



HAL
open science

Enhancing the Behavior of Agents in Social Simulations with Emotions and Social Relations

Mathieu Bourgais, Patrick Taillandier, Laurent Vercouter

► **To cite this version:**

Mathieu Bourgais, Patrick Taillandier, Laurent Vercouter. Enhancing the Behavior of Agents in Social Simulations with Emotions and Social Relations. The 18th Workshop on Multi-agent-based Simulation - MABS 2017, May 2017, São Paulo, Brazil. pp.16, 10.1007/978-3-319-91587-6_7. hal-01573384

HAL Id: hal-01573384

<https://hal.science/hal-01573384>

Submitted on 9 Aug 2017

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Enhancing the Behavior of Agents in Social Simulations with Emotions and Social Relations

Mathieu Bourgeois¹, Patrick Taillandier² and Laurent Vercoeur¹

¹ Normandie Univ, INSA Rouen, UNIHAVRE, UNIROUEN, LITIS, 76000 Rouen, France

² MIAT, INRA, 31000 Toulouse, France

Abstract. Social Simulations need agents with a realistic behavior to be used as a scientific tool by social scientists. When simulating a human society, a realistic behavior implies the use of cognition, social relations between people but also to take into account emotions and the dynamic between these features. However, developing such a behavior is often too complex for people with little knowledge in programming. In this paper, we present a formalism to represent cognition, social relations and emotions, which is integrated in an agent architecture to give a dynamic emotional behavior to social agents. This architecture is implemented in the open-source multi-agent platform GAMA. A use case about evacuation during bush fires in Australia is used to show the possibilities of our work.

Keywords: Social Simulation, Emotions, Cognition, Agent Architecture

1 Introduction

Multi-agent simulation has become an important tool, especially in social sciences where it is used to study complex systems composed of hundreds or thousands of simulated humans. In this particular case, we now speak of social simulation [17] which means simulation featuring social agents.

A social agent can be described as a simulated human. In other words, a social agent is an agent with a cognitive behavior able to interact with its environment and with other agents through a social behavior. This social behavior can feature an emotional engine, a personality, an engine to cope with social norms or social relations.

In order to increase the accuracy of social simulations, these ones have to be as close as possible to real cases studied. This goal for realism leads to the use of believable social agents, i.e. agents with more and more of social features [43].

Let's take the example of bush fires in Australia studied with multi-agent simulation [2]. In this study, the goal was to simulate the evacuation of an area in the context of bush fires in Australia. The authors used a BDI architecture [11] for the modeling of their agent's cognition but in this situation, a person also reacts to his/her emotions and his/her social relations.

In this paper, we tackle this issue by adding dynamic emotions and social relations into an existing cognitive architecture. The main goal is to provide a formalism for the creation of emotions through cognition and for the evolution of social relations according to an agent’s mental state and to integrate it in an agent architecture. However, modeling and simulating such behavior requires high level skills in computer programming and in artificial intelligence, so it is out of reach of most modelers coming from social sciences. Our work was implemented in the open-source simulation platform GAMA [20]. The main benefit of integrating our agent architecture inside this platform is to take advantages of the modeling language offered by GAMA that eases its use for modelers who are not expert in programming.

This paper is structured as follows: in Section 2, we show existing works to create social agents with a cognitive behavior, an emotional engine or social relations. In Section 3, we propose a formalism used to deal with the mental state of the agent in terms of cognition, emotion and social relations. In Section 4, we describe the integration of this formalism into a cognitive agent architecture. In Section 5, we present an example to illustrate how our social architecture can be used on a model of evacuation in the context of bush fires in Australia. Finally, Section 6 serves as a conclusion.

2 Related Works

Creating believable agents is a key point in social simulations. In this section, we present various works dealing with the integration of cognition, emotions or social relations in agents to improve the realism of social simulations.

2.1 Cognition in social agents

Adding cognition is a first step in order to increase the realism of social simulations [3] [5]. To give a cognitive behavior to agents, some cognitive architectures have been proposed such as SOAR [26], ACT-R [13] or BDI [11], which is the most adapted to simulation context [3]. The BDI paradigm uses modal logic [14] to define the concepts of beliefs, desires and intentions that compose the mental state of the agent. It then provides logical links between these concepts and a collection of action plans to give a cognitive behavior to agents.

To ease its use, the BDI architecture has been implemented in different frameworks. A classic one is the Procedural Reasoning Systems (PRS) [28] which is based on three steps: firstly, a perception of the environment to update the belief base, then a deliberation between desires and the state of the world and finally the selection of an action to execute. PRS serves as a base for many other frameworks such as JACK [21], JADE [8] or Jadex [33].

Some researchers have tried to integrate the BDI architecture in modeling and simulation platforms. An extension to NetLogo [47] implements a simplified BDI architecture for educational purposes [38]. Sing and Padgham [40] decided to connect a multi-agent platform with an existing BDI framework (JACK or

Jadex for example) and, in the same spirit, an application connecting the Matsim platform [6] with the GORITE BDI framework [36] has been proposed.

2.2 Emotional architectures

Various works have shown that adding emotions to agents increases the believability of their behavior [7] [29]. This improvement of credibility is useful in social simulations as the main goal of these simulations is to be as realistic as possible.

In psychology, there is no consensus about a unique emotional theory. The most used theory in AI is the cognitive appraisal theory of emotion [4] [41] and more particularly the OCC theory [32] that is specifically developed to integrate emotions in artificial intelligence.

The OCC model of emotions defines twenty-two emotions distributed in eleven pairs according to the cognitive appraisal of a situation by an agent. This cognitive appraisal is made according to three aspects: the consequences of events, the actions of other agents and the aspect of objects.

Different implementations of emotional systems for multi-agent simulations have been proposed. For example, DETT (Disposition, Emotion, Trigger, Tendency) [46] considers the perception of a situation as the triggering condition to the creation of emotions based on the OCC model. Gratch and Marsella proposed a different approach with their EMA model [19] that not only creates emotions based on the appraisal of a situation thanks to appraisal variables, but which also study the coping behavior created by emotions. Finally, eBDI [23] proposes to integrate directly the OCC model into a BDI architecture.

2.3 Social relation in multi-agent simulations

As people create social relations when living with other people, it seems logical to model social relations between agents that simulate humans. In [37], the agent's behavior is computed with a social-psychological model featuring personality, emotions and attitude of the agent against its environment. Gratch [18] adds a social level to modify the behavior of an agent according to the social state of the world.

In those models of social agents, social relations are represented with a finite number of variables, each one of them defining a precise dimension of the relation. As it is pointed out by Ochs [30], there is actually no consensus on the type and number of variables required to correctly model a social relation but four variables seem to be used more frequently than other ones. These variables correspond to the ones present in the dimensional model of interpersonal relation of Svennevig [44]:

- The degree of **liking** for another agent [22] [34].
- The degree of **dominance** one agent has over another one [37] [34]. It represents the degree of power an agent thinks having over another one.

- The degree of **solidarity** also known as the social distance [9] [12]. It indicates the similarity in terms of desires, beliefs and values between two agents.
- The degree of **familiarity** that characterizes the number and the type (private/public) of information that can be transmitted to another agent [9].

In works cited above, social relations are studied as behavioral change factors, so they are only used as static variables. But obviously, a social relation between two people can evolve in time. To tackle this issue, Ochs [30] has proposed an agent architecture incorporating personality and using emotions to add dynamism to social relations to non playable characters of video games.

2.4 Synthesis

These three notions, cognition, emotions and social relations, have never been combined in simulation to provide a realistic behavior to social agents. Moreover, each of them has never been implemented to be easy to use for social scientists who do not have high level skills in computer science.

This paper aims at proposing an architecture for social simulation that deals with cognition, emotions and social relations and that can be used by modelers with low level skills in programming. To do so, we use the principles of the multi-agent simulation platform GAMA [20] that has proved its ease of use [27][35] thanks to its modeling language GAML that we extended to use our architecture.

3 Creating emotions and social relations with cognition

The main contribution of this article consists in defining a formalism to represent and to articulate mental states of a social agent. This mental state is composed of a cognitive state, an emotional state and social relations with other agents.

3.1 Representing mental states of agents

Representing the cognition with predicates The cognitive part of our architecture is based on the BDI paradigm [11], in which agents have a belief base, a desire base and an intention base to store the cognitive states about the world. We also use a base for uncertain beliefs that are expected, called the uncertainty base, and used to create emotions about expected facts.

To represent this knowledge, we use predicates. A predicate unifies the representation of the information about the world so it can represent a situation, an event or an action. As the goal of this work is to create emotions from cognition about events and values of actions from other agents, we represent an information P caused by an agent j with a praiseworthiness value of pr by $\mathbf{P}_{j,pr}$. The praiseworthiness value can be positive (in this case, the information P is praiseworthy) or negative (in this case, the information P is blameworthy). A

predicate \mathbf{P}_j represents an information caused by an agent j with any praiseworthiness value and a predicate \mathbf{P} represents an information caused by any agent with any praiseworthiness value. We represent the opposite of a predicate P by **not P**.

Depending on the storage base, a predicate can be considered as a belief, an uncertain belief or a desire and it is represented as follows:

- **Belief_{*i*}(P)**: indicates that the predicate P belongs to the belief base of the agent i .
- **Expect_{*i*}(P)**: indicates that the predicate P belongs to the uncertainty base of the agent i .
- **Desire_{*i*}(P)**: indicates that the predicate P belongs to the desire base of the agent i .

Formal representation of emotions For the definition of emotion, we base our work on the OCC theory of emotions [32]. According to this theory, an emotion is a valued answer to the appraisal of a situation. As we use emotions to update dynamically social relations, our definition of an emotion also needs to contain the agent causing the emotion. With this definition, we represent an emotion by $\mathbf{E}_i(\mathbf{P}, \mathbf{A}, \mathbf{I}, \mathbf{D})$ with the following elements :

- **E_{*i*}**: the name of the emotion felt by agent i .
- **P**: the predicate that represents the fact about which the emotion is expressed.
- **A**: the agent causing the emotion.
- **I**: the intensity of the emotion.
- **D**: the decay of the emotion’s intensity.

For example, if an agent Alice feels fear about an action P caused by agent Bob with an intensity of 4.5 and a decay value of 0.6, this will be represented by the emotion $\text{Fear}_{Alice}(\mathbf{P}_{Bob}, \text{Bob}, 4.5, 0.6)$. An emotion with no specific intensity nor decay is represented by $\mathbf{E}_i(\mathbf{P}, \mathbf{A})$ and an emotion with no specific agent causing it is represented by $\mathbf{E}_i(\mathbf{P})$.

Formalization of social relations Based on the work of Svennevig [44] exposed in Section 2, we define a social link with another agent as a tuple $\langle \text{agent}, \text{liking}, \text{dominance}, \text{solidarity}, \text{familiarity} \rangle$ with the following elements:

- **Agent**: the agent concerned by the link, identified by its name.
- **Liking**: a real value between -1 and 1 representing the degree of liking with the agent concerned by the link. A value of -1 indicates that the concerned agent is hated, a value of 1 indicates that the concerned agent is liked.
- **Dominance**: a real value between -1 and 1 representing the degree of power exerted on the agent concerned by the link. A value of -1 indicates that the concerned agent is dominating, a value of 1 indicates that the concerned agent is dominated.

- **Solidarity**: a real value between 0 and 1 representing the degree of solidarity with the agent concerned by the link. A value of 0 indicates no solidarity with the concerned agent, a value of 1 indicates a complete solidarity with the concerned agent.
- **Familiarity**: a real value between 0 and 1 representing the degree of familiarity with the agent concerned by the link. A value of 0 indicates no familiarity with the concerned agent, a value of 1 indicates a complete familiarity with the concerned agent.

With this definition, a social relation is not necessarily symmetric. For example, let's take two agents, Alice and Bob, with a social link towards each other. The agent Bob may have a social link $\langle \text{Alice}, 1, -0.5, 0.6, 0.8 \rangle$ (Bob likes Alice with a value of 1, he thinks he is dominated by Alice, he is solidary with Alice with a value of 0.6 and is familiar with Alice with a value of 0.8) and Alice may have a social link $\langle \text{Bob}, -0.2, 0.2, 0.4, 0.5 \rangle$ (Alice dislikes Bob with a value of 0.2, she thinks she is dominating Bob, she is solidary with Bob with a value of 0.4 and she is familiar with Bob with a value of 0.5).

3.2 Creating dynamics in emotions and social relations

Dynamic creation of emotions We base the automatic creation of emotions according to the mental states of the agent on the OCC model [32] and its logical formalism [1], which has been proposed to integrate the OCC model in a BDI architecture.

According to the OCC theory, emotions can be split into three groups: emotions linked to events, emotions linked to people and actions performed by people and emotions linked to objects. In this work, as we focus on relations between social agents, we only work on the first two groups of emotions (emotions linked to events and people), so we do not deal with emotions related to objects.

The twenty emotions defined in this paper can be divided into three parts: eight emotions related to events, four emotions related to other agents and eight emotions related to actions. These twenty emotions are defined according to the logical formalism previously done [1]. Moreover, all the emotions are created without intensity nor decay value as there is no generic way to define them.

The eight emotions related to events have the following definition:

- **Joy** $_i(\mathbf{P}_j, \mathbf{j}) = \text{Belief}_i(\mathbf{P}_j) \ \& \ \text{Desire}_i(\mathbf{P})$
- **Sadness** $_i(\mathbf{P}_j, \mathbf{j}) = \text{Belief}_i(\mathbf{P}_j) \ \& \ \text{Desire}_i(\text{not } \mathbf{P})$
- **Hope** $_i(\mathbf{P}_j, \mathbf{j}) = \text{Expect}_i(\mathbf{P}_j) \ \& \ \text{Desire}_i(\mathbf{P})$
- **Fear** $_i(\mathbf{P}_j, \mathbf{j}) = \text{Expect}_i(\mathbf{P}_j) \ \& \ \text{Desire}_i(\text{not } \mathbf{P})$
- **Satisfaction** $_i(\mathbf{P}_j, \mathbf{j}) = \text{Hope}_i(\mathbf{P}_j, \mathbf{j}) \ \& \ \text{Belief}_i(\mathbf{P}_j)$
- **Disappointment** $_i(\mathbf{P}_j, \mathbf{j}) = \text{Hope}_i(\mathbf{P}_j, \mathbf{j}) \ \& \ \text{Belief}_i(\text{not } \mathbf{P}_j)$
- **Relief** $_i(\mathbf{P}_j, \mathbf{j}) = \text{Fear}_i(\mathbf{P}_j, \mathbf{j}) \ \& \ \text{Belief}_i(\text{not } \mathbf{P}_j)$
- **Fear confirmed** $_i(\mathbf{P}_j, \mathbf{j}) = \text{Fear}_i(\mathbf{P}_j, \mathbf{j}) \ \& \ \text{Belief}_i(\mathbf{P}_j)$

On top of that, according to the logical formalism [1], four rules can be defined:

- The creation of **fear confirmed** or the creation of **relief** will replace the emotion of **fear**.
- The creation of **satisfaction** or the creation of **disappointment** will replace a **hope** emotion.
- The creation of **satisfaction** or **relief** leads to the creation of **joy**.
- The creation of **disappointment** or **fear confirmed** leads to the creation of **sadness**.

The four emotions linked to other agents have the following definition:

- **Happy for_i(P,j)** = i likes j & Joy_j(P)
- **Sorry for_i(P,j)** = i likes j & Sadness_j(P)
- **Resentment_i(P,j)** = i hates j & Joy_j(P)
- **Gloating_i(P,j)** = i hates j & Sadness_j(P)

The terms “ i likes j ” and “ i hates j ” have the following definitions:

- **i likes j** : agent i has a social relation with agent j with a positive liking value.
- **i hates j** : agent i has a social relation with agent j with a negative liking value.

Finally, the eight emotions linked to actions performed by agents have the following definition:

- **Pride_i(P,i)** = Belief_i(P_i) & P_i praiseworthy
- **Shame_i(P,i)** = Belief_i(P_i) & P_i blameworthy
- **Admiration_i(P_j,j)** = Belief_i(P_j) & P_j praiseworthy
- **Reproach_i(P_j,j)** = Belief_i(P_j) & P_j blameworthy
- **Gratification_i(P,i)** = Pride_i(P_i,i) & Joy_i(P_i)
- **Remorse_i(P,i)** = Shame_i(P_i,i) & Sadness_i(P_i)
- **Gratitude_i(P_j,j)** = Admiration_i(P_j,j) & Joy_i(P_j)
- **Anger_i(P_j,j)** = Reproach_i(P_j,j) & Sadness_i(P_i)

The terms “praiseworthy” and “blameworthy” have the following definitions:

- **praiseworthy**: indicates that the fact P has a positive praiseworthiness value.
- **blameworthy**: indicates that the fact P has a negative praiseworthiness value.

Updating automatically social relations As explained in Section 2, some works have shown that social relations are meant to be dynamic. Based on the previous work of Ochs [30], we integrate in our architecture a social engine that updates the social links of an agent according to its cognitive and emotive states.

In the following, we study the update of the social link $\langle j, \text{Liking}, \text{Dominance}, \text{Solidarity}, \text{Familiarity} \rangle$ possessed by agent i . Each variable of this social link evolves according to its own rule.

- **Liking**: according to Ortony [31], the degree of liking between two agents depends on the valence (positive or negative) of the emotions induced by the corresponding agent. In our model, *joy* and *hope* are considered as positive emotions (*satisfaction* and *relief* automatically raise *joy* in our engine) while *sadness* and *fear* are considered as negative emotions (*fear confirmed* and *disappointment* automatically raise *sadness* in our engine). This evolution is made by a fixed level α for each positive or negative emotion involved. Thus, the evolution is not related to the intensity of the concerned emotions as our engine creates emotions without intensities.

Moreover, some works have shown that the degree of liking is influenced by the solidarity value [42]. The computation formulation can be formalized as follows with $nbPE(t)$ the number of positive emotions caused by agent j to agent i at time t , $nbNE(t)$ the number of negative emotions caused by agent j to agent i at time t and α the evolution coefficient between 0 and 1:

$$liking(t + 1) = liking(t) * (1 + solidarity(t)) + \alpha * (nbPE(t + 1) - nbNE(t + 1))$$

- **Dominance** : Keltner and Haid [24] and Shiota *et al.* [39] explain that an emotion of fear or sadness caused by another agent represents an inferior status. But Knutson [25] explains that perceiving fear and sadness in others increases the sensation of power over those persons. The computation formulation can be formalized as follows with $nbONE(t)$ the number of negative emotions caused by agent i to agent j at time t and $nbSNE(t)$ the number of negative emotions caused by agent j to agent i at time t and α the evolution coefficient between 0 and 1:

$$dominance(t + 1) = dominance(t) + \alpha * (nbONE(t + 1) - nbSNE(t + 1))$$

- **Solidarity**: As explained in Section 2, the solidarity represents the degree of similarity of desires, beliefs and expectations between two agents. In our work, the evolution of the solidarity value depends on the ratio of similarity between the desires, beliefs and uncertainties between agent i and agent j . We compare the desire, belief and uncertainty bases of the two agents and look for similarities and differences. If the predicates are equal and have the same truth value, it is a similarity, otherwise, if the predicates are equal but with a different truth value, it is a difference. For each similarity (resp. difference), the solidarity value increases (resp. decreases) of one α level. On top of that, according to de Rivera and Grinkis [16], negative emotions tend to decrease the value of solidarity between two people. The computation formulation can be formalized as follows with $nbS(t)$ the number of similarities at time t and $nbD(t)$ the number of differences at time t , $nbNE(t)$ the number of negative emotions caused by agent j to agent i at time t and α

the evolution coefficient between 0 and 1:

$$\begin{aligned} \text{solidarity}(t + 1) = & \text{solidarity}(t) + \\ & + \alpha(\text{nbS}(t + 1) - \text{nbD}(t + 1) - \text{nbNE}(t + 1)) \end{aligned}$$

- **Familiarity**: In psychology, emotions and cognition do not seem to impact the familiarity. However, Collins and Miller [15] explain that people tend to be more familiar with other people they appreciate. We model this notion by basing the evolution of the familiarity value on the liking value between two agents. The computation formulation can be formalized as follow:

$$\text{familiarity}(t + 1) = \text{familiarity}(t) * (1 + \text{liking}(t + 1))$$

4 An Agent Architecture mixing cognition, emotions and social relations

The formalism proposed in this article has been used to enhance an existing agent architecture in order to ease the definition of social agents with a behavior combining cognition, emotions and social relations. The developed architecture has been implemented in GAMA, a modeling and multi-agent simulation platform [20].

4.1 Presentation of the architecture

We integrate our work as an extension of the architecture presented in Figure 1 which has been defined in [10].

In this architecture, a predicate (see Section 3.1) is implemented with a tuple $\langle \text{Name}, \text{Values}, \text{Priority}, \text{Truth}, \text{Lifetime}, \text{Praiseworthiness}, \text{Cause_Agent} \rangle$ with the following elements:

- **Name** (mandatory): the name of the predicate.
- **Values** (optional): a map containing the values stored by the predicate.
- **Priority** (optional): a real positive priority value to compare two predicates.
- **Truth** (optional): a boolean value that indicates if the predicate is true or false.
- **Lifetime** (optional): an integer value that indicates the number of steps the predicate will exist.
- **Praiseworthiness** (optional): a real value between -1.0 and 1.0 that indicates if the predicate is praiseworthy (a positive value) or blameworthy (a negative value).
- **Cause_Agent** (optional): the agent causing the predicate.

We implement emotions defined in Section 3.1 as a tuple $\langle \text{Name}, \text{Predicate}, \text{Cause_Agent}, \text{Intensity}, \text{Decay} \rangle$ with the following elements:

- **Name** (mandatory): the name of the emotion.

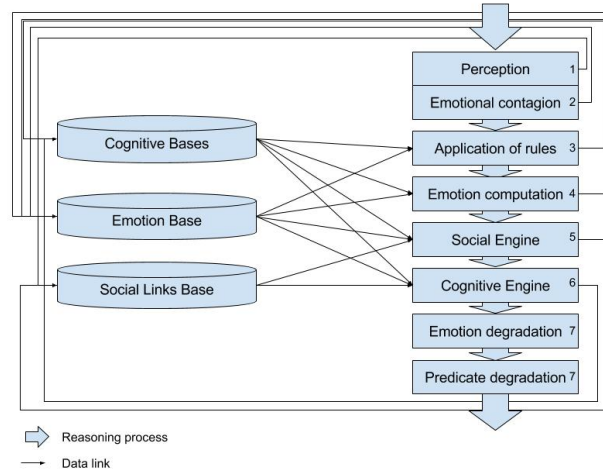


Fig. 1. Schema of our cognitive, emotional and social architecture

- **Predicate** (optional): the predicate, identified by its name, that represents the fact about which the emotion is expressed.
- **Cause_Agent** (optional): the agent causing the emotion.
- **Intensity** (optional): a real positive value of the intensity of the emotion.
- **Decay** (optional): a real positive value which will be subtracted from the intensity value at the end of each time step.

We implement social relations with the tuple $\langle \text{agent, liking, dominance, solidarity, familiarity} \rangle$ as exposed in Section 3.1.

4.2 Reasoning cycle of the agent

The first step in the reasoning cycle of the agent with our architecture is the perception of the environment (step 1 in Figure 1). Perception updates the beliefs of the agent and creates social links with other agents met. On the same model, the emotional contagion module (step 2) serves as an emotional perception of the environment as it updates the agent's emotions according to the emotions of agents nearby. This step of perception can be parametrized with different variables such as a distance value for example. Figure 2 shows a definition of a perception in GAML, the programming language of GAMA. This example enables to perceive "people" agents at a distance of 10 meters.

Then, the agent applies inference rules (step 3), defined by the modeler, to manage the belief base and the desire base according to its previous perceptions. This step gives a dynamic to the overall behavior as the agent can act according to a change in the environment. These inference rules can also be influenced by

```
perceive target:people in:10
```

Fig. 2. Definition of a perception in GAML

```
rule belief:belief1 new_desire:desire1;
```

Fig. 3. Definition of an inference rule in GAML

emotions or social relations. Figure 3 shows the definition of an inference rule in GAML to create the desire "desire1" if the agent possesses the belief "belief1".

The emotion computation module (step 4) is the engine that creates automatically, with no intervention from the modeler, emotions. The emotions are created according to the rules defined in Section 3.2.

The social engine (step 5) is used to dynamically update the social relations of the agent. This evolution of social relations is based on the rules exposed in Section 3.3 and is done only with the other agents perceived. The idea is that a person updates its social relation with someone else only if they both are in contact. The same process is applied through our architecture.

The cognitive engine (step 6) is based on the BDI paradigm [11] and selects a desire to create a current intention. Then it selects a plan described by the modeler to answer the current intention. This whole process is influenced by the cognitive bases but also by the emotions and the social relations of the agent and, through the execution of plans, can influence back these bases. This cognitive engine is described in details in [45].

The final step of the reasoning cycle is the degradation of the agent's knowledge (step 7). The predicates stored in the cognitive bases are reduced in lifetime and the intensity of emotions are reduced by their decay value. This mechanism gives a temporal dynamic to the agent's behavior.

This architecture (with its new extension) is already available within the GAMA platform. Modelers can easily use it - with just few lines of codes - through the GAMA dedicated modeling language.

5 Example Case

The architecture defined in Section 4 has been used on the example case of an evacuation of a large open area during a bushfire in Australia.

5.1 Presentation of the example

Bushfires in Australia Bushfires are a true concern in Australia as they kill people and destroy properties each year. A study has been carried out to simulate the evacuation of an area during a bushfire using a BDI architecture to create the agent's behavior [2]. If the BDI model proposed in the cited paper

shows interesting results in terms of replication of the real situation, it has some limitations.

The goal of this example, in our work, is not to provide a realistic model to simulate the evacuation of an area during a bushfire, but to show how to use emotions and social relations provided by our architecture. We take as a basis the model developed by [2] and we explain how to add emotions and social relations and how these new features can change the agents' behavior. Note that the proposed model of bushfires evacuation was implemented using the BDI architecture of GAMA platform [45], so adding new social and emotional properties to the agents only required to write few lines of code that we present here.

Description of the basic model The basic model is made of an environment composed of buildings, shelters and fires and defines civil agents that will try to survive to fires. Shelters are safe places that can not be damaged by fire while buildings can burn. Fires are placed randomly in the environment and along the simulation, they can grow, propagate, burn buildings and people, and finally disappear.

The civil agents have two main behaviors: either they stay in their house and fight the fire or they escape to the nearest known shelter. In details, civil agents have a probability to be aware of the danger, a motivation value to escape and a motivation value to fight the fire that is different for each agent. When an agent perceives a fire within a certain distance, if this agent is aware of the danger, it will add desires both to stay and to escape. Depending on their motivation, they will chose the intention to fight or to flee. If the motivation to escape becomes bigger than the motivation to stay, the agent can decide to give up its fight and try to flee to a shelter.

5.2 Implementation of the example using the developed architecture

Adding social links between agents We propose to improve the agent's behavior in terms of realism by giving them social relations. Social links between agents can be included at the start of the simulation or can dynamically appear during the simulation.

An example of social relation that can exist at the start of the simulation is a family relation. In the context of bushfires, we can easily imagine that two members of a family will try to help each other surviving the catastrophe.

From the point of view of the implementation, the modeler only need to add the family link in the initialization phase of the agent. This relationship will be represented by a social link with the family member with a familiarity value of 1.0 as shown in figure 4.

Social relations can also be used dynamically when agents are escaping to a shelter but do not know the shelter's location. When meeting another escaping agent, a social link is created. If the solidarity value of the link is high enough, the two agents will help each other going to a shelter.

```
do add_social_link(
  new_social_link(familyMember) set_familiarity 1.0
);
```

Fig. 4. Definition of a family link

After dynamically creating a social relation as shown in Figure 5, the modeler only needs to change the plan to escape to make an agent follow another agent if there is a social link between them with a particular solidarity value.

```
perceive target: pedestrianBDI{
  socialize;
}
```

Fig. 5. Definition of dynamic creation of social links

Creating emotions to change the agents' behavior To add emotions to civil agents, we use the emotional module of our architecture that automatically creates emotions depending on the mental states of the agent.

The simulation runs a first time with no emotions as a training session. An agent that decided to escape after fighting the fire is proud to have fled if it is alive and its house is destroyed. This pride emotion increases its motivation to escape for a future fire. If it is alive and its house is not destroyed, it is ashamed to have fled and this emotion decreases the motivation to escape for a future fire.

The implementation consists in adding the belief that the agent is alive or dead, has tried to flee or not and its house is destroyed or not at the end of the training run of the simulation as shown by Figure 6. Eventually, the modeler defines rules to change the internal motivations of the agent depending on its emotions as explained in Figure 7.

```
do add_belief(flee with_praiseworthiness -1.0);
```

Fig. 6. Adding a belief corresponding to the state of the agent at the end of the training run

```
if(has_emotion(shameFlee)){
  if(escape_motivation>0.0){
    escape_motivation <-
      escape_motivation*(1.0-0.2);
  }
}
```

Fig. 7. Using a shame emotion to update the escape motivation

The complete model can be found at the following site:
<https://github.com/mathieuBourgais/ExampleModels>

5.3 Discussion

As shown by the implementation of the example, a modeler can easily enhance the behavior of his/her agents by just writing few lines of GAML code, the programming language of the GAMA platform. As shown by [27][35], this programming language is quite easy to learn and to use by social scientists who are not expert in computer science.

On top of that, modelers with more knowledge in programming and in AI can use our architecture deeper to create complex behaviors. For example, an expert user in emotions can easily manually redefine emotions thanks to the definition of inference rules or use multiple emotions with different intensities and different social links to create believable agents for social simulations.

6 Conclusion

In this paper, we have presented a formalism to use emotions and social relations in the modeling of cognitive social agents. This formalism has been integrated in an agent architecture and implemented in a multi-agent simulation platform to show its ease of use from the point of view of people not expert in programming. The example of bushfires in Australia indicates a way to use our work on the simulation of an evacuation in an open and large area.

In the future, we want to carry out experiments with modelers to test the ease of use of our architecture. We also plan to improve our work by adding a personality to create social agents more and more realistic while keeping in mind the constraint of easiness for modelers who are not expert in programming.

Acknowledgment

This work is partially supported by two public grants overseen by the French National Research Agency (ANR) as part of the program PRC (reference: ESCAPE ANR-16-CE39-0011-01 and ACTEUR ANR-14-CE22-0002).

References

1. Carole Adam. Emotions: from psychological theories to logical formalization and implementation in a bdi agent, 2007.
2. Carole Adam and Julie Dugdale. Comparing agent architectures in social simulation: Bdi agents versus finite-state machines. In *HICSS*, 2016.
3. Carole Adam and Benoit Gaudou. Bdi agents in social simulations: a survey. *The Knowledge Engineering Review*, 2016.
4. Magda B Arnold. Emotion and personality. 1960.

5. Tina Balke and Nigel Gilbert. How do agents make decisions? a survey. *Journal of Artificial Societies and Social Simulation*, 2014.
6. Michael Balmer, Marcel Rieser, Konrad Meister, David Charypar, Nicolas Lefebvre, Kai Nagel, and K Axhausen. Matsim-t: Architecture and simulation times. *Multi-agent systems for traffic and transportation engineering*, 2009.
7. Joseph Bates. The role of emotion in believable agents. *Communications of the ACM*, 1994.
8. Fabio Bellifemine, Agostino Poggi, and Giovanni Rimassa. Jade—a fipa-compliant agent framework. In *Proceedings of PAAM*. London, 1999.
9. Timothy Bickmore and Justine Cassell. Relational agents: a model and implementation of building user trust. In *Proceedings of the SIGCHI conference on Human factors in computing systems*. ACM, 2001.
10. M. Bourgeois, P. Taillandier, and L. Vercoeur. An agent architecture coupling cognition and emotions for simulation of complex systems. In *SSC*, 2016.
11. M Bratman. Intentions, plans, and practical reason. 1987.
12. Penelope Brown and Stephen C Levinson. *Politeness: Some universals in language usage*, volume 4. Cambridge university press, 1987.
13. MD Byrne and JR Anderson. Perception and action. *The atomic components of thought*, 1998.
14. P. R. Cohen and H. J. Levesque. Intention is choice with commitment. 1990.
15. Nancy L Collins and Lynn Carol Miller. Self-disclosure and liking: a meta-analytic review. *Psychological bulletin*, 1994.
16. Joseph de Rivera and Carmen Grinkis. Emotions as social relationships. *Motivation and emotion*, 1986.
17. Nigel Gilbert and Klaus Troitzsch. *Simulation for the social scientist*. McGraw-Hill Education (UK), 2005.
18. Jonathan Gratch. Socially situated planning. In *Socially Intelligent Agents*. Springer, 2002.
19. Jonathan Gratch and Stacy Marsella. A domain-independent framework for modeling emotion. *Cognitive Systems Research*, 2004.
20. Arnaud Grignard, Patrick Taillandier, Benoit Gaudou, Duc An Vo, Nghi Quang Huynh, and Alexis Drogoul. *GAMA 1.6: Advancing the Art of Complex Agent-Based Modeling and Simulation*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013.
21. Nick Howden, Ralph Rönquist, Andrew Hodgson, and Andrew Lucas. Jack intelligent agents-summary of an agent infrastructure. In *5th International conference on autonomous agents*, 2001.
22. Katherine Isbister. *Better game characters by design: A psychological approach*. Elsevier San Francisco, 2006.
23. Hong Jiang, Jose M Vidal, and Michael N Huhns. Ebd: an architecture for emotional agents. In *Proceedings of the 6th international joint conference on Autonomous agents and multiagent systems*. ACM, 2007.
24. Dacher Keltner and Jonathan Haidt. Social functions of emotions. 2001.
25. Brian Knutson. Facial expressions of emotion influence interpersonal trait inferences. *Journal of Nonverbal Behavior*, 1996.
26. John E Laird, Allen Newell, and Paul S Rosenbloom. Soar: An architecture for general intelligence. *Artificial intelligence*, 1987.
27. EG Macatulad and AC Blanco. 3dgis-based multi-agent geosimulation and visualization of building evacuation using gama platform. *The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 2014.

28. Karen L Myers. User guide for the procedural reasoning system. *SRI International AI Center Technical Report.*, 1997.
29. Ranjit Nair, Milind Tambe, and Stacy Marsella. The role of emotions in multiagent teamwork. *Who Needs Emotions*, 2005.
30. Magalie Ochs, Nicolas Sabouret, and Vincent Corruble. Simulation of the dynamics of nonplayer characters' emotions and social relations in games. *IEEE Transactions on Computational Intelligence and AI in Games*, 2009.
31. Andrew Ortony. Memories, thoughts, and emotions: Essays in honor of george mandler, chapter value and emotion, 1991.
32. Andrew Ortony, Gerald L Clore, and Allan Collins. *The cognitive structure of emotions*. Cambridge university press, 1990.
33. Alexander Pokahr, Lars Braubach, and Winfried Lamersdorf. Jadex: A bdi reasoning engine. In *Multi-agent programming*. Springer, 2005.
34. Helmut Prendinger and Mitsuru Ishizuka. Social role awareness in animated agents. In *Proceedings of the fifth international conference on Autonomous agents*. ACM, 2001.
35. Nur Raihan Ramli, Szalinsyah Razali, and Mashanum Osman. An overview of simulation software for non-experts to perform multi-robot experiments. In *ISAMSR*. IEEE, 2015.
36. Ralph Rönquist. The goal oriented teams (gorite) framework. In *International Workshop on Programming Multi-Agent Systems*. Springer, 2007.
37. Daniel Rousseau and Barbara Hayes-Roth. A social-psychological model for synthetic actors. In *Proceedings of the second international conference on Autonomous agents*. ACM, 1998.
38. Ilias Sakellariou, Petros Kefalas, and Ioanna Stamatopoulou. Enhancing netlogo to simulate bdi communicating agents. In *Hellenic Conference on Artificial Intelligence*. Springer, 2008.
39. Michelle N Shiota, Belinda Campos, Dacher Keltner, and Matthew J Hertenstein. Positive emotion and the regulation of interpersonal relationships. *The regulation of emotion*, 2004.
40. Dharendra Singh and Lin Padgham. Opensim: A framework for integrating agent-based models and simulation components. In *Frontiers in Artificial Intelligence and Applications-Volume 263: ECAI 2014*. IOS Press, 2014.
41. Craig A. Smith and Richard S. Lazarus. Emotion and adaptation. 1990.
42. Eliot R Smith, Diane M Mackie, and Heather M Claypool. *Social psychology*. Psychology Press, 2014.
43. Ron Sun. *Cognition and multi-agent interaction: From cognitive modeling to social simulation*. Cambridge University Press, 2006.
44. Jan Svennevig. *Getting acquainted in conversation: a study of initial interactions*. John Benjamins Publishing, 2000.
45. Patrick Taillandier, Mathieu Bourgeois, Philippe Caillou, Carole Adam, and Benoit Gaudou. A bdi agent architecture for the gama modeling and simulation platform. In *MABS 2016*.
46. H. Van Dyke Parunak, Robert Bisson, Sven Brueckner, Robert Matthews, and John Sauter. A model of emotions for situated agents. In *Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems*. ACM, 2006.
47. Uri Wilensky and I Evanston. Netlogo: Center for connected learning and computer-based modeling. *Northwestern Univ., Evanston, IL*, 1999.