

# A Simulation-Oriented Formalization for a Psychological Theory

Paulo Salem da Silva and Ana C. Vieira de Melo

University of São Paulo  
Department of Computer Science  
São Paulo – Brazil  
salem@ime.usp.br, acvm@ime.usp.br

**Abstract.** In this paper we present a formal specification of a traditionally informal domain of knowledge: the Behavior Analysis psychological theory. Our main objective is to highlight some motivations, issues, constructions and insights that, we believe, are particular to the task of formalizing a preexisting informal theory. In order to achieve this, we give a short introduction to Behavior Analysis and then explore in detail some fragments of the full specification, which is written using the Z formal method. With such a specification, we argue, one is in better position to implement a software system that relates to an actual psychological theory. Such relation could be useful, for instance, in the implementation of multi-agent simulators.

## 1 Introduction

Mathematical approaches have been successful in representing the universe of natural sciences and engineering. Modern Physics is, perhaps, the greatest example of this success. Yet, many important fields of study remain distant from formal structures and reasoning. Among these, we regard Psychology as particularly interesting.

Roughly speaking, Psychology is divided into several schools of thought, and each one adopts its own definitions, methods and goals. As examples, we may cite Psychoanalysis, Cognitivism and Behaviorism. The later is further divided into several approaches, out of which *Behavior Analysis* [1], created by Burrhus Frederic Skinner, stands out. While not strictly built on formal terms, it does bear some resemblance to them through detailed and precise definitions. As a consequence, it suggests the possibility of a complete formalization.

With this in mind, we have designed a formal specification for agent behavior based on the Behavior Analysis theory. Its purpose is twofold. First, it should allow the construction of agent simulators following the principles of this psychological school. Second, it aims at demonstrating the possibility and the value, from a Software Engineering perspective, of formally specifying traditionally informal domains in order to build tools related to these domains.

The specification of the Behavior Analysis theory has been written with the Z formal method [2], and this paper presents its fundamental structure, but

does not go deep into all details. Our aim, here, is to highlight some issues, constructions and insights that, we believe, are particular to the task of formalizing a preexisting informal domain of knowledge. Moreover, we hope that our presentation argue in favor of this kind of formalization.

We are aware of some other works similar to ours either on their purposes or on their methods. A multi-agent specification framework written in Z, called SMART, can be found in [3]. One of the authors of this book is also involved in the formal modelling and simulation of stem cells [4]. Neuron models and simulations are common practice in the field of Computational Neuroscience [5,6]. We do not know, however, of attempts to formalize whole theories about *organism behavior*.

Sect. 2 details the process through which our specification was conceived. Naturally, we assume that the reader is not familiar with Psychology. Therefore, Sect. 3 presents a brief introduction to the fundamental elements of Behavior Analysis. Sect. 4 explores some fragments of the specification in detail, using them to illustrate relevant points. We expect the reader to know the basics of the Z formal method, which can be learned in works such as [2] and [7]. Sect. 5 summarizes our main results and further elaborates on them. Finally, Sect. 5 acknowledges the help we received.

## 2 Formalization Process

Although the formalization process we employed is not precise, it does follow a number of principles and practices which are worth registering. In this section we present this knowledge, as structured as possible.

Let us begin by tracing the two major steps that we went through, namely:

1. Definition of the main entities and relationships in the theory;
2. Addition of restrictions and further structure upon the entities and relationships.

The first step allow us to identify the elements upon which we should focus. This, we believe, is specially important if the domain being formalized is not entirely understood. In our case, we initially built an ontology [8,9] for the concepts of Behavior Analysis as described by Skinner in the book *Science and Human Behavior* [1]. Among the techniques we employed to accomplish this stage, the most relevant ones are the following:

- Map chapters or sections to subsystems. By doing this, we reused the general structure of the original theory;
- Build the ontology as the book is read. We adopted the discipline of editing the ontology at the end of sections or chapters;
- Register concepts in the ontology without structure and later organize them. This is important because sometimes it is not clear what a concept actually means or where it should be positioned in the ontology. As one gains more knowledge about the domain, it becomes simpler to organize the available concepts.

In the second step, then, we can focus our attention on the details of each entity and relationship identified in the previous step. More expressive formalisms might be needed at this point. In our work, we employed the Z formal method to this end. Z was chosen in part because of our prior experience with it, but also owing to the method's emphasis on axiomatic descriptions, refinement and modularization. Moreover, we used the Z/EVES tool [10] to help us write the specification.

To gain a deeper understanding of the identified entities, we also began to study other references, specially the book *Learning* [11], written by Charles Catania, a well known contemporary psychologist. The formal specification, thus, is mostly structured according to the views of Skinner himself, though we have used a modern reference to improve our understanding of specific topics. At this point, we found the following practices to be useful:

- Design subsystems to be as isolated as possible;
- Try to express new things in terms of what is available. We found that once some base concepts are set, much can be expressed using them;
- When defining an operation, try to account for all possible input cases. This helps spot conditions that have not been considered, either by the original theory, or by the formalization. We shall see an example of this in Sect. 4.2;
- When a concept is not clear, leave it as abstract as necessary. By not trying to formalize what is not well understood, one avoids having to change the formalization later on;
- When a concept may have multiple interpretations, provide an abstract definition followed by refinements that specialize it. We shall encounter an example of this in Sect. 4.2;
- Do not attempt to formalize all details of the theory at once. In our experience, such ambition is doomed to failure, for the more details are added, the harder it gets to connect each part of the specification to the others.

Such are the main practices we employed. In Sect. 4 we shall encounter some of them applied to an actual example.

### 3 A Brief Introduction to Behavior Analysis

We now present some fundamental ideas and elements of Behavior Analysis, upon which we have built our formal specification.

Behaviorism is a branch of Psychology created in the beginning of the 20th century. It was born mainly as an opposition to the dominating idea that the objective of Psychology was the study of the mind. Behaviorists rejected this position, claiming that it was too vague and unsuitable for scientific investigation. They asserted that the true purpose of Psychology should be the study of the *behavior* of organisms, which, they thought, was a precise concept and, therefore, within the realm of natural science.<sup>1</sup>

---

<sup>1</sup> See [12] for a classical exposition of these principles.

The Behaviorist tradition produced several important thinkers, from which Burrhus Frederic Skinner was, perhaps, the most notorious one. Between the decades of 1930 and 1950 he developed his own kind of Behaviorism, called Behavior Analysis.

In Behavior Analysis, the fundamental object of study is the *organism*. Organisms perceive their environments through *stimuli* and act upon such environments through *behavior*. Further, a relation is assumed to exist between stimuli and behavior, in such a way that behavior is, ultimately, determined by the stimulation received by the organism. Thus, the purpose of this science is the prediction and control of behavior.

This objective is pursued mainly through the classification of several phenomena concerning stimuli and behavior. The hope is that regularities can be discovered, leading to the formulation of behavioral laws. Let us first examine the ideas concerning stimulation, and then proceed to the points about behavior.

Each stimulus has an utility value. That is, it is either pleasant or painful, desired or feared. Some stimuli, called *primary*, possess utility values *a priori*, independently of prior experience. All others, called *conditioned*, have their utilities determined by primary stimuli during the organism's life.

The relations between primary and conditioned stimuli are modified through the process named *stimulus conditioning*. Essentially, it is a learning process that tries to relate the occurrence of certain stimuli to the occurrence of others. In other words, it allows organisms to formulate causal laws about their environments. As an example, consider a dog that is always fed after a whistle. Initially, only the presentation of food can make the dog salivate. With time, however, the dog learns that the whistle is related to the food, causing him to salivate with the whistle, prior to any food delivery. In this case, food is the primary stimulus, since it is naturally pleasant to the dog. The whistle, on the other hand, is a conditioned stimulus, which becomes related to food.

Stimulus conditioning also works the other way around. If the relation between two conditioned stimuli is not maintained, it tends to disappear. In the previous example, if the whistle is no longer followed by food, it is likely that, after some time, it won't elicit salivation.

Now let us proceed to the study of behavior. Behavior Analysis defines two main classes of behavior, namely, the class of *reflexes* and the class of *operants*. A reflex is characterized by an *antecedent stimulus*, which causes the organism to behave in some way. For instance, salivation is a reflex, since it is caused by the the presentation of food. Reflexes are innate to the organism. That is, they are not learning structures, they cannot be created nor modified in great extent. Operants, on the other hand, are far more flexible behavioral structures. An operant is defined by a *consequent stimulus*. The operant stands for the behavior that leads to this stimulus. That is, the behavior that operates in the environment in order to generate the stimulus. Notice that if a behavior no longer takes to a stimulus, or if the behavior required to reach that stimulus changes, the operant changes as well. They are, therefore, learning structures. As an example, suppose that a dog learns that the push of a button brings

food to him. Then this button pushing behavior becomes an operant, for it is associated with a specific consequent stimulus.

It is through operant behavior that the most interesting issues arise in Behavior Analysis. Organisms can have their behavior changed by operations of *reinforcement* and *punishment*. Reinforcement is the presentation of pleasant stimuli as a reward for particular behaviors. Punishment, in turn, accounts for the presentation of unpleasant stimuli, in order to inhibit specific behaviors. There are many ways to perform these operations, called *schedules of reinforcement*. Each schedule modifies behavior in a distinct way.

There are other interesting concepts, but we shall limit ourselves to these, for they are sufficient to understand the examples that come in the next section. Moreover, most of the concepts discussed above are present explicitly in our specification. And how a simulator based on it could be useful? Once we define an organism, we can perform simulations to determine properties like:

- How frequent should reinforcement be in order to preserve behaviors of interest;
- How much time it takes to teach the organism a new behavior.

In general, simulations could replace some experiments usually done with real animals.

## 4 Results

As stated above, the specification is too large to be completely described in this paper. Therefore, in this section we do not present the whole specification, but some of its most significant parts, from which useful discussion can be drawn. Some schemata used might not be defined for this reason. Sect. 4.1 gives an overview of the specification's general structure, while Sect. 4.2 explores some of its most instructive parts in detail.

### 4.1 Specification Overview

The formalization's main goal is to allow the construction of a system that simulates the behavior of organisms according to the principles of Behavior Analysis. It is natural, therefore, to build a specification centered around the concept of "organism". The main object of our specification is an isolated organism, which receives stimuli from an environment and produces behavioral responses. It is modelled as a state machine according to the following principles:

- Time is discrete;
- At every instant, the state of the organism may change;
- At every instant, the organism may receive one stimulus;
- At every instant, the organism may produce a new behavioral response.

Changes in the state of the organism are given either spontaneously or as consequences of stimulation. These changes are controlled by several mechanisms, which we have divided into subsystems. Each subsystem is responsible for a particular aspect of behavior and is closely related to major concepts in the psychological theory. Thus, formally, an organism is a composition of several subsystems, as the following schema shows.

<i>Organism</i> <i>stimulationSubsystem : StimulationSubsystem</i> <i>respondingSubsystem : RespondingSubsystem</i> <i>driveSubsystem : DriveSubsystem</i> <i>emotionSubsystem : EmotionSubsystem</i>
-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

At every instant, the organism may receive a new stimulus, which is processed by all subsystems in no particular order. How these stimuli are generated or how the organism's behavior changes the environment is out of the specification's scope. Nevertheless, we do provide a simple definition of the simulation process with the following schema.

<i>Simulator</i> <i>organism : Organism</i> <i>currentInstant : Instant</i>
-----------------------------------------------------------------------------------

## 4.2 Specification: Main Elements

Let us now proceed to the detailed examination of some parts of the specification. In what follows, we first explore some of the stimulation subsystem, and then give some details of operant behavior, defined in the responding subsystem.

**Stimulation.** The specification of stimulus processing is particularly suitable for the discussion of how traditional mathematical structures, such as graphs, can be used in formalization processes. The fact that these phenomena can be translated to well studied formal structures sheds new light on them. It allows us to consider possibilities that could have remained hidden prior to the formalization.

We begin by giving the main stimulation subsystem definition.

<i>StimulationSubsystem</i> <i>StimulationParameters</i> <i>StimulusImplication</i> <i>StimulusEquivalence</i> <i>currentStimuli : <math>\mathbb{P}</math> Stimulus</i> <i>stimulus_status : Stimulus <math>\rightarrow</math> StimulusStatus</i>
------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Consider the several schema imports above. The first, *StimulationParameters*, merely defines the parameters that are given as input to the simulation. They define what is particular, *a priori*, to the organism being simulated. We shall not pursue it in detail here. Our interest is in the other two, *StimulusImplication* and *StimulusEquivalence*. They carry the fundamental definitions that allow the formalization of stimulus conditioning operation. As we pointed out earlier, such operation allows organisms to learn about how their environment works. Let us first examine it informally and then, upon that, build a formal definition.

The behavior of organisms depends greatly on their power to learn how environmental stimuli are related. Sometimes, it is useful to consider two stimuli that are, in reality, different, to be equivalent. For example, if, through experimental procedures, we arrange that both the presence of a red light and of a green light are always followed by the same consequences (e.g., food), why should a hungry organism bother to distinguish between the colors? As far as the organism is concerned, the two lights are equivalent.

On the other hand, sometimes the appropriate relation is one that defines causality, not equivalence. In the previous example, we may arrange the procedure so that the red light is always followed by food. In this case, the learning takes the order of stimulation into account: though red light is followed by food, food is not necessarily followed by a red light. That is, the organism may establish an implication between red light and food.

We now proceed to the formalization of these ideas. Notice that causal laws are certainly reflexive, since a stimulus trivially causes itself. They are also transitive, in the sense that causality can be chained (e.g., stimulus  $s_1$  causes  $s_2$  which, in turn, causes  $s_3$ ). Finally, in principle no symmetry is needed (e.g., if  $s_1$  causes  $s_2$ , there is no need, at first, for  $s_2$  to cause  $s_1$ ). We are now in position to specify causality in the *StimulusImplication* schema. It also defines a function called *sCorrelation*, which accounts for the fact that some implications may be stronger than others.

<p style="margin: 0;"><i>StimulusImplication</i> _____</p> <p style="margin: 0;"><math>sCauses : \mathbb{P}(Stimulus \times Stimulus)</math></p> <p style="margin: 0;"><math>sCorrelation : Stimulus \times Stimulus \rightarrow Correlation</math></p> <hr style="border: 0.5px solid black; margin: 5px 0;"/> <p style="margin: 0;"><math>\forall s_1, s_2, s_3 : Stimulus \bullet</math></p> <p style="margin: 0; padding-left: 20px;"><math>(s_1 \underline{sCauses} s_1) \wedge</math></p> <p style="margin: 0; padding-left: 20px;"><math>((s_1 \underline{sCauses} s_2) \wedge (s_2 \underline{sCauses} s_3)) \Rightarrow (s_1 \underline{sCauses} s_3)</math></p> <p style="margin: 0;"><math>\forall s_1, s_2 : Stimulus \mid s_1 \underline{sCauses} s_2 \bullet</math></p> <p style="margin: 0; padding-left: 20px;"><math>\exists c : Correlation \bullet ((s_1, s_2) \mapsto c) \in sCorrelation</math></p>
--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

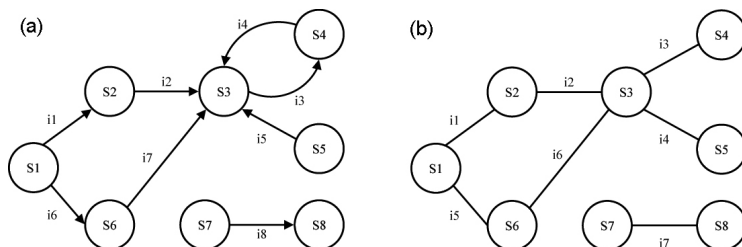
Stimulus equivalence relations, in turn, can be defined in terms of stimulus implication. We merely add the symmetry axiom and require the *sCorrelation* function to have the same value in both directions.

---

*StimulusEquivalence*
*StimulusImplication*
 $equals : \mathbb{P}(Stimulus \times Stimulus)$ 
 $\forall s_1, s_2 : Stimulus \bullet$ 
 $(s_1 \underline{equals} s_2) \Leftrightarrow (s_1 \underline{sCauses} s_2) \wedge (s_2 \underline{sCauses} s_1)$ 
 $\forall s_1, s_2 : Stimulus \mid s_1 \underline{equals} s_2 \bullet$ 
 $sCorrelation(s_1, s_2) = sCorrelation(s_2, s_1)$ 


---

With this, we have achieved a formal specification for the relations among stimuli. But we may continue our analysis, casting this specification in other terms. Notice that stimulus implication may be regarded as a directed graph (Fig. 1(a)), in which vertices represent stimuli and edges are the conditioning between stimuli. Similarly, stimulus equivalence can also be seen as a graph (Fig. 1(b)), but undirected. Furthermore, edges in both graphs might have weight, if the correlation of the conditioning is to be taken into account.



**Fig. 1.** (a) An example of stimulus implication represented as a directed graph; (b) An example of stimulus equivalence represented as an undirected graph

Regarding this stimuli graph, new psychological questions arise. In fact, we can use all our knowledge of Graph Theory and search algorithms to formulate questions, bringing new light to the psychological theory itself. For instance, consider the following:

- When looking for causal relations, which search strategy do organisms employ? Do they execute a depth- or breadth-first search?
- How deep can a search go? Is there some sort of memory limitation that prevents it from being exhaustive?

Answers to these questions, of course, are left to psychologists. We must, however, model this lack of knowledge somehow. Fortunately, the Z formal method allows us to do this easily, as follows. For all operations that deal with stimulus



implication and equivalence, we first define a more abstract version, containing only axioms that we are sure to hold. Then we provide one or more refinements that add assumptions to it. This allows experimentation with several possibilities and makes it easier to update the specification as we learn more about psychological phenomena.<sup>2</sup>

As an example, let us consider the schemata that specify how the utility of a stimulus is calculated. Recall from Sect. 3 that stimuli are divided into two classes, namely, primary and conditioned. Primary stimuli have utility values *a priori*, while conditioned stimuli have their utilities calculated in terms of the primary ones. Moreover, drives and emotions can influence this calculation. The more general version of stimulus utility, *StimulusUtility*, states that there exists a function that calculates the utility in terms of the stimulus, a set of emotions and a set of drives.

$\begin{array}{l} \textit{StimulusUtility} \\ \textit{StimulationSubsystem} \\ \textit{EmotionSubsystem} \\ \textit{DriveSubsystem} \\ \textit{sUtility} : \textit{Stimulus} \rightarrow \textit{Utility} \end{array}$
$\begin{array}{l} \exists f : \textit{Stimulus} \times \mathbb{P} \textit{Emotion} \times \mathbb{P} \textit{Drive} \rightarrow \textit{Utility} \bullet \\ \forall s : \textit{Stimulus} \bullet \\ \quad \textit{sUtility}(s) = f(s, \textit{activeEmotions}, \textit{activeDrives}) \end{array}$

Clearly, this abstract definition does not relate conditioned to primary stimuli. The reason is that, as far as we can see, any such relation must contain assumptions that we are not sure to hold. Thus, the actual relation is given in refinements. A simple one is given by *StimulusUtility\_Ref1* schema, which depends on another schema, *StimulusUtilityBase*. In this refinement, the calculation is performed by locating the best primary stimulus that can be reached through stimulus implication, and then applying emotional and driving filters.

$\begin{array}{l} \textit{StimulusUtility\_Ref1} \\ \textit{StimulusUtilityBase} \\ \textit{StimulusEmotionalRegulator} \\ \textit{StimulusDriveRegulator} \end{array}$
$\begin{array}{l} \forall s : \textit{Stimulus} \bullet \\ \quad \textit{sUtility}(s) = \textit{driveRegulator}(s, \textit{emotionalRegulator}(s, \textit{base}(s))) \end{array}$

<sup>2</sup> Notice that if the specification is implemented in an object-oriented language, this approach can be seen in terms of class inheritance.

$\textit{StimulusUtilityBase}$ <hr/> $\textit{StimulusUtility}$ $\textit{StimulusImplication}$ $\textit{base} : \textit{Stimulus} \rightarrow \textit{Utility}$ <hr/> $\forall s : \textit{Stimulus} \bullet$ $(\exists p : \textit{primaryStimuli} \bullet$ $\quad \textit{base}(s) = \textit{primary\_utility}(p) \wedge$ $\quad (\forall q : \textit{primaryStimuli} \mid s \textit{sCauses} q \bullet$ $\quad \quad \textit{primary\_utility}(p) \geq_1 \textit{primary\_utility}(q) \wedge$ $\quad \quad (s \textit{sCauses} p))) \vee$ $(\forall p : \textit{primaryStimuli} \bullet$ $\quad \neg (s \textit{sCauses} p) \wedge$ $\quad \textit{sUtility}(s) = \textit{neutral})$
-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

In the next section we shall make references to some of the entities presented here in order to show how different subsystems are related.

**Operant Behavior.** Operant behavior, as we have seen in Sect. 3, is the most important behavioral class within Behavior Analysis. We shall study it here from two perspectives. First, its formalization is not straightforward, and we shall examine some of the difficulties. Second, operant processing is not simple, but can be elegantly modeled to some extent.

Let us begin by defining an operant.

$\textit{Operant}$ <hr/> $\textit{StimulusUtility}$ $\textit{antecedents} : \mathbb{P}(\mathbb{P} \textit{Stimulus})$ $\textit{action} : \textit{Action}$ $\textit{consequence} : \textit{Stimulus}$ $\textit{consequenceContingency} : (\mathbb{P} \textit{Stimulus}) \leftrightarrow \textit{Correlation}$ <hr/> $\textit{sUtility}(\textit{consequence}) \neq \textit{neutral}$ $\emptyset \in \textit{antecedents}$ $\text{dom } \textit{consequenceContingency} = \textit{antecedents}$
----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

The above schema states that an operant has an *action* which leads to a *consequence*. There are two important considerations to be made here. First, notice that we introduced the concept of action. From the study of Behavior Analysis, we realized that there are some terminological imprecisions; a behavior (i.e., what is actually performed by the organism) and a behavior *class* (i.e., a set containing behaviors that have some properties) are distinct concepts, but

it is easy to confuse them. Thus, we adopted the notion of action to refer to what would traditionally be called a behavior or even a mechanical property of behavior.

The second consideration regards the fact that Behavior Analysis defines operants solely by a stimulus consequence. Thus, in principle, either no action should be defined within an operant, or all possible actions that lead to the consequence should be present. This approach, however, would neglect the fact that each action takes to the consequent stimulus in a different way. For instance, pushing either a red button or a green one might lead an animal to food. But, perhaps, the red button is more efficient and, hence, will be more strongly correlated with the consequence than the other.

The schema also defines a set of sets of stimuli, *antecedents*. This accounts for the fact that the stimuli currently present in the environment might change the chances of reaching the desired consequence. This is formalized by the function *consequenceContingency*, which takes antecedent stimuli to the probability of success.

Such details show that a formalization process is not just a matter of translation. Sometimes it is necessary to add notions and to infer, from unclear prose, what was actually meant.

We now move on to study some operations. In Z, we say that an operation is *total* if, and only if, its preconditions cover all possibilities. This concept will guide our analysis from here on.

Operants might be either created or modified. Here, we shall focus on operant modification, which can be achieved in four ways.

First, a new environmental condition might be learned. This is called a discrimination operation, for it allows the organism to discriminate among several environmental possibilities. Each possibility is defined by a set of *discriminative stimuli*.

<i>DiscriminationOp</i>
<i>OperantOp</i>
<i>discriminativeStimuli?</i> $\notin$ dom <i>consequenceContingency</i>
<i>consequence?</i> <u>sCauses</u> <i>consequence</i>
<i>discriminativeStimuli?</i> $\in$ dom <i>consequenceContingency'</i>
<i>consequenceContingency'</i> ( <i>discriminativeStimuli?</i> ) $>_1$ <i>min_correlation</i>

In the above schema we import *OperantOp*, which defines a general operation over an operant but is not necessary for the present discussion and, thus, is omitted.

Second, an already known environmental condition might lead to the operant consequent stimulus, which strengthens their relation.

$\frac{\text{OperantConditioningOp}}{\text{OperantOp}}$ $\text{discriminativeStimuli?} \in \text{dom consequenceContingency}$ $\text{consequence?} \underline{s\text{Causes}} \text{consequence}$ $\text{consequenceContingency}'(\text{discriminativeStimuli?})$ $\geq_1 \text{consequenceContingency}(\text{discriminativeStimuli?})$
-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Third, a known environmental state might not lead to the desired consequence, which reduces their relation.

$\frac{\text{ExtinctionOp}}{\text{OperantOp}}$ $\text{discriminativeStimuli?} \in \text{dom consequenceContingency}$ $\neg (\text{consequence?} \underline{s\text{Causes}} \text{consequence})$ $\text{consequenceContingency}'(\text{discriminativeStimuli?})$ $\leq_1 \text{consequenceContingency}(\text{discriminativeStimuli?})$
---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Finally, if neither the environmental condition is known, nor the consequence desired, the operant simply remains unchanged.

$\frac{\text{NeutralOp}}{\text{OperantOp}}$ $\text{discriminativeStimuli?} \notin \text{dom consequenceContingency}$ $\neg (\text{consequence?} \underline{s\text{Causes}} \text{consequence})$ $\text{consequenceContingency}'(\text{discriminativeStimuli?})$ $= \text{consequenceContingency}(\text{discriminativeStimuli?})$
----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Notice that these four definitions form a total operation: they cover all possibilities for the input variables *discriminativeStimuli?* and *consequence?*:

1. *DiscriminationOp* accounts for the case in which *discriminativeStimuli?*  $\notin$  *dom consequenceContingency*.
2. *OperantConditioningOp* handles the case in which *discriminativeStimuli?*  $\in$  *dom consequenceContingency* and *consequence?*  $\underline{s\text{Causes}}$  *consequence*.
3. *ExtinctionOp* occurs when *discriminativeStimuli?*  $\in$  *dom consequenceContingency* and  $\neg (\text{consequence?} \underline{s\text{Causes}} \text{consequence})$ .
4. *NeutralOp* accounts for the remaining case.

This model can be further refined by adding the notions of reinforcement and punishment. Each of these, in turn, can be either *positive* or *negative*. A positive reinforcement accounts for the provision of a pleasant stimulus (e.g., provision

of food), while a negative reinforcement stands for the removal of an unpleasant stimulus (e.g., relief from pain through analgesics). Punishment is analogous. At last, there is the case in which the stimulus is neither pleasant nor painful. Hence, there are five possibilities.

<i>PositiveReinforcement</i>
<i>StimulusUtility</i>
<i>consequence?</i> : <i>Stimulus</i>
$sUtility(consequence?) >_1 neutral$
$stimulus\_status(consequence?) = Beginning$

<i>NegativeReinforcement</i>
<i>StimulusUtility</i>
<i>consequence?</i> : <i>Stimulus</i>
$sUtility(consequence?) <_1 neutral$
$stimulus\_status(consequence?) = Ending$

<i>PositivePunishment</i>
<i>StimulusUtility</i>
<i>consequence?</i> : <i>Stimulus</i>
$sUtility(consequence?) <_1 neutral$
$stimulus\_status(consequence?) = Beginning$

<i>NegativePunishment</i>
<i>StimulusUtility</i>
<i>consequence?</i> : <i>Stimulus</i>
$sUtility(consequence?) >_1 neutral$
$stimulus\_status(consequence?) = Ending$

<i>NeutralReinforcementOp_1</i>
<i>OperantOp</i>
$sUtility(consequence?) = neutral$

Again, these possibilities account for all cases. We can integrate them with the previous schemata using the following formulae.

$$T\_FundamentalOperantOp \hat{=} DiscriminationOp \vee OperantConditioningOp \vee \\ ExtinctionOp \vee NeutralOp$$

$$PositiveReinforcementOp\_1 \hat{=} T\_FundamentalOperantOp \wedge \\ PositiveReinforcement$$

$$PositiveReinforcementOp\_2 \hat{=} OperantFormationOp \wedge PositiveReinforcement$$

$$PositivePunishmentOp\_1 \hat{=} T\_FundamentalOperantOp \wedge \\ PositivePunishment$$

$$PositivePunishmentOp\_2 \hat{=} OperantFormationOp \wedge PositivePunishment$$

We may now combine all of the above and define a rather complex total operation.

$$T\_OperantOp \hat{=} PositiveReinforcementOp\_1 \vee NegativeReinforcementOp\_1 \vee \\ PositivePunishmentOp\_1 \vee NegativePunishmentOp\_1 \vee \\ NeutralReinforcementOp\_1$$

## 5 Discussion

In this paper, we discussed the formalization of the Behavior Analysis psychological theory, a traditionally informal domain of knowledge. We argued that, though informal, such theory is sufficiently precise in order to allow a complete formal specification. Moreover, we tried to show that there is much to gain with such a formalization and that particular issues arise when dealing with it.

The formal specification brings new questions to the knowledge it formalizes. As such, it can be a theoretical tool for Psychology. For instance, we saw that graphs can be used to model certain stimuli properties. Furthermore, once implemented, experiments can be performed to validate the theory. Experiments that yield unexpected results might demonstrate that the underlying theory is not correct. And because a formal specification is responsible for the implementation, the faulty assumptions could be more easily located.

The construction of the ontology described in Sect. 2 was easy and fast. Moreover, its structure was simple enough in order to allow a person not familiar with formal specifications to read it. Therefore, it constituted an useful prototype, which could be used both to determine the value of further formalization and to allow an expert in the theory to validate the model.

The Z formal method allowed useful techniques, such as the definition of total operations and of refinement levels. Looking for total operations forces us to examine all possibilities of transformations. Hence, it helps spotting faults both in the specification and in the original theory. Different levels of refinements allow us to cope with incomplete information. Thus, when we are not sure about the details of a particular concept, we can nevertheless achieve a formalization, by breaking it into several levels of abstraction.

Z also encourages modularization through schemata and integration through schema calculus. We employed this facilities to divide as much as possible the

concepts being formalized. We also found useful to group together into subsystems all concepts that relate to some major division of the theory. One of the advantages of this approach is that modifications in the specification tend to be localized.

The next step regarding this work will be the formal verification of properties of the specification (e.g., consistency). An implementation, then, will follow and conclude the project.

## Acknowledgments

This work has been supported by CNPq (Conselho Nacional de Desenvolvimento Científico e Tecnológico) and FAPESP (Fundação de Amparo à Pesquisa do Estado de São Paulo). We also thank Walkiria Helena Grant, from the Institute of Psychology of the University of São Paulo, for proofreading our initial ontology.

## References

1. Skinner, B.F.: Science and Human Behavior. The Free Press (1953)
2. Jacky, J.: The way of Z: practical programming with formal methods. Cambridge University Press, New York, NY, USA (1996)
3. d’Inverno, M., Luck, M.: Understanding Agent Systems. Springer (2003)
4. d’Inverno, M.: Modelling and simulating the behaviour of adult stem cells using agent-based systems (2006) <http://www2.wmin.ac.uk/~dinverm/cell/index.htm>.
5. Bush, P.C., Sejnowski, T.J.: Simulations of a reconstructed cerebellar purkinje cell based on simplified channel kinetics. *Neural Computation* **3**(3) (1991) 321–332
6. Lytton, W.W., Sejnowski, T.J.: Simulations of cortical pyramidal neurons synchronized by inhibitory interneurons. *Journal of Neurophysiology* **66**(3) (1991) 1059–1079
7. Woodcock, J., Davies, J.: Using Z: Specification, Refinement, and Proof. Prentice Hall (1996)
8. W3C: Web ontology language (2004) <http://www.w3.org/2004/OWL/>.
9. Informatics, S.M.: The protégé ontology editor and knowledge acquisition system (2006) <http://protege.stanford.edu/>.
10. Saaltink, M.: The Z/EVES 2.0 User’s Guide. ORA Canada (1999)
11. Catania, C.A.: Learning. Prentice Hall (1998)
12. Watson, J.B.: Psychology as the behaviorist views it. *Psychological Review* (20) (1913) 158–177