VEsNA-Pro: Exploiting BDI Agents with Propensities for Emergent Narrative

Andrea Gatti
University of Genova
Genova, Italy
andrea.gatti@edu.unige.it

Fabio Casale
Untold Games
Genova, Italy
fabio@untoldgames.com

Viviana Mascardi
University of Genova
Genova, Italy
viviana.mascardi@unige.it

Andrea Stucchi
University of Genova
Genova, Italy
andrea.stucchi02@gmail.com

Angelo Ferrando
University of Modena-Reggio Emilia
Modena, Italy
angelo.ferrando@unimore.it

ABSTRACT

In many real applications including entertainment, training, serious games, immersing users and players in a virtual reality environment and making their interaction with virtual humans engaging and believable is the key for the application success and effectiveness.

Pro-AgentSpeak(L) extends the AgentSpeak(L) language for Belief-Desire-Intention (BDI) agents with the agents' *temperament*, *i.e.* those personality traits and attitudes that constitute their *propensities*. Unlike beliefs, that model the agents' epistemic state, propensities are related with their personality and emotional state. In order for these traits and emotional attitudes to drive deliberation, plans are annotated with weighted propensities and plan selection is guided by compatibility with the agent's current propensities; execution, in turn, updates mutable propensities through post-effects.

Two case studies designed with the Untold Games video game company show how propensities enable lightweight yet expressive agents' behavioural diversity and properly address emergence of realistic and engaging narrative.

KEYWORDS

Belief-Desire-Intention agents, BDI agents, AgentSpeak(L), Pro-AgentSpeak(L), temperament, personality traits, emotional state, propensities, plan selection, intention selection, video games

ACM Reference Format:

Andrea Gatti, Fabio Casale, Viviana Mascardi, Andrea Stucchi, and Angelo Ferrando. 2026. VEsNA-Pro: Exploiting BDI Agents with Propensities for Emergent Narrative. In *Proc. of the 25th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2026), Paphos, Cyprus, May 25 – 29, 2026*, IFAAMAS, 10 pages. https://doi.org/10.65109/

1 INTRODUCTION AND MOTIVATION

Alice and Bob are colleagues. Alice meets Bob in the office at 6.30 AM. She is very surprised, and asks him what he is doing there, so early in the morning. Bob says that he has a deadline: he is struggling to follow his schedule and he is getting almost stressed. Alice answers that she also has a deadline, but she attended a party all night long. She had a lot of fun there, made new friends, and does not care to be behind schedule.

From their interaction, we may guess that Bob's personality is mainly characterised by *conscientiousness* and some *neuroticism*, whereas Alice is highly *extroverted* and neither very conscientious, not very neurotic.

Carol and Dave reach the office at 8.30 AM. When Carol comes to know that Alice and Bob have troubles with their deadlines, she proposes to help them. Dave instead says that he cannot take care of the others' problems, and proposes to see an experimental theatre piece, that night.

From their reactions, we may guess that Carol is *agreeable* and Dave is not, but he is, instead, very *open to experience*.

From a human-agent interaction point of view, having agents equipped with their own personality represents a significant advantage, and several studies have shown that personality traits and emotional models influence users' perceptions and interaction quality [9, 10, 14]. In the specific context of video games, previous works have long emphasized that equipping Non-Player Characters (NPCs) with personality and emotions enhances their believability and contributes to emergent narrative experiences [24, 29, 33].

From the point of view of engineering a multiagent system (MAS), personality traits and attitudes are a suitable tool to drive the agents' behaviour. In particular, if we consider agents driven by Beliefs, Desires, Intentions (BDI agents [6]) implemented using AgentSpeak(L) [32], adding such traits to plans and selecting plans that are more aligned with the current agents' propensities may allow the MAS designers and developers to create compact, maintainable, and reusable libraries of behaviours.

In the above scenario, for example, the explanation for being in the office very early in the morning may be modelled by two different plans, both triggered by a request from a colleague, and both applicable under the condition that current time is earlier than 7.00 AM. One plan might suit highly conscientious and slightly neurotics agents, another might suit highly extroverted agents with low conscientiousness and neuroticism. Other combinations of personality traits might label other plans.

In current BDI-style agent programming, and in the Jason interpreter [3] for the BDI AgentSpeak(L) language in particular, a developer who wants to simulate agents types characterised by different personalities must either create different plan libraries for each different agent personality or keep the same plan library for

Proc. of the 25th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2026), C. Amato, L. Dennis, V. Mascardi, J. Thangarajah (eds.), May 25 – 29, 2026, Paphos, Cyprus. © 2026 International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org). https://doi.org/10.65109/

all of them, but hard-code selection strategies for each agent. This quickly becomes unmanageable: the plan library grows, behaviours become rigid, different plan libraries must be associated with different agent types lowering maintainability and increasing the risk of inconsistencies, and tuning or balancing across large groups of agents becomes costly [21].

We argue that *propensities* provide a lightweight and scalable alternative. Propensities are weighted annotations attached to both agents and plans that express personality traits such as – following the *OCEAN* (or Big Five) model [20] – being *Open*, *Conscientious*, *Extrovert*, *Agreeable*, *Neurotics* (immutable, or mutable only on the very long run), and mutable attitudes such are being upset, sad, cheerful, suspicious.

Such propensities are related in spirit to graded or value-based extensions of BDI agents [8, 11], and to emotion- or personality-based approaches to behavioural diversity [22, 23], but differ in being operationalised directly at the level of AgentSpeak(L) plan-case selection. The same plan library can be reused across all agents, while each agent's behaviour emerges from its current propensity profile. Moreover, some propensities can evolve over time: being often helped by colleagues may soften suspicion, while disdainful interactions reinforce it, echoing work on adaptive preferences and emotions in MAS [26, 39].

This separation of concerns — capabilities encoded in plans, propensities encoded as data — yields a practical engineering pattern. Developers gain behavioural diversity "for free", with less code to maintain and more flexibility to parameterise, tune, or sample entire populations of agents. The approach spans entertainment and serious applications (training, therapy), enabling adaptive, diverse behaviours; propensities provide a principled, lightweight way to achieve variety.

Contributions. This paper is the result of a joint effort involving academic scientists and professional video game developers from the awards-winning Untold Games company, both interested in assessing the potential of propensity-aware BDI agents for emergent narrative video games. We attack the problem from three directions.

(1) From a *theoretical* point of view, we (1.1) formalise the notion of *propensities* at the agent, plan, and intention levels in AgentSpeak(L), resulting into our Pro-AgentSpeak(L) extension; (1.2) extend the standard operational semantics of AgentSpeak(L) with *propensity-aware* plan- and intention-selection rules; (1.3) introduce a feedback loop where executing a plan updates the agent's mutable propensities through post-effects.

(2) From a *practical* point of view, we (2.1) implement Pro-AgentSpeak(L) using the Jason interpreter of AgentSpeak(L); (2.2) exploit Pro-AgentSpeak(L) to design and implement two emergent narrative case studies in the VEsNA environment [16] that bridges Jason with both the Unity [38] and the Godot [19] video game engines; one scenario shows the behaviour of agents joining a concert when unexpected events may take place, and the second is inspired to Alice and Bob office daily life and shows their conversational behaviour.

(3) From an *experimental* point of view, we (3.1) demonstrate the engineering benefits of Pro-AgentSpeak(L) in terms of scalability, in the concert scenario; (3.2) measure the believability of Pro-AgentSpeak(L) agents via an experiment where more than 90 participants assessed the agents' personality based on their conversational behaviour, in the office scenario.

The paper is organised in the following way: Section 2 introduces BDI agents programmed in AgentSpeak(L) and the VEsNA toolkit; Section 3 formalises the theory behind Pro-AgentSpeak(L); Section 4.1 presents design, implementation and experiments in the concert scenario, and Section 4.2 addresses the same aspects in the believable conversations scenario. Section 5 discusses the related work, and Section 6 concludes.

2 BACKGROUND: BDI AGENTS AND VESNA

AgentSpeak(L) is a logic programming language that provides an abstract framework for programming BDI agents [32]. We follow a presentation of AgentSpeak(L) syntax and operational semantics similar to [4] and [40]. Jason[1] is an open source interpreter for AgentSpeak(L).

The beliefs of an agent determine what an agent currently knows about itself, the other agents in the system, and the environment. They are defined as atomic formulae, as follows:

$$b \ ::= \ P(t_1, \dots, t_n) \quad (n \geq 0)$$

where P denotes a predicate symbol, and $t_1, \dots, t_n$ are standard terms of first-order logic. A belief base is a sequence of beliefs:

$$beliefs \ ::= \ b_1 \dots b_n \quad (n \geq 0)$$

The beliefs defined by the programmer at design time make up for the initial belief base. The rest of the beliefs are then added dynamically during the agent's lifetime.

An achievement goal in AgentSpeak(L) is specified as:

$$g \ ::= !at$$

where at is an atomic proposition. A query goal looks like $?at$.

Finally, an action in AgentSpeak(L) is defined as:

$$a \ ::= \ A(t_1, \dots, t_n) \quad (n \geq 0)$$

Plans are used to define the course of action for the agent to fulfil its goals. A plan has three main components: a triggering event te, denoting the event triggering the execution of the plan, a context $ctxt$, denoting the conditions that must hold to consider the plan applicable, and a body h consisting of a sequence of steps to be executed. A plan in AgentSpeak(L) is defined as:

$$p \ ::= \ te : ctxt \leftarrow h$$

The triggering event can be the addition (resp. deletion) of a belief b, and the addition (resp. deletion) of an achievement goal g. A plan is relevant for a triggering event if the event can be unified with the plan's head.

For a plan to be considered applicable the condition $ctxt$ must hold as a logical consequence of the agent's belief.

The body of a plan is composed of actions (a), belief updates ($+b$, $-b$), and goals (g).

An agent program contains a plain library with a set of plans.

Definition 2.1. Given a set of plans of an agent and a triggering event te, the set $RelPlans(plans, te)$ of relevant plans is:

$$\{p\phi \mid p \in plans \ \land \ \phi = mgu(te, TE(p))\}$$

with $\phi = mgu$ the most general unifier and $TE(p)$ the triggering event of the plan p.

Definition 2.2. Given a set of relevant plans R, and the beliefs of an agent, the set of applicable plans $AppPlans(R, beliefs)$ is:

[1] https://github.com/jason-lang/jason/releases

$$\{p\phi \mid p \in R \ \wedge \ \exists_{\phi}.beliefs \models ctxt(p)\phi\}$$
with $ctxt(p)$ being the context of plan p.

Relevant plans are all plans that could be triggered by the triggering event te. Applicable plans are the subset of the relevant plans that could be executed considering the agent's current state of mind.

An AgentSpeak(L) configuration C is a tuple $\langle I, E, A, R, Ap, \iota, \rho, \epsilon\rangle$ where: I is the set of intentions $\{i, i', \ldots\}$, with i as an intention stack of partially instantiated plans $[p_1|p_2 \ldots p_n]$. We use the $|$ symbol to separate plans in an intention stack. E is a set of events $\{\langle te, i\rangle, \langle te', i'\rangle, \ldots\}$. Each event is a pair $\langle te, i\rangle$, where te is a triggering event and i is an intention stack containing plans associated with te. A is a set of actions $\{\langle a, i\rangle, \langle a', i'\rangle, \ldots\}$. Each action is a pair $\langle a, i\rangle$, where a is an action and i is an intention stack containing plans associated with a. R is a set of relevant plans. Ap is a set of applicable plans. ι, ϵ and ρ keep the record of a particular intention, event and applicable plan (respectively) being considered in the current agent's reasoning cycle. To improve readability we rely on a compressed representation of the inference rules used in the operational semantics in [4, 40], moving some of the elements to C and omitting others that were not used or changed in our extended rules.

To keep the notation compact, we adopt the following: (i) if C is an AgentSpeak(L) configuration, we write C_I to make reference to the component I of C (same for the other components of C, such as C_E and so on); (ii) we write $C_\iota = \_$ to indicate there is no intention considered in the agent's execution (same for C_ρ and C_ϵ); and (iii) we write $i[p]$ to denote the intention stack that has p on its top. In the rest of the paper, we follow the standard AgentSpeak(L) notation, expanding only acronyms that may be non-obvious to readers unfamiliar with the language. The AgentSpeak(L) interpreter relies on three selection functions, each defined by the agent programmer. The event selection function S_E chooses an event from the set of events C_E; the applicable plan selection function S_{Ap} selects one plan from the set of applicable plans; and the intention selection function S_I selects an intention from the set of current intentions C_I, which is then executed. Formally, these selection functions are part of the agent's configuration, and will be affected by the Pro-AgentSpeak(L) extension.

VEsNA [16] is a general-purpose, open-source agent-based ecosystem[2] for managing *V*irtual *E*nvironments via *N*atural language *A*gents. It enables Jason agents to be embodied in virtual environments built with Godot (`vesna-light` repository) or Unity (`vesna-unity` repository, [7]), and it supports human–agent natural language interaction via ChatBDI (`chatbdi` repository, [17, 18]). We do not rely on ChatBDI in this work.

The `vesna.asl` file provides plans to `go_to(Target)` (makes the agent move to the target) and `follow_path([Path])` (makes the agent follow a path). Additionally, VEsNA agents can use three actions in the body of their plans:

- `vesna.walk()` (depending on the parameters, makes one step, or makes one step of length n, or moves to the target);
- `vesna.rotate()` (rotates in some direction, or towards a target);
- `vesna.jump()` (makes a jump).

[2]https://github.com/VEsNA-ToolKit

These internal actions are available for both the Godot and the Unity versions, making the same Jason "brain" usable for agents embodied in these two different Virtual Reality environments.

3 THEORY: PRO-AGENTSPEAK(L)

3.1 Formal Notation of Propensities

We now extend the formalism of AgentSpeak(L) with the notion of *propensities*, leading to the definition of Pro-AgentSpeak(L). Pro-AgentSpeak(L) extends the standard AgentSpeak(L) configuration C by associating agents, plans, and intentions with propensity annotations and by introducing propensity-aware selection functions, while leaving the remaining components of C unchanged.

Definition 3.1 (Agent's Propensity). Let $\mathcal{P}r$ be the set of propensities. They include disjoint immutable personality traits, $\mathcal{P}r_{immute}$, like being extroverted, and mutable temporary temper states, $\mathcal{P}r_{mute}$, like feeling uncomfortable.

$$\mathcal{P}r = \mathcal{P}r_{immute} \cup \mathcal{P}r_{mute} \text{ and } \mathcal{P}r_{immute} \cap \mathcal{P}r_{mute} = \emptyset$$

At time t, the *propensity of an agent a* is a mapping

$$A_t : \mathcal{P}r \ \longrightarrow \ [-1, +1],$$

where $A_t(pr)$ denotes the agent's current "amount" of $pr \in \mathcal{P}r$. +1 represents that a is fully pr (for example, fully open to experience, neurotics, sad, suspicious), -1 represents the reverse of pr (for example, being conscientious with weight -1 means being sloppy and unreliable).

This definition provides the baseline: an agent has, at every moment, a vector describing its stance towards all relevant propensities. Since propensities should directly inform deliberation, we annotate plans with them.

Definition 3.2 (Plan Propensity). Each plan p is associated with a subset of propensities $Pr_p \subseteq \mathcal{P}r$. The *propensity of plan p* is a function

$$A_p : Pr_p \ \longrightarrow \ [-1, +1],$$

where for every $pr \in Pr_p$, the value $A_p(pr)$ expresses the stance encoded in plan p toward pr. Dimensions outside Pr_p are considered irrelevant to p and thus unannotated (*i.e.*, $\forall_{pr \in \mathcal{P}r}.(pr \notin Pr_p \Rightarrow A_p(pr) = 0)$).

For instance, a plan to help colleagues to cope with their deadlines might carry a +1 *agreeable* propensity, while a plan to go to a party despite forthcoming deadlines might be annotated with $A_p(conscientious) = -1$.

Plans not only embody a predisposition but can also *change* the agent's mutable propensities once executed. To capture this, we define post-effect profiles.

Definition 3.3 (Plan Post-Effect). Each plan p also specifies a post-effect profile on (possibly different) mutable propensities $Pr_p^{\text{eff}} \subseteq \mathcal{P}r_{mute}$:

$$E_p : Pr_p^{\text{eff}} \rightarrow [-1, +1],$$

where $E_p(pr)$ is the *incremental change* that executing p tends to induce on the agent's mutable propensity pr; $E_p(pr) = 0$ means no change, the sign indicates direction, and the magnitude the strength of the change. Results exceeding the absolute value of 1 are clipped to 1 in absolute value.

In the office example, a plan aimed at harshly blaming colleagues for delays may increase their uneasiness and reduce their self-confidence. Note that discomfort is a mutable feeling, not a personality trait: while certain dispositions may make individuals more prone to specific emotions, anyone can feel sad, happy, relaxed, self-confident, or suspicious depending on context and prior experience. The current propensity is then updated by adding the post-effect to the previous value and clipping the result to the interval $[-1, 1]$.

Definition 3.4 (Propensity Update via Clipped Sum). If agent a selects plan p at time t, for each $pr \in \mathcal{P}r_{mute}$ the next weight of pr is

$$A_{t+1}(pr) \;=\; \begin{cases} \mathrm{clip}\big(A_t(pr) + E_p(pr)\big) & \text{if } pr \in Pr_p^{\mathrm{eff}}, \\ A_t(pr) & \text{otherwise,} \end{cases}$$

where $\mathrm{clip}(x) = \min\{1, \max\{-1, x\}\}$. Note that effects can only be defined on mutable propensities; furthermore, effects update only the mutable propensities of the agent executing the plan.

Lemma 3.5 (Boundedness and Neutrality). *For all $pr \in \mathcal{P}r_{mute}$, if $A_t(pr) \in [-1, +1]$ and $E_p(pr) \in [-1, +1]$, then $A_{t+1}(pr) \in [-1, +1]$. Moreover, $E_p(pr) = 0$ implies $A_{t+1}(pr) = A_t(pr)$.*

So far, we have defined propensities at the agent and plan levels. However, AgentSpeak(L) executes via intentions—stacks of partially instantiated/executed plans representing current commitments. To make propensities operational at this level, we aggregate the plan-level stances across the plans composing an intention.

Definition 3.6 (Intention Propensity). Let $i = [p_1 \mid p_2 \ldots p_n]$ be an intention, *i.e.*, a stack of partially instantiated and possibly partially executed plans. The *propensity of intention i* is a function

$$A_i : Pr_i \;\longrightarrow\; [-1, +1],$$

where $Pr_i = \bigcup_{k=1}^{n} Pr_{p_k}$ is the set of all propensities associated with *plan schemas* p_k from which the elements of the stack are derived, with p_1 being the plan that triggered the creation of the intention i. For each $pr \in Pr_i$, the value $A_i(pr)$ is obtained by aggregating the propensities of the corresponding plan schemas:

$$A_i(pr) = \mathrm{agg}\{A_{p_k}(pr) \mid pr \in Pr_{p_k},\ 1 \le k \le n\}.$$

Here agg denotes an aggregation operator chosen according to the agent design. It might be for example the average, weighted sum, maximum, or, in the simplest case, $A_i(pr) = A_{p_1}(pr)$, meaning that the propensity of an intention is the propensity of its first plan. Dimensions outside Pr_i are considered irrelevant to i and thus unannotated (*i.e.*, $\forall_{pr \in \mathcal{P}r}.(pr \notin Pr_i \Rightarrow A_i(pr) = 0)$).
Note that even when a plan instance p_k in the stack is only partially instantiated or executed, its propensity is inherited from the corresponding plan schema. Thus, propensities act as intrinsic annotations of plan cases, independent of execution progress.

3.2 Propensity-Oriented Plan Selection

Propensities come into play in the reasoning cycle when the agent must choose among alternatives. The first such choice is at the *plan-selection* stage. In the standard semantics [40], this is captured by rule SelAppl, where *AddIM* stands for *Add Intended Means*.

$$(\textsc{SelAppl})\,\frac{S_{AP}(C_{AP}) = \langle p, \phi\rangle}{C, SelAppl \rightarrow C', AddIM}$$

$$\text{where} \quad C'_\rho = \langle p, \phi\rangle$$

Here, S_{AP} is the programmer-defined function that picks one plan among the applicable set C_{AP}.

Semantic parameters. We assume:

- A compatibility measure $\mathrm{Compat}(A_t, A_p) \in [0, 1]$, expressing how well the agent's current propensity aligns with that encoded in a plan.
- A propensity-aware selection function

  $$S^A_{AP}(A_t, C_{AP}) := \arg \max_{\langle p,\phi\rangle \in C_{AP}} \mathrm{Compat}(A_t, A_p),$$

  with deterministic tie-breaking.

A key feature of our approach is that the compatibility function is *parameterisable*. Different compatibility measures can be plugged in depending on the desired behaviour, such as L1 similarity, dot product, cosine similarity.

L1 similarity. This measure favours plans whose stance is numerically *closest* to the agent's current disposition across all relevant objects. It strongly penalises even small mismatches, and treats over- and under-estimation symmetrically.

$$\mathrm{Compat}_{L1}(A_t, A_p) = 1 - \frac{1}{2|Pr_p|} \sum_{pr \in Pr_p} \big|A_t(pr) - A_p(pr)\big|.$$

Dot product. It favours plans that *align in polarity* with the agent's propensities (both positive or both negative), and is tolerant of differences in magnitude as long as the direction is the same.

$$\mathrm{Compat}_{\bullet}(A_t, A_p) = \frac{1}{2|Pr_p|}\left(\sum_{pr \in Pr_p} A_t(pr)\, A_p(pr) + |Pr_p|\right).$$

Cosine similarity. This measure captures the *angle* between the vectors, focusing purely on the degree of directional alignment irrespective of magnitude.

$$\mathrm{Compat}_{\cos}(A_t, A_p) =$$
$$\frac{1}{2}\left(\frac{\sum_{pr \in Pr_p} A_t(pr) \cdot A_p(pr)}{\sqrt{\sum_{pr \in Pr_p} A_t(pr)^2}\;\sqrt{\sum_{pr \in Pr_p} A_p(pr)^2}} + 1\right).$$

These alternatives illustrate that the framework is not tied to any specific metric. Rather, the compatibility function can be instantiated according to the needs of the application: favouring precise matching (Compat_{L1}), polarity alignment ($\mathrm{Compat}_{\bullet}$), or direction-only robustness ($\mathrm{Compat}_{\cos}$). This makes the approach flexible, allowing developers to select the similarity notion that best matches the intended agent behaviour.

This yields the following modified rule:

$$(\textsc{SelAppl}^{\mathrm{Pr}})\,\frac{S^A_{AP}(A_t, C_{AP}) = \langle p, \phi\rangle}{\langle C \mid A_t\rangle, SelAppl \rightarrow \langle C' \mid A_{t+1}\rangle, AddIM}$$

$$\text{where} \quad C'_\rho = \langle p, \phi\rangle$$
$$A_{t+1} \text{ is obtained as in Definition 3.4.}$$

To illustrate the effect of choosing one measure over another, let us consider our running example again. Suppose that Alice is not very conscientious (*con*), say, $A_t(con) = -0.4$. Two candidate plans with context `late_with_dl` are available to manage the currently selected event, `invitation_to_party`: `go_to_party`, annotated with $A_p(con) = -1.0$, and `switch_on_laptop`, annotated with $A_p(con) = +0.05$ (as switching the laptop on does not necessarily imply working). Under Compat_{L1}, switching on the laptop is preferred, as +0.05 is numerically closer to −0.4 than −1.0 is. Under $\text{Compat}_{\bullet}$, going to the party is preferred, because it shares the same non-conscientious polarity as Alice, while the other plan is penalised for pointing in the opposite direction. $\text{Compat}_{\cos}$ behaves similarly, focusing purely on direction and therefore also favouring the hedonistic plan. This example shows how the choice of compatibility metric can materially affect which plan the agent selects.

Stochastic variant. The deterministic definition of S_{AP}^{A} always chooses the plan with the highest compatibility score. While this ensures consistency, it can also make agent behaviour too rigid and predictable: the same circumstances will always trigger the same plan. To increase variability and believability, we introduce a stochastic alternative. Instead of selecting the single best plan, the agent samples from the set of applicable plans according to a weighted distribution where higher-compatibility plans are more likely to be chosen.

Formally, let $C_{AP} = \{\langle p_1, \phi_1\rangle, \ldots, \langle p_m, \phi_m\rangle\}$. In the stochastic variant, each applicable plan $\langle p_k, \phi_k\rangle$ is chosen with probability

$$\mathcal{P}\big(S_{AP}^{\text{Stoch}}(A_t, C_{AP}) = \langle p_k, \phi_k\rangle\big) \;=\; \frac{\text{Compat}(A_t, A_{p_k})}{\sum_{j=1}^{m} \text{Compat}(A_t, A_{p_j})}.$$

That is, plan selection corresponds to sampling from a *categorical distribution* whose weights are given by the compatibility values. Intuitively, this is equivalent to rolling a weighted dice: the more compatible a plan is with the agent's current propensities, the greater its chance of being selected, though less compatible alternatives are never entirely excluded. This randomness injects diversity across repeated runs, making agents appear less deterministic and more believable in domains such as games or training simulations.

The choice of compatibility measure is a modelling decision affecting behavioural style rather than semantics. Compatibility metrics are supplied components of the selector and may differ across agents, e.g. to reflect immutable personality traits: deterministic measures favour stable, personality-consistent behaviour, while stochastic or aggregating variants promote diversity and emergent narrative.

3.3 Propensity-Oriented Intention Selection

A second point where choice arises is the *intention-selection* stage, where the agent decides which active commitment to pursue next. The standard semantics [40] defines two rules, depending on whether the set of intentions is empty.

Standard rules.

$$(\textsc{SelInt}_1)\,\frac{S_I(C_I) = i}{C, SelInt \rightarrow C', ExecInt} \quad \text{where } C'_\iota = i$$

$$(\textsc{SelInt}_2)\,\frac{C_I = \emptyset}{C, SelInt \rightarrow C, ProcMsg}$$

Rule $\textsc{SelInt}_1$ chooses one intention i from C_I according to the programmer-defined function S_I. Rule $\textsc{SelInt}_2$ applies when no intentions are present, leaving the configuration unchanged (the reasoning cycle then restarts). For example, an employee may have to choose between having a longer coffee break and resuming working. Standard semantics requires a choice but does not prescribe how.

Propensity-based compatibility. To bias this choice with propensities, let A_t be the agent's propensity at time t. Each $i \in C_I$ is annotated with a propensity profile A_i (Definition 3.6). We assume a compatibility measure

$$\text{Compat}(A_t, A_i) \in [0, 1],$$

and define the propensity-aware selection function

$$S_I^{A}(A_t, C_I) := \arg\max_{i \in C_I} \text{Compat}(A_t, A_i),$$

with deterministic tie-breaking. Thus, intention selection is no longer neutral: the agent chooses the commitment most compatible with its current profile. For example, Bob might favour continuing an intention involving working alone w.r.t. an intention that is associated with meeting people, since the former aligns more closely with his negative extraversion.

Stochastic variant. Instead of deterministically picking the maximally compatible intention, we may *sample* from the set of intentions proportionally to their compatibility scores:

$$\Pr\big(S_I^{\text{Stoch}}(A_t, C_I) = i_k\big) \;=\; \frac{\text{Compat}(A_t, A_{i_k})}{\sum_{j=1}^{n} \text{Compat}(A_t, A_{i_j})}.$$

This corresponds to rolling a weighted dice: intentions closer to the agent's current profile are more likely to be chosen, but all retain a non-zero chance. Such stochasticity reduces determinism and produces greater behavioural diversity across repeated runs.

Fairness-aware extension. A potential issue with pure compatibility-based selection is *starvation*: intentions misaligned with the current profile may be indefinitely postponed. To address this, we define a hybrid score that combines compatibility with a fairness factor reflecting how long an intention has been waiting:

$$\text{Compat}^F(A_t, A_i, w_i) = (1-\alpha) \cdot \text{Compat}(A_t, A_i) + \alpha \cdot \text{Age}(i, t),$$

where w_i is the number of cycles intention i has been waiting, $\text{Age}(i, t) \in [0, \infty)$ is a monotonically increasing[3] function of w_i, and $\alpha \in (0, 1]$ balances propensities and fairness. The corresponding deterministic selection is

$$S_I^{AF}(A_t, C_I) := \arg\max_{i \in C_I} \text{Compat}^F(A_t, A_i, w_i).$$

A stochastic variant is obtained by sampling proportionally to Compat^F rather than raw compatibility, which preserves fairness (older intentions gain weight) while still reflecting attitudinal bias.

Theorem 3.7 (Fairness Guarantee). *If $\text{Age}(i, t)$ is monotonically increasing while i is not selected, is reset to 0 upon selection, then for any $\alpha > 0$ no intention i that persists in C_I can be indefinitely starved under S_I^{AF}.*

In practice, this means that even if the employee strongly prefers coffee breaks over work, fairness ensures that neglected obligations such as work duties will eventually be resumed.

Together, these extensions ensure that propensities shape agent deliberation at both the plan and intention levels, while still fitting naturally within the standard AgentSpeak(L) reasoning cycle.

[3] Unlike the plain compatibility score, Compat^F is unbounded; this is harmless since selection ($S_I^{AF}(A_t, C_I)$) uses arg max.

Throughout, we assume at least two relevant candidate plans ($k \geq 2$), since with a single plan no selection is needed and the issues we study do not arise. Furthermore, for brevity, the dependence of S^A_{AP} on the set of applicable plans C_{AP} is omitted and assumed to be fixed.

Theorem 3.8 (No finite pure-AgentSpeak(L) encoding of Pro-AgentSpeak(L) plan selection over dense propensities). *Let propensities range over an infinite set $D \subseteq [-1, 1]$. Fix a triggering event te and a finite set of relevant plan schemas $\{p_1, \dots, p_k\}$ with $k \geq 2$, each with fixed plan-propensity annotations A_{p_j}. Let S^A_{AP} be the Pro-AgentSpeak(L) applicable-plan selector for a fixed, well-defined compatibility measure and a fixed deterministic tie-breaking rule. Assume* pure AgentSpeak(L) *means:*

(1) *plan contexts may only test (in)equality of ground beliefs (no numeric comparisons over propensity values); and*
(2) *selection functions S_{AP} and S_I cannot access numeric propensity values.*

Then there is no finite *pure-AgentSpeak(L) plan library (i.e., finitely many plan variants and contexts for te) that realises the same mapping as S^A_{AP} for all $A_t \in D^{\mathcal{P}r}$.*

Corollary 3.9 (Intentions). *If intention selection S_I uses the same class of numeric compatibility measures over intention propensities (aggregated from plans), the analogue of Theorem 3.8 holds for S_I as well.*

Theorem 3.10 (Combinatorial blow-up under discretisation). *Let $d := |\mathcal{P}r|$ be the number of propensity dimensions. Discretise each dimension into $m \geq 2$ levels, so the agent propensity space is $\mathcal{X} = [m]^d$ with $|\mathcal{X}| = m^d$. For a given triggering event te, let $k \geq 2$ candidate plans $\{p_1, \dots, p_k\}$ be relevant. Consider the Pro-AgentSpeak(L) plan-selection function*

$$S^A_{AP} : \mathcal{X} \to \{p_1, \dots, p_k\}, \qquad S^A_{AP}(x) = \arg\max_j \mathrm{Compat}(x, A_{p_j})$$

with deterministic tie-breaking.

Any pure *AgentSpeak(L) program (as in Theorem 3.8) that reproduces the same input–output behaviour of S^A_{AP} on all $x \in \mathcal{X}$ requires, in the worst case,*

$$\Theta(|\mathcal{A}| \cdot m^d)$$

distinct guarded cases, where $\mathcal{A}$ is the set of distinct applicability patterns (i.e. subsets of $\{p_1, \dots, p_k\}$ that may be simultaneously applicable). In particular, $|\mathcal{A}| \leq 2^k - 1$.

In contrast, Pro-AgentSpeak(L) represents the same behaviour using only the k plan schemas annotated by $\{A_{p_j}\}_{j=1}^k$ together with a single shared selection rule, i.e. description size $O(k + d)$.

Proofs of Theorems 3.7, 3.8, and 3.10 are sketched in a technical appendix available online[4].

Remark 1 (On equivalence). *Together, Theorems 3.8 and 3.10 show that: (i) with dense propensities and no numeric tests, Pro-AgentSpeak(L) behaviour is not finitely encodable in pure AgentSpeak(L); and (ii) even after discretisation to m levels on d dimensions, a pure encoding suffers a worst-case blow-up proportional to $|\mathcal{A}| \cdot m^d$, where $\mathcal{A}$ is the set of distinct applicability patterns in the plan library. Since $|\mathcal{A}|$ may itself grow exponentially in the number of candidate plans k, the overhead can be exponential in both d and k. Equivalence can only be recovered by enriching pure AgentSpeak(L) with (a) numeric comparisons in plan contexts that can partition the propensity space, and (b) a selection function that consults propensity values — effectively re-introducing the Pro-AgentSpeak(L) machinery inside the base language.*

[4]https://github.com/VEsNA-ToolKit/vesna-pro/blob/main/docs/appendix.pdf

4 CASE STUDIES

In the following case studies, social and emotional aspects are represented as propensity dimensions, encoding both social tendencies and transient emotional states. These are handled uniformly through compatibility-based plan and intention selection, and no additional semantic machinery is introduced. An implementation of Pro-AgentSpeak(L) integrated with VEsNA is available as open-source software on GitHub[5].

4.1 Case Study I: The Concert Scenario

Design. To evaluate the behaviour of agents endowed with temperament and personality traits, we employ a *concert scenario* as a case study. This setting provides a socially rich yet computationally manageable environment in which individual differences and collective dynamics can be observed under both routine and unexpected conditions. A video of the scenario is publicly available online[6]. Each agent is characterised by a personality profile based on the OCEAN model. For simplicity, propensities do not decay over time in this scenario. These traits shape the agent's propensities, influencing action choice, social interaction, and emotional reactions to external events, so that agents placed in the same situation may behave differently. Agents inhabit a small virtual city and follow autonomous daily routines. Coloured agents implement full temperament modelling and engage in activities such as sleeping, commuting, taking breaks, working, and attending the concert, while grey agents represent simpler baseline entities with limited internal modelling that wander the environment and may also attend the concert. The concert is designed as a focal social event requiring minimal coordination: it only takes place if at least one musician and one spectator decide to participate. Once started, the concert's continuation depends on the agents' evolving conditions and interactions. In this scenario, *unexpected behaviour* is modelled as one or more agents **suddenly falling ill**, caused by the player's intervention via a mouse click. This provides a controllable yet realistic source of disturbance, revealing how the agents' personalities influence emotional contagion, decision adjustment, and collective robustness. Several variations emerge depending on when illness occurs throughout the day. If an agent becomes ill during the concert itself—whether a musician or spectator—the remaining participants adapt according to their social relationships and emotional states. The event continues only if participation requirements remain fulfilled. In a more severe case, when an agent becomes critically ill mid-performance, nearby altruistic agents intervene to assist, briefly interrupting the concert. Once the emergency is resolved, the performance resumes, illustrating how collective behaviour reorganises in response to perturbations.

[5]https://github.com/VEsNA-ToolKit/vesna-pro
[6]https://youtu.be/nKxd_6fw-QE

Implementation. The Pro-AgentSpeak(L) extension was realised in Jason by minimally modifying agent and plan annotations, post-effects, and selection functions. Propensities are stored in Java objects mirrored as beliefs, with plans annotated via `pr(...)` and `eff(...)` tags parsed at load time and updated through the clipped-sum rule. Plan selection applies compatibility-based similarity, while intention choice aggregates plan annotations with a fairness factor; both can run deterministically or stochastically. The implementation integrates seamlessly with VEsNA, enabling Godot or Unity embodiment without altering plan libraries. The concert scenario was implemented with Godot (code available in the Pro-AgentSpeak(L) repository on GitHub) and comprises **784 LoC and 133 plans for coloured agents** and **648 LoC and 96 plans for grey agents**. Core concert functionality includes coordinated events (113 LoC, 29 plans), the concert module (51 LoC, 13 plans), agent and NPC scripts (35 LoC, 2 plans), VEsNA movement (107 LoC, 22 plans), and map-belief construction (165 LoC). Broader behaviours extend this base with office life (43 LoC, 9 plans), friendships (35 LoC, 8 plans), illness handling (59 LoC, 9 plans), home routines (31 LoC, 7 plans), and daily activities (160 LoC, 35 plans for coloured; 107 LoC, 22 plans for grey agents). Overall, Pro-AgentSpeak(L) achieves rich, personality-driven behaviour in a fully modular way.

Results: Scalability Analysis. We evaluated the scalability of Pro-AgentSpeak(L) in the concert scenario by increasing the number of active agents up to **50**. Each agent maintained an individual OCEAN profile and autonomous deliberation loop, while communication and plan selection were executed in real time within the Godot environment through VEsNA. Preliminary results show that the overhead introduced by propensity evaluation and stochastic/fairness-aware deliberation grows approximately linearly with the number of agents. In our prototype, 50 fully autonomous agents ran smoothly at interactive frame rates on a standard laptop. Based on Untold Games experience, we know that 50 fully autonomous NPCs are never displayed together in a real video game. This would in fact confuse the player, whose attention can be conveyed towards less than 10 autonomous NPCs at a time, and waste resources. Our stress test suggests that Pro-AgentSpeak(L) scales efficiently and is suitable for larger multi-agent simulations and interactive games.

4.2 Case Study II: Believable Dialogues

Design. To test whether Pro-AgentSpeak(L) yields psychologically consistent and recognisable personalities in conversational settings, we generated multi-agent interaction logs in which agents with different OCEAN propensities engaged autonomously. The conversations were designed to highlight personality-dependent reactions emerging solely from the agents' internal propensities rather than from scripted rules.

Implementation. The same Pro-AgentSpeak(L)-enabled Jason architecture used for the concert scenario powered the conversational agents in the VEsNA framework. Each agent was configured with distinct OCEAN personality parameters shaping communicative style and plan preferences, while the deliberation mechanism—identical to that of the concert scenario—controlled turn-taking, speech-act selection, and emotional response. The dialogue domain comprises **46 plans** dealing with conversations involving Alice and Bob, taking place early in the morning in the office.

Functionally equivalent plans (e.g., multiple responses to party-related or overload dialogues) share triggering event and context but differ in their *propensity-combination tags*, each defining a temperament-specific variant. This tagging mechanism allows identical conversational structures to produce personality-dependent utterances and emotional nuances. As an example, **Conversation 1** includes the following sentences along with some contextual descriptions integrated in the plans, as in a theatrical script:

```
[alice] (soft smile, cup in hand) Good morning. You're in early.
[bob] (rubbing eyes, jittery) Morning! Barely slept - deadline's
chewing me up. Is this machine even working?
[alice] (steady tone) It is. Let's start with coffee and make a
simple plan - priorities, then small wins.
[bob] (talking fast) Okay... I kind of promised extra features last
night, and I'm on edge. It felt right in the moment.
[alice] (calm) I would trim scope. What's essential for today?
[bob] (shrugs) Uh... all of it? And also the fancy dashboard... I
told the client it would pop.
[alice] (matter-of-fact) Then we'll deliver the core and schedule
the dashboard for next sprint. I'll map tasks right after coffee.
[bob] (half-grin) You're ridiculously composed. How do you not freak
out? ......
```

Results: Believability Assessment. A total of 93 fully anonymous participants, recruited via invitations shared over various mailing lists and media channels, evaluated the conversations through an online questionnaire. Each participant freely selected one of four conversations and was asked to judge the personalities of the agents involved. For each dialogue, participants rated the agents OCEAN dimensions using an adapted version of the short Big Five Inventory [31] yielding a structured set of 93 annotated evaluations.

	Ans.	$L1_A$	$L1_B$	Dot_A	Dot_B	Cos_A	Cos_B
C1	19	0.60	0.76	0.54	0.61	**0.62**	**0.90**
C2	24	0.62	0.73	0.55	0.58	**0.72**	**0.85**
C3	32	0.72	0.72	0.64	0.57	**0.89**	**0.88**
C4	18	**0.55**	0.62	0.47	0.56	0.42	**0.74**

Table 1: Similarity of modelled and perceived propensities.

Figure 1 compares the *modelled* (blue) and *perceived* (red) trait profiles; Table 1 reports similarity scores using the same metrics as Section 3.2. Overall, profiles show good *directional* alignment (cosine): Extraversion and Neuroticism align in 7/8 cases, Conscientiousness in 6/8, Agreeableness in 5/8, while Openness is mostly misinterpreted (1/8), suggesting that curiosity/novelty cues were too subtle or context-dependent. By contrast, L1 values are lower, indicating that participants often inferred the correct polarity but over/underestimated *intensity*—a natural effect of brief exchanges.

Across conversations, **C3** yields the highest mean similarity (~0.75), as it blends task coordination with social cues (planning, reassurance, humour, curiosity), providing richer evidence for personality inference. In comparison, **C1** is cooperative but emotionally flat (moderate alignment), **C2** introduces irritation and anxiety (reduced consistency), and **C4** reverses emotional roles (intermediate coherence). Taken together, these results indicate that dialogues combining task-related and social–emotional signals enable more accurate human reconstruction of the modelled OCEAN profiles.

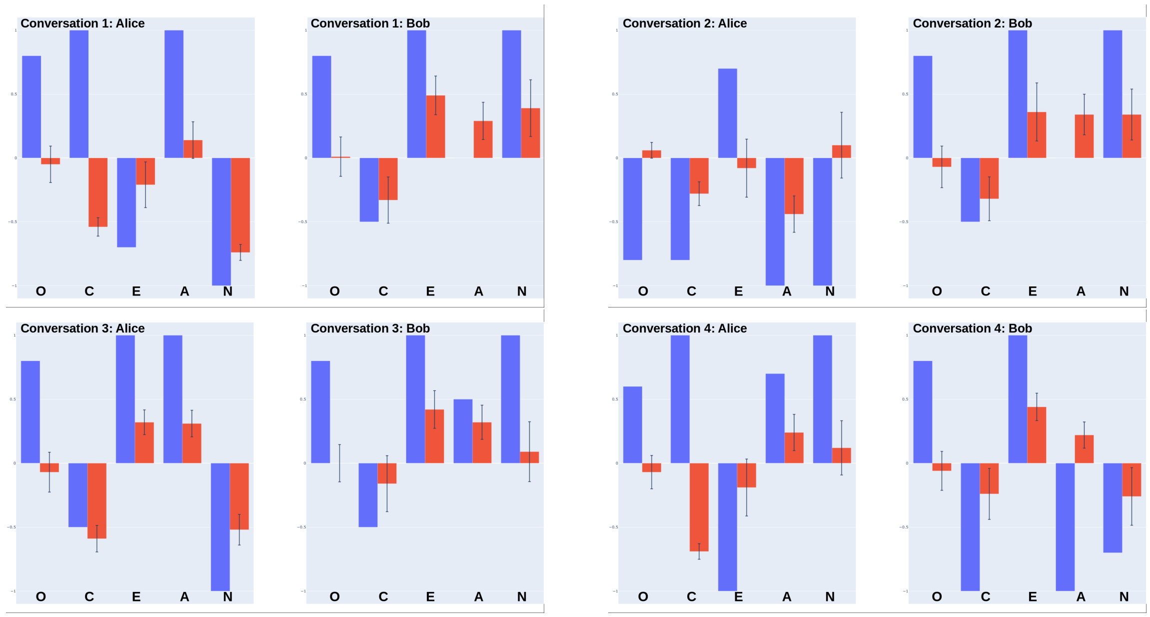

Figure 1: Detailed results on modelled (blue/leftmost bars) against perceived (red/rightmost bars) propensities.

5 RELATED WORK

Extensions of the BDI model have explored the integration of *values*, *preferences*, *norms*, *emotions*, *personality traits*, and *valuings*, each influencing deliberation at different points of the BDI control loop.

Values and valuings. Cranefield et al. operationalise *value*-based plan selection through societal or moral values guiding choice points [11]. Winikoff et al. model *valuings* to explain BDI behaviour [43]. In contrast, we introduce *propensities* as agent-centric, numerically combined dispositions that evolve through plan execution.

Preferences. Preferences have been used to direct plan selection and subgoal ordering [41, 42], or learned from experience—via context conditions [37], experiential plan choice [36], and situational preferences [28]. Dann et al. apply preference-guided Monte Carlo Tree Search to intention scheduling [13]. Unlike these, we attach *propensity weights* to plan *cases*, combine them with an agent's *propensity profile* at run time, and update them post-execution—forming a behaviour–disposition feedback loop rather than learning preference rules.

Graded mental propensities and norms. Casali et al.'s graded BDI model attaches degrees to beliefs, desires, and intentions for preference reasoning [8]. Criado et al. extend graded BDI with normative deliberation [12], and Meneguzzi et al. study BDI reasoning with normative considerations [26]. Our model is orthogonal: we keep mental propensities and norms categorical but weight *plan cases* by evolving *propensities*.

Affect and personality. Emotions have been used to bias intention selection (EBDI) [22], integrate physiology and affect (PEP! → !BDI) [23], or resolve normative conflicts via personality traits [1]. Affect has also informed crowd models [39], emotion regulation [30], and appraisal-based affect control [35]. Surveys highlight open issues in emotional BDI design [34]. While related, these treat affect or personality as global modulators; we make *propensities* first-class, plan-level selectors that update through experience.

Decision-theoretic selection. Decision-theoretic intention scheduling in AgentSpeak(XL) [2] optimises intention choice. We instead focus on *plan-case selection* driven by evolving propensities.

Summary. While prior work shapes BDI decision making via values, norms, affect, and decision-theoretic control, we instead (i) weight *plan cases* with *propensities*, (ii) compose them with agent-specific profiles at run time, and (iii) update them post-execution, forming a behaviour⟶disposition feedback loop.

Beyond BDI, affective and personality-driven agents have been studied through fuzzy emotion, appraisal, and regulation models [5, 15, 27], as well as narrative character-affinity models [25]. Unlike these procedural or story-level approaches, we embed traits and affective states as propensities directly in AgentSpeak(L) plan and intention selection.

6 CONCLUSIONS AND FUTURE WORK

We presented Pro-AgentSpeak(L), an extension of AgentSpeak(L) that equips BDI agents with *propensities*, unifying personality traits and temper states to guide plan and intention selection, producing diverse yet coherent behaviour from a shared plan library. Integrated into VEsNA [16], it enables personality-consistent, adaptive behaviour in interactive environments.

Future work will replace the current post-effect mechanism with adaptive learning, enabling agents to refine temperament through experience. We will integrate affective appraisal models so transient emotions shape deliberation via long-term propensities [22, 30, 35]. Finally, we will extend the framework to domains such as negotiation, teamwork, and education, and formalise its semantics to capture learning and affective dynamics, building on quantitative and value-based BDI models [8, 11, 13, 26, 42].

REFERENCES

[1] Paulo Henrique Cardoso Alves, Marx L. Viana, and Carlos José Pereira de Lucena. 2018. An Architecture for Autonomous Normative BDI Agents based on Personality Traits to Solve Normative Conflicts. In *Proceedings of ICAART 2018, Volume 1*, Ana Paula Rocha and H. Jaap van den Herik (Eds.). SciTePress, 80–90. https://doi.org/10.5220/0006599300800090

[2] Rafael H. Bordini, Ana L. C. Bazzan, Rafael de Oliveira Jannone, Daniel M. Basso, Rosa Maria Vicari, and Victor R. Lesser. 2002. AgentSpeak(XL): efficient intention selection in BDI agents via decision-theoretic task scheduling. In *The First International Joint Conference on Autonomous Agents & Multiagent Systems, AAMAS 2002*. ACM, 1294–1302. https://doi.org/10.1145/545056.545122

[3] Rafael H. Bordini, Hübner, and Michael J. Wooldridge. 2007. *Programming multi-agent systems in AgentSpeak using Jason*. J. Wiley.

[4] Rafael H. Bordini and Jomi Fred Hübner. 2010. Semantics for the Jason Variant of AgentSpeak (Plan Failure and some Internal Actions). In *19th European Conference on Artificial Intelligence (Frontiers in Artificial Intelligence and Applications, Vol. 215)*. IOS Press, 635–640. https://doi.org/10.3233/978-1-60750-606-5-635

[5] Tibor Bosse, Matthijs Pontier, and Jan Treur. 2010. A computational model based on Gross' emotion regulation theory. *Cogn. Syst. Res.* 11, 3 (2010), 211–230. https://doi.org/10.1016/J.COGSYS.2009.10.001

[6] Michael Bratman. 1987. *Intention, Plans, and Practical Reason*. Cambridge, MA: Harvard University Press, Cambridge.

[7] Daniela Briola, Giuseppe Vizzari, Giuseppe Mirra, Andrea Gatti, Matteo Martini, and Viviana Mascardi. 2025. Integrating JaCa-Unity in VEsNA for Behavioural Realistic VR Simulations. In *22nd European Conference on Multi-Agent Systems (EUMAS 2025)*. Springer.

[8] Ana Casali, Lluís Godo, and Carles Sierra. 2011. A graded BDI agent model to represent and reason about preferences. *Artif. Intell.* 175, 7-8 (2011), 1468–1478. https://doi.org/10.1016/J.ARTINT.2010.12.006

[9] Aleksandra Cerekovic, Oya Aran, and Daniel Gatica-Perez. 2014. How Do You Like Your Virtual Agent?: Human-Agent Interaction Experience through Nonverbal Features and Personality Traits. In *Human Behavior Understanding - 5th International Workshop, HBU 2014, Zurich, Switzerland, September 12, 2014. Proceedings (Lecture Notes in Computer Science, Vol. 8749)*, Hyun Soo Park, Albert Ali Salah, Yong Jae Lee, Louis-Philippe Morency, Yaser Sheikh, and Rita Cucchiara (Eds.). Springer, 1–15. https://doi.org/10.1007/978-3-319-11839-0_1

[10] Kushal Chawla, Ian Wu, Yu Rong, Gale M. Lucas, and Jonathan Gratch. 2023. Be Selfish, But Wisely: Investigating the Impact of Agent Personality in Mixed-Motive Human-Agent Interactions. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, Houda Bouamor, Juan Pino, and Kalika Bali (Eds.). Association for Computational Linguistics, 13078–13092. https://doi.org/10.18653/V1/2023.EMNLP-MAIN.808

[11] Stephen Cranefield, Michael Winikoff, Virginia Dignum, and Frank Dignum. 2017. No Pizza for You: Value-based Plan Selection in BDI Agents. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017, Melbourne, Australia, August 19-25, 2017*, Carles Sierra (Ed.). ijcai.org, 178–184. https://doi.org/10.24963/IJCAI.2017/26

[12] Natalia Criado, Estefania Argente, and Vicente J. Botti. 2010. Normative Deliberation in Graded BDI Agents. In *Multiagent System Technologies, 8th German Conference, MATES 2010, Leipzig, Germany, September 27-29, 2010. Proceedings (Lecture Notes in Computer Science, Vol. 6251)*, Jürgen Dix and Cees Witteveen (Eds.). Springer, 52–63. https://doi.org/10.1007/978-3-642-16178-0_7

[13] Michael Dann, John Thangarajah, and Minyi Li. 2023. Feedback-Guided Intention Scheduling for BDI Agents. In *Proceedings of the 2023 International Conference on Autonomous Agents and Multiagent Systems, AAMAS 2023*, Noa Agmon, Bo An, Alessandro Ricci, and William Yeoh (Eds.). ACM, 1173–1181. https://doi.org/10.5555/3545946.3598760

[14] Pierangelo Dell'Acqua and Stefania Costantini. 2023. Empathetic human-agent interaction via emotional behavior trees. *Intelligenza Artificiale* 17, 1 (2023), 89–100. https://doi.org/10.3233/IA-230014

[15] Magy Seif El-Nasr, John Yen, and Thomas R. Ioerger. 2000. FLAME-Fuzzy Logic Adaptive Model of Emotions. *Auton. Agents Multi Agent Syst.* 3, 3 (2000), 219–257. https://doi.org/10.1023/A:1010030809960

[16] Andrea Gatti and Viviana Mascardi. 2023. VEsNA, a Framework for Virtual Environments via Natural Language Agents and Its Application to Factory Automation. *Robotics* 12, 2 (2023), 46. https://doi.org/10.3390/ROBOTICS12020046

[17] Andrea Gatti, Viviana Mascardi, and Angelo Ferrando. 2025. ChatBDI: Think BDI, Talk LLM. In *24th International Conference on Autonomous Agents and Multiagent Systems, AAMAS 2025*, Y. Vorobeychik, S. Das, and A. Nowé (Eds.). International Foundation for Autonomous Agents and Multiagent Systems.

[18] Andrea Gatti, Viviana Mascardi, and Angelo Ferrando. 2025. Let Me Talk to you! Natural Language Interaction between Humans and BDI Agents via ChatBDI. In *28th European Conference on Artificial Intelligence, ECAI 2025*, Inês Lynce, Nello Murano, Mauro Vallati, Federico Chesani Serena Villata, Michela Milano, Andrea Omicini, and Mehdi Dastani (Eds.).

[19] Godot foundation. 2014. *Godot web site*. https://godotengine.org/ Accessed on February 16, 2026.

[20] Lewis Goldberg. 1990. An Alternative "Description of Personality": The Big-Five Factor Structure. *Journal of Personality and Social Psychology* 59 (12 1990), 1216–1229. https://doi.org/10.1037/0022-3514.59.6.1216

[21] Nicholas R. Jennings. 2000. On agent-based software engineering. *Artif. Intell.* 117, 2 (2000), 277–296. https://doi.org/10.1016/S0004-3702(99)00107-1

[22] Hong Jiang, José M. Vidal, and Michael N. Huhns. 2007. EBDI: an architecture for emotional agents. In *6th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2007)*, Edmund H. Durfee, Makoto Yokoo, Michael N. Huhns, and Onn Shehory (Eds.). IFAAMAS, 11. https://doi.org/10.1145/1329125.1329139

[23] Hazaël Jones, Julien Saunier, and Domitile Lourdeaux. 2009. Personality, Emotions and Physiology in a BDI Agent Architecture: The PEP ->> BDI Model. In *Proceedings of the 2009 IEEE/WIC/ACM International Conference on Intelligent Agent Technology, IAT 2009, Milan, Italy, 15-18 September 2009*. IEEE Computer Society, 263–266. https://doi.org/10.1109/WI-IAT.2009.160

[24] Michael Mateas and Andrew Stern. 2003. Façade: An Experiment in Building a Fully-Realized Interactive Drama. https://api.semanticscholar.org/CorpusID:8567404

[25] Gonzalo Méndez, Pablo Gervás, and Carlos León. 2014. On the Use of Character Affinities for Story Plot Generation. In *Knowledge, Information and Creativity Support Systems - Selected Papers from KICSS'2014 - 9th International Conference, held in Limassol, Cyprus, on November 6-8, 2014 (Advances in Intelligent Systems and Computing, Vol. 416)*, Susumu Kunifuji, George Angelos Papadopoulos, Andrzej M. J. Skulimowski, and Janusz Kacprzyk (Eds.). Springer, 211–225. https://doi.org/10.1007/978-3-319-27478-2_15

[26] Felipe Meneguzzi, Odinaldo Rodrigues, Nir Oren, Wamberto Weber Vasconcelos, and Michael Luck. 2015. BDI reasoning with normative considerations. *Eng. Appl. Artif. Intell.* 43 (2015), 127–146. https://doi.org/10.1016/J.ENGAPPAI.2015.04.011

[27] Suman Ojha, Jonathan Vitale, and Mary-Anne Williams. 2021. Computational Emotion Models: A Thematic Review. *Int. J. Soc. Robotics* 13, 6 (2021), 1253–1279. https://doi.org/10.1007/S12369-020-00713-1

[28] Lin Padgham and Dhirendra Singh. 2013. Situational preferences for BDI plans. In *International conference on Autonomous Agents and Multi-Agent Systems, AAMAS'13*, Maria L. Gini, Onn Shehory, Takayuki Ito, and Catholijn M. Jonker (Eds.). IFAAMAS, 1013–1020. http://dl.acm.org/citation.cfm?id=2485080

[29] Ana Paiva, Iolanda Leite, Hana Boukricha, and Ipke Wachsmuth. 2017. Empathy in Virtual Agents and Robots: A Survey. *ACM Trans. Interact. Intell. Syst.* 7, 3 (2017), 11:1–11:40. https://doi.org/10.1145/2912150

[30] Aaron Pico, Joaquín Taverner, Emilio Vivancos, Vicent J. Botti, and Ana García-Fornes. 2024. Towards an Affective Intelligent Agent Model for Extrinsic Emotion Regulation. *Syst.* 12, 3 (2024), 77. https://doi.org/10.3390/SYSTEMS12030077

[31] Beatrice Rammstedt and Oliver P. John. 2007. Measuring personality in one minute or less: A 10-item short version of the Big Five Inventory in English and German. *Journal of Research in Personality* 41, 1 (2007), 203–212. https://doi.org/10.1016/j.jrp.2006.02.001

[32] Anand S. Rao. 1996. AgentSpeak(L): BDI Agents Speak Out in a Logical Computable Language. In *7th European Workshop on Modelling Autonomous Agents in a Multi-Agent World, Eindhoven, The Netherlands, January 22-25, 1996 (Lecture Notes in Computer Science, Vol. 1038)*. Springer, 42–55. https://doi.org/10.1007/BFb0031845

[33] Mark O. Riedl and Andrew Stern. 2006. Believable Agents and Intelligent Story Adaptation for Interactive Storytelling. In *Technologies for Interactive Digital Storytelling and Entertainment, Third International Conference, TIDSE 2006, Darmstadt, Germany, December 4-6, 2006 (Lecture Notes in Computer Science, Vol. 4326)*, Stefan Göbel, Rainer Malkewitz, and Ido Iurgel (Eds.). Springer, 1–12. https://doi.org/10.1007/11944577_1

[34] Yanet Sánchez and Eva Cerezo. 2019. Designing emotional BDI agents: good practices and open questions. *Knowl. Eng. Rev.* 34 (2019), e26. https://doi.org/10.1017/S0269888919000122

[35] Yanet Sánchez, Teresa Coma, Antonio Aguelo, and Eva Cerezo. 2019. ABC-EBDI: An affective framework for BDI agents. *Cogn. Syst. Res.* 58 (2019), 195–216. https://doi.org/10.1016/J.COGSYS.2019.07.002

[36] Dhirendra Singh, Sebastian Sardiña, and Lin Padgham. 2010. Extending BDI plan selection to incorporate learning from experience. *Robotics Auton. Syst.* 58, 9 (2010), 1067–1075. https://doi.org/10.1016/J.ROBOT.2010.05.008

[37] Dhirendra Singh, Sebastian Sardiña, Lin Padgham, and Stéphane Airiau. 2010. Learning context conditions for BDI plan selection. In *9th International Conference on Autonomous Agents and Multiagent Systems, AAMAS 2010*, Wiebe van der Hoek, Gal A. Kaminka, Yves Lespérance, Michael Luck, and Sandip Sen (Eds.). IFAAMAS, 325–332. https://dl.acm.org/citation.cfm?id=1838252

[38] Unity Technologies. 2005. *Unity web site*. https://unity.com/ Accessed on February 16, 2026.

[39] Marion Valette, Benoit Gaudou, Dominique Longin, and Patrick Taillandier. 2018. Modeling a Real-Case Situation of Egress Using BDI Agents with Emotions and Social Skills. In *PRIMA 2018: Principles and Practice of Multi-Agent Systems - 21st International Conference, Tokyo, Japan, October 29 - November 2, 2018, Proceedings (Lecture Notes in Computer Science, Vol. 11224)*, Tim Miller, Nir Oren, Yuko Sakurai, Itsuki Noda, Bastin Tony Roy Savarimuthu, and Tran Cao Son (Eds.). Springer,

3–18. https://doi.org/10.1007/978-3-030-03098-8_1

[40] Renata Vieira, Álvaro F. Moreira, Michael J. Wooldridge, and Rafael H. Bordini. 2007. On the Formal Semantics of Speech-Act Based Communication in an Agent-Oriented Programming Language. *J. Artif. Intell. Res.* 29 (2007), 221–267. https://doi.org/10.1613/jair.2221

[41] Simeon Visser, John Thangarajah, and James Harland. 2011. Reasoning about Preferences in Intelligent Agent Systems. In *IJCAI 2011, Proceedings of the 22nd International Joint Conference on Artificial Intelligence, Barcelona, Catalonia, Spain, July 16-22, 2011*, Toby Walsh (Ed.). IJCAI/AAAI, 426–431. https://doi.org/10.5591/978-1-57735-516-8/IJCAI11-079

[42] Simeon Visser, John Thangarajah, James Harland, and Frank Dignum. 2016. Preference-based reasoning in BDI agent systems. *Auton. Agents Multi Agent Syst.* 30, 2 (2016), 291–330. https://doi.org/10.1007/S10458-015-9288-2

[43] Michael Winikoff, Galina Sidorenko, Virginia Dignum, and Frank Dignum. 2021. Why bad coffee? Explaining BDI agent behaviour with valuings. *Artif. Intell.* 300 (2021), 103554. https://doi.org/10.1016/J.ARTINT.2021.103554