An Inferential Dynamics Approach to Personality and Emotion Driven Behavior Determination for Virtual Animals

Ben Goertzel and **Cassio Pennachin**

Abstract. The problem considered is how to provide virtual animals, living in an online virtual world, with internal personality and emotion structures that will lead them to display behaviors perceived as naturalistic and emotionally compelling by humans controlling avatars that interact with the virtual animals. A novel approach is proposed, in which both spontaneous and goal-driven animal behaviors are governed by a set of probabilistic logic implications, which are forward and backward chained together, both to directly guide action selection, and to adjust the values of internal emotion indicators. The approach has been prototyped in an preliminary version of a system that controls virtual animals in Second Life, and is expected to be included in a commercial virtual-animals product later in 2008.

1 INTRODUCTION

We consider the problem of creating virtual animals, resident in a 3D virtual world such as Second Life, which not only learn and adapt their behavior based on training and experience, but also possess distinctive personalities and fluid emotional responses, sufficient to intrigue and emotionally engage the humans who interact with them (by means of their avatars). The approach we describe here is based on a tight integration of emotion and personality with other aspects of "virtual animal psyche," within an integrative Virtual Animal Brain (VAB) architecture. At present a prototype of this architecture has been constructed and is the subject of testing and experimentation; the ultimate goal of the project within which it has been created is the launch of a commercial virtual animal product within Second Life and potentially other virtual worlds as well.

The approach taken here is novel in several respects, most notably in its integration of logical and dynamical methods. In the VAB, an animal's behavior is controlled by a combination of procedures (represented internally in a dag form, corresponding to human-readable scripts in a LISP-like language), and probabilistic-logical implications. There are methods for converting back and forth between these procedural and declarative representations as necessary. Currently, learned behaviors such as "tricks" are represented procedurally, whereas relationships between personality traits, emotions and behaviors are represented declaratively as implications. The learning aspect of the VAB has been described in detail elsewhere [1]; so here, after a brief review of the VAB overall, we focus on explicating how the system of implications is used to regulate emotion and behavior. Iterated forward and backward chaining probabilistic inference, using these implications, play the role of "update equations" updating the states of internal emotional variables and behavioral propensities. These equations modify behaviors, which in turn lead to shifts in emotional state directly, which affect the outputs of the implications, thus leading to an overall nonlinear dynamic coupling the animal's mind with its behaviors.

This approach is somewhat complex, but the end result of this complexity is a richness of emotion and personality driven behavior that seems (based on our own experimentation) to be more difficult to achieve with simpler and more straightforward approaches. Our preliminary experimentation suggests that animals governed by the approach presented here may be interesting and appealing to interact with; but the final test, of course, will be after product release occurs.

It's worth noting that the approach is also highly configurable, as the basic logical implications on which it is based may be easily customized by nontechnical individuals. This gives rise to the possibility (which will likely not be realized in our initial product releases) that eventually end-users may be able to enter new implications textually or graphically, thus configuring the personality and emotional makeup of animals that serve as their pets or relate to them in other ways. Finally, there are ample possibilities for further extensions, such as using advanced probabilistic logic to learn new emotion/personality/behavior implications via experience, generalization, analogy and so forth.

2 THE NOVAMENTE COGNITION ENGINE

The VAB is a simplified, specialized version of a broader AI architecture called the Novamente Cognition Engine (NCE) [2,3], which is aimed beyond the domain of virtual animals, toward powerful artificial general intelligence [4,5].

One may conceptualize the NCE in the context of the overall task of creating a powerful AGI system, which we decompose into four aspects (which of course are not entirely distinct, but still are usefully distinguished):

1. Cognitive architecture (the overall design of an AGI system: what parts does it have, how do they connect to each other)
2. Knowledge representation (how does the system internally store declarative, procedural and episodic knowledge; and how does it create its own representation for knowledge of these sorts in new domains it encounters)
3. Learning (how does it learn new knowledge of the types mentioned above; and how does it learn how to learn, and so on)

4. Teaching methodology (how is it coupled with other systems so as to enable it to gain new knowledge about itself, the world and others)

We now briefly review how these four aspects are handled in the NCE. For a more in-depth discussion of the NCE the reader is referred to [2,3].

The NCE's high-level cognitive architecture is motivated by human cognitive science and is roughly analogous to Stan Franklin's LIDA architecture [6]. It consists of a division into a number of interconnected functional units corresponding to different specialized capabilities such as perception, motor control and language, and also an "attentional focus" unit corresponding to intensive integrative processing. A diagrammatic depiction is given in [2].

Within each functional unit, knowledge representation is enabled via an AtomTable software object that contains nodes and links (collectively called Atoms) of various types representing declarative, procedural and episodic knowledge both symbolically and subsymbolically. Each unit also contains a collection of MindAgent objects implementing cognitive, perception or action processes that act on this AtomTable, and/or interact with the outside world.

One of the most important types of Atoms is the PredicateNode, which represents a logical predicate evaluated on certain inputs. Emotions, which will play a significant role in our discussion here, are represented as 0-ary predicates, which have a truth value at each time calculated via fixed internal code representing the "biological" grounding of the emotion. Emotion predicates may also be updated via application of logical rules, as will be described below. These logical rules take the role of ImplicationLinks (representing probabilistic logical implications) joining combinations of PredicateNodes to each other, where combinations of PredicateNodes are represented in terms of AndLinks, OrLinks and NotLinks joining PredicateNodes.

In addition to a number of specialized learning algorithms associated with particular functional units, the NCE is endowed with two powerful learning mechanisms embedded in MindAgents: the MOSES probabilistic-program-evolution module (based on [7]), and the Probabilistic Logic Networks module for probabilistic logical inference [8,9]. These are used both to learn procedural and declarative knowledge, and to regulate the attention of the MindAgents as they shift from one focus to another, using an economic attention-allocation mechanism that leads to subtle nonlinear dynamics and associated emergent complexity including spontaneous creative emergence of new concepts, plans, procedures, etc.

Finally, regarding teaching methodology, we advocate a virtually-embodied approach which integrates linguistic with nonlinguistic instruction, and also autonomous learning via spontaneous exploration of the virtual world. And this is where the subject of the present paper comes in: personality and emotion, via their impact on behavior, are key to establishing appropriate interactions with other agents, so as to encourage an embodied AI system's ongoing learning as growth (as well as achieving other goals such as making the AI system more appealing for humans to interact with).

3 AN ARCHITECTURE FOR INTELLIGENT VIRTUAL ANIMALS

Figure 1. Screenshot of a virtual animal in Second Life, controlled by the NCE-based AGI architecture described in this section.

In this section we briefly describe our current, preliminary experimental work using a simplified version of the Novamente Cognition Engine (the so-called "Virtual Animal Brain" or VAB) to control virtual animals in the Second Life virtual world. Figure 1 above shows an example virtual animal controlled by the VAB, interacting with a human-controlled avatar in the context of learning to play soccer. Figure 2 gives a high-level architecture diagram for the VAB, which is a simplification of the overall NCE architecture as diagrammed in [2].

The capabilities of the VAB-controlled virtual animals, in their current form, include

- Spontaneous exploration of the environment
- Automated enactment of a set of simple predefined behaviors
- Flexible trainability: i.e., (less efficient) learning of behaviors invented by teachers on the fly
- Communication with the animals, for training of new behaviors and a few additional purposes, occurs in a special subset of English called ACL (Animal Command Language)
- Individuality: each animal has its own distinct personality
- Spontaneous learning of new behaviors, without need for explicit training
- Capabilities intended to be added in future VAB versions include
- Recognition of novel categories of objects, and integration of object recognition into learning
- Generalization based on prior learning, so as to be able to transfer old tricks to new contexts
- Use of computational linguistics to achieve a more flexible conversational facility

The VAB architecture is not particular to Second Life, but up till now has been guided somewhat by the particular limitations of Second Life. In particular, Second Life does not conveniently lend itself to highly detailed perceptual and motoric interaction, so we have not dealt with issues related to these in the current version of the VAB. However, we have dealt with

some of these issues in a prior version of the VAB, which was connected to the AGISim framework, a wrapper for the open-source game engine CrystalSpace [10].

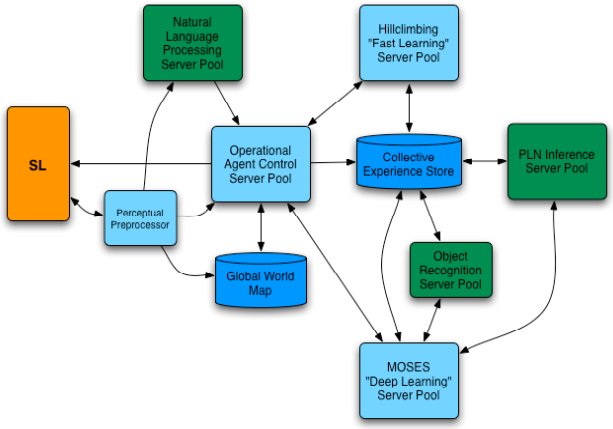

Figure 2. High-level diagram depicting VAB software architecture. The NLP, object recognition and PLN components are missing from the architecture that will initially be commercially deployed but are present in Novamente LLC's internal research codebase.

Instruction of VAB-controlled agents takes place according to a methodology we call IRC learning and is described in detail in [1], involving three interacting aspects:

- *Imitative* learning: The teacher acts out a behavior, showing the student by example what he wants the student to do
- *Reinforcement* learning: The student tries to do the behavior himself, and the teacher gives him feedback on how well he did
- *Corrective* learning: As the student attempts the behavior, the teacher actively corrects (i.e. changes) the student's actions, guiding him toward correct performance

The combination of these three sorts of instruction appears to us critical, for learning of complex embodied behaviors and also, further along, for language learning. Current experimentation with the IRC methodology has been interesting and successful, resulting in a framework allowing human-controlled avatars to teach VAB-controlled agents a variety of behaviors such as fetching objects, delivering objects, going to desired locations, doing dances, and so forth. Further detail is given in [1]; our present treatment is focused on the emotion and personality aspects of the system.

.

4 MODELING EMOTION AND PERSONALITY

Psychological theories of emotion are numerous and diverse, and it seems likely that many of the available theories capture relevant aspects of the emotion phenomenon as it occurs in humans and other animals. The NCE architecture itself is flexible enough to support a variety of approaches to AI emotion; a theoretical analysis of related issues is given in chapter [11]. For the purpose of the VAB project, however, we have opted for a relatively simplistic approach, drawing directly on the ontology of emotions supplied in [12]. Based on a deep and rigorous analysis of the logical structure of emotional experience, [12] propose an emotional ontology that is well summarized in Figure 3:

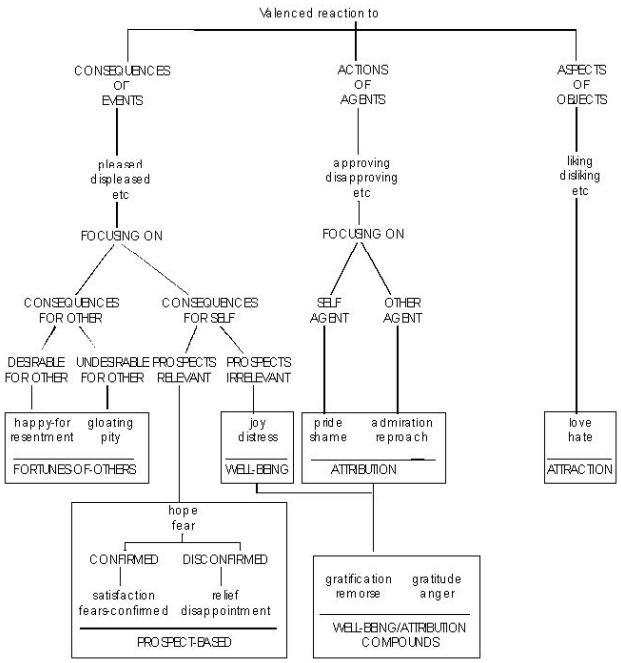

Figure 3. Ortony et al's logic-based ontology of emotions [12]

We have implemented this emotion theory within the VAB via the simple mechanism of associating a PredicateNode with each emotion in the ontology. While this may seem overly simplistic, it's not as bad as it initially seems. As argued in [11], there is not necessarily a dichotomy between localized and distributed representations of knowledge. A PredicateNode associated with an emotion like anger must be considered not in isolation, but rather as a trigger of, and indicator of, broader patterns of activity within the NCE's knowledge base.

Next, regarding animal personality, we have taken a pragmatic approach, including a number of personality parameters drawn directly from the cognitive theory of emotions –

- ill-will, which determines how much resentment/ gloating the pet indulges in
- morality, which determines how much pride/shame/ admiration/reproach the pet indulges in (this is related to obedience)
- goal-orientation, which determines how much joy/distress the pet indulges in, i.e. how much does it care if it gets what it wants or not

- other-orientation, which determines how much the pet indulges in emotions related to others (e.g. happiness-for, admiration/reproach, resentment/gloating)

-- and also a number of personality parameters drawn from qualitative analysis of the psychology of dogs (being the animals we're initially exploring): aggressiveness, curiosity, playfulness, friendliness, fearfulness and obedience. There is also a personality parameter called "emotional expressiveness," which governs how intensely an animal needs to be experiencing an emotion in order to express it externally.

Each animal is assigned a number corresponding to each personality parameter, and the set of these numbers is a crude characterization of the animal's personality. Of course, the actual personality of the animal is more complex than a set of numbers, and consists of a set of complex emergent patterns that are induced by these numbers in the context of the animal's cognitive structures and dynamics and the environment in which it is embedded.

5 INFERENTIAL DYNAMICS FOR EMOTION AND PERSONALITY DRIVEN BEHAVIOR DETERMINATION

Now we describe the scheme via which animal emotions are updated, and used to drive behavior, in the VAB architecture. In short, a collection of probabilistic logic implications are encoded relating emotional states, personality traits and behaviors. Emotional state adjustment and emotion and personality driven behavior determination are then guided by chaining of these implications. In the current, prototype version the implications ("rules") have been hard-coded, but, the overall VAB architecture supports the learning of such rules based on experience and on combination and generalization of pre-programmed rules; and, future work will explore this direction.

The full rule-base used to guide spontaneous behaviors and emotions in the current system version is too large to present here, but we will give a few evocative examples. First, though, we must give a few comments on rule notation. Firstly, the notation ==> in a rule indicates a PredictiveImplication relationship. Rules are assumed to have truth value strengths drawn from a discrete set of values

{0, VERY LOW, LOW, MIDDLE, HIGH, VERY HIGH, 1}

In the following list, all rules should be assumed to have a truth value of HIGH unless something else is explicitly indicated

Also, predicates (including emotions, personality values and others) are assumed to be scalable according to a scaling function called scale(), which takes two arguments: scale(x,c), where both x and c should live in [0,1]. The behavior of this is as follows:

If c=1, then scale(x,c) = x^r
If c=0, then scale(x,c) = x
If c=-1, then scale(x,c) = $x^{1/r}$

(As a default one may choose, say, r=5 for the scaling parameter.) For fixed x, scale(x,c) increases as c increases. The reason to use this function is because if x is trapped in [0,1], one can't scale it by multiplying it by a constant. So we need to scale x nonlinearly, in a way that making c bigger generally makes x bigger. A simple first choice of scaling function is

scale(x,c) = x^{c*r}, for c>0
scale(x,c) = $x^{|c|/r}$, for c<0

For simplicity of notation, scaling by c will be denoted ^^c . For instance

$.5^{\wedge\wedge c}$ = scale(.5,c)
AggressivenessP$^{\wedge\wedge .8}$. = scale(AggressivenessP,.8)

Without scaling, it seems that rules with more factors on the lhs will generally be less often invoked because their lhs has the product of a larger number of terms, all less than 1. So we have introduced a default scaling, so that in a rule with k terms, all terms are scaled by ^^(-k/r), for example.

For clarity, in the following list of rules, we've used suffixes to depict certain types of entities: P for personality traits, E for emotions, C for contexts and S for schemata (the latter being the lingo for "executable procedures" within the NCE). In the case of schemata an additional shorthanding is in place, e.g. barkS is used as a shorthand for (Execution bark) where bark is a SchemaNode. Also, the notation TE<expression>($X) is shorthand for

ThereExists $X
    Evaluation <expression> $X

i.e. an existential quantification relationship.

Example rules from the rule-base are as follows:

- angerToward($X) ==> angry
- loveToward($X) ==> love
- hateToward($X) ==> hate
- fearToward($X) ==> fear
- TEgratitudeToward($X) ==> gratitude
- angerToward($X) ==> ~friend($X) <LOW>
- TE(near($X) & novelty($X)) ==> novelty
- TEloveToward($X) & sleepy ==> gotoS($X)
- TE(loveToward($X) & near($X)) & sleepy ==> sleepS
- gratitudeToward($X) ==> lick($X)
- atHomeC & sleepyB ==> Ex sleepS <.7>
- gotoS($X) ==> near($X) <.6>
- gotoS($X) ==> near($X) <.6>
- AggressivenessP & angryE & barkS => happyE
- AggressivenessP & angryE & barkS ==> proudE
- AggressivenessP & angerToward($X) ==> barkAtS($X) <VH>
- AggressivenessP & angerToward($X) ==> barkAtS($X) <VH>
- AggressivenessP & angerToward($X) ==> nipS($X) <MID>
- AggressivenessP & near($X) & ~friend($X) ==> angerToward($X)
- AggressivenessP & near($X) & enemy($X) ==> angerToward($X) <VH>
- AggressivenessP & near_my_food($X) & ~friend($X) ==> angryToward($X) <VL>

- AggressivenessP & near_my_food($X) ==> angryToward($X)
- AggressivenessP & angerToward($X) & ~friend($X) ==> hate($X)
- AggressivenessP & OtherOrientationP & ownerNear($X) & enemy($X) ==> angerToward($X)
- AggressivenessP & near($X) & enemy($X) & homeC ==> angerToward($X)
- AggressivenessP & ~happyE & ~angryE ==> boredE
- AggressivenessP & jealousE ==> angryE
- AggressivenessP & boredE ==> angryE <LOW>

Spontaneous activity of a virtual animal, governed by the above equations, is determined based on the modeling of habitual activity as the carrying out of actions that the pet has previously carried out in similar contexts. For each schema S, there is a certain number of implications pointing into (Ex S), and each of these implications leads to a certain value for the truth value of (Ex S). These values may be merged together using (some version of) the revision rule.

However, a complication arises here, which is the appearance of emotion values like happyE on the rhs of some implications, and on the lhs of some others. This requires some simple backward chaining inference in order to evaluate some of the (Ex S).

A similar approach applies to the generation of goal-driven activity based on rules such as the above. As an example, suppose we have a goal G that involves a single emotion/mood E, such as excitement. Then there are two steps:

1. Make a list of schemata S whose execution is known to fairly directly affect E
2. For these schemata, estimate the probability of achievement of G if S were activated in the current context

For Step 1, we can look for

- Schemata on the lhs of implications with E on the rhs
- One more level: schemata on the lhs of implications with X on the rhs, so that X is on the lhs of some implication with E on the rhs

7 CONCLUSIONS & FUTURE WORK

We have described an approach to emotion and personality driven behavior determination for virtual animals. The approach has a relatively simple initial incarnation, which has been implemented as described above, and also presents a broad scope of possibilities for future growth. Most notably, since the behavior and emotion determination rules are expressed in the form of probabilistic-logic implications, it will be natural to augment the initial architecture via

- introducing automated mining of rules based on a database of the system's experience (the principle being that rules which the system has implicitly followed in the past, may be explicitly mined as probabilistic implications and then used as explicit behavior determination rules; this process has a deep foundation in cognitive systems theory and is related to the "cognitive equation" articulated in [13]).
- utilizing probabilistic logic to derive new rules from existing ones, based on logic rules described in [9] such as deduction, induction, abduction, analogy and so forth.

While these enhancements will lead to substantially richer behaviors and emotional dynamics, our preliminary experimentation suggests that the initial version is quite sufficient to give rise to a variety of interesting behaviors. The real test, of course, will be when the animals are released in Second Life and other virtual worlds for interaction with end-users.

REFERENCES

[1] Goertzel, Cassio Pennachin, Nil Geissweiller, Moshe Looks, Andre Senna, Welter Silva, Ari Heljakka, Carlos Lopes. An Integrative Methodology for Teaching Embodied Non-Linguistic Agents, Applied to Virtual Animals in Second Life. Proceedings of AGI-08, IOS Press

[2] Goertzel, Ben, Moshe Looks and Cassio Pennachin (2004). Novamente: An Integrative Architecture for Artificial General Intelligence. Proceedings of AAAI Symposium on Achieving Human-Level Intelligence through Integrated Systems and Research, Washington DC, August 2004

[3] Goertzel, Ben (2006). Patterns, Hypergraphs and General Intelligence. Proceedings of International Joint Conference on Neural Networks, IJCNN 2006, Vancouver CA

[4] Goertzel, Ben and Cassio Pennachin, Eds. (2006). Artificial General Intelligence. Springer.

[5] Goertzel, Ben and Pei Wang, Eds. (2007). Advances in Artificial General Intelligence. IOS Press.

[6] Friedlander, David and Stan Franklin (2008). LIDA and a Theory of Mind. In Proceedings of AGI-08, ed. Ben Goertzel and Pei Wang, IOS Press.

[7] Looks, Moshe (2006). Competent Program Evolution. PhD Thesis, Department of Computer Science, Washington University, St. Louis

[8] Ikle', Matt, Ben Goertzel, Izabela Goertzel and Ari Heljakka (2007). Indefinite Probabilities for General Intelligence, in Advances in Artificial General Intelligence, IOS Press.

[9] Ikle', Matt, Ben Goertzel, Izabela Goertzel and Ari Heljakka (2008). Probabilistic Term Logic. Springer.

[10] Heljakka, Ari, Ben Goertzel, Welter Silva, Izabela Goertzel and Cassio Pennachin (2007). Reinforcement Learning of Simple Behaviors in a Simulation World Using Probabilistic Logic, in Advances in Artificial General Intelligence, IOS Press.

[11] Goertzel, Ben (2006). The Hidden Pattern. BrownWalker Press

[12] Ortony, Andrew, Gerald Clore and Allan Collins (1990). The Cgonitive Structure of Emotions. Cambridge University Press.

[13] Goertzel, Ben (1994). Chaotic Logic. Plenum Press.