Using Emotion as Motivation in the Newtonian Emotion System

Valentin Lungu, Andra Băltoiu, and Șerban Radu

Abstract. The main goal of this work has been to develop an emotion simulation model and agent architecture that provides artificial characters in virtual environments with believable behavior in order to enhance virtual environment experiences for human users; this means that emotions should act as motivation for agent behavior, influence and improve perception, reasoning and decision-making.

We put forth the Newtonian Emotion System, a representation model vectorial in nature, easily integrated into computational environments and firmly grounded in Pluthcik's theory of emotion. The system functions according to psychological theory, influencing the way that a character perceives the environment. The system does not influence reasoning. However it does influence the agent's actions through the **motivation mechanism**, which we consider vital for the agent to have believable affective states.

1 Introduction

The main goal of our work is to provide virtual characters with the ability of emotion expression. Several attempts have been made to endow artificial agents with an emotional layer [10, 9, 14, 2, 3]. However, we believe that in order for artificial agents to have deep, meaningful and believable emotions, they need their emotional layer to serve a similar function to that of its natural human counterpart.

Emotion in humans has been strongly linked to motivation. Emotions are a part of the human evolutionary legacy [1, 11, 13, 4, 12], serving adaptive ends, acting as a heuristic in time-critical decision-processes (such as fight-or-flight situations). We believe that emotions act as a subsystem that enhances human behavior, by stepping up brain activity in arousing circumstances, directing attention and behavior,

Valentin Lungu · Andra Băltoiu · Șerban Radu
University POLITEHNICA of Bucharest, Bucharest, Romania
e-mail: `valentin.lungu@cs.pub.ro`
`http://aimas.cs.pub.ro/people/valentin.lungu/`

F. Zavoral et al. (eds.), *Intelligent Distributed Computing VII*,
Studies in Computational Intelligence 511,
DOI: 10.1007/978-3-319-01571-2_41,

establishing importance of events and act as motivation. The Newtonian Emotion System [5, 6, 8, 7] that we developed will influence agent behavior by establishing the importance of events and by influencing knowledge processing, as well as provide the agent with an emotional state that it will be able to express and that will further influence its behavior.

2 The Newtonian Emotion System

This chapter presents our new improved emotion simulation technique based off of Plutchik's emotion taxonomy. However, the emotion representation scheme has been streamlined, the emotion feature being reduced to a four-dimensional vector, and emotion interactions follow Newtonian interaction laws that are easier to learn and understand. The architecture has also been reduced to the bare minimum necessary for emotion simulation and expression while still influencing character behavior and memory in the same way. The new architecture allows for any behavior generation technique (belief-desire-intention, expert systems, behavior trees, finite state machines) and any machine learning algorithm (SVM, linear regression, clustering), in essence, allowing designers to develop a system that suits them best, while incoroprating emotion simulation.

2.1 Newtonian Emotion System Space

In this section we introduce the Newtonian Emotion Space, where we define concepts that allow emotional states to interact with each other and external factors, as well as the two laws that govern the interaction between an emotional influence and an emotional state. The Newtonian Emotion Space is based on the work of R. Plutchik, shown in Fig. 1.

Laws of Emotion Dynamics

The following are two laws that form the basis of emotion dynamics, to be used in order to explain and investigate the variance of emotional states within the emotional space. They describe the relationship between the forces acting on an emotional state and its motion due to those forces.

Theorem 1. *The velocity of an emotional state remains constant unless it is acted upon by an external force.*

Theorem 2. *The acceleration* $\mathbf{a}$ *of a body is parallel and directly proportional to the net force* $\mathbf{F}$ *and inversely proportional to the mass* m*:* $\mathbf{F} = m \cdot \mathbf{a}$

Emotion Center and Gravity

The emotion space has a center, the agent's neutral state, a point in space to which all of the agent's emotional states tend to gravitate (usually (0,0,0,0), however, different characters might be predisposed to certain kinds of emotions, thus, we should be

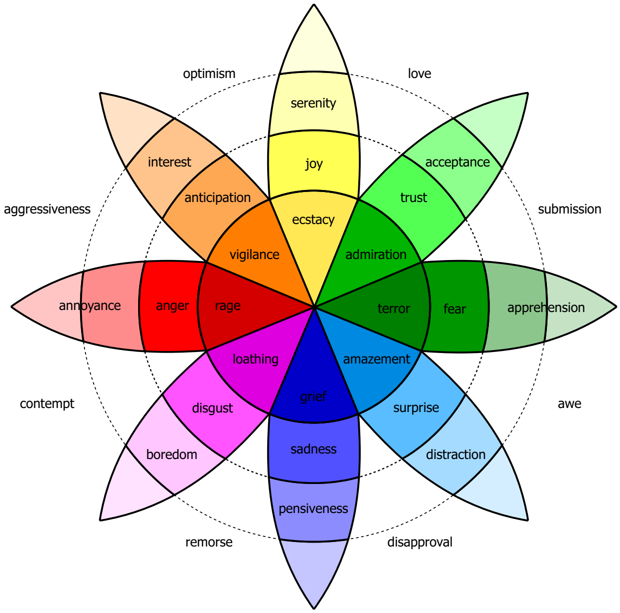

Fig. 1 Newtonian emotion space

able to specify a different emotional centre, and instill a certain disposition, for each character; we also use different emotion state mass to show how easy/hard it is to change an agent's mood). In order to represent this tendency we use gravitational force:

$$\mathbf{G} = m \cdot \frac{\mathbf{p}-\mathbf{c}}{||\mathbf{p}-\mathbf{c}||} \cdot k_g,$$

where p is the current position, c is the center and k_g is a gravitational constant. This force ensures that, unless acted upon by other external forces, the emotional state will decay towards the emotion space center.

2.2 Newtonian Emotion System Architecture

We developed an emotion subsystem architecture (Fig. 2) that interposes itself between the agent's behavior module and the environment. The subsystem models the process described by Lazarus.

Events perceived from the environment are first processed by the appraisal module, where an emotional force is associated with it. The resulting list of events is then sorted in descending order according to magnitude and fed into the agent's perception module. This is done in accordance with the psychological theory of attention narrowing, so that in a limited attention span scenario, the agent will be aware of the more meaningful events.

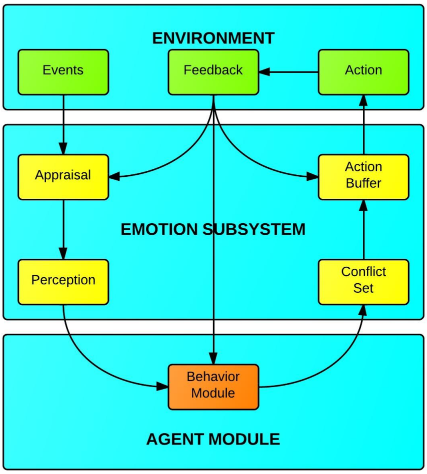

Fig. 2 Emotion system architecture

We treat the agent behavior module as a black box that may house any behavior generation technique (rule based expert system, behavior trees, etc.). The interface receives as input events perceived from the environment and produces a set of possible actions. The conflict set module takes this set of rules and evaluates them in order to attach an emotional state to each. Out of these, the action whose emotion vector most closely matches the agent's current state is selected to be carried out.

$$\underset{i \in \{conflictset\}}{\arg\min} \quad arccos \frac{e_{agent} \cdot e_i}{||e_{agent}|| \cdot ||e_i||}$$

Feedback from the environment has an associated emotion force vector that actually affects the agent's emotional state. Feedback is distributed to the appraisal and conflict set modules as well as the agent's behavior module. Actions undertaken are temporariliy stored in an action buffer for feedback purposes.

2.3 Test Scenario

Feedback from the environment has an associated emotion force vector that affects the agent's state. The agent attempts to maximize its Gain (joy and truth axes) and minimize its Risk (fear and surprise axes), and thus will choose actions that increase one while decreasing the other; however, the agent does not know what the feedback will be. It is the learning's module task to attempt to predict it, based on the current context. The better the prediction, the better informed a decision the agent can make.

We chose to use linear regression to predict feedback in our application. Linear regression is a way of modelling the relationship between a dependent variable and one or more explanatory variables. The data is modelled using a linear combination of the set of coefficients and explanatory variables. Linear regression is a supervised machine learning technique that estimates unknown model parameters from the data.

In our case, the algorithm attempts to learn the feedback method explained. The dependent variable is the emotional force received by the agent, while the explanatory variable is a feature vector constructed based on the action's context (Fig. 3). The structure of this array is made up of three segments:

- **Base.** Contains the basic information about the action.
- **Agents.** Contains information about the agents involved in the action.
- **Objects.** Contains information about the objects (other than equipment, i.e. containers, traps) involved in the action.

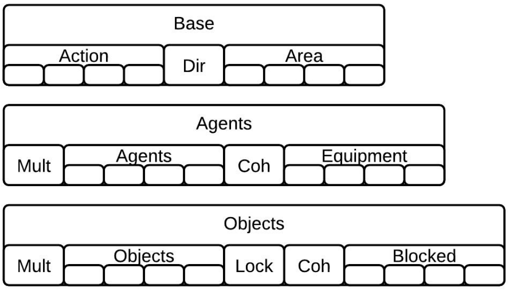

Fig. 3 Action context

The algorithm then receives the correct feedback from the environment which modifies the weights in the algorithm according to the difference between the estimated feedback force and the actual feedback force.

2.4 Learning in the Newtonian Emotion System

Both appraisal (percept and action appraisal) modules use machine learning techniques in order to learn to predict the outcome of events and actions, respectively. As there are many viable options for which learning technique to use (hidden markov models, neural networks, q-learning, POMDP), we use a plug-and-play model where the technique can be replaced [7]. The appraisal module attempts to predict the emotional feedback received from the environment based on characteristics of the event to classify and previous feedback. The goal of the appraisal module is to better label (and thus establish priority of) events for the perception model. The conflict set module works in a similar way based on the characteristics of actions taken. The goal of the conflict set module is to settle conflicts between competing actions by selecting the most appropriate action to be performed. The action that has the greatest Gain-Risk ratio will be chosen. Both modules treat the event, rule respectively, configuration as the observable state of the model, and attempt to predict the next hidden state (the emotional feedback force). Let us take the following example: character A's behavior module generates as the next action, an attack on character B. The following simplified feature vector for the action is built, that encodes information about the current context (feelings about the action, feelings about our target and action direction). From previous experience, the agent knows that the action would raise it's aggression level (feelings about the attack action are on the anger and anticipation axes [0,0,-.9,-.3]); we also assume that we are somewhat afraid of the character that we are attacking [0,0,.7,0], and we know that we are performing the action, so action direction is 1. Based on this feature vector, the appraisal module predicts an emotional feedback with a dominant on the anger and anticipation axes of [0,0,-.2,-.2]. We assume this action has a satisfactory Gain/Risk ratio and gets selected and executed. We also assume that it succeeds. According to the environment feedback formula we used, the result would be [0,0,-.2,-.3] which means that the estimation was close, but could be better. The feedback result is added to the learning module data set, and the results are nudged in the direction of [0,0,-.2,-.3] - [0,0,-.2,-.2] = [0,0,0,-.1], for a better future prediction. The perception appraisal module works in a similar way to appraise perceived actions, however the percepts are sorted according to a factor based on their distance to the agent's current emotional state (minimum) or their distance to the average emotional state of events perceived this turn (maximum). This is so that an agent will prioritize events that it resonates with or that stand out from the norm.

2.5 Personality Filter

The personality filter allows an agent's designers to skew the agent's perception of events. It defines a perception in interpreting emotional forces when applied to feedback received from the environment. The personal filter consists of a four dimensional vector (each element matching an axis of the Newtonian Emotion representation scheme) with values between -1 and 1 that scales the agent's emotional feedback according to personal criteria. For example, an agent may feel fear more accurately than joy and as such, its personal filter should reflect this, having a lower scaling factor for joy than for fear. Another agent may be extremely slow to anger, this would be reflected in the personal filter as a very low scaling factor for the anger axis. It would even be possible to change the sign altogether so that some agents may feel anger instead of fear or joy instead of sadness (leaving room for the representation of psychologically damaged individuals).

3 Emotion as Motivation

The emotional feedback that an agent receives from the environment influences its behavior. This is achieved by selecting an action from the conflict set provided by the behavior model according to the feedback that the agent predicts from the environment.

In order to evaluate its effect on the agent's current state, we put forth the notion of tensor product between two states, because of the information this gives about the influence that one axis has on another. This product has an interpretation as stress produced by an emotional force on the current emotional state, showing how the force is, relative to the current state of the agent. This is called the emotional impact and the tensor product is computed between the two forces acting on the agent: the Gravity force, pulling the agent towards its center, and the learned action force (estimated feedback).

$$\begin{aligned} Stress &= [Feedback] \cdot [Gravity] \\ &= \begin{bmatrix} f_{joy} \\ f_{trust} \\ f_{fear} \\ f_{surprise} \end{bmatrix} \cdot \begin{bmatrix} g_{joy} & g_{trust} & g_{fear} & g_{surprise} \end{bmatrix} \end{aligned} \tag{1}$$

This allows the assessment of the influence that each emotion axis of the action has on those of the current state. In line with Lazarus' appraisal theory, emotional evaluation implies reasoning about the significance of the event, as well the determination of the ability to cope with the event. Therefore we define two metrics for assessing emotional experiences. Well-being measures the effect the action has on the positive axes (Joy-Trust) of the current state, thus establishing the gain implied by the action; while Danger describes the risk, as it evaluates the action in terms of its influence on the Fear-Surprise axes.

When choosing an action, an agent estimates how much its current well-being is influenced by the action in regard to how much danger the action implies, a gain-risk ratio. In terms of the tensor product, this involves a column-wise computation with respect to the Well-being axes of the agent's current state (for Gain) and a row-wise computation with respect to the Danger axes of the action (for Risk).

$$\begin{aligned} Gain = {} & g_{joy} \cdot (f_{joy} + f_{trust} + f_{fear} + f_{surprise}) \\ & + g_{trust} \cdot (f_{joy} + f_{trust} + f_{fear} + f_{surprise}) \end{aligned} \tag{2}$$

$$\begin{aligned} Risk = {} & f_{fear} \cdot (g_{joy} + g_{trust} + g_{fear} + g_{surprise}) \\ & + f_{surprise} \cdot (g_{joy} + g_{trust} + g_{fear} + g_{surprise}) \end{aligned} \tag{3}$$

Thus the *Impact* factor can be used to assign priorities to all actions in order to resolve the conflict set.

$$Impact = Gain - Risk \tag{4}$$

Let's assume that a character's current state is [.3,.5,-.1,-.1] (gravity = [-.5,-.8, .16,.16]). We will further assume that the agent finds itself in a conflictual situation surrounded by other agents. We will also assume that the behavior module will only generate options to attack the other characters, and that the appraisal module predicts the correct feedback. In the table, the possible feedback options are various degrees of

- **fear and surprise (alarm)** - the attack fails, but was not expected to
- **anger and surprise (outrage)** - when the attack succeeds, but was not expected to
- **anger and anticipation (aggression)** - when the attack succeeds and was expected to

According to our feedback formula, we have the following impact factors:

Table 1 Attack impact table

Gravity	Feedback	Gain	Risk	Impact
[-0.5 -0.83 0.16 0.16]	[0 0 0.2 0.3]	0.4	0.3	0.10
[-0.5 -0.83 0.16 0.16]	[0 0 -0.6 -0.6]	1.56	1.176	0.38
[-0.5 -0.83 0.16 0.16]	[0 0 -0.4 0.3]	0.13	0.098	0.03
[-0.5 -0.83 0.16 0.16]	[0 0 -1 -1]	2.6	1.96	0.64
[-0.5 -0.83 0.16 0.16]	[0 0 1 0.6]	-2.08	-1.568	-0.51
[-0.5 -0.83 0.16 0.16]	[0 0 1 0.3]	-1.69	-1.274	-0.42

In table 1) we can see that the agent will prefer the two options that will increase it's aggression, will be neutral towards attempting to attack an agent that will suprise it, and will be reluctant to attack when it might fail.

As a further example, let us assume that the agent is in the same state and that it is surrounded by objects that it can pick up, the only actions generated are actions to pick up objects and the joy felt when picking up an object is equal to $\frac{value(object)}{value(object)+wealth(agent)}$.

Table 2 Take impact table

Gravity	Feedback	Gain	Risk	Impact
[-0.5 -0.8 0.16 0.16]	[0.1 0 0 0]	-0.13	0	0.13
[-0.5 -0.8 0.16 0.16]	[0.2 0 0 0]	-0.26	0	0.26
[-0.5 -0.8 0.16 0.16]	[0.3 0 0 0]	-0.39	0	0.39
[-0.5 -0.8 0.16 0.16]	[0.4 0 0 0]	-0.52	0	0.52
[-0.5 -0.8 0.16 0.16]	[0.5 0 0 0]	-0.65	0	0.65
[-0.5 -0.8 0.16 0.16]	[0.6 0 0 0]	-0.78	0	0.78

As we can see in the second table (Fig. 2), for varying item values, we get different priorities. The agent would prefer to pick up the most valuable item first.

4 Conclusion

We have provided virtual characters with the means of artificial expression and methods for its integration into intelligent artificial agent architectures. The system is firmly grounded in theory, respecting PLutchik's taxonomy, Lazarus' appraisal model, replicating the way emotions influence human perception and motivating a character's actions in the environment. We have developed a game artificial intelligence framework and demonstrated how easily the Plug-and-play architecture is integrated with the AI framework as well as the game engine used. We have also succeeded in keeping the system complexity down in order to make the system more accessible and easy to use and adopt. Last, but not least, the Plug-and-Play system is scalable (as the system only intervenes within the agent's context, the overhead is linear) and both the New Newtonian Emotion System and the Plug-and-Play interface are easily expandable.

5 Future Work

There are several application domains that the model can easily expand to. The first among these would be as a trust and reputation model. Emotions play such a role in humans. The process is called reciprocal altruism, an emergent norm in human systems that states that one person will help another, seemingly without any possibility of reciprocation, in the hopes that the initial actor will be helped themselves at a later date - in artificial intelligence terms, this means that an agent would act in a manner that temporarily reduces its fitness while increasing another agent's fitness, with the expectation that the other agent will act in a similar manner at a later time. Emotions serve as an evaluator to help spot cheaters that try to abuse the system. We could use the Trust-Disgust axis in order to quantify how a given agent performs when a contract has been agreed upon, and use the Surprise-Anticipation axis in order to evaluate how an agent conforms to emergent but non-contractual norms. In order for the system to be useful in spotting *cheaters*. This can be achieved in a centralized or distributed manner, depending on goals.

Acknowledgements. The work has been funded by Project 264207, ERRIC-Empowering Romanian Research on Intelligent Information Technologies/FP7-REGPOT-2010-1.

References

[1] Heuer, F., Burke, A., Reisberg, D.: Remembering emotional events. Memory and Cognition 20, 277–290 (1992)
[2] Clore, G., Ortony, A., Collins, A.: The cognitive structure of emotions. Cambridge University Press, Cambridge (1988)
[3] Pereira, D., Oliveira, E., Moreira, N.: Formal modelling of emotions in BDI agents. In: Sadri, F., Satoh, K. (eds.) CLIMA VIII 2007. LNCS (LNAI), vol. 5056, pp. 62–81. Springer, Heidelberg (2008)
[4] Kensinger, E.A.: Remembering emotional experiences: The contribution of valence and arousal. Reviews in the Neurosciences 15, 241–251 (2004)
[5] Lungu, V.: Rule-based system for emotional decision-making agents. In: Sisteme Distribuite, Suceava (2009) ISSN 2067–5259

[6] Lungu, V.: Artificial emotion simulation model. In: 7th Workshop on Agents for Complex Systems (ACSYS 2010) held in conjuction with 12th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing, SYNASC 2010, Timisoara (2010)
[7] Lungu, V.: Newtonian emotion system. In: Proceedings of the 6th International Symposium on Intelligent Distributed Computing - IDC 2012. Springer Series on Intelligent Distributed Computing, vol. VI, pp. 307–315 (September 2012)
[8] Lungu, V.: Artificial emotion simulation model and agent architecture. Advances in Intelligent Control Systems and Computer Science 187, 207–221 (2013)
[9] Sabouret, N., Ochs, M., Corruble, V.: Simulation of the dynamics of non-player characters' emotions and social relations in games. IEEE Transactions on Computational Intelligence and AI in Games 1(4), 281–297 (2009)
[10] Ioerger, T.R., El-Nasr, M.S., Yen, J.: Peteei: a pet with evolving emotional intelligence. In: Proceedings of the third annual conference on Autonomous Agents, Seattle, Washington, United States, pp. 9–15 (April 1999)
[11] Schachter, S., Singer, J.: Cognitive, social, and physiological determinants of emotional state. Psychological Review 69, 379–399 (1962)
[12] Scherer, K.: What are emotions and how can they be measured? Social Science Information 44(4), 695–729 (2005)
[13] Sharot, T., Phelps, E.A.: How arousal modulates memory: Disentangling the effects of attention and retention. Cognitive, Affective and Behavioral Neuroscience 4, 294–306 (2004)
[14] Strauss, M.: Eric: a generic rule-based framework for an affective embodied commentary agent. In: AAMAS 2008: Proceedings of the 7th International Joint Conference on Autonomous Agents and Multiagent Systems, pp. 97–104 (2008)