

Document downloaded from:

<http://hdl.handle.net/10251/102091>

This paper must be cited as:



The final publication is available at

<http://doi.org/10.1145/3001584>

Copyright Association for Computing Machinery

Additional Information

Towards Formal Modeling of Affective Agents in a BDI Architecture

BEXY ALFONSO, Universitat Politècnica de València
EMILIO VIVANCOS, Universitat Politècnica de València
VICENTE BOTTI, Universitat Politècnica de València

Abstract

Affective characteristics are crucial factors that influence human behavior, and often the prevalence of either emotions or reason varies on each individual. We aim to facilitate the development of agents that simulate humans' reasoning in equilibrium with their affective characteristics. We first identify core processes in an affective BDI agent, and we integrate them into an agent architecture (*GenIA*³). These tasks include the extension of the BDI agent reasoning cycle to be compliant with the architecture, and the extension of the agent language (Jason) to support affection-based reasoning, as well as the parametrization of the affection-rationality equilibrium.

Keywords— Agents, Affective characteristics, Emotions, Semantic, Formalization, Jason

1 Introduction

Research on Multi-Agent Systems (MAS) has traditionally focused on the search for rational solutions that maximize the quality or utility of the result. However, when an agent needs to simulate human behavior, this kind of approach is not the most appropriate. Studies demonstrate that, when facing alternative choices, emotions guide decision making towards an advantageous direction, influencing the subjective utility of the choices [12]. Also studies show that, in human-machine applications, the human-machine interaction is improved when virtual agents express emotions, enhancing human satisfaction [27], and believability [11, 41], among others. These results highlight the importance of affective characteristics in social and cognitive functions, becoming required characteristics for believable intelligent agents. As it has been addressed by recent approaches, several applications can benefit from affective agents, for example video games, education, health care, and simulation of decision-making [10].

Many computational approaches that model affective agents are based on the cognitive perspective of emotions. These proposals may model interconnected phenomena such as the appraisal process, the emotions dynamics, and/or the influence of affect on the cognitive processes and agent behavior. For example, an embodied virtual character that reacts emotionally to some external stimulus requires, not only an emotion-reaction mechanism, but also a mechanism for

“interpreting” the stimulus and generating emotions; or for example, an agent that imitates humans when playing a card game that involves gambling, needs mechanisms for making decisions about what play make next, possibly biased by the current affective state of the agent, and mechanisms for emotionally react to what happens in the game¹. Thus, when modeling a single affect-related phenomena, researchers often have to deal with modeling all related processes (and hence making greater “unnecessary” efforts), or focusing on modeling the required phenomena, paying less attention to the rest of processes (and hence maybe missing important details). On the other hand, studies argue for the prevalence on each individual of either emotions or rationality on his behavior, and for the relation between these two aspects [15]. Nevertheless, computationally modeling this relation is difficult, maybe because, to the best of our knowledge, current computational architectures do not allow to parametrize this relation to create artificial entities more rational or more emotive.

In order to address these issues we propose an approach that 1) allows to implement various psychological theories relative to: individual differences, affect generation, affect dynamics, and affect influence on cognition and behavior, and that comes with a default implementation that can be used in several domains. Our approach also aims to 2) facilitate to set an equilibrium between the rational and the affective sides of an agent according to different psychological theories. Specifically we propose to establish this equilibrium by offering means for adjusting: the level of rationality of an individual, the frequency of rational and affective processes, the way the affective state influences decisions, the way the affective state influences individual’s believes, and how changes on the affective state generate behaviors. We have designed *GenIA*³ [3], a General-purpose Intelligent Affective Agent Architecture, which is inspired by O3A [1]. *GenIA*³ is based on widely accepted psychological and neurological theories, and it is built over a traditional BDI (Beliefs-Desires-Intentions) architecture, offering components to represent affective traits like personality, emotions and mood. Being based on a BDI architecture, *GenIA*³ follows theories of motivation and action generation, where the course of actions to be executed is decided according to the agent’s goals. In *GenIA*³ the affective state also has a motivational function, guiding goal’s priorities and/or generating new intentions. The interaction among the components of *GenIA*³ produces an agent behavior biased by the agent affective state. An architecture like *GenIA*³ facilitates “the creation of computational models of specific psychological phenomena of interest” [45], by relieving the modeler of irrelevant implementation choices or design specifications, and providing plausible default values.

Our aim is to offer a feasible and comprehensive way of developing affective agents on the base of *GenIA*³. We start from an existing BDI platform and language (Jason [9]), and we extend the reasoning cycle of AgentSpeak (which is the base of Jason), in order to introduce new affect-related steps. The new components that we propose are flexible enough to be committed to several psychological theories, and the modifications we propose for Jason can be adapted to other BDI agent languages.

¹Literature argues for the influence of emotions on decisions in this kind of games [7, 13]

2 Related Work

Several authors have proposed mechanisms to incorporate affective components into intelligent agents. Usually these components represent an appraisal derivation model, an affect derivation model, and a behavior consequent model [31]. These models have been used in several computational models like EMA [30], FLAME [17], MEA [6], or the Ochs' *et. al* model of emotions for an empathic dialog agent [38] (see [45] or [46] for a detailed review of affective computational models). According to these models, affective architectures have also been proposed. Probably the approach most similar to ours is FATiMA [16], a BDI architecture that uses emotions and personality to influence agents' behavior. FATiMA is a modular approach that defines a general processing cycle for the agent by using generic functions. These generic functions are realized through specialized components. Specific components can also be implemented with additional functionalities, or different psychological theories. Nevertheless FATiMA only solves some of the issues we aim to solve. For example it offers a general architecture and platform that allows to model different affective phenomena with different psychological theories (which was part our first requirement). But, for example, in relation to our second requirement of facilitating to set an equilibrium between the rational and the affective sides, in FATiMA it is not possible to set a level of rationality of an individual, to set specific synchronization relationships among affective and practical reasoning cycles, or to specify how changes on the affective state generate behaviors through an explicit and integrated way of defining a coping model.

On the other hand, some works propose to incorporate affective traits into agent-based systems in a formal way [36, 54, 55]. Our aim is to reach a higher level of formalization of affective agents, offering a flexible approach where the processing of the emotions can be easily adapted to the specific emotional characteristics of the problem to be solved, and to particular psychological theories. Therefore, the formalization of our *GenIA*³ architecture uses general components in order to integrate the affective components with the BDI agent cognitive process. A *GenIA*³ agent has also a personality that contains the agent individual traits and the agent common responses to deal with a change in the affective state. Thus, the affective state and agent personality may not only drive goals, but also generate new ones (related to these common responses). In this article we extend the syntax and operational semantics used in Jason [8, 57], a well know agent-oriented programming language grounded in a logical computable language (AgentSpeak [44]). Jason is widely accepted on the agents community due to its versatility to be adapted to several kinds of agent applications, thus it becomes a suitable choice for building "customizable" affective agents able to represent a wide set of situations.

3 Extension of the Jason Platform and Language to Include Affective processes

By extending the operational semantics of AgentSpeak and the syntax of the Jason agent language and platform, we want to illustrate how to develop BDI agents endowed with affective characteristics and behavior. Thus, we provide the agent programmer with psychologically grounded tools for creating human-

like agents. To this end, we first identify the core processes that may be included in the agent affective and reasoning cycles. Then we propose a way of integrating these processes on a Jason agent, and we describe new attributes that should be included in the Jason language. Finally we formalize the way each step of the agent reasoning and affective cycles works through an extension of the AgentSpeak’s operational semantics, providing also the description of the default design for the processes involved.

Hereinafter we refer to *affective state* as a generalized representation of all agent attributes that characterize one or more aspects of the agent state in line with the definition of *core affect* in [48]: “A neurophysiological state that is consciously accessible as a simple, nonreflective feeling (...)”; *affective processes* as the new processes added to the original BDI processes in order to consider affective characteristics; *affective cycle* as the cycle which modifies or generates the *affective state*; and *reasoning cycle* as the cycle that represents the agent practical reasoning (see Figure 1). Moreover we refer to the ranges of values for the variables that define the *affective state* as *affective categories*, and to a single emotion as an *emotion category* (e.g. joy, fear, or anger, in line with classifications like the one of OCC [40]).

3.1 Core Processes of an Affective BDI Agent

Our first goal was to propose an approach that allows to implement different theories relative to: individual differences, affect generation, affect experience, and affect influence on cognition and behavior. Therefore the proposed architecture (*GenIA*³) includes the core processes that fulfill this requirement. First, individual differences are represented through personality traits that may influence the processes of the agent reasoning and affective cycles. Besides, affect generation is represented through the **appraisal** process, affect experience through the **affect generator** and **affect temporal dynamics** processes, affect influence on cognition through the **affective modulator of beliefs**, and affect influence on behavior is represented through the **coping** and **filter** processes. *GenIA*³ also includes the processes of a traditional BDI agent architecture. Figure 1 shows the structure of *GenIA*³. The reasoning cycle includes the main processes of a BDI agent (bottom side of Figure 1). The architecture has two other cycles: one has only one process that is executed continuously, the **affect temporal dynamic**, and the other (**affective cycle**) includes the rest of affective processes. In the implementation of the architecture (and fulfilling part of our second requirement of allowing to set an equilibrium between the affective and rational sides of an agent), it is possible to set a synchronization between the reasoning and affective cycles. This is described on Sections 3.3 and 4. The theories that support this customization are the appraisal theories. Some of them such as Scherer’s appraisal theory [49], state that the appraisal is performed at several levels and that several appraisal evaluation checks are performed sequentially. Next we describe the core processes of an affective BDI agent following the *GenIA*³ structure [3].

Five core affective processes are included in *GenIA*³: **appraisal**, **affect generator**, **coping**, **affective modulator of beliefs**, and **affect temporal dynamics**. In order to illustrate how each of these processes work, we will take up the example introduced in Section 1, of an agent that imitates humans when playing a card game that involves gambling. The evaluation of the current

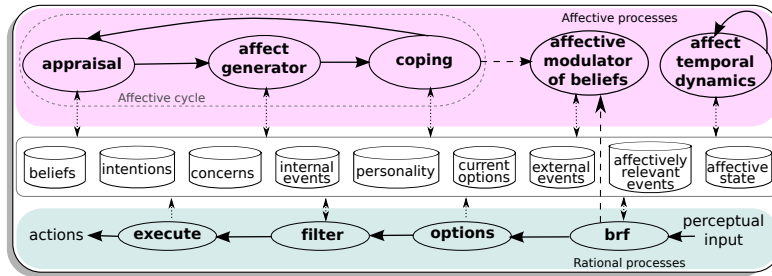


Figure 1: *GenIA*³: a General-purpose Intelligent Affective Agent Architecture that integrates BDI and affective processes. Sequences are represented as solid line arrows, subprocess as dashed line arrows, and exchange of information as dotted line arrows.

situation according to the current state of the world and the agent’s *concerns* (i.e., interests, motivations, ideals, or standards) is performed in the **appraisal** process. In *GenIA*³ this process can be committed to any particular appraisal theory (e.g., [53], [49], [47], [40]), and several parameters can be used in order to perform this appraisal (e.g., the agent’s *beliefs*, *concerns*, *internal events*, *external events*, memories of *affectively relevant events*, *current options*). A set of “appraisal variables” result from this evaluation. Consider, for example, that our agent concerns involve to win every single hand, and he loses the current hand; an appraisal variable “desirability” may result from the evaluation of this event in the appraisal process, whose value will be negative. New relevant events (according to their impact on the affective state of the agent), are also stored in the memory of *affectively relevant events* during the **appraisal** process. These affectively relevant events may be used in a future appraisal². An example of affectively relevant event could be to lose a risky hand where a high bet was made. Although the appraisal process has several parameters, not all of them are necessarily used. For example, following the EMA model [30], the appraisal variable *desirability* is determined by assessing the value of a proposition. It may imply the use of the agent *concerns*, and the agent *beliefs* (in order to evaluate the agent concerns). The **affect generator** is in charge of generating the agent *affective state* by using the current *affective state* and the appraisal variables generated by the **appraisal** process. The **affect generator** process can be committed to any psychological theory and the agent *affective state* can be represented either as a set of emotion categories, appraisal variables, or mood dimensions. For example, when representing *affective state* through mood dimensions, the Pleasure-Arousal-Dominance or PAD model of mood [35], or the Russell’s bipolar dimensions (hedonism and arousal³) [48] can be used. In our example, if our agent is very depressed, and he wins a hand, he may feel more happy, but probably not as happy as if his previous affective state were happy already. Determining the way the agent affective state changes is

²We allow this possibility on the base that past personally significant events (which are stored in the autobiographical memory), can have a significant impact on human life, shaping the perception of the upcoming tasks and modifying actual behavior [51, 50, 14].

³According to Russell’s definition of affective state: “A neurophysiological state that is consciously accessible as a simple, nonreflective feeling that is an integral blend of hedonic (pleasure–displeasure) and arousal (sleepy–activated) values” [48].

what the **affect generator** does. The **coping** process is tightly linked to the agent personality, since this process determines whether some agent responses or reactive behavior should be generated, and what should be these responses or reactive behavior. These agent responses can represent “response tendencies” in line with [39], or can be oriented to take back the agent *affective state* to a desired state⁴ (also called “coping strategies” [30]). Examples of possible reactive behaviors are facial expressions or body gestures that are involuntary and individual. The agent can have coping strategies like “shift responsibility” (e.g., to think that he lost a hand because the dealer gave him bad cards), or “wishful thinking” (e.g., to think that he will win all subsequent hands) [30]. These coping strategies may imply a modification on the beliefs of the agent, what involves the process **affective modulator of beliefs** (described below). The process for controlling the **affect temporal dynamics** is in charge of determining the temporal variation of the *affective state*, specifically its duration and decay. These dynamics vary from one individual to another, in such a way that some *personality* traits can determine the way that these variations are produced.

The **appraisal**, **affect generator**, and **coping** processes are part of the appraisal-reappraisal cycle (also called affective cycle) that is represented in most appraisal theories⁵. The **affective modulator of beliefs** is not executed as an independent process but as a subprocess of either the **coping** process or the **brf** process. The **brf** process corresponds to the “belief revision function” which is explained next. The **affect temporal dynamics**, on the other hand, is not included in this affective cycle, because it doesn’t depend on any other process and no other process depends on it. Thus, the **affect temporal dynamics** is controlled in an independent cycle.

*GenIA*³ also contains the cognitive processes that take place in a typical BDI agent reasoning cycle. These processes are summarized in **brf**, **options**, **filter**, and **execute** [58]. The **brf** process uses a perceptual input along with current *beliefs* in order to determine the agent’s new *beliefs*. As a result of this process, new *external events* (one per percept) may be generated. In line with the idea that the agent *affective state* contributes to the maintenance of *beliefs* [29, 42, 25] (e.g., a negative affective state induces an individual to question more his or her beliefs, making him or her more susceptible to accept new information), the **brf** can use the **affective modulator of beliefs** to determine how the *beliefs* are maintained. In order to understand the function of the **affective modulator of beliefs** let’s consider the agent perceived self-efficacy in the card game: the belief related to “the conviction that one can successfully execute the behavior required to produce the outcome” [5]. If he continuously losses several hands his perceived self-efficacy may be affected. See [42, 20] for a more detailed description of affective beliefs revision. The tasks for *options* generation are performed in the **options** process. These *options* (or desires) are generated on the base of the agent’s current *beliefs*, *external events*,

⁴The specification of a “desired state” depends on the assumptions or psychological theories used in particular designs. For example according to [24], a desired state is that where emotional distress is reduced, and according to [28] a state where the negative emotional responses associated with stress are reduced.

⁵Appraisal-reappraisal is the term used in the Scherer’s appraisal theory [49], which is considered one of the most complex and hence, the representation of other appraisal theories could be easily done through it [16].

internal events, and *intentions*. These *options* represent the means whereby the agent can achieve its *intentions* (e.g., doubling the bet, standing, or hitting a card). The **filter** process determines what to do by generating the agent’s *intentions* (e.g., rising profits). To this end, a deliberation process is performed that considers previously-held *intentions*, current *beliefs*, and *options*. Also, as part of this deliberation process, the current agent *affective state*, and some aspect of the agent *personality*, are considered. Specifically, in relation to the agent personality, the extent to which the agent decisions are influenced by its *affective state* can be taken into account. We call it *rationality level*, which has been included to fulfill our second requirement of facilitating to set an equilibrium between the rational and the affective sides. Including the affective state and personality aspects on the filter process relies on theories that argue for the need of considering the influence of emotions and individual differences for behaving either rationally or emotionally/intuitively/unconsciously, in order to properly model human behavior [26]. Besides, experimental studies offer evidences that emotions drive deliberative decision making [13]. The **execute** process contains the “action selection function”, so it uses the current *intentions* to determine the next action to be executed. The execution of actions can produce *internal events* that are related to, for example, the action failure or success. For example, if the action “rise bet” is executed, a possible reason for it to fail could be that the bet reached a top value.

3.2 Extension of the Jason Agent Reasoning Cycle

We have extended the reasoning cycle of a Jason agent [8], in order to build human-like agents whose execution and representation consider both affective and rational processes. Figure 2 shows the steps of the three cycles that are part of the agent execution, as well as the relationship between these steps. The colored steps are either new or modified, while the non-colored steps are the ones proposed in [57]. Similarly, the transitions with dashed lines are new or modified, and their corresponding transitions rules are presented in Section 3.4. There is a clear correspondence between these steps and the processes of *GenIA*³. The affective processes **appraisal**, **affect generator**, **coping**, **affective modulator of beliefs**, and **affect temporal dynamics** of *GenIA*³ are performed in the steps `Appr`, `UpAs`, `Cope`, `AffModB`, and `AsDecay` of Figure 2 respectively. The process `AffModB` has been integrated in the reasoning cycle, since it is closely linked to the addition and deletion of beliefs what takes place in the reasoning cycle. The process `SelCs` is part of the **coping** process of the *GenIA*³ architecture, and it is in charge of determining the coping strategies that need to be executed in the current affective cycle. The **brf** process is performed through the `Perceive`⁶ and `ProcMsg` steps; the steps `SelEv`, `RelPl`, and `ApplPl` perform the **options** process; the **filter** process is performed through the `SelAppl`, `AddIM`, and `SelInt` steps; and the `ExecInt` and `ClrInt` steps are in charge of the **execute** process.

Figure 2 also shows that three possible cycles can take place during the agent execution. These cycles control the affective processes (affective cycle), the ra-

⁶Although the formalization of AgentSpeak considers that an agent can perceive new information from the environment, to the best of our knowledge, there is no explicit step in the reasoning cycle for this task. We have decided to make this step explicit with the initial step `Perceive`.

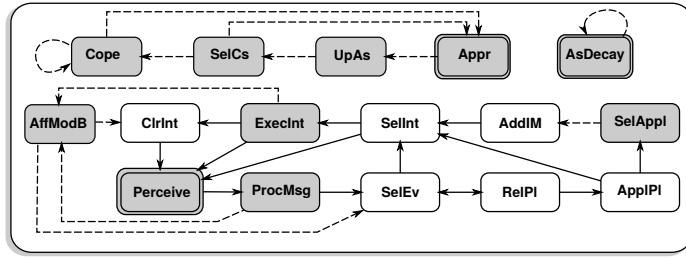


Figure 2: Extension of the reasoning cycle of AgentSpeak. New and modified steps are colored and new and modified transitions are dashed lines.

tional processes (reasoning cycle) and the temporal dynamics of affect (affect temporal dynamics cycle). Next we describe the steps for each cycle. Before starting with the affective cycle it is worth mentioning that, in an initialization stage, the *affective state* has an initial value. This value is also the agent “equilibrium state”, which is a neutral state where the agent doesn’t experience any significant emotion. The affective cycle starts with the `Appr` step, where the **appraisal** process is performed on the base of several parameters including the agent *concerns*, *personality*, and the *probabilities* associated with agent *beliefs* (if prospect-based emotions are generated). The `Appr` also determines whether the event is relevant for the agent from an affective point of view (in case that the new *affective state* has an event associated). Then, in the `UpAs` step, the *affective state* is updated by using the appraisal variables generated in the `Appr` step. After the updating of the *affective state*, the `SelCs` step verifies whether it is necessary to generate new behaviors in the agent according to this change on the *affective state*, verifying which of the agent *coping strategies* are applicable. The `Cope` step performs the tasks required to execute the selected coping strategies. The intentions derived from the execution of coping strategies are added as intended means at the end of the base of intentions, which is shared by both reasoning and affective cycles. Both reasoning and affective cycles generate their own intentions independently which are included in this common base of intentions. Intentions generated by the affective cycle are added at the end of the current intentions, as well as the intentions generated by steps of the reasoning cycle. In the default design all intentions are executed by their insertion order in the `SelInt` step of the reasoning cycle. The default implementation of this step is explained in Section 3.4.2. The reasoning cycle contains two new steps (which are the steps `Perceive` and `AffModB`), and three modified steps (`ProcMsg`, `SelAppl`, and `ExecInt`). `Perceive` is the initial step of the reasoning cycle. In this step the agent *beliefs* are modified according to what can be observed from the environment and/or according to external events. The `Perceive` step is followed by the `ProcMsg` step, which is in charge of processing the messages received from other agents. Next, the information coming from the received messages, and from the perception of the environment (on the `Perceive` step), can be modified in the step `AffModB`, which follows the `ProcMsg` step. In *GenIA*³ the agent *affective state* also influences agent’s decisions. The `SelAppl` step performs this task by selecting the next applicable plan; thus it has been modified to consider the agent current *affective state*, and also the agent *rationality level*. The step `ExecInt` may require to execute

<u>agent</u>	→ (init_bels init_goals)*plans concerns personality
<u>concerns</u>	→ “concerns__”“(” <var> “)”“-” log_expr “.”
<u>personality</u>	→ “personality__:”“(” traits [“,” rat_level] [“,” coping_strats] “)”“.”
<u>traits</u>	→ “[” <number> (“,” <number>)* “]”
<u>rat_level</u>	→ <number>
<u>coping_strats</u>	→ “copingst__:” (cs)+
<u>cs</u>	→ “cs__”“(” context “,” aff_categ “)”“-” body “.”
<u>aff_categ</u>	→ <atom>
<u>beliefs</u>	→ ((literal literal_prob) “.”)*
<u>literal_prob</u>	→ [“~”] atomic_formula_prob
<u>atomic_formula_prob</u>	→ (<atom> <var>) [“(” list_of_terms “)”] [“[” list_of_terms “]”] [time_point_range]
<u>time_point_range</u>	→ “<” time_point “,” time_point “>”
<u>time_point</u>	→ (<number> <atom>)

Figure 3: Simplified extension of the EBNF for the Jason Agent Language.

intentions that imply adding or removing beliefs. Thus it can also be followed by the step `AffModB` to this end. Finally the affect temporal dynamics cycle contains one step: `AsDecay`. This step determines the tendency of the *affective state* to return to its “equilibrium state”. This task could use some trait of the agent *personality*.

3.2.1 Affect-Related Agent Attributes

In Section 3.2 new cycles were introduced that contain new steps in the AgentSpeak agent execution and some existing steps were modified as well. These steps use some parameters that are agent attributes and that do not have a corresponding representation in the AgentSpeak agent language. For example, aspects of the agent *personality*, the agent *affective state*, the agent *concerns*, and the *probabilities* of the beliefs are used in several steps. A formal syntax for these attributes is presented in Section 3.3 as an extension of the Jason agent language.

3.3 Extension of the Jason Language

In this section we briefly describe the extension of the specification of the Jason agent language. To this end we extend the EBNF used in [9] (see Figures 3 and 4). Figure 3 shows an extract of the new syntax of the agent program, and Figure 4 shows an extract of the new syntax of the MAS project configuration.

In Figure 3 the Jason agent syntax has been extended with two new components: *concerns* and *personality*. Thus, when writing the agent code, the programmer is able to define the agent *concerns*⁷ and *personality*. We also propose new structures in order to express the probabilities of the agent

⁷This view of *concerns* can be confused with declarative goals. Unlike declarative goals, which are goals that explicitly represent a state of affairs to be achieved, *concerns* are a way of evaluating how “good” the state of affairs is. In other words, they offer a direct assessment of the state of affairs without needing to represent declarative goals. For example, a situation where an agent has declarative goals, and where the current state of affairs doesn’t satisfy any of them, would be represented with agent *concerns* with a very low value.

	Concept	Involved components and meaning
Agent EBFN	concerns	concerns: A rule, whose left hand side is a literal that contains one numerical term, indicating the extent to which the current beliefs are aligned with the agent <i>concerns</i> . The right hand side of the rule expresses the way this numerical value is calculated according to the agent current beliefs.
	personality	personality: The agent personality with three components: <i>traits</i> , <i>rat_level</i> , and <i>coping_strats</i> . The last two are optional.
		traits: In line with the traits perspective of personality [4, 18, 23]. It is a list of numerical values, each representing the extent to which its corresponding personality trait is present in the agent personality.
		rat_level: A numerical value that indicates the extent to which an individual is more rational, and hence, less emotive when making decisions.
		coping_strats: Represents the agent set of coping strategies.
		cs: A single coping strategy. Its left hand side has two components: a context and an affective category (<i>aff_cat</i>), and its right hand side represents a list of actions ^a (that represent the agent behavior for this coping strategy). Actions in <i>cs</i> , are executed if the context is a logical consequence of the agent belief base and if the agent current affective state matches <i>aff_cat</i> ^b .
		aff_cat: A label indicating one of the affective categories defined in the project configuration (<i>affect_categories</i>).
		literal_prob: A literal for representing beliefs with probabilities. Its atomic formula (<i>atomic_formula_prob</i>) contains a time range.
	beliefs probabilities	time_point_range: Temporal interval that the agent thinks the corresponding belief holds and/or will hold.
		time_point: Represents the lower or upper bound of a time range.
Project Configuration EBFN	affect	affect_categories: Represents a set of affective categories.
		aff_category: A single affective category. It has a functor and a list of components, where each component can be either a numerical value or a range of values. These ranges contain the values that the affective state should have to match this affective category ^c .
		range: A range that contains the minimum and maximum values for each affective component.

^aThe right hand side of a coping strategy is similar to the body of a Jason plan (as shown in the extension of the EBNF of Jason on Figure 3). The body of a Jason plan can include conditional statements, loops, etc.

^bDue to their nature, coping strategies are not processed as Jason plans or rules. That's why a special structure has been created for them.

^cThe number of components of an affective category (and hence, of an affective state), depends on the psychological theory used.

Table 1: Explanation for some of the new components of the EBNF extension for a Jason agent.

<u>mas</u>	→ “MAS” <id> “{” [<u>infrastructure</u>] [<u>environment</u>] [<u>exec_control</u>] [<u>affect_categories</u>] “}”
<u>agent</u>	→ <asid> [<u>filename</u>] [<u>options</u>] [“agentArchClass” <ID>] [“beliefBaseClass” <ID>] [“agentClass” <ID>] [“emEngineClass” <ID>] [“personalityClass” <ID>] [“affStateClass” <ID>] [“#” <number>] [“at” <id>] “;”
<u>affect_categories</u>	→ “aff_categories:” <u>aff_category</u> (, <u>aff_category</u>)* “,”
<u>aff_category</u>	→ <atom> “(” <u>aff_components</u> “)”
<u>aff_components</u>	→ (<number> <u>range</u>) (“,” <number> <u>range</u>)*
<u>range</u>	→ “<” <number> “:” <number> “>”

Figure 4: Simplified extension of the EBNF for the Multi-Agent Systems Language.

beliefs. In the new belief syntax each belief can be written either in the traditional way or with an annotation indicating its probability. We defined the new reserved word `prob__` that represents the functor of a term (with one numerical term as argument), that indicates the probability of the corresponding belief (when the term is one of the annotations of the belief). This new kind of beliefs has also a new component which indicates the temporal interval that the agent thinks this belief holds and/or will hold. If a belief doesn't have any probability annotation, the default probability 1 will be used; if a belief doesn't have an extra temporal interval, it expresses that the belief holds in that moment. In the reasoning process, probabilities are propagated following the Bayes theory. We also defined the reserved word `affect__` as the functor of a literal with one 'Atom' term as argument. This argument represents an affective category that must match with one of the `affect_categories` included in the project configuration (see Figure 4). This literal can be used as a plan annotation, and it will be considered as an additional criteria in the selection of an applicable plan.

The syntactic specification for the MAS project configuration on Figure 4 has been extended in two directions. First, the agent specification includes three new optional components: `emEngineClass`, `\personalityClass`, and `affStateClass`. The reserved word `emEngineClass` allows to assign the agent the emotion engine class, which contains the functions that implement the main affective processes; `\personalityClass` allows to assign the agent the class that manages its personality; and `affStateClass` allows to assign the agent the class that manages its affective state. Besides, it is possible to use a new reserved word in the agent options, which is `nacprc` (number of affective cycles per reasoning cycle). This option allows to establish the relation between the affective and reasoning cycles (by default it is 1-1). Second, the new component `affect_categories`, represents a list of affective categories. Section 4 shows an example using this grammar. Table 1 shows a more detailed explanation of some of the components of Figures 3 and 4.

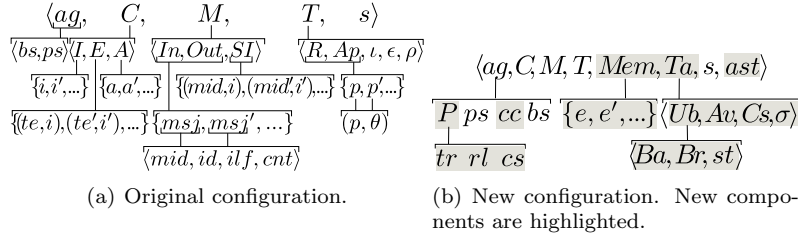


Figure 5: Configuration of a Jason agent.

3.4 Extension of the AgentSpeak Operational Semantics

In order to build an extension of the Jason platform whose agents try to simulate a human-like behavior, with affect-related processes and characteristics, we extended the AgentSpeak operational semantics, considering that it is the base of the Jason operational semantics. The AgentSpeak agent configuration is defined by a tuple $\langle ag, C, M, T, s \rangle$, whose values can be modified after a transition among two steps. Figure 5(a) shows this configuration as well as the structure of each one of its components. The new Jason agent configuration has the form $\langle ag, C, M, T, Ta, Mem, s, ast \rangle$. The new components of this configuration are highlighted in Figure 5(b), and are described next.

- ag represents the agent program, which originally contains a set of beliefs (bs), and a set of plans (ps). Additionally a set of concerns (cc), and a personality (P) has been included in the agent program ag .
 - The agent concerns cc is an agent attribute which is in line with the concerns in *GenIA*³, and reflects the agent’s ideals, motivations, interests, and/or standards.
 - Personality P includes the personality traits tr . tr contains a set of numerical values representing the agent personality traits (e.g., the Five Factor Model of personality [32] argues that the traits openness, conscientiousness, extraversion, agreeableness, and neuroticism can differentiate an individual from the rest). In line with the “rationality level” of *GenIA*³ we propose rl , which is also part of the agent personality. The rationality level states the extent to which agent decisions are influenced by its affective state⁸. cs represents the agent coping strategies, which relates a particular state (represented through a set of beliefs) and an affective state with a set of actions that generate intentions to be included in the agent current intentions.
- C , M , and T were originally part of AgentSpeak and represent the agent circumstance, communication parameters, and temporary information for a reasoning cycle.
- Mem contains a set on events $\{e, e', \dots\}$ that have been relevant for the agent from an affective point of view. We consider this set as a kind of

⁸This personality aspect is inspired in psychological tests like the Cognitive Reflexion Test (CRT) [19]

“autobiographic memory”, where the meaningful experiences are stored as proposed by [37]. This events are determined and *Mem* is updated in the appraisal process.

- *Ta* is a tuple $\langle Ub, Av, Cs, \sigma \rangle$, which represents the temporary information used by the affective processes in a cycle. Its components are:
 - *Ub* is a tuple $\langle Ba, Br, st \rangle$ which contains those beliefs to be added to or removed from the agent belief base. *Ba* and *Br* represent the set of beliefs to be added and the set of beliefs to be removed respectively; *st* contains the label of the step that requires to add and/or to remove beliefs in *Ba* and *Br*⁹.
 - *Av* contains the set of numerical values for the appraisal variables in the current affective cycle.
 - *Cs* contains the set of coping strategies to be executed in the current affective cycle.
 - σ represents the agent current affective state. It contains a set of variables $\{v, v', \dots\}$ where each variable contains a numerical value representing the intensity or the presence or not (in the case a bivalent variable) of either an emotion category (e.g., sad, happy, angry), an appraisal variable (e.g., desirability, controllability), or a mood dimension (e.g., the dimensions of the PAD model [34]).
- *s* is a label annotating the current step in the reasoning cycle, where $s \in \{Perceive, ProcMsg, SelEv, RelPl, ApplPl, SelAppl, AddIM, SelInt, ExecInt, ClrInt, AffModB\}$. A new label *AffModB* has been included in *s* which corresponds to the new step *AffModB* in the reasoning cycle (see Figure 2).
- *ast* is a label annotating the current step in the affective cycle, where $ast \in \{Appr, UpAs, SelCs, Cope\}$ (see Figure 2).

Using a similar notation to that used in [57], we refer to attributes with a subindex. For example we refer to the appraisal variables *Av* that are part of the affective temporary information *Ta*, as Ta_{Av} . Similarly we refer to the traits *tr* of the agent personality ag_P as $ag_{P_{tr}}$. We have also defined the structure of new functions that are part of the agent configuration and whose content must be specified by the agent programmer¹⁰. By offering a way of customizing these function we fulfill our first requirement, where the possibility of implementing various psychological theories should be offered. An example of how these functions can be implemented is described in more detail in Section 3.4.2. One of these functions is $ASDEC(\sigma, P_{tr})$, which controls how the affective state σ decays over time¹¹, and obtains new values for this affective state considering the personality traits P_{tr} . Appraisal variables are generated through the

⁹*Ub* is used by the step *AffModB*.

¹⁰The selection of the type and number of the parameters of these functions is based on those most commonly used in related computational approaches, and not all of them should be necessarily used in every case.

¹¹By default it is considered that the affective state decays, because in general psychological theories argue for this decay when dealing with the affect temporal dynamic [21]. Nevertheless, the function $ASDEC(\sigma, P_{tr})$ can be customized to include any other behavior.

function $\text{Appraise}(\varepsilon, bs, cc, Mem, Ap)$, which considers the event to evaluate (ε)¹², a set of beliefs (bs), concerns (cc), the affective relevant events for the agent (Mem), and options that the agent has available (Ap) [40, 29, 30]. The function $\text{AffRelEv}(\varepsilon, Av)$ evaluates if the event ε is relevant for the agent from an affective point of view, by using the appraisal variables in Av . The function $\text{UpAffSt}(\sigma, Av)$ determines a new affective state (which contains a set of variables), given the affective state σ and according to a set of appraisal variables Av [22, 30, 38]. Function $\text{modB}(AddB, DelB, \sigma)$ determines what beliefs, from the sets $AddB$ and $DelB$, need to be added or modified and which beliefs, from the set $DelB$, need to be removed, according to the affective state σ . We have also modified the AgentSpeak selection function S_{Ap} for selecting an applicable plan from the set of applicable plans Ap . It has the form $S_{Ap}(Ap, \sigma, P_{rl})$ where the affective state σ and an agent rationality level P_{rl} are new parameters. Besides, a new selection function $S_{cs}(Cs)$ has been created in order to select a coping strategy from a set of coping strategies Cs . The selection functions S_{Ap} , S_{cs} , and S_M , are defined at design time by the agent programmer, according to the desired behaviors for the agents. We do not include the selection functions in the configuration for a better readability. Nevertheless, Section 3.4.2 offers an informal description of them in our default design.

Additionally we have defined the $\text{EvalP}(PSet, bs)$, $\text{match}(\sigma, ac)$, and $\text{SelCopeSt}(P_{cs}, bs, \sigma)$ functions to determine: changes on percepts in the environment, whether a particular affective state matches an affective category, and applicable coping strategies respectively. Definitions 2, 3, and 4 propose a formalization for these functions. The function $\text{agperc}(bs)$ of definition 1 is an auxiliary function that determines the agent current percepts.

Definition 1 *Given the set bs of agent beliefs, the set of beliefs that correspond to the agent percepts is defined as follows:*

$$\text{agperc}(bs) = \{b[\text{annot}] \mid b[\text{annot}] \in bs \text{ and } \text{source}(\text{percept}) \in \text{annot}\}$$

Definition 2 *Given the set bs of agent beliefs, and the set of percepts $PSet = \{pc, pc', \dots\}$ observable in the environment (where each pc is a literal), the set of new percepts $NewP$ is calculated as the set difference $PSet \setminus \text{agperc}(bs)$. Also the set $RemP$ of percepts no longer existing in bs , is calculated as the set difference $\text{agperc}(bs) \setminus PSet$. The function $\text{EvalP}(PSet, bs)$ performs this task. It is defined as follows:*

$$\text{EvalP}(PSet, bs) = \{\langle NewP, RemP \rangle \mid NewP = \{b \in PSet \mid b \notin \text{agperc}(bs)\} \text{ and } RemP = \{b \in \text{agperc}(bs) \mid b \notin PSet\}\}$$

Definition 3 *Let be $\sigma = \{a_1, a_2, \dots, a_k\}$ a set of k numerical values, each corresponding to an affective label, and let be $ac = \{r_1, r_2, \dots, r_k\}$ a set of k ranges of values for the same affective labels, where $r_i = [rmin_i, rmax_i]$. The $\text{match}(\sigma, ac)$ function determines whether a particular affective state σ matches an affective category ac or not, and it is defined as follows:*

$$\text{match}(\sigma, ac) = \begin{cases} TRUE & \text{if } a_i \geq rmin_i \text{ and } a_i \leq rmax_i \quad \forall i \in \mathbb{Z} \mid 1 \leq i \leq k \\ FALSE & \text{otherwise} \end{cases}$$

¹²Events in AgentSpeak include the addition and deletion of beliefs (from the environment or own), addition of goals, and failure of goals.

As shown in Section 3.3, a coping strategy has three components: *context*, *affective category*, and *body*. Both *context* and *body* have the same meaning and structure that a context and body of a plan, where *context* represents a set of conditions that must hold, and *body* contains a set of actions to be performed [57]. Also, for the *body*'s actions to be executed, the agent current affective state must match *affective category*. If a coping strategy cs has the form $(ct, ac) \rightarrow h$, where ct is the context, ac is the affective category, and h is the body, the function $\text{SelCopeSt}(P_{cs}, bs, \sigma)$ is defined as follows:

Definition 4 *Given a set of coping strategies P_{cs} , a set of beliefs bs and a particular affective state σ , the set of applicable coping strategies is defined as follows:*

$$\text{SelCopeSt}(P_{cs}, bs, \sigma) = \{(cs, \theta) \mid cs \in P_{cs} \text{ and } \theta \text{ is s.t. } bs \models ct\theta \text{ and } \text{match}(\sigma, ac)\}$$

$$\text{where } ct = \text{CsCtxt}(cs) \text{ and } ac = \text{CsAc}(cs)\}$$

In the definition 4, the functions $\text{CsCtxt}(cs)$ and $\text{CsAc}(cs)$ return the context and the affective category of a given coping strategy cs , and θ is the most general unifier.

3.4.1 New Transition Rules

In this section we present the transition rules for the updated or new steps of the agent cycles (see Figure 2), with respect to [57] using the Structural Operational Semantics (SOS) [43]. Note that the initial state of the reasoning cycle is $\langle ag, C, M, T, Mem, Ta, \text{Perceive}, \text{Appr} \rangle$. At this point, the steps of the cycles are able to update one or more components of the agent configuration. Next we describe these transition rules individually. We start by the transition rules for the steps of the affective cycle.

Appraisal The process of appraisal takes place in this transition rules through the function $\text{Appraise}(T_\varepsilon, ag_{bs}, ag_{cc}, Mem, T_{Ap})$, which evaluates the current event T_ε . If the function $\text{AffRelEv}(T_\varepsilon)$ returns TRUE, the current event T_ε is added to the set of affectively relevant events Mem (rule Appr_1). The next step in this transition is the UpAs step.

$$\frac{\text{AppVar} = \text{Appraise}(T_\varepsilon, ag_{bs}, ag_{cc}, Mem, T_{Ap}) \quad \text{AffRelEv}(T_\varepsilon, \text{AppVar})}{\langle ag, C, M, T, Mem, Ta, s, \text{Appr} \rangle \rightarrow \langle ag, C, M, T, Mem', Ta', s, \text{UpAs} \rangle} \quad (\text{Appr}_1)$$

$$\text{where: } Ta'_{Av} = \text{AppVar} \quad Mem' = Mem \cup T_\varepsilon$$

$$\frac{\text{AppVar} = \text{Appraise}(ag_{bs}, ag_{cc}, T_\varepsilon, Mem, T_{Ap}) \quad \neg \text{AffRelEv}(T_\varepsilon, \text{AppVar})}{\langle ag, C, M, T, Mem, Ta, s, \text{Appr} \rangle \rightarrow \langle ag, C, M, T, Mem, Ta', s, \text{UpAs} \rangle} \quad (\text{Appr}_2)$$

$$\text{where: } Ta'_{Av} = \text{AppVar}$$

Update Affective State In this transition the agent affective state is updated through the function $\text{UpAffSt}(Ta_\sigma, Ta_{Av})$. The next step after this transition is the SelCs step.

$$\frac{\langle ag, C, M, T, Mem, Ta, s, \text{UpAs} \rangle}{\langle ag, C, M, T, Mem, Ta', P, s, \text{SelCs} \rangle} \quad (\text{UpAffState})$$

$$\text{where: } Ta'_\sigma = \text{UpAffSt}(Ta_\sigma, Ta_{Av})$$

Select Coping Strategies In this transition the agent applicable coping strategies are determined through the function $\text{SelCopeSt}(ag_{P_{cs}}, ag_{bs}, Ta_\sigma)$. Ta_{Cs} is updated with the result of $\text{SelCopeSt}(ag_{P_{cs}}, ag_{bs}, Ta_\sigma)$ and the cycle goes on with the step Cope (transition rule SelCs_2). If no coping strategy is applicable, the cycle returns to the step Appr (transition rule SelCs_1).

$$\frac{\text{SelCopeSt}(ag_{P_{cs}}, ag_{bs}, Ta_\sigma) = \{\}}{\langle ag, C, M, T, Mem, Ta, s, \text{SelCs} \rangle \rightarrow \langle ag, C, M, T, Mem, Ta, s, \text{Appr} \rangle} \quad (\text{SelCs}_1)$$

$$\frac{\text{SelCopeSt}(ag_{P_{cs}}, ag_{bs}, Ta_\sigma) \neq \{\}}{\langle ag, C, M, T, Mem, Ta, s, \text{SelCs} \rangle \rightarrow \langle ag, C, M, T, Mem, Ta', s, \text{Cope} \rangle} \quad (\text{SelCs}_2)$$

$$\text{where: } Ta_{Cs} = \text{SelCopeSt}(ag_{P_{cs}}, ag_{bs}, Ta_\sigma)$$

Cope In the step Cope the function S_{cs} selects a coping strategy from the current set Ta_{Cs} of applicable coping strategies. A plan p is created whose head is a TRUE value and whose actions (which are the body h of the plan p) are those of the selected coping strategy. The plan p and the unifier θ are added as an intention to the set of current intentions C_I and the selected coping strategy is removed from the set of applicable coping strategies Ta_{Cs} . The intention added can lead to the addition or dropping of beliefs, goals, and to a variety of actions (in general all actions that Jason allows to perform in a plan body). This step is repeated until Ta_{Cs} is empty, and then, the cycles goes on with the step Appr .

$$\frac{Ta_{Cs} \neq \{\} \quad S_{cs}(Ta_{Cs}) = (cs, \theta) \quad cs = (ct, ac) \rightarrow h}{\langle ag, C, M, T, Mem, Ta, s, \text{Cope} \rangle \rightarrow \langle ag, C', M, T, Mem, Ta', s, \text{Cope} \rangle} \quad (\text{Cope}_1)$$

$$\text{where: } p = \text{true} \leftarrow h \quad C'_I = C_I \cup \{[p\theta]\} \quad Ta'_{Cs} = Ta_{Cs} \setminus \{(cs, \theta)\}$$

$$\frac{Ta_{Cs} = \{\}}{\langle ag, C, M, T, Mem, Ta, s, \text{Cope} \rangle \rightarrow \langle ag, C, M, T, Mem, Ta, s, \text{Appr} \rangle} \quad (\text{Cope}_2)$$

Perceive This is the initial step of the reasoning cycle. The agent checks the environment for determining changes on percepts ($PSet$) through the function $\text{EvalP}(PSet, ag_{bs})$. $NewP$ contains new percepts to be included in, and $RemP$ contains percepts to be removed from the agent belief base ag_{bs} . The next step in the cycle is ProcMsg , and both $NewP$ and $RemP$ are stored in the affective temporal information of the agent configuration as Ta_{Ub} for them to be processed later in the step AffModB .

$$\frac{\text{EvalP}(PSet, ag_{bs}) = \langle NewP, RemP \rangle}{\langle ag, C, M, T, Mem, Ta, \text{Perceive}, ast \rangle \rightarrow \langle ag, C, M, T, Mem, Ta', \text{ProcMsg}, ast \rangle} \quad (\text{Perc})$$

$$Ta'_{Ub} = \langle NewP, RemP, Perceive \rangle$$

The next four rules are related to the processing of received messages. In these rules the functions $S_M(M_{In})$ and $SocAcc(id, ilf, at)$ are used. The first selects a message from the messages set M_{In} , and the second determines if a message is “socially acceptable”, where id is the message identifier, ilf is the illocutionary force of the message, and at is the propositional content of the message. More details of these functions can be found in [57].

Receiving a Tell message This transition has been modified in the same way as other transitions in which beliefs were added to the agent belief base. Thus, instead of adding them directly to the agent belief base, they are added to the affective temporal information of the agent configuration Ta_{Ub} , for them to be processed in the step $AffModB$.

$$\frac{S_M(M_{In}) = \langle mid, id, Tell, Bs \rangle \quad (mid, i) \notin M_{SI} \text{ (for any intention } i) \\ SocAcc(id, Tell, Bs) \quad Ta_{Ub} = \langle NewP, RemP, Perceive \rangle}{\langle ag, C, M, T, Mem, Ta, ProcMsg, ast \rangle \rightarrow \langle ag, C, M', T, Mem, Ta', AffModB, ast \rangle} \text{ (Tell')}$$

$$\text{where: } M'_{In} = M_{In} \setminus \{ \langle mid, id, Tell, Bs \rangle \} \quad Bs' = NewP \\ \text{and for each } b \in Bs: Bs' = Bs' \cup \{ b[\text{source}(id)] \} \\ Ta'_{Ub} = \langle Bs', RemP, ProcMsg \rangle$$

Receiving a Tell message as Reply Similarly, in this transition, beliefs sent by another agent as reply, are added to Ta_{Ub} for them to be processed in the step $AffModB$. Also, the required actions to resume the required intention are performed.

$$\frac{S_M(M_{In}) = \langle mid, id, Tell, Bs \rangle \quad (mid, i) \in M_{SI} \text{ (for any intention } i) \\ SocAcc(id, Tell, Bs) \quad Ta_{Ub} = \langle NewP, RemP, Perceive \rangle}{\langle ag, C, M, T, Mem, Ta, ProcMsg, ast \rangle \rightarrow \langle ag, C', M', T, Mem, Ta', AffModB, ast \rangle} \text{ (TellRepl')}$$

$$\text{where: } M'_{In} = M_{In} \setminus \{ \langle mid, id, Tell, Bs \rangle \} \quad M'_{SI} = M_{SI} \setminus \{ (mid, i) \} \\ C'_I = C_I \cup \{ i \} \quad Bs' = NewP \\ \text{and for each } b \in Bs: Bs' = Bs' \cup \{ b[\text{source}(id)] \} \\ Ta'_{Ub} = \langle Bs', RemP, ProcMsg \rangle$$

Receiving an Untell message In this transition, beliefs that need to be removed as the result of a message of other agent, are added to Ta_{Ub} for them to be processed in the step $AffModB$.

$$\frac{S_M(M_{In}) = \langle mid, id, Untell, ATs \rangle \quad (mid, i) \notin M_{SI} \text{ (for any intention } i) \\ SocAcc(id, Untell, ATs) \quad Ta_{Ub} = \langle NewP, RemP, Perceive \rangle}{\langle ag, C, M, T, Mem, Ta, ProcMsg, ast \rangle \rightarrow \langle ag, C, M', T, Mem, Ta', AffModB, ast \rangle} \text{ (Untell')}$$

$$\begin{aligned}
&\text{where: } M'_{In} = M_{In} \setminus \{\langle mid, id, Untell, ATs \rangle\} \\
&\quad DelB = \{at\theta \mid \theta \in \text{Test}(ag_{bs}, at) \wedge at \in ATs\} \\
&\quad DelB' = RemP \\
&\text{and for each } b \in DelB: DelB' = DelB' \cup \{b[\text{source}(id)]\} \\
&\quad Ta'_{Ub} = \langle NewP, DelB', ProcMsg \rangle
\end{aligned}$$

Receiving an Untell message as Reply This rule is similar to the previous one where beliefs that need to be removed as the result of a reply message of another agent, are added to Ta_{Ub} for them to be processed in the step $AffModB$. Also, the required actions to resume the required intention are performed.

$$\frac{S_M(M_{In}) = \langle mid, id, Untell, ATs \rangle \quad (mid, i) \in M_{SI} \text{ (for any intention } i) \quad \text{SocAcc}(id, Untell, ATs) \quad Ta_{Ub} = \langle NewP, RemP, Perceive \rangle}{\langle ag, C, M, T, Mem, Ta, ProcMsg, ast \rangle \rightarrow \langle ag, C', M', T, Mem, Ta', AffModB, ast \rangle} \text{ (UntellRepl')}$$

$$\begin{aligned}
&\text{where: } M'_{In} = M_{In} \setminus \{\langle mid, id, Untell, ATs \rangle\} \quad M'_{SI} = M_{SI} \setminus \{(mid, i)\} \\
&C'_I = C_I \cup \{i\} \quad DelB = \{at\theta \mid \theta \in \text{Test}(ag_{bs}, at) \wedge at \in ATs\} \quad DelB' = RemP \\
&\text{and for each } b \in DelB: DelB' = DelB' \cup \{b[\text{source}(id)]\} \\
&\quad Ta'_{Ub} = \langle NewP, DelB', ProcMsg \rangle
\end{aligned}$$

Selection of an Applicable Plan This transition rule has been modified so that the S_{Ap} function has two additional parameters: the agent current affective state Ta_σ , and the agent rationality level $ag_{P_{rl}}$. Thus the plan that the agent selects to execute, will be influenced by this two parameters. This is another function that can be customized by the programmer, nevertheless its default implemented mechanism is described in Section 3.4.2.

$$\frac{S_{Ap}(T_{Ap}, Ta_\sigma, ag_{P_{rl}}) = (p, \theta)}{\langle ag, C, M, T, Mem, Ta, SelAppl, ast \rangle \rightarrow \langle ag, C, M, T', Mem, Ta, AddIM, ast \rangle} \text{ (SelAppl')}$$

$$\text{where: } T'_p = (p, \theta)$$

Executing an Intention Following with the notation used in [57], $i[p]$ denotes an intention i with the plan p on top of it. Similarly to other transition rules above, in the next two rules, if the intention to be executed implies adding or removing a belief, these beliefs are stored in Ta_{Ub} for them to be processed in the step $AffModB$.

$$\frac{T_i = i[\text{head} \leftarrow +b; h]}{\langle ag, C, M, T, Mem, Ta, ExecInt, ast \rangle \rightarrow \langle ag, C', M, T, Mem, Ta', AffModB, ast \rangle} \text{ (AddBel')}$$

$$\text{where: } Ta'_{Ub} = \langle \{b[\text{source}(self)]\}, \{\}, ExecInt \rangle \quad C'_I = (C_I \setminus \{T_i\}) \cup \{i[\text{head} \leftarrow h]\}$$

$$\frac{T_i = i[\text{head} \leftarrow -at; h]}{\langle ag, C, M, T, Mem, Ta, ExecInt, ast \rangle \rightarrow \langle ag, C', M, T', Mem, Ta', AffModB, ast \rangle} \quad (\text{DelBel}')$$

where: $Ta'_{Ub} = \langle \{\}, \{at[\text{source}(\text{self})]\}, ExecInt \rangle$ $C'_I = (C_I \setminus \{T_i\}) \cup \{i[\text{head} \leftarrow h]\}$

Affective modulator of beliefs In this transition beliefs to be added and removed in tuple Ta_{Ub} are modulated according to the agent affective state Ta_σ (by the function $\text{modB}(AddB, DelB, Ta_\sigma)$), where a new set of beliefs to be added ($MAddB$) and removed ($MDelB$) are obtained. The third component of Ta_{Ub} indicates the step that requires the addition or deletion of beliefs, helping to determine the next step in the cycle (i.e., $SELv$ or $CLRInt$). The corresponding additions and deletions are performed, and the corresponding events of belief addition or deletion are created.

$$\frac{\begin{array}{c} Ta_{Ub} = \langle AddB, DelB, ProcMsg \rangle \\ \text{modB}(AddB, DelB, Ta_\sigma) = (MAddB, MDelB) \end{array}}{\langle ag, C, M, T, Mem, Ta, AffModB, ast \rangle \rightarrow \langle ag', C', M, T, Mem, Ta, SELv, ast \rangle} \quad (\text{ModB}_1)$$

where for each $mb \in MDelB$: $ag'_{bs} = ag_{bs} \setminus \{mb\}$ $C'_E = C_E \cup \{-mb, \top\}$
and for each $mab \in MAddB$: $ag'_{bs} = ag_{bs} \cup \{mab\}$ $C'_E = C_E \cup \{+mab, \top\}$

$$\frac{\begin{array}{c} Ta_{Ub} = \langle AddB, DelB, ExecInt \rangle \\ \text{modB}(AddB, DelB, Ta_\sigma) = (MAddB, MDelB) \end{array}}{\langle ag, C, M, T, Mem, Ta, AffModB, ast \rangle \rightarrow \langle ag', C', M, T, Mem, Ta, CLRInt, ast \rangle} \quad (\text{ModB}_2)$$

where for each $mb \in MDelB$: $ag'_{bs} = ag_{bs} \setminus \{mb\}$ $C'_E = C_E \cup \{-mb, \top\}$
and for each $mab \in MAddB$: $ag'_{bs} = ag_{bs} \cup \{mab\}$ $C'_E = C_E \cup \{+mab, \top\}$

Mood temporal dynamic A single cycle controls how the affective state decays over time. It contains the single step $ASDecay$, which is continuously executed. This task is performed by the $ASDec(Ta_\sigma, ag_{P_{tr}})$ function.

$$\frac{}{\langle ag, C, M, T, Mem, Ta, s, ast \rangle \rightarrow \langle ag, C, M, T, Mem, Ta', s, ast \rangle} \quad (\text{DecAffState})$$

where: $Ta'_\sigma = ASDec(Ta_\sigma, ag_{P_{tr}})$

The transition rules that correspond to the steps not previously addressed, have also been modified so that the structure of the agent configuration has been adapted to the new configuration. They are not presented for simplicity.

3.4.2 Default design

In Section 3.4 we introduced a set of functions that are used in the agent execution cycles, and that can be customized by the programmer. The existence of this set of functions adds flexibility to the agent programmer to adapt the agents behavior to several psychological theories and application domains. We have implemented these functions on the base of widely used psychological theories. This default implementation has been used in several scenarios, and can be extended if required. Next we perform a general description of these functions in order to offer a global understanding of the default implementation, avoiding specific details for simplicity. An alternative design can be found in [2].

In our default design, the affective state Ta_σ is represented as the agent mood in a dimensional way, where three values describe the agent mood in a particular moment: pleasure, arousal, and dominance (or PAD, according to Mehrabian’s model [34]). Appraisal variables Ta_{Av} can take three possible values (`desirability`, `likelihood`, or `causal attribution`), which were selected from the EMA model proposed in [30]. The traits of the agent personality follow the Five Factor Model [32], which describes individual traits through five dimensions (openness, conscientiousness, extraversion, agreeableness, and neuroticism). We propose a design for the affective cycle steps inspired by the Gebhard’s ALMA model [22]¹³. The initial (and also equilibrium affective state) of the agent is calculated following Mehrabian’s work [33], which proposes a mapping of the agent five dimensions of personality to the three dimensions of the PAD space. The function `Appraise`($\varepsilon, bs, cc, Mem, Ap$) evaluates the event ε when this event implies the addition or deletion of a belief. This function determines the `desirability`, `likelihood`, and `causal attribution` of the resulting state after the addition or deletion of the belief. `Desirability` is determined according to the agent concerns (by using its numerical value), `likelihood` is determined according to the probability of the agent belief, and the `causal attribution` can be the `environment` (if the belief to be added or removed is a percept) `other agent` (if the belief to be added or removed is a message), or `self` (if the belief to be added removed is a mental note). For example, consider an agent that represents a student who wants to pass an exam, and whose concerns value can be calculated as $V = Note/MaxNote$, where $Note$ is the exam result and $MaxNote$ is the maximum possible result (lets say 5). If he is told by the *teacher* that he passed with 4, `desirability` will be $\frac{4}{5}$, `likelihood` will be 1, and the `causal attribution` will be `other agent`. Also, in our default implementation, the function `AffRelev`(ε, Av) determines that the event ε is relevant when the `desirability` in Av is not in a range of “average desirabilities” (i.e., when it is extremely undesirable or extremely desirable). Besides the affective state is updated through the function `UpAffSt`(σ, Av) in three steps. First, five possible emotion categories can be derived (`hope`, `joy`, `fear`, `sadness`, and `guilt`), starting from the appraisal variables Av following [30]. Secondly, each emotion is mapped into the three PAD dimensions following [22]. Thirdly, mapped emotions are averaged in a single value for each dimension according to [22]¹⁴. The function $S_{Ap}(Ap, \sigma, P_{rl})$

¹³In the implementation of the steps we avoided to introduce too much execution complexity selecting as default mechanisms, those most commonly implemented in computational approaches.

¹⁴Considering that the set of emotions in [22] doesn’t contain all emotions in [30] we carefully

```

1 MAS prisoners_dilemma {
2   infrastructure: Centralised
3   agents: prisonerA [nacprc=1:1]
4     agentClass jason.Em.asSemantics.EmAgent
5     agentArchClass jason.Em.architecture.EmAgArch
6     emEngineClass
7     jason.Em.architecture.DefaultEmotionEngine
8     personalityClass jason.Em.architecture.OCEANPersonality
9     affStateClass jason.Em.architecture.PAD;
10  prisonerB
11  agentClass jason.Em.asSemantics.EmAgent
    agentArchClass jason.Em.architecture.EmAgArch;}

```

Figure 6: Extract of the code for the MAS project configuration.

uses the affective state σ and the rationality level of the agent personality P_{rl} (`rat_level` in the grammar of Figure 3) in order to select the next actions to be performed (by selecting the next applicable plan). It selects an applicable plan by ranking applicable plans with and without considering the affective state; then a general ranking is assigned to each plan by weighing up the two first rankings (the weight for the rank without affect is P_{rl} and the weight for the rank with affect is $1 - P_{rl}$). The applicable plan with the minimum value in the general ranking is selected. The function $S_{Cs}(Cs)$ always selects by default the first coping strategy from the set of coping strategies to be executed. Also by default the function `modB(AddB, DelB, σ)` adds beliefs to *AddB* and removes beliefs from *DelB*. Offering additional mechanisms to determine the way beliefs may be modulated would make this approach too complex for being included as a default implementation. The default design for the rest of the functions that haven't been described (such as $S_M(M_{In})$ or `SocAcc(id, ilf, at)`), follows the default design of a Jason agent, which can be found in [9]. Some of these functions of the original Jason agent (such as the function $S_I(C_I)$ for selecting the next intention to be executed), could also be customized by using the tools offered by the Jason original platform.

4 A Simple Example

In order to illustrate how the use of affective characteristics can produce a more believable behavior than the behavior of an agent with only a practical reasoning we use the classical “Prisoner’s Dilemma” game, typical of decision theories. In the prisoner dilemma there are two participants representing two prisoners that have committed a crime. We denote them as ‘prisonerA’ and ‘prisonerB’. A deal is proposed to the prisoners by the police, where each prisoner has two possible options: to keep silence (thus cooperating with the other prisoner), or to betray the other prisoner (revealing the details of the crime to the police). Each possible combination of decisions has a punishment of a number of years in prison for the participants. If ‘prisonerA’ and ‘prisonerB’ cooperate each other and they keep silence, then they receive both a year in prison as punishment; if one betrays and the other keeps silence, then the first receives the temptation

looked for a similarity assuming sadness as distress and guilt as remorse.

```

1 maxround(3).
2 deal(cooperationpunishment(1),defectpunishment(2),suckerpunishment(3),
   temptationpunishment(0)).
3 concern1(C1):-years_prison(YP)&((.ground(YP) & YP>0 & C1=1/YP)|(C1=0))
   .
4 concern2(C2):-(perceived_em(police,positive) & C2=1)|
5   (perceived_em(police,negative) & C2=-1)|(C2=0).
6 concerns__(V):-concern1(C1) & concern2(C2) & V=C1+C2.
7 !decide.
8 @p_betray[atomic,affect__(excited)]
9 +!decide: .my_name(Me) & partner(Other)
10 <- .print("I'm ",Me," and I betray ",Other," in round ",C);
11   +betray(Me); .send(Other,tell,betray(Me)).
12 @p_shut_up[atomic,affect__(relaxed)]
13 +!decide: .my_name(Me) & partner(Other)
14 <- .print("I'm ",Me," and I decide to shut up in round ",C);
15   +shut_up(Me); .send(Other,tell,shut_up(Me)).
16 +betray(P)[source(P)]:partner(P)
17 <- ?deal(cooperationpunishment(CP), defectpunishment(DP),
   suckerpunishment(DP), temptationpunishment(TP));
18   !partner_betray(DP,SP);
19   !next_round.
20 +!partner_betray(DP,SP): .my_name(Me) & (betray(Me)|shut_up(Me))
21 <- if (betray(Me)){+years_prison(DP); };
22   if (shut_up(Me)){+years_prison(SP); }.
23 +!partner_betray(CP,TP):.my_name(Me) & not (betray(Me)|shut_up(Me))
24 <- !partner_betray(CP,TP).

```

Figure 7: Extract of the code for ‘prisoner_template.asl’.

punishment of zero years (i.e., no punishment at all), and the other receives the sucker punishment of three years in prison; if both prisoners betray each other, then they both receive the defect punishment of two years in prison [56]. The economic theory predicts the third option (prisoners betray each other) as the most rational for both acting as a group [52]. Nevertheless, results of experiments systematically demonstrate that humans tend to keep silence showing aims of cooperation. This evidences that not only rational, but other factors influence human decisions. We used the iterated version of the prisoner dilemma, where there are N rounds of decisions before the final decision. This helps to observe how the participants’ decisions change when observing the other participant’s behavior. Figure 6 shows the project configuration of the example. Lines 3-8 contain the configuration for the agent “prisonerA”, and lines 9-11 contain the configuration for the agent “prisonerB”. The configuration of “prisonerB” is the required for an affective agent with the default design. The configuration of “prisonerA” is equivalent to the configuration of “prisonerB” but we have made some other parameters explicit to help to understand how to use a different design. Figure 8 shows an extract of the code for one of the agents (‘prisonerA’), which is written following our extended syntax of Jason. It includes the code of Figure 7 (see line 1 of Figure 8), which is common for both prisoners.

The code of Figure 7 shows the maximum number of rounds (line 1), the terms of the deal (line 2), and that both agents individual concerns involve the years to spend in prison (line 3), and the emotions expressed by the police

```

1  {include ("prisoner_template.asl")}
2  partner(prisonerB). round(0). years_prison(_).
3  betray(prisonerB) [prob__(0.6)]<r2,r2>.
4  shut_up(prisonerB) [prob__(0.4)]<r2,r2>.
5  personality__: {[ -0.8, 0, 0.1, -0.8, -0.1],
6                  0.8,
7  copingst__:    cs__(.my_name(Me) & partner(Other) &
8                  betray(Me) & shut_up(Other), neutral) -> .smile.
9                  cs__(.my_name(Me) & partner(Other) &
10                 betray(Other) & shut_up(Me), neutral) -> .frown.}

```

Figure 8: Extract of the code for the agent ‘prisonerA’.

officer (line 4). More specifically, the agents’ concerns highest value (and hence the best situation for the agents) is reached when the agents get the lowest value of years in prison and the emotions perceived are positive. Moreover, the concerns’ lowest value (and hence the worst situation for the agents) is reached when the agents get the highest value of years in prison and the emotions perceived are negative. Line 6 shows how concerns are calculated. Also, when it is time to take a decision, both participants can betray each other (plan of lines 8-11), or keep silence (plan of lines 12-15). Both plans have the same triggering events and contexts, nevertheless the current agent affective state will determine which one has priority (see the plan annotations of lines 8 and 12). At the end of each plan, a message containing the decision is sent to the other participant. Figure 7 also shows three additional plans in lines 16-24. The first plan is executed when a “betray” message is received from the other prisoner. As part of the actions of this plan, a goal is added. This goal triggers a plan that updates the number of years in prison for the current round if the agent has already made its decision (see plan of lines 20-22). If the agent hasn’t made any decision yet (lines 23-24) a goal is added to try to execute the plan of lines 20-22 again. The goal `!next_round` of line 19 removes all decisions (own and other’s decisions), updates the current round, and, if the maximum number of rounds hasn’t been reached, starts the decision process again. Figure 8 shows the specific configuration for the ‘prisonerA’ agent, which is similar to ‘prisonerB’. For example, the beliefs that contain the name of the other participant, the current round, the years in prison for the current round (line 2), and also, what the agent thinks the other participant will do in the second round as a distribution of probabilities¹⁵ (lines 3-4). The agent personality contains the list of values for the agent personality traits (line 5), its rationality level (line 6), and the agent coping strategies. The agent has two coping strategies. In the first, the agent smiles if it’s affective state is `neutral`, and if it betrays and the other keeps silence (lines 7-8). In the second coping strategy the agent frowns if it’s affective state is `neutral`, and if it keeps silence and the other betrays him (lines 9-10). The affective categories `neutral`, `relaxed`, and `excited` are represented in the project configuration (e.g. `excited` was defined as `excited(<0:0>, <0.8:0.9>, <0.1:0.3>)`). In the first cycle of iterations the initial affective state of ‘prisonerA’ is `neutral`, its decision is to keep silence, and ‘prisonerB’ betrays him. As a consequence

¹⁵We use the round number $rn = \{r1, r2, \dots, rn\}$ as the time unit.

‘prisonerA’ becomes `excited`, and in the second iteration he decides to betray ‘prisonerB’.

5 Conclusions

In this work we offer a formalization of *GenIA*³, a General-purpose Intelligent Affective Agent Architecture which is based on the BDI agent architecture. This formalization includes an extension of the AgentSpeak reasoning cycle, the definition of its operational semantics, and the extension of the syntax of an AgentSpeak-based agent language (Jason) to include affect-related attributes. With this kind of formalization, comparisons of different psychological theories can be easily performed (thus approaches can be adapted to specific application domains requirements and psychological theories); also more flexibility is provided to the language, offering a compact and elegant notation that avoids ambiguities. *GenIA*³ is grounded on widely studied psychological and neurological theories and offers an integral vision of the agent and its behavior considering both rational and affective attributes and processes. In order to offer this integral vision we’ve not only modeled emotions. We also address the agent affective state in a more generic way allowing to use different psychological theories for its representation. Also different psychological theories can be used in other affect-related processes, since our formalization allows customizing several steps in the reasoning and affective cycles. On the other hand, when defining the agent personality, it is possible to include other characteristics, besides the personality traits, traditionally addressed by computational approaches. In our approach it is also possible to define a “rationality level” for the agent, and the agent response tendencies (or coping strategies) for a given situation and affective state. Thus, the affective state not only drives goals but also may generate new ones through the agent coping strategies.

We propose a formalization that allows to apply our approach to other BDI based agent languages. Besides, as we propose an integral view of the agent, agents of several domains can be implemented, such as cognitive agents or virtual characters. This implies the validation of our approach in as many application domains as possible, which is one of our future tasks. Nevertheless, we provide useful and flexible tools and a default implementation, that any expert on fields like psychology or behavioral computing, could use in order to provide more precise and refined ways of describing each particular affective process.

6 ACKNOWLEDGMENTS

This work was supported by the Spanish government grant MINECO/FEDER TIN2012-36586-C03-01 and the HUMBACE project (Human and Social Behaviour Models for Agent-Based Computational Economics).

References

- [1] B. Alfonso, E. Vivancos, and V. J. Botti. An Open Architecture for Affective Traits in a BDI Agent. In *Proceedings of the 6th ECTA 2014. Part of the 6th IJCCI 2014*, pages 320–325, 2014.

- [2] B. Alfonso, E. Vivancos, and V. J. Botti. Design of an Affective Intelligent Agent on *GenIA*³. Technical report, DSIC, UPV, Spain, 2016.
- [3] B. Alfonso, E. Vivancos, and V. J. Botti. Toward a Systematic Development of Affective Intelligent Agents. Technical report, DSIC, UPV, Spain, 2016.
- [4] G. W. Allport. *Personality: A psychological interpretation*. Henry Holt New York, 1937.
- [5] A. Bandura. Self-efficacy: Toward a Unifying Theory of Behavioral Change. *Psychological review*, 84(2):191, 1977.
- [6] C. Battaglino, R. Damiano, and L. Lesmo. Emotional Range in Value-sensitive Deliberation. In *Proceedings of AAMAS '13*, pages 769–776, Richland, SC. International Foundation for Autonomous Agents and Multiagent Systems.
- [7] A. Bechara, H. Damasio, and A. R. Damasio. Emotion, Decision Making and the Orbitofrontal Cortex. *Cerebral cortex*, 10(3):295–307, 2000.
- [8] R. H. Bordini and J. F. Hübner. Semantics for the Jason Variant of AgentSpeak (Plan Failure and Some Internal Actions). In *Proceedings of ECAI'10*, pages 635–640, Amsterdam, The Netherlands, The Netherlands, 2010. IOS Press.
- [9] R. H. Bordini, J. F. Hübner, and M. Wooldridge. *Programming Multi-Agent Systems in AgentSpeak Using Jason*. Wiley, 2007.
- [10] T. Bosse, J. Broekens, J. Dias, and J. van der Zwaan. *Emotion Modeling*. Springer, 2014.
- [11] S. Brave, C. Nass, and K. Hutchinson. Computers that Care: Investigating the Effects of Orientation of Emotion Exhibited by an Embodied Computer Agent. *International journal of human-computer studies*, 62(2):161–178, 2005.
- [12] J. R. Busemeyer, E. Dimperio, and R. K. Jessup. *Integrating Emotional Processes Into Decision-Making Models*, pages 29–44. Oxford University Press, 2007.
- [13] C. F. Camerer, G. Loewenstein, and M. Rabin. *Advances in Behavioral Economics*. Princeton University Press, 2011.
- [14] M. A. Conway. *Autobiographical Memory: An Introduction*. Open University Press, 1990.
- [15] R. De Sousa. *The Rationality of Emotion*. Mit Press, 1990.
- [16] J. Dias, S. Mascarenhas, and A. Paiva. *FAtiMA Modular: Towards an Agent Architecture with a Generic Appraisal Framework*, chapter Generic Models and Frameworks, pages 44–56. Springer International Publishing, Cham, 2014.

- [17] M. S. El-Nasr, J. Yen, and T. R. Ioerger. Flame—fuzzy logic adaptive model of emotions. *Autonomous Agents and Multi-agent systems*, 3(3):219–257, 2000.
- [18] H. J. Eysenck. *Personality, Genetics, and Behavior: Selected Papers*, chapter Development of a Theory. Centennial psychology series. Praeger, 1982.
- [19] S. Frederick. Cognitive Reflection and Decision Making. *The Journal of Economic Perspectives*, 19(4):25–42, 2005.
- [20] N. Frijda, A. Manstead, and S. Bem. *Emotions and Beliefs: How Feelings Influence Thoughts*. Studies in Emotion and Social Interaction. Cambridge University Press, 2000.
- [21] N. H. Frijda. *The Laws of Emotion*. Lawrence Erlbaum Associates, Incorporated, 2007.
- [22] P. Gebhard. ALMA: A Layered Model of Affect. In *Proceedings of the 4th AAMAS*, pages 29–36, NY, USA, 2005. ACM.
- [23] L. R. Goldberg et al. An Alternative “Description of Personality”: The Big-Five Factor Structure. *Journal of personality and Social Psychology*, 59(6):1216–1229, 1990.
- [24] J. J. Gross and R. A. Thompson. Emotion Regulation: Conceptual Foundations. In *Handbook of Emotion Regulation*, Guilford Publications. Guilford Publications, 2011.
- [25] J. Ito, D. Pynadath, and S. Marsella. Modeling Self-Deception Within a Decision-Theoretic Framework. *AAMAS*, 20(1):3–13, 2010.
- [26] W. G. Kennedy. Modelling Human Behaviour in Agent-Based Models. In *Agent-based models of geographical systems*, pages 167–179. Springer, 2012.
- [27] J. Klein, Y. Moon, and R. W. Picard. This Computer Responds to User Frustration: Theory, Design, and Results. *Interacting with computers*, 14(2):119–140, 2002.
- [28] R. S. Lazarus and S. Folkman. *Stress, Appraisal, and Coping*. Springer publishing company, 1984.
- [29] S. Marsella and J. Gratch. Modeling Coping Behavior in Virtual Humans: Don’t Worry, be Happy. In *Proceedings of AAMAS’03*, pages 313–320. ACM, 2003.
- [30] S. C. Marsella and J. Gratch. EMA: A process model of appraisal dynamics. *Cognitive Systems Research*, 10(1):70–90, 2009.
- [31] S. C. Marsella, J. Gratch, and P. Petta. Computational Models of Emotion. In *A Blueprint for Affective Computing: A Sourcebook and Manual*, Affective Science, pages 21–46. OUP Oxford, 2010.
- [32] R. R. McCrae and O. P. John. An Introduction to the Five-Factor Model and its Applications. *Journal of personality*, 60(2):175–215, 1992.

- [33] A. Mehrabian. Analysis of the Big-Five Personality Factors in Terms of the PAD Temperament Model. *Australian Journal of Psychology*, 48(2):86–92, 1996.
- [34] A. Mehrabian. Pleasure-Arousal-Dominance: A General Framework for Describing and Measuring Individual Differences in Temperament. *Current Psychology*, 14(4):261–292, 1996.
- [35] A. Mehrabian and J. A. Russell. *An Approach to Environmental Psychology*. MIT Press, 1974.
- [36] J.-J. C. Meyer. Reasoning About Emotional Agents. *International Journal of Intelligent Systems*, 21(6):601–619, June 2006.
- [37] K. Nelson. The Psychological and Social Origins of Autobiographical Memory. *Psychological science*, 4(1):7–14, 1993.
- [38] M. Ochs, D. Sadek, and C. Pelachaud. A Formal Model of Emotions for an Empathic Rational Dialog Agent. *AAMAS*, 24(3):410–440, 2012.
- [39] A. Ortony. On Making Believable Emotional Agents Believable. In R. P. Trappé, P. Petta, and S. Payer, editors, *Emotions in Humans and Artifacts*, chapter 6, pages 189–212. MIT Press, 2003.
- [40] A. Ortony, G. L. Clore, and A. Collins. *The Cognitive Structure of Emotions*. Cambridge University Press, July 1988.
- [41] R. W. Picard and K. K. Liu. Relative Subjective Count and Assessment of Interruptive Technologies Applied to Mobile Monitoring of Stress. *International Journal of Human-Computer Studies*, 65(4):361–375, 2007.
- [42] C. Pimentel and M. Cravo. Affective Revision. In C. Bento, A. Cardoso, and G. Dias, editors, *Progress in Artificial Intelligence*, volume 3808 of *LNCS*, pages 115–126. Springer Berlin Heidelberg, 2005.
- [43] G. D. Plotkin. A Structural Approach to Operational Semantics. Technical Report DAIMI FN-19, Aarhus University, 1981.
- [44] A. S. Rao. AgentSpeak(L): BDI Agents Speak Out in a Logical Computable Language. In R. Van Hoe, editor, *Seventh European Workshop on Modelling Autonomous Agents in a Multi-Agent World*, Eindhoven, The Netherlands, 1996.
- [45] R. Reisenzein, E. Hudlicka, M. Dastani, J. Gratch, K. Hindriks, E. Lorini, and J.-J. Meyer. Computational Modeling of Emotion: Toward Improving the Inter- and Intradisciplinary Exchange. *IEEE Transactions on Affective Computing*, 4(3):246–266, 2013.
- [46] L.-F. Rodríguez and F. Ramos. Development of Computational Models of Emotions for Autonomous Agents: A Review. *Cognitive Computation*, 6(3):351–375, 2014.
- [47] I. J. Roseman. *A Model of Appraisal in the Emotion System: Integrating Theory, Research, and Applications*, pages 68–91. Oxford University Press, 2001.

- [48] J. A. Russell. Core Affect and the Psychological Construction of Emotion. *Psychological review*, 110(1):145–172, 2003.
- [49] K. R. Scherer. Appraisal Considered as a Process of Multilevel Sequential Checking. *Appraisal processes in emotion: Theory, methods, research*, 92:120, 2001.
- [50] N. Schwarz. Emotion, Cognition, and Decision Making. *Cognition & Emotion*, 14(4):433–440, 2000.
- [51] L. Selimbegović, I. Régner, P. Huguet, and A. Chatard. On the Power of Autobiographical Memories: From Threat and Challenge Appraisals to Actual Behaviour. *Memory*, pages 1–8, 2015.
- [52] M. Sewell. Emotions Help Solve the Prisoner’s Dilemma. In *Behavioural Finance Working Group Conference: Fairness, Trust and Emotions in Finance, London*, pages 1–2, 2010.
- [53] C. A. Smith and R. S. Lazarus. Emotion and Adaptation. In L. A. Pervin, editor, *Handbook of Personality: Theory and Research*, pages 609–637, 1990.
- [54] B. R. Steunebrink, M. Dastani, and J.-J. C. Meyer. A Formal Model of Emotion-Based Action Tendency for Intelligent Agents. In *Proceedings of EPIA’09*, pages 174–186, Berlin, Heidelberg, 2009. Springer-Verlag.
- [55] B. R. Steunebrink, M. Dastani, and J.-J. C. Meyer. A Formal Model of Emotion Triggers: an Approach for BDI Agents. *Synthese*, 185:83–129, 2012.
- [56] A. Tucker. The mathematics of Tucker: A sampler. *The Two-Year College Mathematics Journal*, 14(3):228–232, 1983.
- [57] R. Vieira, Á. F. Moreira, M. Wooldridge, and R. H. Bordini. On the Formal Semantics of Speech-Act Based Communication in an Agent-Oriented Programming Language. *J. Artif. Intell. Res.(JAIR)*, 29:221–267, 2007.
- [58] G. Weiss. *Multiagent Systems*. Intelligent robotics and autonomous agents. MIT Press, 2013.