# Mixed Agent/Social Dynamics for Emotion Computation

Julien Saunier
LITIS, INSA-Rouen
Avenue de l'Université - BP 8, 76801
Saint-Étienne-du-Rouvray Cedex
julien.saunier@insa-rouen.fr

Hazaël Jones
Montpellier SupAgro, UMR ITAP
Bâtiment 29, 2 place Pierre Viala, 34060
Montpellier
hazael.jones@supagro.inra.fr

## ABSTRACT

Affective computing is the study and development of systems and devices that can recognise, interpret, process, and simulate human affects. In this context, computational modelling of emotion is a major challenge in order to design believable virtual humans. This factor has an impact on both the individual behaviour and the collective one. Recently, researchers have shown an increased interest in the emotion contagion phenomenon in order to model emerging group behaviour.

Stemming from works on multi-agent systems environments, we propose an architecture to manage both internal and external emotion dynamics. Emotions evolve in function of three influences: punctual events, temporal dynamics and external influences. In an embodied agent approach, the first is the responsibility of the agent's mind, the second of the agent's body, and the third of the environment. This functional architecture is then adapted to a multi-agent architecture, adding a control responsibility to the agent body. Finally, we show the results of several experiments to examine the properties of the architecture and its efficiency by comparing it to a full agent approach.

## Categories and Subject Descriptors

Computing Methodologies [**Artificial intelligence**]: Distributed artificial intelligence, Multi-agent systems

## Keywords

Multi-agent Systems, Embodied agent, Emotional contagion, Architecture

## 1. INTRODUCTION

Human behaviour simulation has to take into account the role of emotions in the decision process [11]. Emotions have an impact on the whole cycle of the agent: perception, decision and action are driven by the agent's emotional state. Emotions are also used as a metaphor of social constructs in agent learning, trust, norm following and text analysis. In this article, we focus on the agents emotion computation, specifically the different influences generating their emotional state and the underlying MultiAgent System (MAS) architecture.

The emotional contagion theme has recently emerged to explain a number of collective phenomena such as crowd behaviour [4] or effectiveness in performing group tasks [1]. Collective behaviour is not a simple aggregation of individual independent behaviours, especially due to the human ability to synchronise their emotional state with the one of their peers. This phenomenon takes place through two mechanisms: empathy and emotional contagion [1]. Empathy is a high-level cognitive phenomenon, while emotional contagion is a reactive phenomenon described as "a process by which a person or group of people influence the emotions or behaviour of another person or another group by the conscious or unconscious induction of emotional states and behavioural attitudes" [21], encountered for example in crowds.

The computation of the emotional state of an agent depending on its perceptions has been studied extensively in the literature, but emotional contagion has not received the same attention. Furthermore, the literature on emotional contagion [2–5, 8, 23, 24] generally does not explain the underlying MAS architecture, leaving open the question of which multi-agent architecture has to be used to allow the introduction of massive simulations with sensory emotional agents.

In this article, we propose an hybrid architecture where a part of the emotional dynamics is delegated to the multi-agent environment, in accordance with the actual underlying mechanisms. This allows to alleviate a part of the agent complexity in terms of modelling and a part of the computation cost. Considering the latter, this hypothesis is based on the fact that a part of the calculus are repeated similarly in several agents, using at least partly the same data. Hence, using the environment enables the designer to reuse a part of the computation and reduce data duplication and transmission, since only the result is transmitted to the agents.

In Section 2, we detail the motivations for our emotional dynamics management architecture and discuss its impact on the autonomy of the agent. In Section 3, we introduce the architecture, MA/SDEC (*Mixed Agent/Social Dynamics for Emotion Computation*), and the corresponding formulas for emotion computation to illustrate our approach. In Section 4, we give the results of experiments to verify the properties of our model. In Section 5, we compare the efficiency of our MA/SDEC architecture to that of a full agent approach. Finally we discuss our approach and propose some perspectives in Section 6.

**Appears in:** *Alessio Lomuscio, Paul Scerri, Ana Bazzan, and Michael Huhns (eds.), Proceedings of the 13th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2014), May 5-9, 2014, Paris, France.*

## 2. RELATED WORKS AND MOTIVATION

There are two main architectural approaches for emotional contagion: agent-only approaches, and state-sharing approaches. Most of the articles describing the whole architecture (e.g. [14]) use an agent-only solution, transmitting the state of all agents to their neighbours and then calculating the emotional contagion in each agent. This solution has two limits: firstly, each agent has to display its emotional state and have the knowledge of contagion moderators to successfully compute the contagion result, and secondly similar calculus are done in every agent.

Broekens *et al.* [5] have compared several architecture for group emotions. In this work, the computational model of emotion is separated in three steps (appraisal, emotional state maintenance and emotional behaviour), and the authors show how the choice of which part of the computation is shared impacts both computation time and simulation quality. In the same way, mental states may be spread to obtain shared beliefs, emotions and group decision-making [4]. This approach enables to share efficiently computation costs and obtain consistent behaviour. However, this modelling requires sharing an important part of the agents' private states. Furthermore, in the case of explicit groups, group management adds several difficulties such as leader choice and group membership although it enables to determine which agents perceive which pieces of information.

Emotions evolve according to three influences [7]: one-off events, temporal dynamics and emotional contagion. Traditionally in agent modelling, all processes are integrated into the architecture of the agent, see e.g. [9]. If the evaluation of the impact of one-off events is necessarily managed by the cognitive process of the agent, we propose to decentralise the other processes in the software body of the agent and in the environment.

Although there is no consensus on the way emotions are processed in biological systems, many computational models have been proposed. In the following, we base our modelling on the thesis whereby the computation of emotions is the result of an intuitive (*appraisal*) and cognitive dual process [20]. The first is semi-automatic and often unconscious. It represents the change resulting from an immediate emotional percept and concerns the so-called primary emotions (such as joy). The second is a cognitive evaluation deriving from the consistency between beliefs, goals, and percepts of the agent and the emotions he feels, with emotions both primary and secondary (such as shame). As we mentioned in the introduction, the emergence of consistent collective behaviour requires the modelling of empathy and emotional contagion. If empathy requires a symbolic representation of the other, Hatfield *et al.* [10] showed that emotional contagion takes place at a significantly lower level of consciousness than empathy, via uncontrolled automatic processes.

In order to propose an adequate architecture for emotional contagion, we rely on two concepts: the active environment and the body/mind separation. The notion of explicit environment has long been associated with the reactive agent paradigm, but recent works [25] have shown the benefits of the use of this abstraction in the general framework of MAS. These studies highlight the interest to delegate some responsibilities of agents to the environment. In particular, the environment may be in charge of accessing and spreading a part of the agent states. In the context of emotion modelling, the environment can get the agents emotional states and compute the emotional contagion in their stead.

In the same logical way, we consider that the agent consists of two parts: its mind and its body (which may possibly be a software body) [18]. In this embodied agent framework, the mind contains the decision process of the agent and is autonomous, and the body is influenced by the mind, but controlled by the environment. This corresponds to human functioning: although the mind may take any arbitrary decision, the limits to the realisation of these decisions are imposed by both the body capacities and its environment rules. In practice, our proposal implies that the body states of the agent are observable and that their access is controlled by the environment, including for the agent itself. For the calculation of emotions, we propose that the perception of events is the responsibility of the mind of the agent, the temporal dynamics managed by the body and the emotional contagion by the environment.

Such modelling can be considered as violating the principle of agent autonomy. Quite the opposite, we believe it provides a clearer separation between the responsibilities of each of the system components, based on the mechanisms involved in the real world. Any agent is always situated in an environment (that can be software, real or simulated), and therefore an agent is never independent of it. One objective of the body/mind separation is to clearly delineate the agent autonomy between its mind (full autonomy) and the rest of the MAS (including actions and actions results).

## 3. MA/SDEC ARCHITECTURE

The Mixed Agent/Social Dynamics for Emotion Computation model is a high-level model which defines global mechanisms for emotion calculus and their dependencies. Emotions evolve in function of three influences [7]: punctual events, temporal dynamics and external influences. The first is the responsibility of the agent's mind, the second of the agent's body, and the third of the environment. The MA/SDEC model describes the dynamics and responsibilities of each MAS component, but does not rely on a particular representation of emotions and personality.

For each emotion $e$, the update formula is composed of three terms:

$$e_{t+1} = e_t + \Psi(b, i, p, e_t) + \Phi(p, e_t) + \Omega(p, e_t)$$

with

- $b$, $i$, $p$: beliefs, intentions and personality of the agent,
- $\Psi(b, i, p, e_t)$ the event dynamics: emotions evolve in function of the stimuli (stored in the belief set) and of its internal state,
- $\Phi(p, e_t)$ the internal dynamics: emotions tend to decay in function of the agents' personality traits towards an equilibrium,
- $\Omega(p, e_t)$ the external dynamics: emotions vary in function of the other agents and of the sensitivity of the agent.

In Figure 1, we give an overview of the architecture and how it relates to the associated model. The emotions are stored in the body of the agent. The events dynamics $\Psi$ are an influence of the mind on the body. The internal dynamics $\Phi$ are managed by the body itself. The emotional

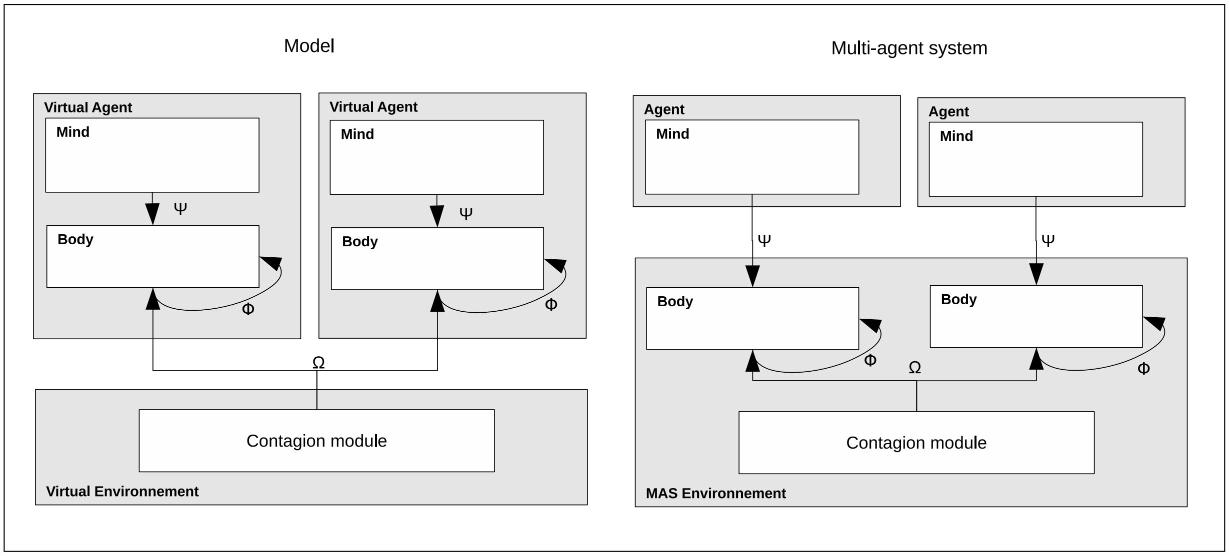



**Figure 1: MA/SDEC model and architecture**

contagion Ω is managed by the environment. In the multi-agent system, the agents are used to implement only the mind of the agents, while both the body and the virtual environment are managed by the MAS environment.

## 3.1 Agent's mind: Event Dynamics

Figure 2 shows a generic agent architecture with emotion support, such as [12] and [13]. The agent gets new information (perception, message and body) from the environment. This new information generates instant emotions through a primary emotion update function, and the agent changes its beliefs in function of its emotions.

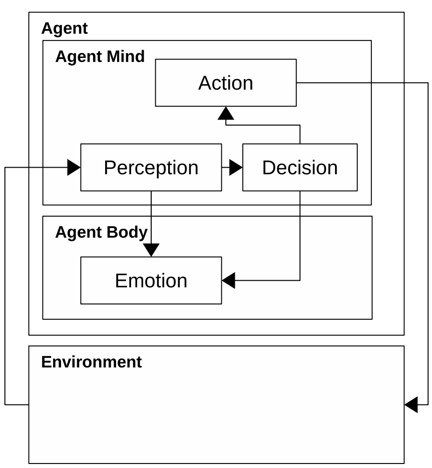



**Figure 2: Agent Architecture**

The selection of desires and intentions is similar to the classical BDI scheme except for the emotion and personality influence. Once intentions are selected, the agent updates its emotions through a secondary emotion update function. If this update modifies its emotions, it updates again its beliefs, desires and intentions. Finally, it plans its actions and executes its new plan.

The general function for emotion update is defined as:

$$\Psi : B \times I \times P \times E \rightarrow E$$

with $B$ the set of beliefs, $I$ the set of intentions, $P$ the set of personalities and $E$ the set of emotions.

We have proposed an agent architecture that illustrates this BDI scheme and manages emotions in [13]. In this work, the perception and emotion computation are processed thanks to fuzzy rules.

## 3.2 Agent's body: Internal Dynamics

Internal temporal dynamics (Figure 3) are managed by the agent itself or by the environment via the body of the agent. It represents the tendency of emotions to stabilise over time. A second module inside the body allows the temporal control of emotions dynamics. It limits emotions variation in order to make smoother state modification for the agent. This module limits the oscillation risk in case of contrary stimuli.

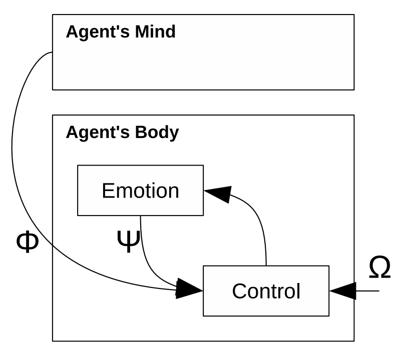



**Figure 3: Temporal dynamics**

Several authors have observed that emotions tend to decay over time, either towards a neutral state [6,24], or towards a baseline [20] which depends on the personality of the agent. Since the equation depends on the emotion representation, it has rarely been made explicit in the literature. In [7], emotions are tri-modal ($\{-1, 0, 1\}$) and the emotion decay parameter represents the number of time steps before returning to a neutral state if no event impacting this emotion occurs in the meantime. However, this discrete representation does not fit fine-grained emotion simulation.

For emotions represented as values in $[-1, 1]$, the emotion variation is calculated as:

$$\begin{aligned} \Phi &: P \times E \to E \\ \Phi(p, e_t) &= (1 - \alpha_e)e_{base} + (\alpha_e - 1)e_t \end{aligned}$$

with $e_{base}$ the personality-based emotion baseline, $e_t$ the emotion level and $\alpha_e$ the decay speed parameter for emotion $e$. The same formula manages the internal dynamics of all emotions, parameters are set for each agent according to their personality traits.

The control module limits emotional fluctuations from one step to another. It allows emotions stabilisation and smooth transitions. The $\Gamma$ function of the control module is:

$$\begin{aligned} \Gamma &: E \to E \\ \Gamma(\delta_e) &= \begin{cases} \delta_e \text{ if } |\delta_e| < \sigma \\ sgn(\delta_e)\ \sigma \text{ otherwise} \end{cases} \end{aligned}$$

Function $sgn$ gives the sign of a real number. If the modification of the emotional state $\delta_e = e_{t+1} - e_t$ is greater than a threshold $\sigma$, then, this modification is limited by $\sigma$.

## 3.3 Environment: Emotional Contagion

Emotional contagion allows agents to be influenced by other agents states. Spatial and/or psychological proximity is mandatory for emotional contagion. The emotion propagation manager is a module of the MAS environment (Figure 4). It updates cyclically agent's bodies states. It gets (1) the current state of the agent, here $a_1$. It updates (2) accordingly its state of the world. The state of the world contains the body properties of all the agents. Then, the emotion propagation manager calculates the effects of emotion propagation on the agents' neighbours in function of their previous state and of their tendency to empathy. Finally, the MAS environment spreads (3) these into the concerned agents' bodies, $a_2$ in our example.

The emotion contagion calculus is inspired from several works in the modelling of agents influences on each other. A majority of contagion models derive from [1], considering the following factors as impacting the contagion strength [3,4,14]: the level of the sender's emotion, the sender's emotion expression, the receiver's openness for received emotion and the strength of the channel from sender to receiver. We simplify this approach by using the physical distance to qualify the strength of the emotion contagion:

$$\begin{aligned} \Omega : P \times E &\to E \\ \Omega(p, e_t) &= \delta_R \times \gamma_R \end{aligned}$$

with $\delta_R$ the receiver agent openness and $\gamma_R$ the influence of the other agents on agent $R$. The agent openness can be derived from personality traits (Agreeableness, Openness and Extraversion) of the Big Five model [14].

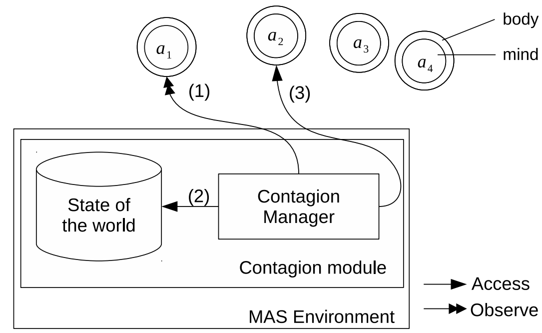



**Figure 4: Environment emotion propagation module and agents' interactions**

The influence $\gamma_R$ is defined as inversely proportional to the distance between the agents:

$$\gamma_R = \sum_{\forall A \neq R | dist(A,R) < \tau} (e_A - e_R) \times \frac{\beta}{dist(A, R)}$$

with $e_A$ the emotion level of agent $A$ and $dist(A, R)$ the Euclidean distance between $A$ and $R$. $\tau$ and $\beta$ are parameters used to define respectively the maximum emotion perception distance and the influence strength. Using a circle of influence centred on the agent is a simplification : an agent perceives the emotions of all agents situated in this circle, including agents it may not be able to see (e.g. behind it).

Let us notice that in case of sequential computation of the emotional contagion, the order of computation impacts the result. In practice, it implies that we update simultaneously all the states based on the previous states, in order not to introduces biases in the results.

# 4. EXPERIMENTAL PROPERTIES

In this article, our goal is not to validate the formulas, but to examine the properties of our architecture, and the effect of each module. Hence, we have chosen to separate the validation of the mechanism from the study of its properties. The first experiment aims at verifying the mechanism by comparing it to data published in a social psychology study on little groups. In the following section, the second set of experiments is designed to show the properties of the model in terms of run time.

## 4.1 Reference Data and Parameters

Computational models of emotional contagion are all based on the seminal work of Barsade [1]. In this work, the author did a series of experiments on groups of 3 to 5 students, who do not know each other beforehand, and have to achieve a semi-cooperative task. One of the students was an actor who had to act constantly during the exercise a particular emotional state (joyful/high energy, joyful/low energy, sad/high energy, and sad/low energy). The emotional state of the other individuals of the group was then scored in two ways: a questionnaire answered by the participants, and video analysis of their behaviour. We propose to verify our model thanks to the qualitative and quantitative data given by Barsade in his article.

In Barsade's experiments, the students do not know each other, and the distance is not relevant. Hence, we initialise the distance between all agents as a constant which is static throughout the experiment. The parameters are initialised as shown in Table 1. We consider that the emotional level of the actor does not change during the experience. We have run experiments using the MadKit[1] general-purpose multi-agent system platform.

| Parameter | Domain | Default value |
|---|---|---|
| $\delta_R$ | $[0.75, 1.25]$ | Random uniform |
| $e$ | $[-1, 1]$ | Random uniform |
| $e_{base}$ | $[-1, 1]$ | Random uniform |
| $\alpha_e$ | $[0, 1[$ | 0.9 |
| $\beta$ | $\mathbb{R}^+$ | 1 |
| $\sigma$ | $[0, 1]$ | 1 |

**Table 1: Simulation parameters**

## 4.2 Verification

In this first experiment, we study the emotion evolution pattern over time. The simulated actor is an attractor, since it stimulates constantly the other agents. Figure 5 illustrates this experiment with a group of 5 agents. The emotional level is normalised on a 1-9 scale (instead of [-1,1]), in order to be consistent with Barsade. We observe that the emotional levels of the agents change toward equilibriums around the actor's state. The states of the agents do not converge to the same equilibrium, since these depend on the $e_{base}$ parameter, which are respectively 5.8, 6.6, 1 and 1.8 for the agents 1, 2, 3 and 4.

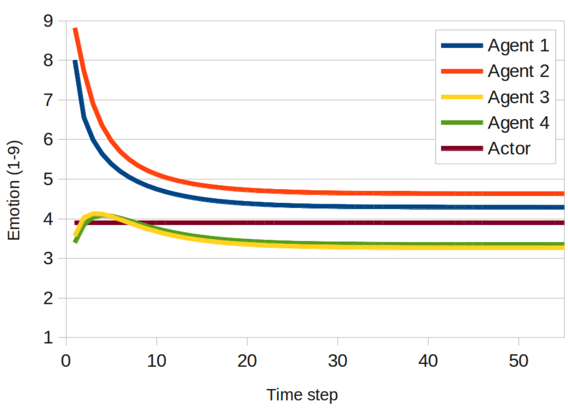



**Figure 5: Emotional convergence in a group of 5 agents.**

With respect to the previous studies of Bosse *et al.* [3], Lhommet *et al.* [14] and Coenen and Broekens [6], we find the same pattern, with the difference of the internal dynamics, which implies that these authors find a convergence in one equilibrium state of all the agents perceiving the emotions. Let us study the extremums. For $\alpha = 1$ the temporal dynamics is cancelled, hence the curves converge. For $\alpha = 0$ the temporal dynamics modifies the emotion level to its baseline in one step, so that the other stimuli (events and/or contagion) have to be constant to modify this level.

[1]http://www.madkit.org

Our pattern is consistent with the results of Barsade, who found both emotional convergence and a high disparity of individual emotions (the standard deviation of the questionnaire results varies between 0.99 and 1.16 on 5-points scale). It implies in our model the equilibrium of external and internal dynamics.

The $\Gamma$ function modifies the emotional states variations. Hence, lowering $\sigma$ causes a lengthening of the convergence period, but doesn't impact the curve shape. It allows the representation of emotional changes that take into account emotional contagion but are not immediate. This behaviour, which exists even if $\Gamma$ function is removed, is a phenomenon of *hysteresis*, i.e. a dynamic lag between cause and effect.

To illustrate this phenomenon, we conducted an experiment in which the stimulus is not constant. An agent changes state between two modalities ($-0.5$ and $0.5$) at a rate of 0.05 per time step, then waits for the stabilisation of the other agents before changing modality again.

The corresponding hysteresis loops are plotted in Figure 6. The studied causal relationship is the effect of the gap (the difference between the emotional state of the stimulating agent and the emotional state of the agent studied) on the emotional state. Both equilibriums shown for the agent 3 are stable states, when the stimulating agent is respectively at $-0.5$ and $0.5$. The parts of the curves that are outside the loops are those corresponding to the initial situation, when the agents first join their stable state. The direction of the transitions between the two stable states is noted in the figure.The gradient of the loop increases as the stimulus (here represented by the gap) is stronger, taking a more and more significant delay. The gradient then reaches the $\sigma$ limit given by the $\Gamma$ function. Then, when the emotional state of the agent causing the stimulation ceases to decrease, the state of the stimulated agents joins the stable state, which depends on each agent's personality and on the external dynamics.

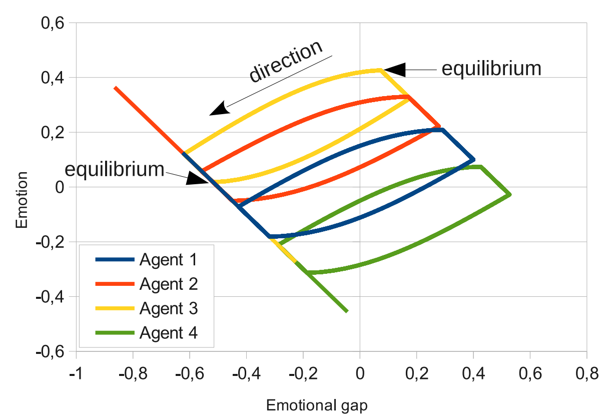



**Figure 6: Hysteresis loops in a group of 5 agents with $\sigma = 0.1$.**

# 5. EFFICIENCY OF THE MA/SDEC ARCHITECTURE

In this section, we compare the cost in computation time of our MA/SDEC architecture with a purely agent solution. For this purpose, we implemented a C++ simulator, allowing a better memory use control than Java (MadKit). This agent-based simulator is pseudo-parallel (hence sequential) and centralised in order to compare the total execution time without taking into account thread management and communication costs.

In the agent solution, in order to avoid systematic spread of all information to all agents, the environment delivers to each agent the emotion status of other agents located in their perception distance. We consider the perception distance and the propagation distance $\tau$ to be the same, *i.e.* the minimum to implement the emotional dynamics while not creating undue overhead. The agents then use this information to update their emotions using the same formula of emotional contagion, once per time-step.

The initialisation of the simulation parameters are the same as for the previous series. The agents are situated on a two-dimensional discrete space and do not move throughout the simulation. Simulation ends when the emotional state of all agents is stable. Each simulation is run 100 times. The same pseudo-random values generator seeds are used to initialise the two simulations (computation by the environment and by the agents) in order to check the comparability of situations. Since the information used to compute emotional contagion is the same, the two solutions produce the same result at each time step.

We compare the performance of these two architectures in terms of run-time in function of the number of agents and of the perception distance.

## 5.1 Number of agents

Figure 7 shows the execution time of the whole simulation until stabilisation for a number of agents ranging from 25 to 1521. The computation time is always lower using the environment than through the agents. The time savings associated with the use of the environment is 52% for 144 agents and only 11% for 1521 agents.

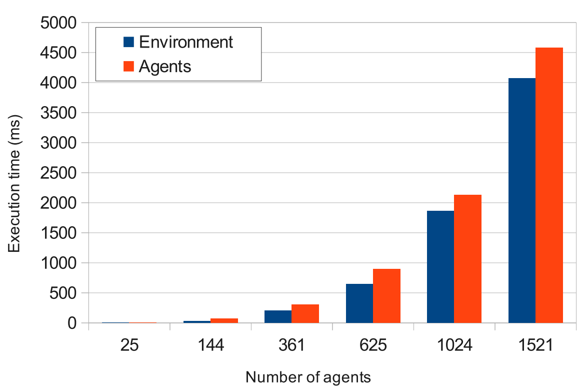



**Figure 7: Total run-time in function of the number of agents**

In value terms, the gain increases with the size (36 ms for 144 agents to 503 ms for 1521 agents), but the portion of this cost in the overall execution time is low compared to the overload caused by the number of agents, regardless of the emotional contagion method used.

As noted in the introduction of this section, the number of emotion contagion updates is the same in both solutions, as is the internal dynamics calculus (once per time-step). To explain our experimental result, the two main differences between the agent-based approach and the environment-based approach are :

1. the amount of data transmitted between the environment and the agents. In the agent-only solution, all the agents have to perceive the emotional states of each of their neighbours to calculate their effect on its own emotional state, while the environment gets once the emotional state of each agent.

2. the emotional contagion calculus, specifically the neighbours influence $\gamma_R$. In the agent-only solution, all the agents calculate independently all their neighbours influences. The environment can reuse a part of the computation in mutual influences, *e.g.* between agent A and B, $\gamma_A$ from B to A is equal to $-\gamma_B$ from A to B.

## 5.2 Perception distance

We then focus on the effect of the perception distance on the execution time. The results are summarised in Figure 8. Figure 8(a) shows two trends: up to 50, the execution time lowers, then it increases sharply for $\tau$ equal to 60. The general shape of the curves corresponds to the mechanism itself: at the beginning, the more $\tau$ increases, the more the simulation stabilises quickly. This is because the calculation is less local, and the agents immediately take into account more of their neighbours. This effect has an impact on the final proportions of influenced agents (when $\tau$ increases, the emotional contagion is logarithmically stronger). Once above a threshold, here 50, the execution time increases sharply. This is explained by a bigger difficulty of stabilisation when each agent is influenced by many other agents (about 25 agents for $\tau = 50$), which are themselves subject to other influences.

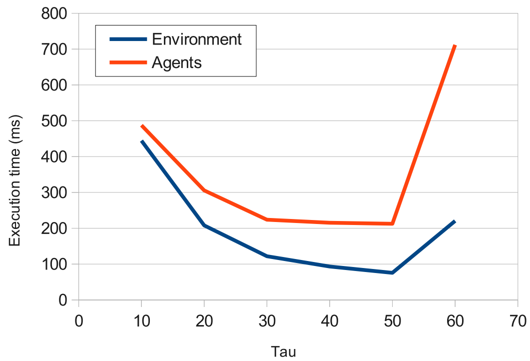
Because the first curve does not allow an easy comparison between the two architectures, we generated a second figure (Figure 8(b)) with standardised results (100% is the execution time of the longest simulation). Regarding the comparison between the environment computation and the agents computation, we observe that the more $\tau$ increases, the more it is interesting to perform the calculations with the environment. Predictably, the computation time is directly related to the size of information exchanged and computed between agents.

As for the experiments on the number of agents, it is always more efficient to make the computation performed by the environment than by the agents. The gain ranges from 8.5% for $\tau = 10$ to 69% for $\tau = 60$.

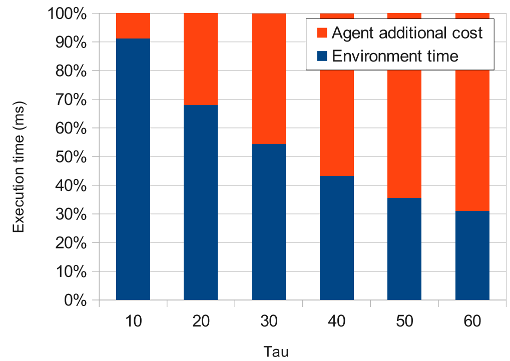
# 6. DISCUSSION AND CONCLUSION

In this paper, we proposed to calculate the emotional dynamics within a multi-agent architecture. This mechanism is based on three dynamics: event, temporal and external. Events impact the emotions depending on the internal state of the agent and its perception of the event. Temporal phenomenon represents the dynamic stabilisation of emotions over time. External dynamic is the emotional contagion between agents.

The calculation of a part of the internal state of the agent based on its equivalent in the internal state of other agents




(a) Run time in function of $\tau$




(b) Normalized computation cost via the environment and additional cost via the agents

**Figure 8: Total run time in function of $\tau$**

has been used in other social simulations based on reactive agents. For example, the model satisfaction/altruism [15] in which the spread of agents' state improves cooperation and conflict resolution between agents. The IRM4S model [16] is an adaptation of an influence-reaction model for simulation, the body responsibilities being separated between the agent and the environment.

Several other frameworks for improved environment and body architectures have been proposed [17,18,25]. Platon et al. [18] have introduced the concept of over-sensing: agents have soft-bodies that have public states, which are mediated (both for visibility and accessibility) by the environment. The information on modifications to the public states is spread throughout the environment. Nevertheless, the model does not explicitly take architectural issues into account, and even if the agents are observable, the body model is in fact empty, relying on the environment for the calculus. Programmable tuple-spaces such as TuCSoN [17], or artifacts [19] could be a way to implement the MA/SDEC architecture, by programming the body and environment responsibilities as reactions.

About the MA/SDEC architecture, compared with a calculation in each agent, this modelling of embodied agent thanks to the environment has two advantages: first, the agent architecture is focused on high-level decisions, while the environment manages a part of the complexity of the agent regarding the low-level mechanisms. Then, regarding the computational cost, the encapsulation of this service in the environment reduces the overall cost. It permits to share part of the calculations, instead of recalculating them in each agent.

Our simulations showed a gain in run-time execution (depending on the parameters and the size of the MAS) of the approach via the environment in comparison with purely agent approach. However, using the environment can create a bottleneck if the environment is not itself distributed. It is important to note that delegating part of the computation to the environment (considered as an architectural abstraction) does not necessarily imply a centralisation of execution. In the case of the simulation of a physical space, it can be divided into areas distributed across multiple hosts where synchronisation is facilitated by the introduction of a distance perception.

The bodies of the agents can be controlled by the environment in which services are in the MAS platform. However, some authors propose to create software body that are not part of the environment, e.g. [22] in order to control the animation of conversational agents. Such a component could be used to manage the temporal dynamics of emotions, used as an interface with the mechanism of emotional contagion and emotional variations control mechanism. It would also allow a looser coupling between agents and environment.

The MA/SDEC architecture describes the dynamics and responsibilities of each MAS component, but does not rely on a particular representation of emotion and personality. Although we propose functions for the dynamics of the emotions, those can be replaced for particular settings, for example groups with known social topology. Hence any group contagion model such as [3] can be used to manage external dynamics. The performance gain of our architecture then depends on whether parts of the computation can be reused for several agents.

Further works include the study of the sensibility of the model to other parameters. A recent review of psychological studies [6] has shown the existence of moderating factors of emotional contagion, such as social power or gender, which were simplified in this article. From the architecture viewpoint, these moderators should be included in the bodies (for individual moderators) and environment (for social moderators). Furthermore, we plan to replicate other psychological phenomena such as the impact of emotional contagion on cooperative decision-making, where the interplay with higher cognitive functions is more complex.

Finally, the approach based on embodied agents, in which the relationship between mind, body and environment are strictly formalised, should facilitate the modelling of human processes. In particular, the choice of removing from the control module (*i.e.* from the mind / autonomous agent) some low level calculations, such as emotional contagion or the physical aspects, can help simplifying its design. More research is needed to better understand how MAS designers can apply this principle to all cases where the agents are situated and can therefore interact with an environment, and propose an adequate methodology.

# 7. REFERENCES


[1] S. Barsade. The ripple effect: Emotional contagion and its influence on group behavior. *Administrative Science Quarterly*, pages 644–675, 2002.

[2] T. Bosse and F. P. J. de Lange. Estimating emotion regulation capabilities. In *Proceedings of the 1st international conference on PErvasive Technologies Related to Assistive Environments*, pages 93:1–93:8, New York, NY, USA, 2008. ACM.

[3] T. Bosse, R. Duell, Z. Memon, J. Treur, and C. Van Der Wal. A multi-agent model for mutual absorption of emotions. In *Proceedings of the 23rd European Conference on Modelling and Simulation, ECMS*, pages 212–218. Citeseer, 2009.

[4] T. Bosse, M. Hoogendoorn, M. C. Klein, J. Treur, C. N. van der Wal, and A. van Wissen. Modelling collective decision making in groups and crowds: Integrating social contagion and interacting emotions, beliefs and intentions. *Autonomous Agents and Multi-Agent Systems*, 27(1):52–84, 2013.

[5] J. Broekens, N. Netten, and D. DeGroot. Consistent dynamic-group emotions for virtual agents. In *Proceedings of the 16th Belgium-Netherlands Conference on Artificial Intelligence*, 2005.

[6] R. Coenen and J. Broekens. Modeling emotional contagion based on experimental evidence for moderating factors. In *Proceedings of Cognitive Agents in Virtual Environments (CAVE), Workshop held at AAMAS'12*, pages 26–33, 2012.

[7] A. Czaplicka, A. Chmiel, and J. A. Holyst. Emotional agents at the square lattice. *ACTA PHYSICA POLONICA A*, 117:688–694, 2010.

[8] A. García-Rojas, M. Gutiérrez, and D. Thalmann. Simulation of individual spontaneous reactive behavior. In *Proceedings of the 7th international joint conference on Autonomous agents and multiagent systems*, pages 143–150, 2008.

[9] J. Gratch and S. Marsella. A domain-independent framework for modeling emotion. *Cognitive Systems Research*, 5(4):269 – 306, 2004.

[10] E. Hatfield, J. Cacioppo, and R. Rapson. *Emotional contagion*. Cambridge Univ Pr, 1994.

[11] R. Imbert, A. De Antonio, et al. Agents that combine emotions and rationality: a context independent cognitive architecture. *WSEAS Transactions on Computers*, 4(9):1202–1209, 2005.

[12] H. Jiang, J. M. Vidal, and M. N. Huhns. Ebdi: an architecture for emotional agents. In *Proceedings of the 6th international joint conference on Autonomous agents and multiagent systems*, pages 11:1–11:3, New York, NY, USA, 2007. ACM.

[13] H. Jones, J. Saunier, and D. Lourdeaux. Fuzzy rules for events perception and emotions in an agent architecture. In *EUSFLAT 2011*, Advances in Intelligent Systems Research, pages 657 – 664. Atlantis Press, 2011.

[14] M. Lhommet, D. Lourdeaux, and J.-P. Barthès. Never alone in the crowd: A microscopic crowd model based on emotional contagion. In *Web Intelligence and Intelligent Agent Technology (WI-IAT), 2011 IEEE/WIC/ACM International Conference on*, volume 2, pages 89 –92, aug. 2011.

[15] P. Lucidarme, O. Simonin, and A. Liégeois. Implementation and evaluation of a satisfaction/altruism based architecture for multi-robot systems. In *IEEE International Conference on Robotics and Automation (ICRA'02)*, volume 1, pages 1007–1012. IEEE, 2002.

[16] F. Michel. The irm4s model: the influence/reaction principle for multiagent based simulation. In *Proceedings of the 6th international joint conference on Autonomous agents and multiagent systems*, pages 903–905, 2007.

[17] A. Omicini and F. Zambonelli. Tuple centres for the coordination of internet agents. In *SAC '99: Proceedings of the 1999 ACM symposium on Applied computing*, pages 183–190. ACM Press, 1999.

[18] E. Platon, N. Sabouret, and S. Honiden. Tag interactions in multiagent systems: Environment support. In *Proceedings of Environment for Multi-Agent Systems, Workshop held at the Fifth Joint Conference in Autonomous Agents and Multi-Agent Systems*, volume 4389 of *Lecture Notes in Artificial Intelligence*, pages 106–123. Springer Verlag, 2007.

[19] A. Ricci, M. Piunti, M. Viroli, and A. Omicini. Environment programming in cartago. In *Multi-Agent Programming:*, pages 259–288. Springer, 2009.

[20] F. Rosis, C. Castelfranchi, P. Goldie, and V. Carofiglio. Cognitive evaluations and intuitive appraisals: Can emotion models handle them both? *Emotion-Oriented Systems*, pages 459–481, 2011.

[21] G. Schoenewolf. Emotional contagion: Behavioral induction in individuals and groups. *Modern Psychoanalysis*, 15(1):49–61, 1990.

[22] M. Thiebaux, S. Marsella, A. Marshall, and M. Kallmann. Smartbody: Behavior realization for embodied conversational agents. In *Proceedings of the 7th international joint conference on Autonomous agents and multiagent systems-Volume 1*, pages 151–158, 2008.

[23] J. Tsai, N. Fridman, E. Bowring, M. Brown, S. Epstein, G. Kaminka, S. Marsella, A. Ogden, I. Rika, A. Sheel, et al. Escapes: evacuation simulation with children, authorities, parents, emotions, and social comparison. In *The 10th International Conference on Autonomous Agents and Multiagent Systems-Volume 2*, pages 457–464, 2011.

[24] L. Van Minh, C. Adam, R. Canal, B. Gaudou, H. Tuong Vinh, and P. Taillandier. Simulation of the emotion dynamics in a group of agents in an evacuation situation. *Principles and Practice of Multi-Agent Systems*, pages 604–619, 2012.

[25] D. Weyns, A. Omicini, and J. Odell. Environment as a first-class abstraction in multi-agent systems. *Autonomous Agents and Multi-Agent Systems*, 14(1):5–30, Feb. 2007. Special Issue on Environments for Multi-agent Systems.