION Framework with all the benefits we will explain in the next section. We can take a look at our simple architecture in figure 1.

In the next section we will take a closer look to ION Framework and all its benefits, and after it, explain how we distributed ION Framework.

3.3 ION Framework

Now that we have chose the technology for the graphics, we have to take care of the development of the agents. Agents cannot be decoupled from their environment. To help us connecting Unity3D with our agents we will use a framework developed at INESC-ID, GAIPS named ION

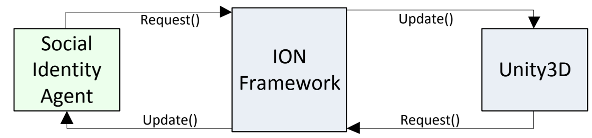


Figure 1: Our Agents and Unity3D communicating via ION Framework

Framework [14]. ION Framework is a framework for simulating virtual environments which separates the simulation environment from the realization engine[4]. In doing so, it facilitates the integration and reuse of the several components of the system.

As said in ION Framework paper, a virtual agent is by definition an entity that senses, reasons and acts within an environment [14]. Using ION Framework we can take advantage of many work done and tested by PhD students. With ION we can have a coherent access to information because all the modifications are made at the same time and because of that last point we can also have a mediation of conflicts and only commit that changes after applying some rule defined by us. With ION we can subscribe a particular bit of information which will be delivered later. They[5] use the observer pattern [15] to provide an event-driven paradigm. Similarly to what happens with Requests, Events are not processed immediately. Likewise, their handling is performed by Event handlers at a specific phase of the update cycle, the Process Events Phase. At that time all interventionists registered to get a particular Event are notified if that Event happened.

While Requests are the desired changes to the simulation state, Events are the information of which changes effectively took place.

The ION Framework allows dynamic configuration changes and it is possible to completely change the simulation state behaviour in runtime. We can add or remove Elements to the simulation, but also change how these Elements inherently behave by modifying their Request Handlers.

Unfortunately ION Framework is not ready to work in a LAN(Local Area Network) with a server and multiple clients connected to the server. For this reason we invested a lot of time in "ION Framework Remoting". "ION Framework Remoting" which is a normal ION Framework but distributed to many computers. In the next section we explain how we achieved this goal.

3.4 ION Framework Remoting

As we said in the last section we invested a lot of our time to transform ION Framework from a single-computer framework to a multi-computer framework. To achieve this we used Microsoft Remoting, a framework to access objects from another computer.

One of our biggest problems was that "not all objects can be serialized". Another of our biggest problems was how to throw events from the server to the client, because C# Remoting has numerous limitations. Those limitations were the reason why Microsoft created Windows Communication

[3]"Unity is the development environment that gets out of your way, allowing you to focus on simply creating your game. Developing for web, mobile, or console? Unity is the tool for the job" [13]

[4]Unity 3D

[5]ION Framework team

Foundation (WCF). We tested WCF but we have not used WCF, because Unity 3D does not really uses Microsoft C# but instead an open-source implementation called Mono, and Mono has some limitations with WCF that are not yet solved, but will be in a near future, so probably then we could migrate all of our work to use the new technology (WCF).

Throwing events from the server to the clients was our biggest problem, which we solved with a really simple solution: Remoting. We just put at the same time every client being also a server, but in another port, and when the real server needs to throw an event to the client, the server just calls the client in that server port.

When a client wants to be informed about a change in the server, that client should register his intention in the server giving it a function to be called, and when that action happens in the server, the server should call all the registered clients in that event/action. This works like a simple publisher-subscriber system, which is not possible in a standard C# Remoting application, without all the clients being at the same time "false" servers.

3.5 ION Framework in Unity3D

The ION Framework is integrated in Unity3D by adding the ION Framework's DLLs or source files (in C#) directly into Unity3D's application assets. Further, this integration is supported by script Components (in C#) whose purposes are to define the Entities, Properties and Actions and maintain a link with those elements in the ION simulation.

Therefore, we can design the simulation's environment by attaching these Components to Game Objects. In the next section we describe some of the most important concepts (from ION) that we have used in INVITE.

3.5.1 ION Framework concepts and abstraction

In this section we will try to be as simple as possible.

ION Framework has lots of concepts, but the most important ones for our game are those related with *Locales*, and *Effectors*. In ION Framework a *Locale* is a physical space, ION has a default place called '*World*. In INVITE we have created more *Locales*. We created inside the *World* a Locale called *Island*, and inside the Island we have created *CampSite* Locales (one for each team) and *ResourceSite* Locales, here again, one for each team.

ION Framework has another important concept, *Effector*. In INVITE every action we do is an *Effector*. We have *MultipleStepEffector* and *SingleStepEffector*. A *MultipleStepEffector* is something that persists in the time, like walking from site "A" to site "B", it takes some seconds. For walking we have an Effector called *Mover* and it receives an ION *Action* called MoveTo. We have also other *MultiStepEffector* called *MiniGamePlayer*, this is used by our agents, to play a MiniGame with a given delay. That *MiniGamePlayer* calls other two *SingleStepEffector* called *GoldMiner* and *WoodCutter* and both receive actions with the values of gold/wood.

We have also much more *Actions* and *Effectors*, and for more details you should read the INVITE Project page [2].

4. SOCIAL MODEL AND AGENT IMPLEMENTATION

In the previous chapter we explain our game. In this chapter describe the model we have implemented to overcome the problem our work tries to solve:

> "How can autonomous agents be believable with a sense of social identity?"

Each element in the model will be described and representative examples will also be given to illustrate their mechanism.

4.1 Agent Model

As many agents based models our model has sensors and effectors as described in figure 2.

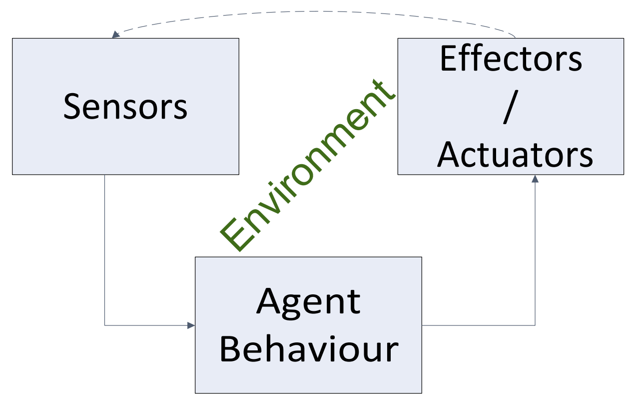


Figure 2: Sensors and Effectors

The sensors feel the world modifications and the actuators/effectors make changes in the world. Every single action that the agent does with its effectors will be felt in his sensors. That is also a simple way to check if the agent modifications have some effect in the world where the agent is.

At this point we have sensors and actuators, but we need something to process the inputs, felt by the sensors, and make a decision in what action should be done by the effectors. For now we will explain our model in a very simple way and we will explain how each sub-module relates to each other and what each module specifically does.

Our model is presented in figure 3.

As promised, in the next sections we will explain each module and how they relate to each other.

4.1.1 Personal Identity / Social Identity

The social identity theory says that, when we are part of a group, we discover the good values and all the good things about the group we belong to, but we try to find negative aspects in the groups we do not belong to, sometimes only to enhance our own self-esteem [3][7].

People do not act only as individuals but also as group members, sharing the same perceptions, goals and identity, the social identity [8][9]. It confers a shared or collective representation of who one is and involves self-categorization (cognitive), self-esteem (evaluative), and commitment (psychological) components [7].

As persons we can have multiple social identities depending on the context or group. We can say that salience of an entity is a product of accessibility and fit [12]. We studied this in section **??** where we explained what salience, fit and accessibility are with some examples.

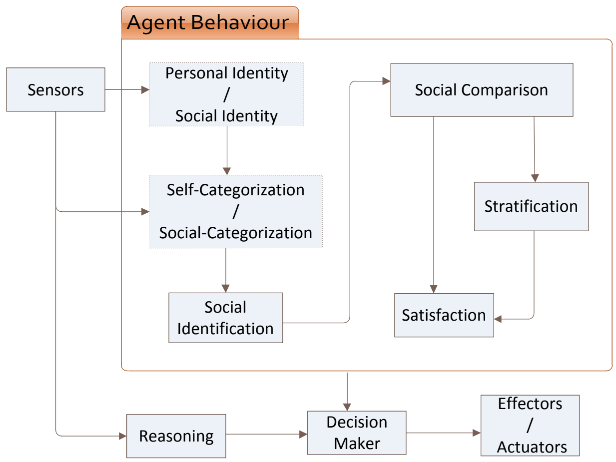


Figure 3: Our Agent Model Proposal

After this brief introduction we should say that this two modules are not part of this thesis work. We just use parameters to simulate this modules. This two modules are part of Joana Dimas' PhD work and when that work is done, we can "easily" merge the two solutions. That's why we included those two modules in our architecture. In this thesis those two modules are simulated by a parameter that describes the agent cooperation. That parameter has a range from zero to two, $[0, 2]$. If we used the value zero our agent only play gold, but if we chose 2 it only plays wood. This parameter acts like a multiplier in our solution, after all the other modules calculates his values we multiply this parameter with the other parameters result and then we obtain our agent output, before the reasoning module, but we will talk about our reasoning module later in this chapter. Just to clarify, if we want to play with our agent in "neutral" mode we should use the value one (1) for this parameter.

4.1.2 *Self-Categorization*

As we studied in our related work Self-Categorization causes people to think of themselves less as unique individuals and more as relatively typical members of a group, and they act accordingly[10]. They see themselves as having the characteristics associated with group memberships, and they act as they believe group members should act [10].

We categorize people in general, including ourselves, in order to understand the social environment we are in. We do this everyday because if we put people in categories we know what type of person he or she is. We try to do the same exercise in ourselves and observe in what categories we fit in, to really understand ourselves, and what type of person we are. This as the last two modules is part of building a personality and is not part of this thesis. This thesis only uses existing personalities and play games based on that personalities. In this module our agent

4.1.3 *Social Identification*

This is where our thesis really starts, we have an entity of a person formed, so our agents can use it and play a complete game. The first module of our chain of modules is Social Identification.

As we stated in section 4.1.2 we categorize ourselves in groups, and this is what social identification is all about. Sometimes we adopt attitudes (and identity) based on the groups we think we belong to. When our identification with a particular group is salient we start to talk, to act, and to do a lot of things like the group [11].

In our implementation to the INVITE Framework we take basic approach to this issue. As we studied, social identification is where we identifies (or not) with our group. In this particular game the most evident aspects of comparison are the wood and the gold. Because we don't have access to our team mates gold we should compare our wood with the average of wood gathered by our team mates.

So, based on the last paragraph our agent in this module look at his contribution to the team wood and sees how much wood (in average by player) the rest of the team have gathered. Obviously if this module is just that, when we put more than one agent playing against each other in the end they are all playing the same amount of wood. We don't want that, so to solve that issue we gave different percentages depending in how much days have passed.

Just to clarify, let's imagine that we are at the end of the second day, we are in a team of three players and our team have a total of 30 units of wood. For this example our team have gathered 15 units of wood in the first day and 15 in the second day. Let's imagine that we played in the first day 7 units of wood and in the second day we have played 1 unit of wood, for example. In this scenario most of us will decide what to play in the third day looking at the day number two with a higher importance, and probably we will play more wood than in the second day. To do that type of human behaviour in this module our agent does the following solution. Our agent does $7 + 1 = 8$ which is the amount of collaboration that we have in the team. Our agent does another math it see that to be in the average he should have played $(15 + 15)/2 = 15$. That our agents sees that in the last round he has played just 1 unit of gold and is team mates have an average of 7.5, $(15 - 1)/2 = 7,5$. Then our agent converts everything for percentage ($results/hoursintheday$) and multiplies by $0,7$ the percentage of the last day and by $0,3$ the percentage of the all time. That result of that math will be a result between zero and one, which will be multiplied by the total of hours in a day and passed to the next module. The output of our modules are always the value of wood they will play if "they are alone" in the conceptual module, without any other modules.

4.1.4 *Social Comparison, Stratification and Satisfaction (Self-Esteem)*

Social comparison happens when we have categorized ourselves as part of a group and have identified with that group, then we tend to compare that group with other groups. If our self-esteem is to be maintained our group need to compare itself with other groups. This is critical to understand the prejudice, because once two groups identify themselves as rivals they are forced to compete in order for the members to maintain their self-esteem. Competition and hostility between groups is thus not only a matter of competing for resources like jobs, but also the result of competing identities [8].

Status is the outcome of intergroup comparison, it reflects a group's relative position on some evaluative dimensions [11], this is when we see our group as a superior group or inferior group, that is also called social stratification.

If we are in a low-status group (inferior group), and we compare it with another out-group (not the high-status out-group), the relevant inferiority should decrease in salience and self-esteem should recover [11]. Competition between subordinate groups is sometimes more intense than between subordinate and dominant groups [11].

The Tajfel experiments showed that individuals achieve positive self-esteem by positively differentiating their in-group from a comparison out-group on some valued dimension [8][5][9].

In our agents we take into account the amount of wood we have and compare it with the other groups. With that comparison we obtain the position of our group in the game. If we are in the first position we are more satisfied with our group and by the theory we should cooperate more (play more wood).

Let's start by the easiest module to understand, the Stratification module. In this module our agent just take into account all teams "raft completion" and calculates by how much our team is in the lead or behind. This value will be the simplest to calculate. Imagine that our team has an advantage of 25 units of wood and the the amount of wood necessary to complete the raft is 255 for example, this module will produce an output of $25/255 = 0,098$. This value will be used later to calculate the difference to the other teams and if we are losing by an higher difference our agents will tend to play more and more wood. If our team is winning and the advantage is huge our agents will tend to play less and less wood. Obviously if an agent or player play less wood he/she will play more gold. Just to clarify, if we are losing the game by 25 units of wood the result will be $-25/255 = -0,098$. In this module we don't take into account the amount of wood needed to complete the raft, that factor will be calculated in our agent's Social Comparison module and also in the Reasoning Module.

Our Social Comparison module has almost the same maths of our agent's social identification module, but this time our agent compare his/her team with the other teams. Our agent's Social Comparison module only sees if his/her team is in the leading. If it is leading the game the math will be $value = (0.5 - advantage) * cooperationVariable$, if the agent's team is in the second position (of lower) the math will be $value = (1.2 + advantage) * cooperationVariable$. The advantage variable is what we have calculated in the stratification Module and cooperationVariable is what we have defined as our parameters (from Joana work).

Our Satisfaction module will only gather the values calculated in our social comparison module (that uses the Stratification module) and our Social Identification module and multiply that factors (percentages from 0 to 100%) by the amount of hours in a day and obtain the value to be played by our agent, but this could not be the final output of our agent, because the Reasoning module can change the values.

4.1.5 Reasoning

This module as stated in the previous sections has his opportunity to take into action only after all the other modules have done its work. This module has a complicated task, but we will try to explain it in a simple way. This module take into account the values calculated in the stratification module (not the final value in percentage, but the real values, for example, 175 units of wood "vs" 200 units of wood). This module will see if our team can end the game in this day, and if it's possible this module will calculate if it is possible to end the day with just one "player" and if its the case, this module will pass an information to the decision maker to ignore all the other information and use the Reasoning information.

4.1.6 Decision Maker

Taking in count all the previous modules the agent will decide what is his next decision choosing between the best possible solution (Reasoning) and the agent behaviour decision, depending on his social feelings. The Decision Maker module will be like a police officer controlling the traffic in a road. In other words, the decision maker will choose between the agent behaviour or reasoning. As a summary we can say that our decision maker will only choose reasoning when our reasoning module passes the information that our team can end the game in this day. But again, the cooperation variable that we have described some sections before, has an important role, because the result given by the Reasoning module will be multiplied by that factor and the result of the other modules will be multiplied by $1 - cooperationVariable$.

4.2 Summary

Our agent is an extension of ION Framework, more accurately from IPlayer presented in INVITE Framework. We have implemented our agent on our related work. We have two points that we want to explain. The first one is the cooperation parameter and the second one is how our social comparison module works in more detail.

For the cooperation parameter it is only a parameter between 0 and 2 that will make the agent be more or less cooperative with his/her team mates. When all the calculations are done we multiply the value obtained by the cooperation parameter, if the parameter is 0, the agent will always play 0 wood and 12 gold. If the parameter is 1 then the output is the "neutral" values. If we put the parameter to 2, that doesn't mean that the agent will play always 12 wood and 0 gold. That is not true!

The person who is controlling the agents can always increase/decrease this value. Let's now talk about some values to have a clear idea what this parameter does.

We hope that what the cooperation parameter does is now clear. The second point that we want to explain is the social comparison module. Just to simplify the explanation we calculate in what position our team is, and if we are in the lead we tend to be happier with our team but at the same time something tells us to play more gold than wood. To overcame this, in the social comparison module we included a parameter that varies with our leading distance to the second team. If our distance is huge we will tend to play more and more gold, but if the difference is smaller or we are losing the game we tend to play more and more wood. Obviously this will be balanced with the cooperation parameter and hopefully we can have the same behaviour with our agents as the players that we observed.

5. COMPLETE SCENARIO FOR TESTS

Now that we have described all the aspects of the game, the agents' architecture, the application and the system that integrates these two aspects, we can present an example of a complete scenario of the working system.

5.1 Standard Scenario Description

In the game we can have a lot of configurable parameters as we have explained, but most of them were the same between our different two tests. The Following parameters were calculated by our team based on some user tests. Obviously that these values can change with different sets of users. These values are the same between the two scenarios.

- PlayerTimePerDay = 12; This is the number of hours that one player has to spend in a single day.
- WoodPerTimeUnit = 4; The player can achieve 4 units of wood, for each hour.
- GoldPerTimeUnit = 2; The player can achieve 2 units of gold, for each hour.
- WoodRequiredForRaftCompletion = 255; Number of wood units required to build the raft.
- NumberDaysUntilEruption = 8; Number of days before the volcano explodes the whole island.
- DistributionRule = Egalitarian; This is the distribution rule. When a team ends a raft, the amount of wood will be divided equally by all team elements. Imagine a team of three elements: when the team ends the raft, all the three players in their final score will have 85 units of wood.
- NumberOfPlayersByTeam is two (2).
- NumberOfTeams is two (2).

In the next section we will only explain what is different from the standard scenario.

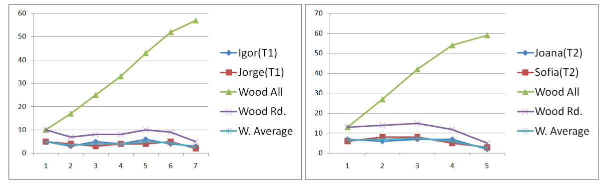
5.2 First Scenario Description

In this scenario the users just have the result of the other team in the end of the game. For the winning team the value of the raft will be multiplied by five (5). The amount of wood to complete the raft is two hundred and twenty five (225). With this kind of scenario we want to see how the Human persons react only with what we called social identification, in another words, we want to measure how the players react to the fact of only knowing that they have x days for the end of the game and his/her team mates are playing y units of wood. In this scenario the players have anything to measure their social comparison, and stratification. For us this scenario is really useful, because the players are just using social identification. Again, we should refer that the only feedback that the players had during the game was his/her team mate wood. They obviously know after one game ends, their score and the other team score, so that worked as a priming message for the next game.

After analysing all the three values in the graphics we can see that our agents have really nice results. We should refer that our best results are obtained with our value of cooperation in 0,7.

In Igor all the three approaches have good results but we can see that the Igor(0,7) has the closer line to Igor(T1), so this one is the one we choose to represent Igor.

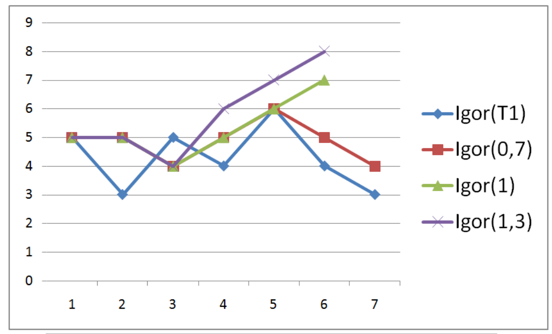
For example with our agent results for Jorge we should prefer the Jorge(1) rather than Jorge(0.7). Even if Jorge(0.7) has closer results to the Jorge(real player) the curve is more precise in Jorge(1) and Jorge(1.3), even with higher values in module.



Igor(T1)	5	3	5	4	6	4	3	
Jorge(T1)	5	4	3	4	4	5	2	
Wood All	10	17	25	33	43	52	57	
Wood Rd.	10	7	8	8	10	9	5	57
W. Average	5	3,5	4	4	5	4,5	2,5	4,1

Joana(T2)	7	6	7	7	2	
Sofia(T2)	6	8	8	5	3	
Wood All	13	27	42	54	59	
Wood Rd.	13	14	15	12	5	59
W. Average	6,5	7	7,5	6	2,5	5,9

Figure 4: The game we will try to simulate



Igor(T1)	5	3	5	4	6	4	3
Igor(0,7)	5	5	4	5	6	5	4
Igor(1)	5	5	4	5	6	7	
Igor(1,3)	5	5	4	6	7	8	

Figure 5: Igor with three possible values of cooperation

For Joana we should choose Joana(1) because she has the most similar line to Joana(T2). It is almost the same explanation we gave to choose Jorge(1) and Jorge(1,3) instead of Jorge(0,7).

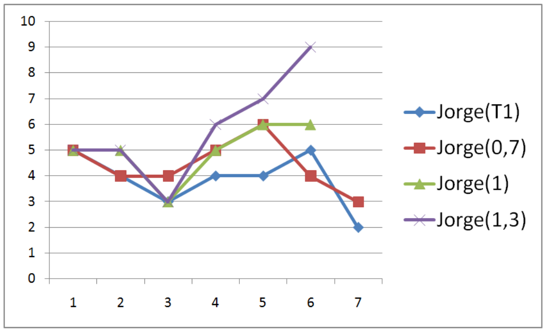
With the same logic in Sofia (T2) we should choose Sofia(0,7) because she has a line almost identical to Sofia(T2), in values and also in the line (ups and downs).

5.3 Results - Scenario 2

In this scenario the users will have the results of the other teams after each round. For the winning team the value of the raft will be multiplied by two (2). All players are Human Players. The amount of wood to complete the raft is two hundred and fifty five (255). The players have social identification between each team mates and social comparison between different teams.

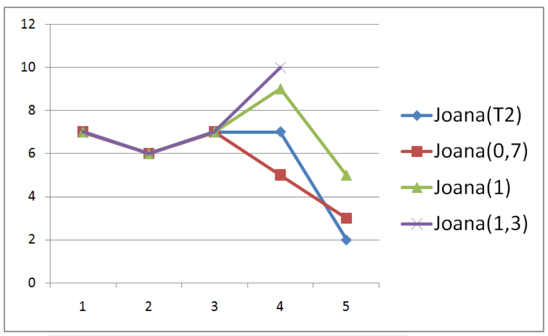
After analysing the second scenario results, with the three values, we concluded that we have a pretty nice model, at least for the type of tests we did. As we said in the last chapter we could not simulate the last two days of Marco and Ruben when they played 0, 0, even if we decrease the cooperative value, because our module doesn't cover that type of actions. In the next few paragraphs we will study this results in detail and explain why we should choose a "line" instead of another one.

For the first agent (Joao) until the fifth day all the lines are decreasing and increasing values, like Joao did in the real world, but after the fifth day the closest agent is Joao(1,3). Joao(1,3) is the highest cooperative agent in this test, so



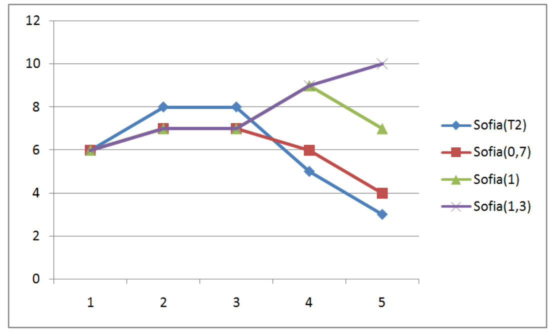
Jorge(T1)	5	4	3	4	4	5	2
Jorge(0,7)	5	4	4	5	6	4	3
Jorge(1)	5	5	3	5	6	6	
Jorge(1,3)	5	5	3	6	7	9	

Figure 6: Jorge with three possible values of cooperation



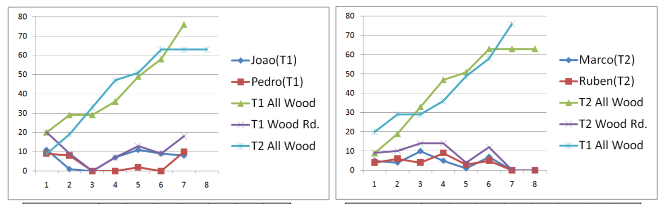
Joana(T2)	7	6	7	7	2
Joana(0,7)	7	6	7	5	3
Joana(1)	7	6	7	9	5
Joana(1,3)	7	6	7	10	

Figure 7: Joana with three possible values of cooperation



Sofia(T2)	6	8	8	5	3
Sofia(0,7)	6	7	7	6	4
Sofia(1)	6	7	7	9	7
Sofia(1,3)	6	7	7	9	10

Figure 8: Sofia with three possible values of cooperation



Joao(T1)	11	1	0	7	11	9	8
Pedro(T1)	9	8	0	0	2	0	10
T1 All Wood	20	29	29	36	49	58	76
T1 Wood Rd.	20	9	0	7	13	9	18

Marco(T2)	5	4	10	5	1	7	0	0
Ruben(T2)	4	6	4	9	3	5	0	0
T2 All Wood	9	19	33	47	51	63	63	63
T2 Wood Rd.	9	10	14	14	4	12	0	0

Figure 9: The base game.

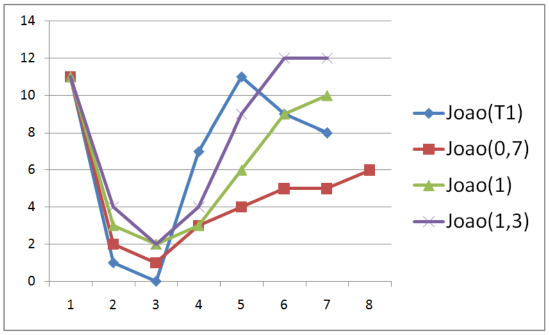
we can say that, based in our model, Joao is a cooperative person.

Now we will analyse Pedro's behaviour during its game and our agents. We can see that all of our agents have values of a really nice behaviour compared to Pedro(T1), but the closer line to Pedro is Pedro(0,7). Maybe if we decrease the cooperation variable a little bit we can be even closer to Pedro(T1), but with a closer cooperation variable the second day will have a lower result that it is right now with Pedro(0,7).

In our opinion the Marco(1,3) was quite impressive. It was incredibly closer to Marco(T2) results. The Line of Marco(T2), Marco(1) and Marco(1,3) are quite similar where they increase and decrease values, but Marco(1,3) has almost the same values of Marco(T2), the only "big" difference is in day 2 where Marco(T2) played 4 and Marco(1,3) played 6, but in the other values the difference between Marco(T2) and Marco(1,3) was never bigger than one unit, which is quite impressive.

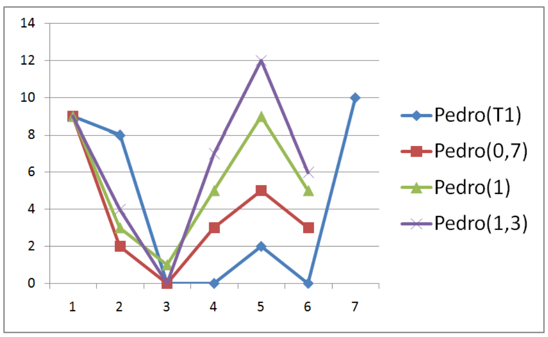
In Ruben agents we cannot simulate as accurately as we did with Marco, but we have not a bad set of results for an agent without particular cases in the code. To be honest the only really bad value that we have in Ruben is day four (4). So lets split this graphic and analyse it separately before and after day 4. Before day 4 we that Ruben(1,3) even if the values are very high the curve is the same as Ruben(T2). It increases between day one to day two, decreases from day two to the day three. In day four(4) our agents failed miserably, because all of them decreased their values and Ruben increased his values. After studying Ruben action we concluded that this is a particular decision that he made with logic, but he did it one day after he should have done it. After analysing the graphs we see that our agents played in the right place (where team 2 has little advantage against team 1) and when the advantage increased (day 4) our agents reduced their cooperation, but Ruben increased his collaboration in this exact moment. If we have added a particular rule to our code we would pass this test, but our goal is to have a generic agent that performs quite well with all human beings and not a particular agent that simulates a human with an efficiency of 100%, but does everything wrong with other humans, that is why we do not want to have particular rules in our model. After day four (4) we can see that our agents and Ruben (T2) take the same type of decisions and it is quite interesting that in this particular situation our Ruben(1,3) would lose the game by just one unit of gold to his team mate. Ruben(1) and Ruben(0,7) would lose the game individually, but his team will also lose the game against team 1.

6. CONCLUSIONS



Joao(T1)	11	1	0	7	11	9	8	
Joao(0,7)	11	2	1	3	4	5	5	6
Joao(1)	11	3	2	3	6	9	10	
Joao(1,3)	11	4	2	4	9	12	12	

Figure 10: Joao with three possible values of cooperation



Pedro(T1)	9	8	0	0	2	0	10
Pedro(0,7)	9	2	0	3	5	3	
Pedro(1)	9	3	1	5	9	5	
Pedro(1,3)	9	4	0	7	12	6	

Figure 11: Pedro with three possible values of cooperation

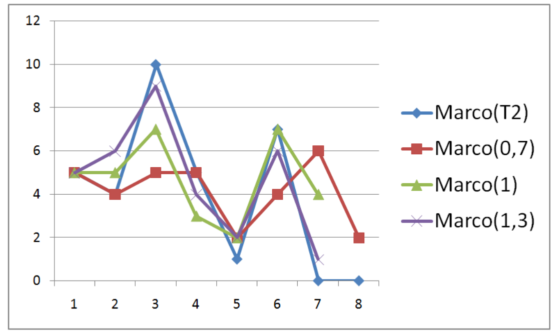
In this thesis we achieved our main goal, we have created believable synthetic characters to play our game. That was not an easy journey.

We started to design the first drafts in paper and imagining what the agents should do when programmed.

We have focused our energy and our attention in a really concrete case scenarios like the scenarios we have described in this thesis, even thought our agents work in almost every scenario.

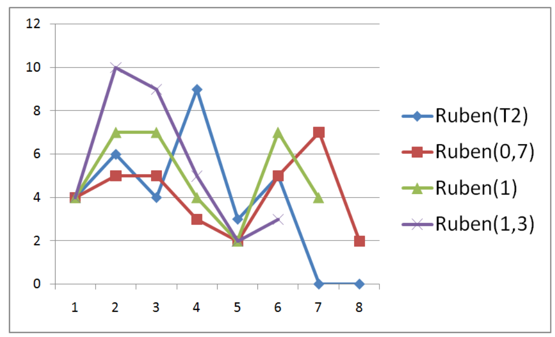
At the end we concluded that they are almost perfect for the type of tests that we have imagined, but they have some failures that could be considered in a future work. For example, if we put many of our agents playing again each others without any Human player after some days (40/50 days) they tend to start playing all the same values (if they have the same input parameter). That could be considered a problem by some people, but in our opinion it is not a big problem, because a normal game rarely has more than 10 days (in average), so we won't see any problem in that range of days.

We know that we have some faults in our agents that we will be describe in the future work section, but it is important to say that this is an area with an enormous effort from the industry and universities to understand the human behaviour and try to put that behaviour into "computers". We think that we have achieved all of that in about a year, which is an impressive result for us. We are very proud of ourselves. We started a really complicated job with our related work and ended this thesis with all the tests that we have made that proved our model.



Marco(T2)	5	4	10	5	1	7	0	0
Marco(0,7)	5	4	5	5	2	4	6	2
Marco(1)	5	5	7	3	2	7	4	
Marco(1,3)	5	6	9	4	2	6	1	

Figure 12: Marco with three possible values of cooperation



Ruben(T2)	4	6	4	9	3	5	0	0
Ruben(0,7)	4	5	5	3	2	5	7	2
Ruben(1)	4	7	7	4	2	7	4	
Ruben(1,3)	4	10	9	5	2	3		

Figure 13: Ruben with three possible values of cooperation

6.1 Future work

In a future work we can improve a lot our agents, in many different ways, but one of the most important areas to improve is add an emotional module like the figure 14 represents.

For that we could use Fatima a framework developed at GAIPS - Inesc - ID. In gaips some people have done the integration between ION and Fatima, so in a near future we could have our model with emotions from Fatima. That will be a great challenge, but we know that the results will overcome all the effort needed to build such a solution.

In a near future we can also prevent another scenarios in our module, because not all the games are as we described, some people have completely different strategies that we have not take into account, and that type of new scenarios will make our model even better. We hope that this work is the first step for many more works in the area in the upcoming years.

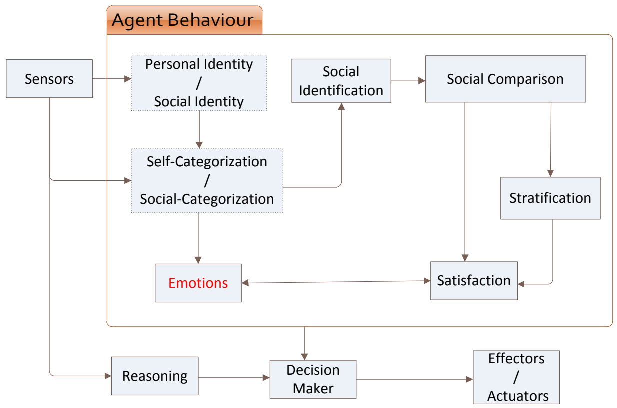


Figure 14: Our Agent Model Proposal

7. ACKNOWLEDGMENTS

I would like to thank everyone who helped me accomplish this work, so I am grateful to my family, my friends and mostly to my girlfriend, who helped me a lot during this process. She was always present in my good and bad moments to take care of me, helping me maintain my mental sanity and giving me the strength I was needing through this process. I am also really grateful to my advisor Professor Carlos Martinho and my co-advisor Professor Rui Prada, who gave me the opportunity to work in INVITE project.

8. REFERENCES

[1] A. Soares, "A multi-dimensional relationship model," Master's thesis, Instituto Superior TÃl'cnico de Lisboa, 2010.

[2] R. Prada, C. Martinho, P. Santos, J. F. PeÃśa, L. L. Ribeiro, J. Dimas, G. Raimundo, M. Baptista, and A. Barata, "http://project-invite.eu/."

[3] J. Dimas, "Social identity - aamas 2012," 2011.

[4] R. S. Stephen Reicher and S. A. Haslam, "The social identity approach in social psychology."

[5] H. Tajfel, "Social identity and intergroup behaviour," Social Science Information. SAGE, April 1974.

[6] E. Smith and D. Mackie, *Social Psychology*. Psychology Press, 2000. [Online]. Available: http://books.google.pt/books?id=ufkfBSyoRl4C

[7] K. Wang and C.-F. Tai, "The influence of social presence on continual participation in online communities: The relational view based on social identity theory." IEE Computer Society, 2011, pp. 110–115.

[8] J. Dimas, "Social identity - psychological approach," january 2011.

[9] M. Shen-Jiang, "Congruence of business ethics in supplier-buyer relationships: A social identity approach." IEE Computer Society, 2009.

[10] D. M. Mackie, E. R. Smith, and D. G. Ray, "Intergroup emotions and intergroup relations," *Journal Compilation - Blackwell Publishing Ltd*, 2008.

[11] H. Tajfel and J. C. Turner, *An integrative theory of intergroup conflict*. Brooks/Cole, 1979, pp. 33–47. [Online]. Available: http://books.google.com/books?hl=en&lr=&id=Q3ClkIkk 4C&oi=fnd&pg=PA56&dq=An+Integrative+Theory-

[12] J. E. Stets and P. J. Burke, *Identity Theory and Social Identity Theory*. Washington State University, 2000, pp. 224–237.

[13] U. Technologies, "http://unity3d.com/unity/."

[14] M. Vala, G. Raimundo, P. Sequeira, P. Cuba, R. Prada, C. Martinho, and A. Paiva, "Ion framework - a simulation environment for worlds with virtual agents."

[15] E. Gamma, R. Helm, R. Johnson, and J. Vlissides, "Design patterns: elements of reusable object-oriented software," 1995.