An approach to knowledge dynamic maintenance for emotional agents

S. E. Fulladoza Dalibón[1,2,3], D. C. Martinez[1,3], and G. R. Simari[1]

[1] Artificial Intelligence Research and Development Laboratory - Department of
Computer Science and Engineering, UN del Sur, Bahía Blanca
[2] Agents and Intelligent Systems Division, UN de Entre Ríos, Concordia, Entre Ríos
[3] CONICET - Consejo Nacional de Investigaciones Científicas y Técnicas, Argentina
{sef,dcm,grs}@cs.uns.edu.ar

Abstract. In this work we present an approach to emotional reasoning for believable agents, by introducing a mechanism to progressively build a map of knowledge for reasoning. We present the notion of inference graph for progressive reasoning in an emotional context. In this model, knowledge is partially highlighted and noticed by the agent.

1 Introduction

A believable character is a synthetic actor in a virtual environment that is able to provide an illusion of life [10]. This kind of agents are interesting in entertainment (Mainly role-playing games) [6, 2] and tutoring applications [1], where there is a strong connection and interaction with the human user for long periods of time. Many aspects of believable characters are being studied, such as graphical aspects, user interaction and behaviour modelling.

One of the properties rarely addressed in this scenario is the connection between emotions and reasoning. In this area, we are particularly interested in how emotions can conduct and affect the process of reasoning towards a believable behaviour [7]. In most KR&R models, decision and reasoning processes are done by contrasting, according to different mechanisms, all the information that an agent possesses, regarding the size of the knowledge base. This is mandatory in many domains, but is not a natural behaviour in believable agents. In fact, humans do not take *all* of its knowledge into account to make a decision, or to become credulous or not about some claim. In a real-time fashion, the human way of thinking uses only information that is relevant or *highlighted* in a particular moment. As knowledge is used in a thinking process, additional information may get highlighted and new mental elaborations are possible. For instance, people usually remember some anecdotes just when seeing a photo of an old friend. This process of retrieving internal knowledge may be conducted by emotions. A person enjoying in a party is not likely to remember sad things. A flight passenger feeling fear may not remember crew instructions during an emergency, while other passengers may only think about protecting their cellphone first. Different emotional states lead to different reasoning and then to different reactions.

We are interesting in the modelling of reasoning mechanisms that include emotions as relevant conductors. In this work we present an approach to emotional reasoning for believable agents, by introducing a mechanism to progressively build a map of defeasible knowledge for reasoning. Since emotions may affect reasoning and reasoning may affect emotions, the highlighted knowledge evolves trough time. In this approach, a logic program is used to represent the knowledge of the agent, and the reasoning of the agent is presented as the evolution of an inference graph similar to those used in various works in TMS [4], and argumentation [8, 11, 9]. This work is based on the formalism presented in [3], but instead of using abstract arguments as information pieces, we add more detail by the specification of the agent's knowledge as a set of inference rules, what allows us to model the reasoning process more closely to a incoming implementation.

We have organized the paper as follows: Section 2 introduce the notion of an agent whose reasoning is influenced by emotions. The components of the model as well as the basic principles and interactions are explained. In Section 3 a cycle of the reasoning process of the agent is described, showing how the reasoning model evolves due to the consideration of new information. Section 4 is a discussion about some ideas to make the model more dynamic through the continuous evolution of emotions during the process.

2   An Emotional Reasoning Agent

Our Emotional Reasoning Agent is equipped with a set of emotions engaging a process of reasoning in a virtual environment. These emotions influence over the selection of rules in such a process, since the knowledge of the agent can be partially highlighted according to the current emotional profile. In this work emotions and arguments are treated abstractly, since we are interested in the general model of reasoning for an actor in a virtual scenario.

Emotional factors are denoted as positive literals in the form $\epsilon_x(p_1, \ldots, p_n)$, where x is usually the name of the emotion (such as love or hate) and p_i are literals. The set of all the emotional factors is denoted as $\mathbb{E}$. For instance, the set $E = \{fear(fire), fear(pet\_dies), courage()\}$ is a set of emotions. The intensity of an emotion is denoted by the function $e_v : \mathbb{E} \to \mathbb{Z}^+ \cup \{0\}$. An emotional state, denoted as ξ, is a set of emotional factors with intensity. For instance, the pair $\langle E, e_v \rangle$, with $e_v(fear(fire)) = 5$, $e_v(fear(pet\_dies)) = 20$, $e_v(courage()) = 1$ is a valid emotional state.

We use Defeasible Logic Programming (DeLP) [5] to model the knowledge of an agent, as a tuple $\mathcal{P} = (\Delta, \Sigma)$, where Δ is a set of defeasible DeLP rules and Σ is a set of presumptions. For instance, the following is a valid knowledge base of an emotional agent:

- $\Delta = \{fuego(Y) \prec fuego(X), next(X, Y); \quad dies(X) \prec in(X, Y), fuego(Y); \sim dies(X) \prec in(X, Y), fuego(Y), escapes(X)\}$
- $\Sigma = \{fuego(cocina) \prec; \quad next(cocina, living) \prec; \quad in(living, tom) \prec; escapes(tom) \prec\}$

2.1 Emotional reasoning

As stated before, we are interested in a model of reasoning influenced by emotional factors for an actor in a virtual, dynamic environment towards a believable behaviour. An agent of this kind does not use his knowledge base as a whole, but only a highlighted part of it, according to its own emotional state. Thus, in our model it is possible that an inference rule is not noticed by the agent in a particular moment and then some conclusions may not be reached. Different emotional states may have different results over the *state of awareness* of the inner knowledge. Knowledge awareness is then subject to the emotional state of the agent. Given a program (P) and a set of valued emotional factors ξ, we define a function $S : \mathcal{P} \times \xi \to \mathbb{R}$ such that, for each rule $r \in \mathcal{P}$, $S(r, \xi)$ represents the intensity of the rule r according the emotional state ξ.

For simplicity, we will define an *intensity requirement* for a rule r, representing the intensity that r needs to reach in any emotional state in order to be highlighted. A rule with intensity below this threshold will not be noticed by the emotional agent. Under this binary interpretation, the higher this requirement is, the more difficult is to fulfil the conditions to use the rule.

Definition 1. *(Intensity Requirement) Let r be a rule in $\mathcal{P}$. The intensity requirement of r, is denoted as a function $\mu : \mathcal{P} \to \mathbb{Z}$. If $S(r, \xi) \geq \mu(r)$ then the rule is said to be highlighted, otherwise it is said to be concealed.*
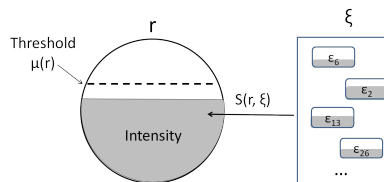

Fig. 1: Emotional intensity of rules

For illustrative purposes, the intensity of rules can be graphically described as in Figure 1, where the circle represents a rule, the doted line in the circle is the intensity requirement, and the filled part of the circle is the intensity level. In Figure 2 a knowledge base is shown. Rules with intensity level above the intensity requirement are highlighted rules, painted in dark grey. Concealed rules are painted in light grey.
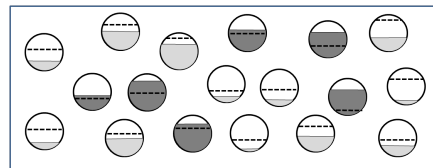

Fig. 2: A graphic representation for the awareness of program rules

2.2 Inference Graph

Similar to the structures used in truth maintenance systems [4], the mental activity of the agent is represented using an *inference graph*, which evolves through the time as new inference rules are used and then new conclusions are obtained. This evolution is modelled in [3] using abstract arguments. An inference graph is composed by two types of nodes: the inference rule nodes, and the conclusion nodes. Inference rule nodes represent defeasible rules or presumptions from a subjacent program $\mathcal{P}$, while conclusion nodes represent literals of the language. Literals are not replicated in the graph, in such a way that every reference of it by an inference rule (either as a conclusion or as a premise) remit to the same conclusion node. This shared use of premises by part of different inferences (or arguments, in an argumentative view) has been called as a *divergent argument structure* by D. Walton [12].

Definition 2. *(Inference Graph) An inference graph $\mathcal{G}$ is a tuple (CN, IN, SR, NR, AR), where CN is a set of conclusion nodes, IN is a set of inference rule nodes, $SR \in IN \times CN$ is a set of support relations, $NR \in CN \times IN$ is a set of necessity relations, and $AR \in CN \times CN$ is a set of attack relations.*

Nodes, either inference or conclusion ones, will be denoted with brackets. If x is an element of $\mathcal{P}$ or a conclusion literal, then $[x]$ is the corresponding node.

The *support* relation links inference nodes to the conclusion node containing the head of its rule. The rule is understood as a reason to believe in the conclusion. For instance, there is a support relation between the inference node $[c{\prec}a, b]$ and the conclusion node $[c]$. The *necessity* relation links conclusion nodes with inference nodes to whose rule it is a premise, and it reflects the fact that the former is needed to apply the inference step defined by the latter. For instance, there is a necessity relation between the conclusion node $[b]$ and the inference node $[c{\prec}a, b]$. The *attack* relation exists between every pair of conclusion nodes containing complementary literals. For instance, there is an attack relation between the conclusion nodes $[q]$ and $[\neg q]$.

There are two types of marking for a node: *support* markings and *acceptance* markings. The support markings are SUPPORTED and UNSUPPORTED. The acceptance marking are ACCEPTED, REJECTED and UNDECIDED. A conclusion node has support and acceptance markings, while an inference node has only acceptance markings. If $[c]$ is a node, then $[c]^S$ is the corresponding support marking and $[c]^A$ is the acceptance marking of $[c]$.

A conclusion node $[c]$ is marked as $[c]^S$ =SUPPORTED if there is at least one inference node $[r]$ such that $[r]^A$ =ACCEPTED, and $(r, c) \in SR$. A conclusion node $[c]$ is marked as $[c]^A$ =ACCEPTED if $[c]^S$ =SUPPORTED, and either it is preferred (according to some preference criteria) over the conclusion node $[\neg c]$, or the confronting node $[\neg c] \notin CN$. $[c]^A$ =REJECTED if the conclusion node $[\neg c]$ is in CN and it is preferred over $[c]$. In any other case, $[c]^A$ =UNDECIDED.

An inference node $[r]$ is marked as $[r]^A$ =ACCEPTED if for every necessity relation (b_i, r) in NR, the conclusion node $[b_i]$ is marked as $[b_i]^A$ =ACCEPTED. In the
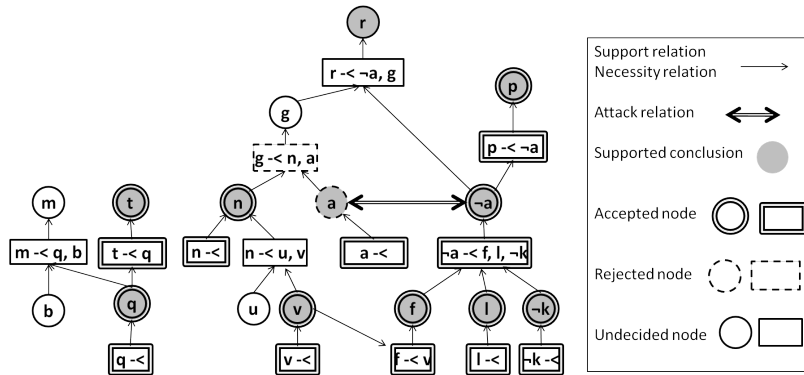
Fig. 3: An example of an inference graph

case that for at least one of the $[b_i]$ is $[b_i]^A$ =REJECTED, then $[r]^A$ =REJECTED. Otherwise, $[r]^A$ =UNDECIDED.

Consider the inference graph of Figure 3. Conclusion nodes that has not an accepted inference node as support, as it is the case of $[m]$, $[u]$ and $[g]$, are UNDECIDED. Unless there is a confrontation of complementary literals, a SUPPORTED conclusion node is also ACCEPTED. This is not the case of $[a]$ and $[\neg a]$ where, although both are SUPPORTED, $[\neg a]$ is preferred over $[a]$ (according to some preference criteria), and thus $[a]^A$ =REJECTED and $[\neg a]^A$ =ACCEPTED. As $[g \prec n, a]$ requires from $[a]$, and $[a]^A$ =REJECTED, then $[g \prec n, a]^A$ =REJECTED.

All of this marking is made during the reasoning cycle of the actor in the virtual environment, as explained in the following section.

3   The reasoning cycle

The goal of the framework is to provide a basis for the characterization of reasoning influenced by emotions in an agent or synthetic actor in a virtual scenario. We are interested in a dynamic, continuous process of reasoning that provides a believable illusion of human thinking. Thus, a reasoning cycle is modelled. Although not shown in this paper, this cycle could be suspended or resumed when needed in the virtual simulation. It is sufficient by now to present the model of knowledge processing, which is sketched in Algorithm 1. An inference graph is dynamically constructed by selecting a highlighted rule of the knowledge base. Intensity of rules must be updated accordingly.

Algorithm 1 Reasoning Cycle

 1: Obtain the expansion border, composed by all selectable rules and presumptions.
 2: Select a rule or presumption r from the expansion border.
 3: Expand the inference graph with r.
 4:         Create the inference node $[r]$
 5:         Connect $[r]$ to each $[b_i]$, being b_i a premise of r.
 6:         Connect $[r]$ to $[c]$, being c the conclusion of r.
 7:         Propagate acceptability and support states

The expansion border is formed by highlighted rules for further reasoning. These rules and presumptions are candidates to be integrated in the inference graph in the current stage. In order to avoid erratic reasoning, there must be a relation between candidate rules and the existing inference graph.

Definition 3. *(Expansion Border) Given an inference graph $\mathcal{G} = (CN, IN, SR, NR, AR)$ and a program $\mathcal{P}$, the expansion border is a set $\mathcal{E}_{P,G}$ that contains all the rules or presumptions $r \in \mathcal{P}$ in the form $h \multimap b_1, \ldots, b_n$ or $h \prec$ respectively, such that r is highlighted, and either (a) r is a rule $h \multimap b_1, \ldots, b_n$ and there exist a conclusion node $[h]$, $[\neg h]$ or $[b_i]$ in CN, or (b) r is a presumption.*

The agent selects a rule or presumption from $\mathcal{E}_{P,G}$ based on some preference criteria[4]. The stimuli of the rules and presumptions can be used for that purpose. Under such criteria, the agent selects the first rule found among the most stimulated ones.

Once a new rule or presumption $r \in \mathcal{P}$ is selected for the expansion of the inference graph, an inference node $[r]$ is created. Then, the new inference node needs to be integrated in the inference graph, so it is connected with the pre-existent pieces of information.

An inference node $[r]$ is connected by a *necessity link* to all of the conclusion nodes $[x]$ such that x is a premise of r. If there is no such a conclusion node in the graph, it is created and marked as UNSUPPORTED and UNDECIDED. After making the connections, the marking of the inference rule will depend on the marking of its premises. For example, if all the conclusion nodes containing the premises for the rule exist previously and in ACCEPTED state, the inference node will be updated to ACCEPTED. Instead, if at least one of these conclusion nodes is REJECTED, the inference node will be REJECTED as well.

As the rule or presumption represents a new reason to believe in the conclusion that is supporting, it is necessary to update the state of acceptability of the supported conclusion node, and of further conclusions that it is indirectly supporting. This update is done through a bottom-up propagation, starting from the point in which the rule was included.

The inference node $[r]$ is then connected by a support link to the conclusion node $[h]$, being h the head of r. If $[h]$ is not in the graph, the system creates the node which is, by default, UNSUPPORTED and UNDECIDED. As $[r]$ represents a reason to believe in $[h]$, if the former is ACCEPTED, then $[h]^S =$ SUPPORTED. A SUPPORTED conclusion $[h]$ will be ACCEPTED, unless there is in the graph an opposed conclusion node $[\neg h]$ that is preferred to it, in such a case $[h]^A =$ REJECTED.

Afterwards, changes should be propagated over the graph, in order to preserve consistency. During the propagation, the process may reach two confronting conclusion nodes (with complementary literals), and must to decide which one to accept and which one to reject, according to some preference criteria. Then, the markings are propagated over the ancestors of both nodes, in such a way that dependant conclusions will always behave according to the state of acceptability of the conclusions they are depending on.

---

[4] DeLP uses *specificity*, but any other preference could be applied.

For example, imagine that there is an inference node $[r]$ such that all the conclusion nodes which are premises to it are ACCEPTED, excepting for one of them, $[b_n]$, so $[r]$ can not be ACCEPTED. Suppose then that in a following cycle we add to the graph an inference node supporting $[b_n]$ in such a way that it makes $[b_n]$ to be ACCEPTED. In this new state, $[r]$ has all their necessary conclusion nodes ACCEPTED, so it also needs to be updated to ACCEPTED. Let h be the head of r; then $[h]$ has gained a support (the recently accepted $[r]$), and thus $[h]^A$ (and $[\neg h]^A$, if $[\neg h]$ is in the graph) must be reviewed. If any change in $[h]^A$ and $[\neg h]^A$, such a change must be propagated to every rule using these literals as premises, and so on. We can observe that the propagation describe a tree path, starting a new branch each time two complementary conclusion nodes are found. The process ends when every branch of propagation reach a node with no change in its acceptance state.

4   Making the framework dynamic

The framework provides an approach to an emotional-based control mechanism for the inferences made by the agent. However, the emotional state of the agent remains the same during the process. We are also interested in the dynamic of emotions. We want the emotional state of an agent to evolve as consequence of the reasoning process, in such a way that the intensity level of the rules in the program is dynamic, and so the reasoning flows as the emotional state changes. In order to model this, the following two steps must be added at the end of the reasoning cycle: (1) Trigger the emotional effects due to the expansion by r, thus changing values in ξ. (2) Update the intensity of the rules in $\mathcal{P}$.

For the realization of the former, we define a function that triggers collateral emotional effects due to the use of inference rules, in such a way that the emotional state of the agent is modified dynamically through the process. This modular construction of effects is interesting as provides it spots for behaviour refinement in the simulation at the design stage.

Definition 4. *(Inference Side Effect) Let $r \in \mathcal{P}$ be a rule, Gs the set of all inference graphs, and ξs the set of all the emotional states. The inference side effect $s : \mathcal{P} \times Gs \times \xi s \to \xi s$ is the effect that the use of a rule of $\mathcal{P}$ in a given inference graph, over a given emotional state, has over such an emotional state.*

Concerning the second step, as the intensity level of the rules on the program depends on the emotional state of the agent, and as the emotional state variates due to the inference side effect, the intensity of the rules must be updated. This can be done by a new computation of the intensity levels. As result, it may happen that some rules in the program that were concealed, now become highlighted, and previously highlighted rules become concealed. Moreover, inference rules that are already used in the inference graph may become concealed. In this case, even though this will not cause the conclusion of the rule to lose its support state in the graph, it can trim the expansion of the graph, making this expansion point a non interesting one; *i. e.,* when the expansion border is being

obtained, the rules of the program containing the literals on the now concealed rule are not taken into account.

5 Conclusion

In this work we presented an approach to emotional reasoning for believable agents, by introducing a mechanism to progressively build a map of knowledge for reasoning. We presented the notion of inference graph for progressive reasoning in an emotional context. In this model, knowledge is partially highlighted and noticed by the agent. Since emotions may affect reasoning and reasoning may affect emotions, both elements evolve dependently over each cycle in the process. Both the emotional state guide the agent to make adequate inferences, as the inferences made by the agent trigger relevant changes over the emotional state. This allows the modelling of believable reasoning behaviour, where emotions are determinant in the recovery and use of knowledge in a given situation.

References

1. Aylett, R., Vala, M., Sequeira, P., Paiva, A.: Fearnot! an emergent narrative approach to virtual dramas for anti-bullying education. In: Cavazza, M., Donikian, S. (eds.) Virtual Storytelling. Using Virtual Reality Technologies for Storytelling, Lecture Notes in Computer Science, vol. 4871, pp. 202–205. Springer Berlin Heidelberg (2007)
2. Bailey, C., You, J., Acton, G., Rankin, A., Katchabaw, M.: Believability through psychosocial behaviour: Creating bots that are more engaging and entertaining. In: Believable Bots, pp. 29–68. Springer (2012)
3. Dalibón, S.E.F., Martinez, D.C., Simari, G.R.: Emotion-directed argument awareness for autonomous agent reasoning. Inteligencia Artificial, Revista Iberoamericana de Inteligencia Artificial 15(50), 30–45 (2012)
4. Doyle, J.: A truth maintenance system. Artif. Intell. 12(3), 231–272 (1979)
5. García, A.J., Simari, G.R.: Defeasible logic programming an argumentative approach. TPLP pp. 95–138 (2004)
6. Lim, M.Y., Dias, J., Aylett, R., Paiva, A.: Creating adaptive affective autonomous npcs. Autonomous Agents and Multi-Agent Systems 24(2), 287–311 (2012)
7. Nabi, R.L.: Exploring the framing effects of emotion do discrete emotions differentially influence information accessibility, information seeking, and policy preference? Communication Research 30(2), 224–247 (2003)
8. Pollock, J.L.: Cognitive Carpentry: A Blueprint for How to Build a Person. MIT Press, Cambridge, MA, USA (1995)
9. Tang, Y., Sklar, E., Parsons, S.: An argumentation engine: Argtrust. In: Ninth International Workshop on Argumentation in Multiagent Systems (2012)
10. Togelius, J., Yannakakis, G.N., Karakovskiy, S., Shaker, N.: Assessing believability. In: Believable Bots, pp. 215–230. Springer (2012)
11. Verheij, H.: Rules, Reasons, Arguments: Formal Studies of Argumentation and Defeat. Universiteit Maastricht (1996)
12. Walton, D.: Argument Structure: A Pragmatic Theory (Toronto Studies in Philosophy). University of Toronto Press (Aug 1996)