BEN: An Agent Architecture
for Explainable and Expressive Behavior
in Social Simulation

Mathieu Bourgais[1]([✉]), Patrick Taillandier[2], and Laurent Vercouter[1]

[1] Normandie Univ, INSA Rouen Normandie, LITIS,
76000 Rouen, France
`mathieu.bourgais@insa-rouen.fr`
[2] MIAT, INRA, 31000 Toulouse, France

Abstract. Social Simulations are used to study complex systems featuring human actors. This means reproducing real-life situations involving people in order to explain an observed behavior. However, there are actually no agent architectures among the most popular platforms for agent-based simulation enabling to easily model human actors. This situation leads modelers to implement simple reactive behaviors while the EROS principle (Enhancing Realism Of Simulation) fosters the use of psychological and social theory to improve the credibility of such agents. This paper presents the BEN architecture (Behavior with Emotions and Norms) that uses cognitive, affective and social dimensions for the behavior of social agents. This agent architecture has been implemented in the GAMA platform so it may be used by a large audience to model agents with a high level explainable behavior. This architecture is used on an evacuation case, showing how it creates believable behaviors in a real case scenario.

Keywords: Social simulation · Agent architecture · Cognition · Emotions · Evacuation

1 Introduction

These last years, agent-based simulation has been used to study complex systems featuring human actors; the community is now speaking of social simulation [19]. The main goal is to reproduce real life situations involving hundreds or thousands of simulated humans in order to better understand interactions leading to an observed result.

To be as close as possible to case studied, social simulations have to integrate social agents with a behavior as close as possible to the human behavior. The creation of believable social agents implies the reproduction of complex processes simulating the human reasoning [44]. Such systems lead to a closer behavior to the one expected. However, the obtained behavior may be hard to explain and to express with high level concepts.

This is the meaning of the KISS (Keep It Simple, Stupid) principle [6] which invites modelers to keep a simple behavior model so it can be explainable by simple rules at any moment in the simulation. This principle has been discussed over the years, leading to the KIDS (Keep It Descriptive, Stupid) principle [18] that favors more descriptive models to gain realism and then the EROS (Enhancing Realism Of Simulation) principle [22] which calls for the use of cognitive, affective and social dimensions to improve the credibility of social agents. Therefore, the problem is to model realistic simulated humans with an explainable behavior at a high level.

To tackle this issue, this paper presents BEN (Behavior with Emotions and Norms), a modular agent architecture integrating cognition, emotions, emotional contagion, personality, norms and social relations. Each of these components relies on psychological or social theories, helping a modeler to improve the credibility of simulated humans and ensuring an explainable behavior with high level concepts [24].

This architecture is implemented and integrated within GAMA [48], a modeling and simulation platform aiming to be used by a large audience. The goal is to create a tool that may even be used by modelers who are not expert in programming, without loosing the expressivity for the behavior developed. This implementation is explained in this paper through the example case of the evacuation of a nightclub, showing it succeeds to handle a real-life scenario and still provide a behavior with a high degree of explainability.

This paper is structured as follows: Sect. 2 reviews existing works to create social agents with a cognitive behavior, an emotional engine or social relations but also the existing agent architecture in popular simulation platforms. In Sect. 3, a formalism is proposed to deal with the mental state of the agent in terms of cognition, emotion and social relations. Section 4 describes the BEN architecture which relies on the aforementioned formalism. Section 5 presents the implementation of BEN through an example to illustrate how it can be used on a model of evacuation to create an explainable and believable behavior for agents simulating humans. Finally, Sect. 6 serves as a conclusion.

2   Related Works

Creating a believable social agent with an explainable behavior may be complex [29]. To ease this process, simulation platforms and behavior architecture have been developed by the community. These existing works are presented in this section.

2.1   Frameworks and Platforms for Simulations

Among the various agent-based platforms [27], some like JACK [21] or Jadex [38] implement the BDI (Belief Desire Intention) [10] paradigm, giving a cognitive behavior to agents, based on modal logic [15]. The addition of cognition

helps creating more believable agents [2] but these platforms are not suitable for thousands of agents required in simulation.

To overcome this problem, Sing and Padgham [42] propose to connect a simulation platform to an existing BDI framework (like JACK or Jadex) and, with the same idea [37], the Matsim platform [8] has been linked with the GORITE BDI framework [40]. These works require a high level in computer science, making it difficult to use by modelers with low level programming skills.

Frameworks like Repast [16] or MASON [31] are dedicated simulations tools which improve existing programming languages to ease the development of agent-based simulations. Agents are described by Java classes and the framework is used to describe the scheduling of the execution of all the classes and the output of the simulation. This means these softwares do not offer specific agent architecture to control or to explain the agents' behavior.

On the other hand, simulation platforms like Netlogo [50] or GAMA [48] are dedicated softwares with their own programming language, their own interface and their own interpreter and compiler. They are made to be easy to use by people with low level in programming skills and they can handle thousands of agents during simulations, making them usable for the definition of explainable models made by experts of the studied fields.

By default, these platform do not propose any particular architecture for the agent behavior: modelers have to define these behaviors with "if-then-else" rules. However, there exist plugins, for NetLogo [41] and for GAMA [47], to use agent architectures based on the BDI paradigm in order to create simulated humans with a more complex and more believable behavior. They both provide the agents with high level concepts such as beliefs and intentions and GAMA's plugin goes beyond, offering a reasoning engine, leading agents to make decisions based on the perception of its environment.

2.2   Agent Architectures for Social Simulations

Using behavioral architectures enables modelers to define more easily credible and explainable social agents as these architectures offer high level concepts from works in psychology and sociology for the decision making. Among the numerous agent architectures [7], some of the most known are presented in this section.

SOAR [28] and ACT-R [13] are two cognitive architectures, grounded on works from psychology. Agents have access to a long and a short term memory, making a decision based on the previous experiments in a given context. These approaches are more complex than the BDI paradigm, making the decision making process more credible. However, they require heavy computation time, which makes them less pertinent for social simulations involving thousand simulated actors.

CLARION [45] represents another proposition of cognitive architecture. The agent's reasoning is divided between four sub-systems, each one manipulating explicit and implicit elements to make a decision in a given context. To our knowledge, CLARION is still a theoretical architecture which has not been implemented in any simulation platform.

Another approach consists in building the reasoning engine around the emotions of the agents. For example, EMA [20] is based on the cognitive appraisal theory of emotion [4] developed by Smith and Lazarus [43] while DETT [49] is based on the OCC [36] theory of emotions. Both those systems creates emotions by assessing the perceptions of the environment and then infer a behavior from the emotional state of the agent.

eBDI [23] relies on OCC theory too but it also uses a BDI architecture to make decisions. This means emotions are created through perceptions and then act upon beliefs, desires and intentions. Finally, these modified mental states are used to make a decision. This proposition has not been yet integrated to a simulation platform.

Finally, some researchers propose to rely on the social context of the agent to describe its behavior: this is done with normative architectures. EMIL-A [3] and NoA [26] describe the agent's behavior with social norms, obligations and sanctions. In other words, an agent makes a decision depending on the state of the normative system at the level of a society of agents.

BOID [12] and BRIDGE [17] propose to combine a normative architecture with a BDI paradigm, leading the agent to take into account the social system when making a decision. However, contrary to EMIL-A and NoA, the agent has personal beliefs, desires and intention, creating a more heterogeneous and credible behavior. But, to our knowledge, these architectures have not been implemented in simulation platform in order to deal with thousand of simulated actors.

2.3   Synthesis

To comply with the EROS principle, modelers need architectures proposing as much psychological and social dimensions as possible. Currently, as shown in this section, there does not exists a single architecture proposing at the same time, cognition, affective dimensions and social dimension for simulation. The only attempts, to our knowledge, to combine more than two traits have used the notion of personality, to combine cognition with emotions and emotional contagion [30] or to combine cognition with emotions and social relations [35].

In this paper, we tackle this issue by proposing BEN (Behavior with Emotions and Norms), an agent architecture featuring cognition, emotions, personality, emotional contagion, social relations and norm management. To implement it, we have based our work on the existing cognitive architecture provided by GAMA. To ease the use of BEN, we have implemented it using the principles of GAMA that has proved its ease of use [32,39] thanks to its modeling language GAML that we extended.

3   Formalization of Mental States

With the BEN architecture, an agent manipulates cognitive mental states, emotions and social relations to make a decision. These notions, and the formalism used to represent them, are presented in this section.

3.1   Representing the World with Predicates

Predicates are used to unify the representation of the information from the world from the agent's point of view. $\mathbf{P}_j(\mathbf{V})$ represents a general predicate with the following elements:

– **P:** the identifier of the predicate.
– **j:** the agent causing the information.
– **V:** a set of values stored by the predicate.

Depending on the context, this general representation may change. $\mathbf{P}_j(\mathbf{V})$ represents an information with no particular value attached, $\mathbf{P(V)}$ represents an information caused by no particular agent and $\mathbf{P}$ stands for an information with no particular value, caused by no particular agent.

For example, the information there is a fire in the environment is represented by the predicate **fire**. If this fire is caused by agent *Bob*, it is represented by $\mathbf{fire}_{Bob}$. Finally, if this fire caused by Bob is at a location (x;y), this information is represented by $\mathbf{fire}_{Bob}(\mathbf{location} :: (\mathbf{x}, \mathbf{y}))$.

3.2   Reasoning According to Cognitive Mental States

With BEN, an agent has cognitive mental states representing its thoughts on the world. $\mathbf{M}_i(\mathbf{PMEm}, \mathbf{Val}, \mathbf{Li})$ represents a general cognitive mental state possessed by agent i with the following elements:

– **M:** the modality, indicating the type of the cognitive mental state (e.g. a belief).
– **PMEm:** the object of the cognitive mental state. It could be a predicate (P), another cognitive mental state (M) or an emotion (Em).
– **Val:** a real value which meaning depends on the modality. It enables to compare two cognitive mental states with the same modality and the same object.
– **Li:** a lifetime value indicating how long the cognitive mental state stays in the agent's knowledge.

In BEN, cognition is based on the BDI model [10], stating an agent needs beliefs, desires and intentions. Also, to link cognition with affective and social dimensions, BEN features 6 different modalities:

– **Belief:** represents what the agent beliefs about the world. The meaning of the attached value is the strength given by the agent to the belief.
– **Uncertainty:** represents an uncertain information about the world. The meaning of the attached value is the importance given to that uncertainty by the agent.
– **Desire:** represents a state of the world the agent want to reach. The meaning of the attached value is the priority of this desire, compared to other desires.
– **Intention:** represent a state of the world the agent is willing to achieve. The meaning of the attached value is the priority of this intention, compared to other intentions.

– **Ideal:** represents an information which is socially appraised by the agent. The attached value is a praiseworthiness value given by the agent.
– **Obligation:** represents a state of the world the agent ought to reach. The meaning of the attached value is the priority of this obligation, compared to other obligations.

3.3   Representing Emotions and Social Relations

Emotions in BEN are based on the OCC theory [36] which means emotions are valued answers to the appraisal of a situation. $\mathbf{Em}_i(\mathbf{P, Ag, I, De})$ represent an emotion possessed by agent i with the following elements:

– **Em:** the name of the emotion.
– **P:** the predicate about which the emotion is felt.
– **Ag:** the agent responsible for the emotion.
– **I:** the intensity, positive or null, of the emotion.
– **De:** the decay value for the emotion's intensity.

This representation enables the agent to have multiple emotions at the same time, all on different predicates. Also, this representation can be adapted, with $\mathbf{Em}_i(\mathbf{P, Ag})$ representing an emotion with no particular intensity nor decay value.

Finally, each agent may store social relations with other agents simulating human actors. These relations are based on the work of Svennevig [46] who identifies four minimal dimensions to describe a social relation between two people. In BEN, $\mathbf{R}_{i,j}(\mathbf{L, D, S, F, T})$ represents a social relation, from agent i towards agent j with the following elements:

– **R:** the identifier of the social relation.
– **L:** a real value between -1 and 1 standing for the degree of liking. A value of -1 indicates agent j is hated, a value of 1 indicates agent j is liked.
– **D:** a real value between -1 and 1 standing for the degree of dominance. A value of -1 indicates agent j is dominating, a value of 1 indicates agent j is dominated.
– **S:** a real value between 0 and 1 standing for the degree of solidarity. A value of 0 indicates no solidarity with agent j, a value of 1 indicates a complete solidarity with agent j.
– **F:** a real value between 0 and 1 standing for the degree of familiarity. A value of 0 indicates no familiarity with agent j, a value of 1 indicates a complete familiarity with agent j.
– **T:** a real value between -1 and 1 standing for the degree of trust. A value of -1 indicates mistrust against agent j, a value of 1 indicates a complete trust towards agent j.

With this definition, social relations do not have to be symmetric; between two agents i and j, $R_{i,j}(L,D,S,F,T)$ is not obviously equal to $R_{j,i}(L,D,S,F,T)$.

4 An Agent Architecture with Cognitive, Affective and Social Dimensions

The BEN architecture represents the main contribution of this article. In this section, we explain how an agent using BEN makes a decision with cognition, emotions, emotional contagion, personality, social relations and norms. With these dimensions, an agent simulating an actor may react to a change in the environment and still explain its behavior with high level concepts.

4.1 Global Presentation of the Architecture

Figure 1 represents the theoretical BEN architecture, providing cognitive, affective and social dimensions to agents simulating human actors. It is made up of four modules, each composed of several processes, communicating with the agent's knowledge, all of this seating on the agent's personality.
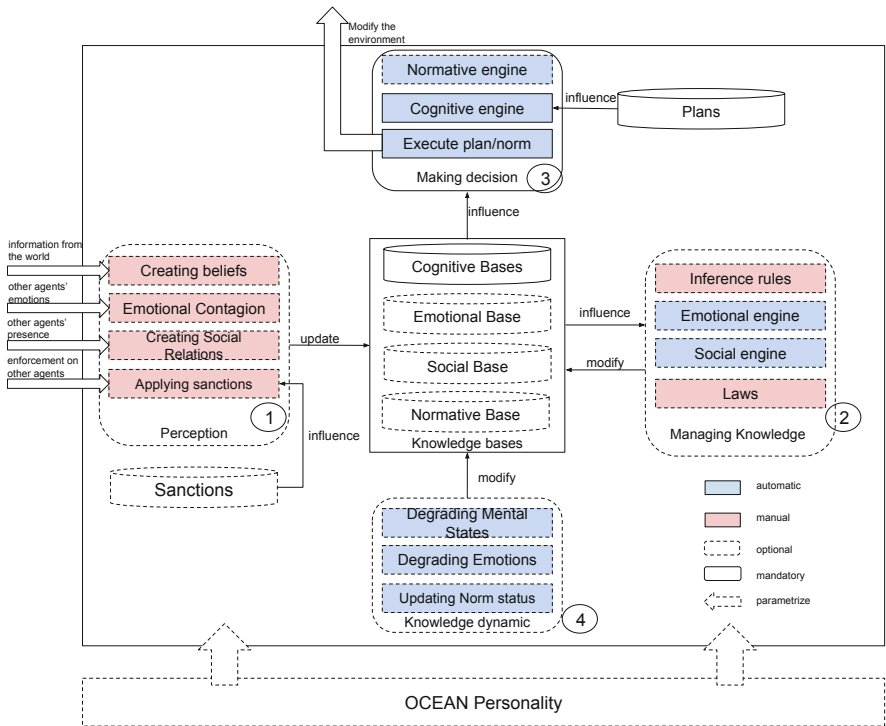

Fig. 1. The BEN architecture (Color figure)

The personality component is based on the OCEAN [33] theory which defines fives parameters (Openness, Consciousness, Extroversion, Agreeableness, Neurotism) that are sufficient to represent a personality. To ease the use of BEN, these

personality traits are the only parameter a modeler may access; they are used to compute all the other parameters needed by the various processes: probability to remove a plan or an intention unfulfilled in the cognitive part, charisma and emotional receptivity for the emotional contagion, initial intensity and decay for emotions created by the engine, update values of social relations obtained with the social engine, and obedience for the normative engine.

The agent's knowledge is composed by cognitive bases, containing cognitive mental states as formalized in section, an emotional base, a social base and a base of norms. This knowledge may evolve through the simulation, which is not the case of plans for the cognitive engine and sanctions for the normative engine, which are stored in dedicated bases, out of the agent's knowledge, as seen on Fig. 1.

Each module, and each process of each module, may be mandatory (in plain line on Fig. 1) or optional (in dash line on Fig. 1); the optional modules and processes may be deactivated by modelers if not necessary in the case studied. For example, if the social engine is not useful from the modeler's point of view, this process is not used in the definition of the agent's behavior as it is an optional process.

Finally, some processes are executed automatically (in blue on Fig. 1) and some others need to be defined manually by the modeler (in pink on Fig. 1). This manual definition enables the architecture to be adapted to each case study while the automatic processes ease the use of BEN as the modeler does not need to be an expert in emotional creation or social relations update.

4.2   Making Decisions in an Evolving Environment

On Fig. 1, every module has a number, indicating its order during the execution. Every time an agent is activated, it perceives the environment, it manages its knowledge based on the new perceptions, it makes a decisions and finally it gives a temporal dynamism to its knowledge. This section explains briefly how each process of each module works, more details can be found in a previous work [9].

Perceptions. The First step in BEN, corresponding to the module number 1 on Fig. 1, consists in perceiving the environment. This step is used to make a link between the world and the agent's knowledge, by creating beliefs and uncertainties on information from the environment, by defining emotional contagion with other agents or by creating new social relations. These three processes are defined manually which means the modeller has to indicate what information is transformed as a predicate and which cognitive mental state is build upon that predicate, which emotion is subjected to an emotional contagion and what is the initial value for each dimension of a new social relation. The last process of this module enables an agent to execute sanctions during the enforcement done on the other agents perceived.

Adding a belief is an important process in BEN as it triggers different rules. Precisely, adding a belief $Belief_A(X)$:

– removes belief $Belief_A(notX)$.
– removes intention $Intention_A(X)$.
– removes desire $Desire_A(X)$ if intention $Intention_A(X)$ has just been removed.
– removes uncertainty $Uncertainty_A(X)$ or $Uncertainty_A(notX)$.
– removes obligation $Obligation_A(X)$.

With the same principle, adding uncertainty $Uncertainty_A(X)$:

– removes uncertainty $Uncertainty_A(notX)$.
– removes belief $Belief_A(X)$ or $Belief_A(notX)$

All these processes are defined inside a perception, which may be parameterized. A modeller indicates a distance of perception or a geometry inside which the perception is done, but also specifies which agents are perceived. As it is, this module and all its processes may adapt to any case study in social simulation.

Managing Knowledge. The second step of BEN, corresponding to the module number 2 on Fig. 1, enables the agent to manage its knowledge after the perception and before making decision. In this phase, modelers may define inference rules, which enable to create or remove any cognitive mental state depending on the actual status of the agent's knowledge. For example, a modeler may define that an agent has a new desire $Desire_A(Y)$ if this agent has a belief $Belief_A(X)$. On the same model, laws may be defined to create obligations if the obedience value of the agent, computed from its personality, is great enough.

During this second step, an emotional engine creates emotions based on the agent's knowledge. This process is done according to the OCC theory [36] and its formalism with the BDI model [1]. For example, an emotion of joy about a predicate P is created according to the following rule: $Joy_i(P_j, j) \overset{\text{def}}{=} Belief_i(P_j)$ & $Desire_i(P)$. The complete process and all the rules to create 20 emotions automatically, with no intervention from the modeler, are detailed in a previous work [9].

Finally, a social engine may be executed during this second step of the architecture. It updates the social relation with the other agents perceived, depending on the knowledge previously acquired. All this process and the equations to compute automatically the new value for each dimension of a social relation are explained in a previous work [9].

Making Decision. The third step of BEN, corresponding to the module number 3 on Fig. 1, is the only mandatory part. This module enables the agent to make decisions and then execute an action, all of this through a cognitive engine over which a normative engine may be added. It is executed automatically, with no need of intervention from the modeler.

The cognitive architecture is based on the BDI model [10]: the agent has intentions based on its desires and one of the intentions as a current intention. The modeler defines plans of action that indicates what action an agent has to

do for a particular current intention in a given context; the plan chosen is kept as the current plan. The normative engine works the same way as the cognitive engine, with obligations as desires and norms as plans. The only difference is an obedience value that can be added to norms and obligations. More details may be found in previous works [9].

Temporal Dynamics. The final part of the architecture, corresponding to the module number 4 on Fig. 1, gives a temporal dynamic to the agent's behavior. This is done automatically by degrading the cognitive mental states and the emotions and by updating the status of each norm.

The degradation of cognitive mental states decreases the lifetime of each cognitive mental state stored by the agent. This mechanism enables an agent to forget, after a certain time, a belief, a desire, etc. The degradation of emotions consists in subtracting the decay value of each emotion to each intensity. With this process, an emotion fades away, unless it is created again, for example with the emotional engine or through the emotional contagion process.

Finally, the last process updates the status of each norm, indicating if it was usable in the current context, and in the case it was usable, has it been used or not by the agent. This system enables to ease the enforcement of norms in a later perception, as each norm indicates its status instead of computing this status in another context.

5   Simulating the Evacuation of a Nightclub

The architecture defined in Sect. 4 has been implemented in the modeling and simulation platform GAMA [48], extending the GAML programming language to help modelers define social agents with cognitive, affective and social dimensions to express their behavior. This implementation is used on the example case of an evacuation of a nightclub in fire, as detailed in this section.

5.1   Presentation of the Example Case

The 27th of January 2013, the Kiss Nightclub in Santa Maria, Rio Grande do Sul state is Brazil, was set in fire at the end of a show lead by a local music band. The ceiling caught fire because of fireworks, emitting toxic smokes which lead to the death of 242 people. The official investigation put light on various factors which aggravated the tragedy: there were between 1200 and 1400 people in the building that could normally handle 641 people, there was only one entrance/exit door, there was no smoke detector nor alarms and finally, the exit signs were showing the direction of the restrooms. The investigation also shows that most of the deceases were due to asphyxia, near the restrooms [5].

Our goal here is to reproduce the behavior of people caught in this tragedy in the most credible way possible. In other words, we are using the BEN architecture to create the agents' behaviors in order to get a result as close as possible as what happened in this nightclub during the fire.

5.2   Modeling the Behavior of Human Actors with BEN

The initial agent's knowledge, at the start of the simulation, can be divided into three types: beliefs about the world, initial desires and social relations with friends. Also, each agent has a personality. Table 1 indicates how few of these initial knowledge are formalized with BEN.

Table 1. Example of agent's initial knowledge

Statement	Formalisation	Description
A belief on the exact position of the exit door	$Belief_i(exitDoor, lifetime1)$	Each agent has a belief about the precise location of the exit door with a lifetime value at $lifetime1$
A desire there is no fire	$Desire_i(notFire, 1.0)$	Each agent wish there is no fire in the nightclub with a priority of 1.0. This desire cannot lead to an action (no action plan are defined to answer it)
A relation of friendship with another agent	$R_{i,j}(L, D, S, F, T)$	Each agent i is likely to have a social relation with agent j, representing its friend

The first step of BEN is the perception of the environment. We need to define what an agent perceives and how it affects its knowledge. Here are examples of the agent's perceptions:

– Perceiving the exit door updates the beliefs related to it.
– Perceiving the fire adds the belief there is a fire.
– Perceiving the smoke adds the belief about the level of smoke perceived.
– Perceiving other agents enables to create social relations with them. An emotional contagion about the fear of a fire is also defined.

Once the agent is up to date with its environment, its overall knowledge has to adapt to what it has perceived. This is done with the definition of inference rules and laws:

– A law creates the obligation to follow the exit signs if there is a reasonable doubt (modeled by the obedience value attached to the law and the quantity of smoke perceived) of a catastrophe.
– An inference rule adds the desire to flee if the agent has a belief there is fire.
– An inference rule adds an uncertainty there is a fire if the agent has a belief there is smoke.
– An inference rule adds the desire to flee if the agent has a fear emotion about the fire with an intensity greater than a given threshold.

With the execution of inference rules and laws, each agent creates emotions with the emotional engine. In this case, the presence of an uncertainty about the fire (added through the inference rule concerning the belief about smoke) with the initial desire that there is no fire produces an emotion of fear, which intensity is computed depending on the quantity of smoke perceived.

Once the agent has the desire to flee (because it perceived the fire or its fear of a fire had an intensity great enough), it needs action plans and norms to indicate what it has to do. Table 2 shows the definition of some action plans and norms used by the agent to answer its intention to flee, depending of the context it perceives.

Table 2. Action plans and norms answering the fleeing intention

Conditions	Actions	Commentaries
The agent has a good visibility and has a belief on the exact location of the exit door	The agent runs to the exit door	In this plan, the agent runs to the exit door following the shortest path
The agent has a good visibility and has no belief about the location of the door	The agent follows the agent in its field of view with the highest trust value among its social relations	This norm works with the trust value of social relations created during the simulation
The agent has a bad visibility and has the obligation to follow signs	The agent goes to the restrooms	In this norm, the agent comply with the law that indicates to follow exit signs
The agent has a bad visibility and has a belief exit signs are wrong	The agent moves randomly	In this plan, the agent moves randomly in the smoke

The social relation defined with a friend may also be used to define plans to help one's friend if it is lost in smoke. This plan consists in finding the friend and telling him the location of the exit door.

As the situation evolves during the simulation, an agent may change its current plan. For example, if an agent leaves the smoke area while fleeing to the restroom, it may perceive the exit, and go there instead of following the signs.

The complete model can be found at this address: https://github.com/mathieuBourgais/ExempleThese.

5.3   Results and Discussion

At the start of the simulation, agents are placed randomly in the recreated Kiss Nightclub with a personality initialised by a Gaussian distribution centered on 0.5 and with a standard deviation of 0.12 for each dimension. The spread of

the smoke is modeled according to an official report from the french government [14]; an agent is considered dead after 50 s in the heavy smoke.

Figure 2 shows a visual result of the simulation where the black lines represent the walls of the nightclub, the grey squares represent the smoke and the triangles represent the simulated actors. The color of each triangle indicates the plan followed. A video of the simulation can be found on the following address: https://github.com/mathieuBourgais/ExempleThese.
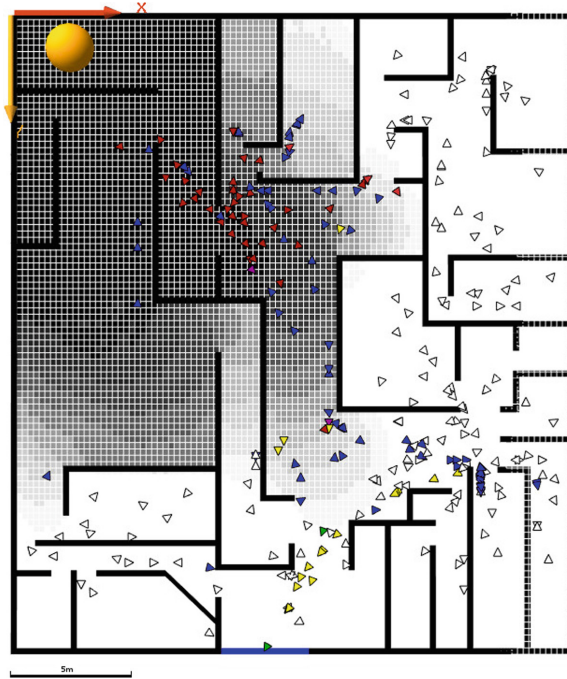

Fig. 2. Simulation of the Kiss Nightclub's evacuation

As the exact number of people in the nightclub is not known, we tested three cases: 1200 people at the beginning, 1300 people at the beginning and 1400 people at the beginning. The statistical results obtained in Table 3 are computed from 10 simulations for each scenario.

Table 3. Number of agents dead in the simulation of the Kiss Nightclub fire

Number of agents	1200	1300	1400
Mean value	230.2	237.7	249.4
Standard deviation	20.1	15.6	32.6

Statistical results indicate our model is well calibrated to reproduce the real life case where 242 people died. However, the main result concerns the explainability and the expressivity of the model. The video of the simulation shows various behavior patterns which may be expressed with high level concepts thanks to BEN.

For example, a lot of agents leave the club at the beginning of the simulation because they directly perceived the fire. This behavior seems corresponding to a real life case where people seeing a fire in a nightclub would flee. On the other hand, agents which do not perceive the smoke or the fire are fleeing later. During that time, they forgot the location of the exit so they had to follow the official exit signs, leading them, in this case, to the restrooms.

Thanks to the BEN architecture, we were able to translate a behavior expressed in common language into an actual behavior for simulated actors. At any moment, it is possible to pause the simulation to inspect the behavior of an agent; this behavior will be expressed in terms of cognitive mental states, emotions, social relations, norms and plans which is, from our point of view, easier to read and understand than equations. This point is supported by the fact that BEN and its cognitive part relies on folk psychology [34]. Also, some works [11,25] have shown that using BDI and emotions helps explaining the agents' behavior.

6   Conclusion

This article presents the BEN architecture, which enables to model agents simulating human actors with cognitive, affective and social dimensions. All the features of the architecture are based on theories coming from psychology and social sciences and are formalised in the same frame to interact between each other without being dependent. This allows the architecture to be domain independent and modular, so it can be used and adapted on different contexts.

This architecture is currently implemented in the modeling and simulation platform GAMA and this implementation is used, in this article, on the case study of the evacuation of a nightclub in fire. This example shows BEN achieves to produce a complex and credible behavior but maintaining its high level explainability.

References

1. Adam, C.: Emotions: from psychological theories to logical formalization and implementation in a BDI agent. Ph.D., thesis, INP Toulouse (2007)
2. Adam, C., Gaudou, B.: BDI agents in social simulations: a survey. Knowl. Eng. Rev. **31**, 207–238 (2016)

3. Andrighetto, G., Conte, R., Turrini, P., Paolucci, M.: Emergence in the loop: simulating the two way dynamics of norm innovation. In: Dagstuhl Seminar Proceedings. Schloss Dagstuhl-Leibniz-Zentrum für Informatik (2007)
4. Arnold, M.B.: Emotion and Personality. Columbia University Press, New York (1960)
5. Atiyeh, B.: Brazilian kiss nightclub disaster. Ann. Burns Fire Disasters. **26**, 3 (2013)
6. Axelrod, R.: Advancing the art of simulation in the social sciences. In: Conte, R., Hegselmann, R., Terna, P. (eds.) Simulating social phenomena. LNE, vol. 456, pp. 21–40. Springer, Berlin (1997). https://doi.org/10.1007/978-3-662-03366-1_2
7. Balke, T., Gilbert, N.: How do agents make decisions? A survey. J. Artif. Soc. Soc. Simul. **17**(4), 13 (2014)
8. Balmer, M., et al.: MATSim-T: architecture and simulation times. In: Multi-agent Systems for Traffic and Transportation Engineering (2009)
9. Bourgais, M.: Towards cognitive, affective and social agents in the simulation. Theses, Normandie Université (2018)
10. Bratman, M.: Intentions, Plans, and Practical Reason. Harvard University Press, Cambridge (1987)
11. Broekens, J., Harbers, M., Hindriks, K., van den Bosch, K., Jonker, C., Meyer, J.-J.: Do you get it? User-evaluated explainable BDI agents. In: Dix, J., Witteveen, C. (eds.) MATES 2010. LNCS (LNAI), vol. 6251, pp. 28–39. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-16178-0_5
12. Broersen, J., Dastani, M., Hulstijn, J., Huang, Z., van der Torre, L.: The BOID architecture: conflicts between beliefs, obligations, intentions and desires. In: Proceedings of the Fifth International Conference on Autonomous Agents, pp. 9–16. ACM (2001)
13. Byrne, M.D., Anderson, J.R.: Perception and action. In: The Atomic Components of Thought (1998)
14. Chivas, C., Cescon, J.: Formalisation du savoir et des outils dans le domaine des risques majeurs (dra-35) - toxicité et dispersion des fumées d'incendie phénoménologie et modélisation des effets. Technical report, INERIS (2005)
15. Cohen, P.R., Levesque, H.J.: Intention is choice with commitment. Artif. Intell. **42**(2–3), 213–261 (1990)
16. Collier, N.: RePast: an extensible framework for agent simulation. Univ. Chicago's Soc. Sci. Res. **36**, 2003 (2003)
17. Dignum, F., Dignum, V., Jonker, C.M.: Towards agents for policy making. In: David, N., Sichman, J.S. (eds.) MABS 2008. LNCS (LNAI), vol. 5269, pp. 141–153. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-01991-3_11
18. Edmonds, B., Moss, S.: From KISS to KIDS – *An 'Anti-simplistic' modelling approach*. In: Davidsson, P., Logan, B., Takadama, K. (eds.) MABS 2004. LNCS (LNAI), vol. 3415, pp. 130–144. Springer, Heidelberg (2005). https://doi.org/10.1007/978-3-540-32243-6_11
19. Gilbert, N., Troitzsch, K.: Simulation for the Social Scientist. McGraw-Hill Education (UK), New York (2005)
20. Gratch, J., Marsella, S.: A domain-independent framework for modeling emotion. Cogn. Syst. Res. **5**, 269–306 (2004)
21. Howden, N., Rönnquist, R., Hodgson, A., Lucas, A.: Jack intelligent agents-summary of an agent infrastructure. In: 5th International Conference on Autonomous Agents (2001)

22. Jager, W.: Enhancing the realism of simulation (EROS): on implementing and developing psychological theory in social simulation. J. Artif. Soc. Soc. Simul. **20**(3), 14 (2017)

23. Jiang, H., Vidal, J.M., Huhns, M.N.: EBDI: an architecture for emotional agents. In: Proceedings of the 6th International Joint Conference on Autonomous Agents and Multiagent Systems. ACM (2007)

24. Kaptein, F., Broekens, J., Hindriks, K., Neerincx, M.: Self-explanations of a cognitive agent by citing goals and emotions, October 2017

25. Kaptein, F., Broekens, J., Hindriks, K., Neerincx, M.: The role of emotion in self-explanations by cognitive agents. In: 2017 Seventh International Conference on Affective Computing and Intelligent Interaction Workshops and Demos (ACIIW), pp. 88–93. IEEE (2017)

26. Kollingbaum, M.J.: Norm-governed practical reasoning agents. Ph.D., thesis, University of Aberdeen Aberdeen (2005)

27. Kravari, K., Bassiliades, N.: A survey of agent platforms. J. Artif. Soc. Soc. Simul. **18**(1), 11 (2015)

28. Laird, J.E., Newell, A., Rosenbloom, P.S.: Soar: an architecture for general intelligence. Artif. Intell. **33**, 1–64 (1987)

29. Langley, P., Meadows, B., Sridharan, M., Choi, D.: Explainable agency for intelligent autonomous systems. In: Twenty-Ninth IAAI Conference (2017)

30. Lhommet, M., Lourdeaux, D., Barthès, J.-P.: Never alone in the crowd: a microscopic crowd model based on emotional contagion. In: Web Intelligence and Intelligent Agent Technology (WI-IAT). IEEE (2011)

31. Luke, S., Cioffi-Revilla, C., Panait, L., Sullivan, K., Balan, G.: Mason: a multiagent simulation environment. Simulation **81**, 517–527 (2005)

32. Macatulad, E.G., Blanco, A.C.: 3DGIS-based multi-agent geosimulation and visualization of building evacuation using GAMA platform. Int. Arch. Photogrammetry Remote Sens. Spatial Inf. Sci. **40**, 87 (2014)

33. McCrae, R.R., John, O.P.: An introduction to the five-factor model and its applications. J. Pers. **60**(2), 175–215 (1992)

34. Norling, E.: Folk psychology for human modelling: extending the BDI paradigm. In: Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems-Volume 1, pp. 202–209. IEEE Computer Society (2004)

35. Ochs, M., Sabouret, N., Corruble, V.: Simulation of the dynamics of nonplayer characters' emotions and social relations in games. IEEE Trans. Comput. Intell. AI Games **1**(4), 281–297 (2009)

36. Ortony, A., Clore, G.L., Collins, A.: The Cognitive Structure of Emotions. Cambridge University Press, Cambridge (1990)

37. Padgham, L., Nagel, K., Singh, D., Chen, Q.: Integrating BDI agents into a MAT-Sim simulation. In: Proceedings of the Twenty-First European Conference on Artificial Intelligence, pp. 681–686. IOS Press (2014)

38. Pokahr, A., Braubach, L., Lamersdorf, W.: Jadex: a BDI reasoning engine. In: Bordini, R.H., Dastani, M., Dix, J., El Fallah Seghrouchni, A. (eds.) Multi-Agent Programming. MSASSO, vol. 15, pp. 149–174. Springer, Boston (2005). https://doi.org/10.1007/0-387-26350-0_6

39. Ramli, N.R., Razali, S., Osman, M.: An overview of simulation software for non-experts to perform multi-robot experiments. In: ISAMSR. IEEE (2015)

40. Rönnquist, R.: The goal oriented teams (GORITE) framework. In: Dastani, M., El Fallah Seghrouchni, A., Ricci, A., Winikoff, M. (eds.) ProMAS 2007. LNCS (LNAI), vol. 4908, pp. 27–41. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-79043-3_2

41. Sakellariou, I., Kefalas, P., Stamatopoulou, I.: Enhancing NetLogo to aimulate BDI communicating agents. In: Darzentas, J., Vouros, G.A., Vosinakis, S., Arnellos, A. (eds.) SETN 2008. LNCS (LNAI), vol. 5138, pp. 263–275. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-87881-0_24
42. Singh, D., Padgham, L.: 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)
43. Smith, C.A., Lazarus, R.S., et al.: Emotion and adaptation. In: Handbook of Personality: Theory and Research, pp. 609–637 (1990)
44. Sun, R.: Cognition and Multi-agent Interaction: From Cognitive Modeling to Social Simulation. Cambridge University Press, Cambridge (2006)
45. Sun, R.: The importance of cognitive architectures: an analysis based on clarion. J. Exp. Theor. Artif. Intell. **19**(2), 159–193 (2007)
46. Svennevig, J.: Getting Acquainted in Conversation: A Study of Initial Interactions. John Benjamins Publishing, Amsterdam (2000)
47. Taillandier, P., Bourgais, M., Caillou, P., Adam, C., Gaudou, B.: A BDI agent architecture for the GAMA modeling and simulation platform. In: Nardin, L.G., Antunes, L. (eds.) MABS 2016. LNCS (LNAI), vol. 10399, pp. 3–23. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-67477-3_1
48. Taillandier, P., et al.: Building, composing and experimenting complex spatial models with the GAMA platform. GeoInformatica **23**, 299–322 (2018)
49. Van Dyke Parunak, H., Bisson, R., Brueckner, S., Matthews, R., Sauter, J.: A model of emotions for situated agents. In: Proceedings of the Fifth International Joint Conference on Autonomous Agents and Multiagent Systems. ACM (2006)
50. Wilensky, U., Evanston, I.: NetLogo: Center for connected learning and computer-based modeling, Northwestern University, Evanston, IL (1999)