

Behaviorist Agent Architecture

Paulo Salem da Silva
University of São Paulo
Department of Computer Science
São Paulo – Brazil
salem@ime.usp.br

Ana C. V. de Melo
University of São Paulo
Department of Computer Science
São Paulo – Brazil
acvm@ime.usp.br

Abstract—In this article we propose an agent architecture based on Behavior Analysis, a behaviorist psychology theory. The main characteristic of this theory is the description of complex behavior exclusively in terms of stimulation and behavioral responses. That is to say, in terms of observable and controllable environmental events, instead of the internal mental mechanisms typically found in other approaches to psychology. In this way, not only trivial reflexive actions, but also a number of sophisticated phenomena can be effectively modeled, such as learning and emotional states. Our work, which we call the Behaviorist Agent Architecture, provides a computational framework with which to build agents according to these principles. It accounts for five aspects of agent behavior in an integrated manner: (i) stimulus conditioning; (ii) reflexes; (iii) operant behavior; (iv) drives; and (v) emotions. The objective of this text and of the architecture itself is to show that a consistent and coherent behaviorist point of view may be useful, despite its current lack of popularity. Here we present our overall rationale and the main ideas underlying our agent architecture, including illustrative examples.

Keywords—multi-agent systems; agent architecture; behaviorism; formalization; simulation.

I. INTRODUCTION

An *agent* can be roughly defined as an entity which acts autonomously within an environment, subject to stimulation therein, and possibly interacting with other agents. But such a definition is too abstract to provide the basis of an agency theory. Therefore, it is necessary to define particular kinds of agents, so that their structure can be detailed, further theory can be built upon it and implementations may have a comprehensive specification to follow.

Agent architectures aim at providing such structure, by defining, in a computational way, the main elements that constitute a certain kind of agent, and showing how these elements relate. Concrete agents can then be obtained by specializing some architectural elements or merely instantiating them. Examples of agent architectures include the BDI approach [2], SOAR [3] and ACT-R [4].

In this article we present an agent architecture based on core principles of Behavior Analysis¹, a well-known behaviorist psychology theory, as proposed by B. F. Skinner [5]. In this theory, an agent is seen as an *organism* capable of being *stimulated* and of producing *behavioral responses* according to certain mechanisms. The hallmark of the approach lies in the definition of complex behavior in terms of these two externally observable aspects, through abstract laws and

constructs, without making references to internal mental and organic characteristics of the organism.²

Our *Behaviorist Agent Architecture*³, as we call it, is designed to fulfill an architectural role with respect to Behavior Analysis. This architecture was formally specified in the Z Notation [8], and is fully available in a PhD thesis [9], as part of a wider framework for the modeling, simulation and verification of multi-agent systems. Moreover, part of this formalization is given by [10], where we focused both on the difficulties and on the value of formalizing informal theories. In the present article, owing to space limitations and to reach a wider readership, we do not include the formalization itself, but only its underlying ideas. Here our objective is to show how such a computational agent architecture is designed in the first place, and what are the main characteristics necessary to endow agents with learning and adaptive mechanisms according to the Behavior Analysis tradition. In this manner, the article establishes some key concepts that not only describe our own work, but also serve as a theoretical framework for other attempts to provide computational accounts of behaviorist theories.

The architecture developed here is not capable of modeling all behavioral phenomena found in the literature – it is merely an approximation and an interpretation of the actual psychological theory. Nonetheless, it is capable of modeling many of them, and in such a way that they relate to each other in a coherent whole. Indeed, it is largely because of this coherence that our formalization is suitable as an agent architecture, since it allows different mechanisms to operate together in creating several aspects that contribute for an interesting agent (e.g., a certain autonomy, learning capabilities, interaction with the environment).

As an example of what our architecture can represent, let us consider a behavior analytic description of a typical laboratory experiment that one could perform on, say, a pigeon. The pigeon is put on a cage, where both a button and a light bulb are present. Before giving food to the pigeon, and only then, the experimenter tuns the light on. After some time, the

²This goes far beyond reflexive actions, as we explain later.

³We realize that the word “behavior” and its variants are quite broad and have many intuitive meanings. Nevertheless, we have chosen to keep them as technical terms here in order to remain faithful to the naming conventions usually employed in the behaviorist literature we draw from. Thus, all of our references to “behavior” should be seen from this perspective, unless explicitly noted otherwise. In particular, our use of the term should not be confused with the use made in behavior-based robotics [6] and approaches derived from the subsumption architecture of Brooks [7], which are biologically inspired, but do not have a direct relation to behaviorist psychology.

¹Or, as it is also sometimes called, Analysis of Behavior.

pigeon learns that light is followed by food. So every time the light is on, the pigeon acts as if the food has arrived. This is an example of *classical conditioning*. Moreover, the pigeon initially does only random actions, because it does not know how its environment works. But eventually it discovers that by pushing a button, the light is turned on. This is an example of *operant conditioning*. By combining these two conditionings, the pigeon then becomes likely to *emit* the behavior of pushing the button when it wants to eat. Its hunger, in turn, is given by a *drive*, which changes the utility of stimuli according to how much the pigeon has already eaten. Finally, the experimenter might decide that no food shall be given in association with the light. In this case, the pigeon will be gradually unconditioned, and the behavior of pushing the button will be *extinct*. This causes *frustration* on the pigeon.

The meaning of the emphasized terms will be given later in the article, but for the moment it suffices to note that they provide a relevant vocabulary to describe the experiment. Remarkably, it is a vocabulary whose expressions are ultimately defined in terms of externally observable behavioral responses and stimulation.

We avoid introducing constructs which we do not find necessary for the computational formalization of the original definitions of Behavior Analysis, thus upholding its values as much as possible. In particular, though agents thus defined have state, which is necessary for computation, we do not ascribe usual mental qualities to them (e.g., beliefs, intentions, rationality). Behavior Analysis rejects these usual explanations of behavior, and puts in their place a different set of concepts, focused on the properties of externally observable events – that is to say, behavioral responses and stimulation. In this way, we differentiate our architecture from most others, which draw inspiration from different branches of psychology, specially Cognitive Psychology [11].

We have implemented our architecture in the form of a simulator, with which a number of experiments can be done using a “virtual organism”. In this article the focus is on the architecture, hence we do not investigate the implementation. Nevertheless, it is worth noting that both the agent architecture’s implementation and overall simulator in which it is used, as well as the actual input files for the various examples presented, are all available for download.⁴

This text is organized as follows. Section II reviews the existing work more directly related to our approach. There aren’t many such works, however, partly because Behaviorism has largely fallen out of favor within psychology. Hence, a defense of the behaviorist perspective itself is required, which is provided in Section III. There, we argue that a behaviorist point of view can be, in fact, an interesting research direction for autonomous agent technology, specially in light of certain current trends. Section IV presents the main characteristics of the Behaviorist Agent Architecture itself. This constitutes the main technical contribution of this article. Section V, in turn, illustrates the use of the architecture through concrete case studies, thereby showing its practical application. Finally, Section VI concludes.

⁴This is part of the Formally Guided Simulator (FGS) project. Sources, input files and executables for all of this work, including the agent architecture, can be found at the project’s main repository: <http://github.com/paulosalem/FGS>.

II. RELATED WORK

Computational models for Behavior Analysis are scarce in the literature, and none of the existing ones gives an unified account of its several aspects, which we classify as the following: (i) stimulus conditioning; (ii) respondent behavior (i.e., reflexes); (iii) operant behavior; (iv) drives; and (v) emotions. The approaches that do exist, such as the work of Touretzky and Saksida [12] (whose agents are called *skinnerbots*), focus mostly on algorithmic aspects of operant conditioning, a form of learning by reinforcement. Similarly, Gaudio and Phone [13] propose a particular version of operant conditioning using neural networks intended specifically to allow robots to avoid obstacles. Important as these results may be, they are not sufficient as an architectural basis, which requires a more extensive and structural specification of what constitutes an agent. It must be extensive because there is great dependency among the several behavioral phenomena, and to represent one it is often necessary to represent another. In particular, operant conditioning itself depends on other aspects of the agent, such as drives and emotions. It must also be structural because it serves as a fundamental basis for both implementation and further theoretical development. Therefore, its elements must be organized in such a way that they can be easily identified, analyzed, related, changed and extended – that is to say, highly structured.

There exists a program called *Sniffy, the virtual rat* which aims at providing an interactive simulation of a rat for the purpose of teaching classical and operant conditioning [14]. However, neither the underlying computational model nor the actual source code are provided, so one cannot understand precisely how the simulation works. It seems, though, that much of it is hard-coded for very specific tasks, since, for example, possible actions and stimuli are all fixed, as are also the experiments that can be conducted. Therefore, despite being a program, *Sniffy* does not provide an actual computational account of behavioral phenomena, but merely a tool for teaching known concepts in an interactive manner [15].

Though the Behavior Analysis perspective to agent modeling is uncommon, some specific ideas concerning learning by reinforcement, originated on this behaviorist literature, have been widely employed in Artificial Intelligence [16]. In particular, Q-learning theory [17] seeks to abstract the notion that an action’s value may change over time according to experience, similarly to the operants of Behavior Analysis. However, Q-learning formulation assumes a particular calculation strategy when seeking the optimal action, which is not necessarily employed by agents (e.g., for efficiency reasons, or other idiosyncrasies, agents might not perform the kind of optimization postulated by Q-learning). Furthermore, it is not directed towards obtaining some particular stimulus (i.e., utility is calculated over states, not over stimuli). In any case, specific reinforcement learning techniques and algorithms can be complementary to our proposed architecture, since they can be used as refinements of operant behavior and thus integrate with the overall agent.

III. THE CASE FOR A BEHAVIORIST POINT OF VIEW

Behaviorist psychology was a strong and influential school of thought mostly during the first half of the XXth century.

Since then, however, its reach has faded considerably, giving place to other approaches, specially Cognitive Psychology [11]. Indeed, it is from this latter movement that cognitive agent architectures take inspiration. Hence, one may ask why research in autonomous agents should even consider the behaviorist point of view at all.

A first reason to study Behaviorism here is that, for historical reasons, it had little influence on early computer scientists. The fundamental ideas of Artificial Intelligence are related to the view that human intellect can be understood as an information processing device, much like a computer (e.g., Simon [18]). This view finds considerable support on approaches to psychology that seek to identify and analyze these internal information processing mechanisms, such as Cognitive Psychology and mere folk psychology (as pointed out by Bratman [1]).

Hence, much of the developments from behavioral psychology are either misunderstood or forgotten, which is of course intellectually undesirable. For instance, the well-known text of [16] reduces the behaviorist view of agency to simple reflexive agents, whose actions are elicited by stimulation according to very direct relationships. Wooldridge [19] seems to share a similar opinion. While reflexes are part of Behaviorism, they are merely one component of a much wider framework. This confusion is particularly vexing since the idea of reinforcement learning, widely used in Artificial Intelligence, comes directly from the work of behaviorists such as Skinner.⁵ Moreover, other aspects of Behaviorism, such as the modeling of drives (e.g., thirst) and emotions (e.g., anger) in terms of action and stimuli, are mostly forgotten. It should be clear, therefore, that a better understanding of Behaviorism in the context of autonomous agents is desirable at least to bring some intellectual order to its legacy.

A more practical and recent reason to adopt a behaviorist-oriented outlook can be found in the Internet and related technologies. As the world becomes more interconnected by networks, devices and applications, it becomes increasingly relevant to model the agents therein. How should that be done? A natural approach would be to use the quantifiable information available about them, which most often can be reduced to two classes: the actions that an agent performs and the stimuli they receive. In this context, actions are things like link clicks, access websites and applications, make phone calls, send messages, and so on. Stimuli, in turn, are interface elements, menu options, automated messages, actions by other agents, and so on. It is therefore convenient to have an agent architecture designed from the ground up to make the most of these quantifiable *interactions*, and not so much of what goes on within the agent's "mind" – that is to say, a behaviorist agent architecture.

This is not to say that one can describe an agent completely in behaviorist terms. Indeed, higher cognitive functions (e.g., logical reasoning) are probably beyond the scope of such a model. However, because it is so difficult to infer more detailed cognitive information about an agent based solely on such simple observations (e.g., clicks, messages, etc), these higher functions become a much harder goal to attain.

⁵More precisely, from the notion of operant behavior, which we explain later.

Beyond the practical realm, one can also find academic motivations in areas that increasingly reject the premise that humans act (mostly) rationally. Economics is a case in point. Traditionally, economists assume that people are rational and act in their best interest, given the resources available. Financial crises and panics, though, are often pointed out as evidence against this orthodox view. In recent years, a number of economists have been systematically developing an alternative point of view, aptly named *Behavioral Economics* [20]. A behaviorist agent architecture would provide a framework in which to develop some aspects of this discipline, specially with respect to quantifiable observations (e.g., purchases, savings).

Academic motivations can also be found in the Multi-Agent Systems (MAS) community itself. While Artificial Intelligence is historically concerned with an individual agent's reasoning, deduction and so on, MAS have widened the applications of the agent concept. In a MAS setting, by definition one is concerned not only with the isolated agent, but also with how it interacts with other agents through some environment. It is very natural, then, to seek agency theories that can leverage this more externalized view, which, again, is something that Behaviorism can provide. In fact, we have applied this insight in the development both of a framework for the representation of environments [21] and of a technique for the verification of MAS [22].

Finally, behaviorist notions are also of interest to the purely theoretical aspects of autonomous agents research, independently of any application or use in other areas. Traditionally, most developments in such research concern the inner workings of agents, how it reasons and decides what to do. When external aspects are considered, they relate mostly to how agents communicate among themselves. A behaviorist outlook brings a different perspective. It asks what kinds of concepts and phenomena can be formulated and investigated by looking at what is outside the agent, in its interaction with the surrounding environment. What are the possibilities and limits of such a view? What can and cannot be described in this manner? What sort of algorithms can leverage such modeling? These are questions that arise naturally from such a different perspective, and however applicable they are, they are interesting in themselves, for they help to shed light in the very core of what it *means* to be an autonomous agent.

The agent architecture presented in this text is a step in rectifying these issues. By providing a general blueprint of what a computational behaviorist agent can look like, along with some concrete examples, we clarify and bring under a coherent whole some of the possibilities offered by the behaviorist point of view.

IV. AGENT ARCHITECTURE

The hallmark of Behaviorism is the description of complex behavior in terms of elements external to the organism (i.e., environmental stimulation and observable actions). However, an agent architecture, by definition, must provide the internal structure of the class of agents it represents. Therefore, a behaviorist agent architecture must provide the internal structure of agents so that, once executed, these agents behave in accordance with behaviorist principles. Whereas Behavior Analysis is concerned with inferring laws from the observation

of mechanisms (i.e., actual organisms), the development of a corresponding agent architecture is concerned with inferring mechanisms from those laws. That is to say, it is the attempt to make the reverse path, from the generalizations back to mechanisms that fit those generalizations.

This does not mean that such an architecture is meant to describe the internals of actual organisms. Rather, it should be seen merely as a technical device to generate behavior that comply with certain behaviorist principles. To achieve this, the architecture defines a number of elements to compute these behaviors. Figure 1 depicts these elements and the relations between them. In what follows we present each one, in a way that emphasizes their computational characteristics and their roles in the overall architecture.

A. Basic Assumptions

Organisms are assumed to be constantly seeking pleasure and avoiding pain. This search is the basis for most of the organism's behavioral responses. And while at first it might seem a rather simple motivation, it turns out that it can be used in order to describe a number of interesting phenomena. This can be achieved computationally by defining and manipulating an *utility function*, which is used in various ways, as described in the next sections.

Time is assumed to flow in discrete steps. At each moment, the organism: (i) Receives the current *stimulation* from the environment; (ii) process the received stimulation; and (iii) defines which *behavioral responses* are being performed.

B. General Responding Characteristics

Organisms influence their environment by producing *behavioral responses*. Each such response is an instance of a *class of behavior*. In Behavior Analysis, two such classes are defined, namely, *respondent* (also known as *reflexive*) and *operant* behavior. But before describing what is unique to each, it is necessary to establish what elements of behavior are general and thus common to all such classes.

First, it is useful to define a set of *actions* which are not, in principle, part of any behavior class. These actions represent basic capabilities of organisms (such as the ability of, say, moving one's arm). More complex behavior can then be defined using these basic actions. The following points should be considered when defining actions:

- Some actions may *conflict*, so that they cannot be performed simultaneously. This allows the modeling of behaviors that use the same physical resources. For example, a dog may bark or eat, but not both.
- Some actions can only be used in specific behavior classes. For example, salivation can only be produced by a reflex.

Given all the possible behaviors of an organism, at any given moment it is necessary to determine which among them to perform. This *behavior selection* problem must be addressed by providing:

- Conditions that determine whether isolated behaviors can be performed, disregarding their relation to other behaviors.

- *Conflict resolution* operations, which determine which behavior to perform when two or more conflict. Such conflicts may arise because, as we have just seen, actions themselves may conflict.

Once behaviors have been selected and conflicts have been solved, it is then necessary to *schedule* a behavioral response for *emission*. This scheduling is necessary because certain behaviors may require time to pass before they really happen. This models the fact that organisms, owing to their biological constitution, do not respond immediately, but always within a certain delay of whatever event that justifies the response.

C. Respondent Behavior

Respondent behavior (also known as reflexes or *reflexive behavior*) is the simplest kind of behavior that an organism possess. A reflex is, essentially, a reliable causal relation between a stimulus and an action. That is, the presence of the stimulus triggers, with high probability, the emission of the action.

Reflexes are innate, so organisms can neither gain nor lose them. Nonetheless, reflexes can be *adjusted*. This means that the way in which they are performed can change. A reflex that initially produces an intense action, may produce a less intense one after being elicited several times. Each particular reflex may have its own specific *adjustment functions* that define how the properties of the associated responses should change.

D. Learning

In Behavior Analysis, learning has two main realizations. The first, *stimulus conditioning* (better known as *classical conditioning*), relates the perceived stimuli to each other in order to understand how the surrounding environment works if left to itself. The second, *operant behavior*, relates the actions of the organism to the effects that they may have in the environment. For computational purposes, however, a third mechanism is required as well, that of *spontaneous actions*. Each of these topics are explored below.

1) *Stimulus Conditioning*: Stimuli whose utilities are inherently pleasant or unpleasant are called *primary*. Stimuli that are not primary gain their utility through association to primary stimuli. In general, this process is known as *stimulus conditioning*. The stimulus that has its utility modified is called a *conditioned stimulus*, and the other is called an *unconditioned stimulus*.

As a learning process, stimulus conditioning has two fundamental operations:

- The *conditioning* operation takes two consecutive stimuli, and relates the first to the second.
- The *unconditioning* operation takes an existing conditioning between two stimuli and weakens it if the second stimulus was not perceived after a perception of the first one. Eventually, if this weakening repeats many times, the conditioning is deleted.

The effect of having a stimulus s_1 related to another s_2 is that the utility of s_1 will be calculated as a function of that of s_2 . Thus, an organism can learn how one stimulus somehow

Environment

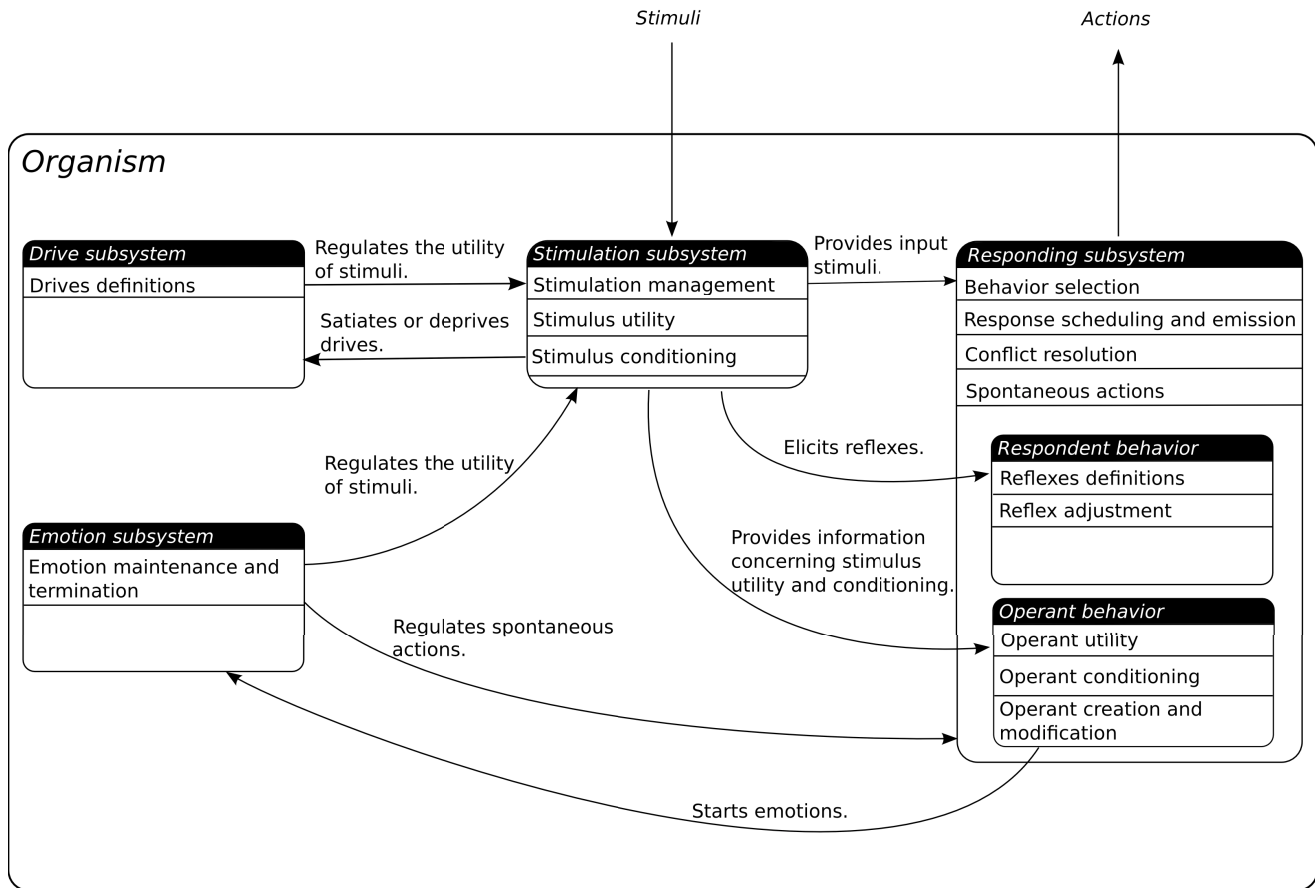


Figure 1. Overview of the Behaviorist Agent Architecture.

causes or precedes another – and unlearn this once such a relation no longer holds.

Computationally, the relation between stimuli can be represented as a graph, in which vertices are stimuli and directed arcs indicate that one stimulus lead to another. By transitivity, indirect relations between stimuli are also defined. Such a structure can be seen as a type of semantic network [23] specialized for behaviorist phenomena.

2) *Operant Behavior*: Organisms seek pleasure and avoid pain in an ever changing world. The consequences of their actions change constantly, in such a way that what used to be an applicable behavior may no longer be appropriate, and useless actions may become interesting. Learning is, therefore, a necessary virtue. In Behavior Analysis, operant behavior is the kind of behavior that accounts for this.

Because the consequences that organisms seek are always reinforcing, operants are also known as *contingencies of reinforcement*. However, by no means reinforcing stimuli are the only consequences that matter. It is equally important to know actions that lead to aversive stimuli in order to avoid them.

The main computational elements of operant behavior are the following:

- The *operant* entity, which records how an action, under certain environmental stimulation, may lead to a consequent stimulus.
- Operations for the *creation*, *update* and *extinction* of operants. Creation takes place whenever a stimulus is observed to happen after an action. Update happens whenever this observation happens again (thereby strengthening the link between them) or the action is actually not followed again by the stimulus (thereby weakening their relation). Finally, extinction takes place when the organism ceases to link an action to a stimulus after a number of updates.
- An *operant utility* function. That is to say, given an operant, this utility function determines the expected stimulus utility to be achieved by emitting the operant. Since one operant may set the antecedents of another, such an operant utility must take into account all possible sequential composition of operants.
- *Emission conditions* for operants.

3) *Spontaneous Actions*: An operant may be created when an action is executed and it is observed that a stimulus happens after it, suggesting therefore a causal relation between the

action and the stimulus. So in the future, should the organism desire the stimulus, it can try to perform the learned operant. But a problem remains: what justified the emission of an action in the first place, before the causal relation was discovered? Clearly, some mechanism to produce unjustified actions is necessary, since otherwise an ignorant organism would never learn any operant behavior. We call this mechanism the *spontaneous action emission*.

A way to provide such a mechanism is by associating a probability of spontaneous emission to each available action. So while selecting behaviors, the agent should also use these probabilities to randomly decide whether any action is to be emitted spontaneously. Note that this can be seen as a way to model curiosity.

In Behavior Analysis, these spontaneous actions are merely assumed to exist. It is assumed that the rat or the pigeon will act somehow, and then one may setup appropriate rewards or punishments for these actions. However, if one is concerned with the implementation of the agents, it is necessary to provide an explicit computational mechanism to account for these initial actions.

E. Behavior Regulation

Behavior, either learned or innate, is made more sophisticated by taking in account certain regulators. These modify the standard behavior of an organism in order to model influences that are, so to speak, orthogonal. This can be achieved in two manners:

- *Response regulators*. They modify the behavioral responses directly.
- *Utility regulators*. They modify the utilities of stimuli, thereby indirectly changing the way in which operant behavior is emitted.

In the framework proposed here, *drives* and *emotions* provide such behavior regulators. In what follows we examine their particularities.

1) *Drives*: In order to stay alive, organisms constantly consume environmental resources. For instance, water, food, air, and so on. Clearly, the utility of these resources must vary over time. An animal that has just drank a lot of water most likely won't be thirsty. On the other hand, an animal that has not drank anything for a day or two will do anything for water.

The mechanisms that control these variations are called *drives*. A drive can be thought of as an appetite for a particular stimulus. The longer one stays without this stimulus, the stronger the appetite for it will be. Conversely, the more one has of the stimulus, the less one will want it. This is modeled by specifying two functions:

- A *satiation* operation, which takes the current importance of the drive and reduces it, if the organism perceives the desired stimulus.
- A *deprivation* operation, which takes the current importance of the drive and increases it, if the organism has not perceived the desired stimulus for some time.

There can be many types of function that fits these criteria (e.g., linear, quadratic, exponential).

2) *Emotions*: Emotions are usually thought of as subjective and private events. Still, one can often guess what a person is feeling by watching his or her behavior. Aggressiveness, for instance, usually indicates a state of anger.

From a behaviorist point of view, though, private events are only relevant to the extent that they produce observable behavior. So aggressiveness is not just a consequence of anger in a behaviorist theory; rather, it is taken to be anger itself.

In the present work, an emotion is defined as a temporary modification in operant behavior that is not explained by the organism's drives. The purpose of emotions is to fine-tune the organism's behavior to match the needs of a given situation. "Pure" operant behavior would only record the relations among actions and stimuli. However, the fact that sometimes actions must be, for example, specially vigorous (e.g., when fighting an opponent), would not be captured. "Pure" stimuli conditioning would be incapable of modifying the utility of primary reinforcers. And that might be exactly what is required sometimes, in order to explain certain kinds of behavior (e.g., depression).

Similarly to what we have seen for drives, this fine-tuning is achieved by providing special regulators. However, differently from what we did for drives, emotions are not defined in a very general manner. They encompass any temporary behavioral modification, and therefore we cannot provide a single mechanism to account for all possible emotions. Instead, here we define three representative emotions, namely:

- *Depression*. A reduction in the utility of all stimuli.
- *Anger*. An increase in the utility of all stimuli that signal pain in another organism (e.g., the sight of blood).
- *Frustration*. An increase in the probability of emitting an action which is not associated neither with a reflex nor with an operant. Frustration is an attempt to learn more about an environment that has not responded properly using the previously available behavior.

Other emotions can be defined in a similar manner.

For each emotion, two operations must be provided:

- A *start operation*, which activates the emotion. Since each emotion is unique, the use of this operation is also particular to each emotion.
- A *stop operation*, which deactivates the emotion. Each emotion, once activated, lasts for a fixed duration, so this operation should be called whenever this duration expires.

F. A Coherent Whole

Adaptation and learning experiences are rather interesting in themselves. However, much of the strength of such a behaviorist architecture comes from their combined effect, since these several experiences influence each other. For instance, an agent may know how to get food (through learning), but because it has eaten too much already, it has no interest in doing so (an adaptation to having eaten). This brings unity to these experiences, as they affect the agent as a whole.

V. EXAMPLES

Let us now present some illustrative examples of models that have been constructed using the proposed architecture. We explore one example at some length, and then summarize a number of other case studies. We do not have the space for detailed developments here. However, full inputs (and corresponding simulation outputs) are provided in [9], where these examples are described in detail.

A. Behavior Elimination

It is often the case that one wishes to eliminate some operant behavior. An obvious way to do it is to punish the agent when it performs such an operant. But punishment has its own undesirable consequences, such as the emotional influence that it brings.

Interestingly, there is an alternate way of eliminating operants, by using reinforcement instead. This approach consists in reinforcing some other behavior which: (i) is something desirable; and (ii) is in conflict with the operant to be eliminated (i.e., the agent cannot perform both at the same time). By this method, one both avoids the problems with punishment and creates a new and valuable behavior.

As an example, let us consider a child who often misbehaves by beating her dog. She does so because she finds the resulting dog's scream amusing. This child, moreover, sometimes do caress the dog, albeit rarely, when she gets tired of hearing the dog's scream.

Clearly, one cannot beat and caress the dog at the same time, since these actions depend on the same mechanism (i.e., the child's hand). So to eliminate the behavior of beating the dog, we reinforce the action of caressing. If the reinforcement is more pleasant to the child than the dog's scream, then the behavior is successfully changed.

We can use our agent architecture to model both agents. Each agent is an instance of the architecture supplied with detailed parameters. For our present illustration purposes, though, it suffices to note some of the key parameters employed. The child is parametrized with the following elements:

- *Stimuli.* We assume that this child enjoys receiving candy, as well as hearing the dog's scream. But for the technique to work, we assume that the child actually prefers the candy.
- *Actions.* The child may either beat or caress the dog, and these are conflicting actions
- *Operant.* The child already knows that when she beats the dog, it will scream.
- *Drive.* The child periodically gets tired of hearing the dog's scream. This is modeled as a drive.

The dog, in turn, is modeled as follows:

- *Stimuli.* Both of the relevant stimuli (i.e., punch and caress) for the dog are primary, but one is unpleasant, whereas the other is pleasant.
- *Actions.* The dog can scream in pain. Moreover, to make the example more complex, the dog also barks spontaneously.

- *Reflex.* The dog has a reflex to model the fact that it can complain when it receives a punch.

By simulating this two agent system, and providing the appropriate rewards when the opportunity arises (i.e., when the child caress the dog), we successfully modeled the phenomenon. This experiment puts in evidence the interaction between drives, stimulation and operant behavior in order to model a behavioral phenomenon. This shows the value of the integrating the different aspects of an organism's behavior in a coherent whole. The example also shows that agents can relate to each other by transforming the actions of one in stimuli to the other.

B. Further Examples

The architecture can model various situations. These are some further examples that we built and simulated:

- *Worker: operant chaining.* The consequences of an operant can set the conditions for another operant. When this happens, we say that *operant chaining* takes place. This notion can explain why an agent acts even though no immediate reward is given. To show this, we model a worker, whose several actions depend on each other in order to lead to what he really desires, such as food. So waking up, going to work, getting payed and buying provisions are all part of one operant chain.
- *Factory: rearranging a social network.* An assembly line is modeled in such a way that any agent can occupy any position in the production chain. However, some workers cannot work close to one another, because they get distracted (e.g., by chatting). Here, our objective is to find a configuration of the social network that allows the assembly line to work. In this network, nodes are particular tasks on the assembly line and relations indicate the dependency between these tasks.
- *School children: from chaos to order.* Children receive homework from a teacher, but are distracted by a number of environmental factors. For example, they might prefer to watch TV or annoy other children. One can then study appropriate manners to organize the overall system so that the children act as desired.
- *Online social network: spreading a message.* In an online social network, users can register themselves and send messages to each other through a website. In particular, they can forward advertisement received from the website. The operator of the network, then, can reward users with points for doing that. The objective is to check how the advertisement spreads in the network.

VI. CONCLUSION

Owing to its focus on the organism as a whole (i.e., not on isolated details of particular internal structures), Behavior Analysis incidentally provides a useful basis for a computational agent architecture. That is to say, a framework with which to define agents capable of receiving stimuli and performing actions in a rather general manner.

In this article a number of architectural elements were defined to model the notions found in Behavior Analysis. Some of them come directly from the behaviorist literature (e.g., operant behavior), whereas others were introduced to allow a proper computational model (e.g., utility functions, spontaneous actions). Their architectural role is important: particular implementations of the architecture may vary widely how each element is actually implemented. For instance, operant behavior can be implemented using many different existing algorithms for reinforcement learning. The architecture provides a general blueprint, which can be changed and extended in many ways. Our particular implementation, used to run the examples, is but one possibility.

The examples given illustrate how several mechanisms interact in order to produce the agent's final behavior. Stimulus conditioning changes the relation between stimuli, and therefore changes the way in which reflexes and operant behavior happen. Drives can alter the utility of stimuli temporarily, which affects behavior as well. Operants can be composed in chains in order to accomplish more complex tasks. Other similar interactions exist, which shows that a behaviorist architecture, such as the one we propose, can model interesting agent behaviors.

The choice of examples was guided by a desire to stay close to the original behaviorist literature (hence, animals and situations from daily life were employed). They show that the architecture can account for important psychological phenomena, thereby validating it. It would be interesting, now, to see how applicable the architecture is in areas of more current interest, such as Behavioral Economics and software applications.

Finally, though coherent and expressive, the architecture developed here is not a complete formalization of Behavior Analysis. A number of interesting notions were not accounted for, and therefore are left as research possibilities. Of these, behavior shaping and schedules of reinforcement are perhaps the most important ones. Behavior shaping consists in teaching a behavior by rewarding successive approximations of it. It therefore requires the introduction of some notion of behavior approximation, which we did not attempt. Schedules of reinforcement, in turn, concern how to deliver reinforcement in order to change the rate of responses over time [24]. The difficulty in this case is in finding a computational formulation that correctly models the empirically observed phenomena.

ACKNOWLEDGMENTS

This project benefited from the financial support of *Conselho Nacional de Desenvolvimento Científico e Tecnológico* (CNPq), *Coordenação de Aperfeiçoamento de Pessoal de Nível Superior* (CAPES), and *Fundação de Amparo à Pesquisa do Estado de São Paulo* (FAPESP).

We also thank Walkiria Helena Grant, from the Institute of Psychology of the University of São Paulo, for proofreading some of our very initial ideas.

REFERENCES

[1] M. E. Bratman, *Intention, Plans, and Practical Reason*. Center for the Study of Language and Information, 1987.

[2] A. S. Rao and M. P. Georgeff, "Bdi agents: From theory to practice," in *ICMAS*, V. R. Lesser and L. Gasser, Eds. The MIT Press, 1995, pp. 312–319.

[3] J. E. Laird, A. Newell, and P. S. Rosenbloom, "SOAR: an architecture for general intelligence," *Artif. Intell.*, vol. 33, no. 1, pp. 1–64, 1987.

[4] J. R. Anderson, D. Bothell, M. D. Byrne, S. Douglass, C. Lebiere, and Y. Qin, "An integrated theory of mind," *Psychological Review*, vol. 111, no. 4, pp. 1036 – 1060, 2004.

[5] B. F. Skinner, *Science and Human Behavior*. The Free Press, 1953.

[6] M. J. Mataric, "Behavior-based robotics as a tool for synthesis of artificial behavior and analysis of natural behavior," *Trends in Cognitive Sciences*, vol. 2, no. 3, pp. 82 – 86, 1998.

[7] R. A. Brooks, "Intelligence without representation," *Artif. Intell.*, vol. 47, no. 1-3, pp. 139–159, 1991.

[8] J. Woodcock and J. Davies, *Using Z: Specification, Refinement, and Proof*. Prentice Hall, 1996.

[9] P. S. da Silva, "Verification of behaviourist multi-agent systems by means of formally guided simulations," Ph.D. dissertation, Universidade de São Paulo and Université Paris-Sud 11, November 2011, joint thesis between the two universities. The text can be obtained at <http://tel.archives-ouvertes.fr/tel-00656809/>.

[10] P. S. da Silva and A. C. V. de Melo, "A simulation-oriented formalization for a psychological theory," in *FASE 2007 Proceedings*, ser. Lecture Notes in Computer Science, M. B. Dwyer and A. Lopes, Eds., vol. 4422. Springer-Verlag, 2007, pp. 42–56.

[11] U. Neisser, *Cognitive psychology*. Appleton-Century-Crofts, 1967.

[12] D. S. Touretzky and L. M. Saksida, "Operant conditioning in skinner-bots," *Adapt. Behav.*, vol. 5, no. 3-4, pp. 219–247, 1997.

[13] P. Gaudiano and C. C. Phone, "Adaptive obstacle avoidance with a neural network for operant conditioning: Experiments with real robots," in *CIRA '97: Proceedings of the 1997 IEEE International Symposium on Computational Intelligence in Robotics and Automation*. Washington, DC, USA: IEEE Computer Society, 1997, p. 13.

[14] T. Alloway, *Sniffy: The Virtual Rat: Pro Version 2.0*. Belmont, CA, USA: Wadsworth Publ. Co., 2005.

[15] J. J. Jakubow, "Review of the book sniffy the virtual rat pro version 2.0," *Journal of the Experimental Analysis of Behavior*, vol. 87, pp. 317 – 323, 2007.

[16] S. J. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*. Prentice Hall, 2002.

[17] C. J. C. H. Watkins, "Learning from delayed rewards," Ph.D. dissertation, Cambridge University, 1989.

[18] H. A. Simon, *Sciences of the Artificial*, 3rd ed. The Mit Press, 1996.

[19] M. Wooldridge, *An Introduction to MultiAgent Systems*, 2nd ed. Wiley Publishing, 2009.

[20] R. H. Thaler and S. Mullainathan, "Behavioral economics," in *The Concise Encyclopedia of Economics*. Library of Economics and Liberty, 2008, retrieved May 17, 2013 from: <http://www.econlib.org/library/Enc/BehavioralEconomics.html>.

[21] P. S. da Silva and A. C. V. de Melo, "A formal environment model for multi-agent systems," in *Formal Methods: Foundations and Applications*, ser. Lecture Notes in Computer Science, J. Davies, L. Silva, and A. Simao, Eds. Springer Berlin / Heidelberg, 2011, vol. 6527, pp. 64–79.

[22] —, "On-the-fly verification of discrete event simulations by means of simulation purposes," in *Proceedings of the 2011 Spring Simulation Multiconference (SpringSim'11)*. The Society for Modeling and Simulation International, 2011.

[23] J. F. Sowa, *Semantic Networks*. Wiley, 1987, an updated version of this article can be found online on <http://www.jfsowa.com/pubs/semnet.htm>.

[24] C. B. Ferster and B. F. Skinner, *Schedules of reinforcement*. Appleton-Century-Crofts, 1957.