Building a Motivational Subsystem
for the Cognitive Systems Toolkit

Eduardo de Moraes Froes and Ricardo Ribeiro Gudwin
DCA-FEEC-UNICAMP
Emails: edufroes@dca.fee.unicamp.br and gudwin@dca.fee.unicamp.br

***Abstract*—Motivations and emotions are intrinsically embedded in animal cognition and behavior, particularly in humans. They are responsible for supporting decision making, stimulating different behaviors such that their internal needs are satisfied. These needs can be of physiological origin (hunger, thirst, sleep and damage avoidance) or social (friendship, curiosity, honor, etc.). This work proposes the design and implementation of a motivational system endowed with motivational and emotional capacities for the ”Cognitive System Toolkit” (CST), a Java-based software toolkit for cognitive computing being developed by our research group, based on studies of theories on motivational and emotional behavior available in the literature and the different implementations of motivational and emotional systems in known cognitive architectures.**

***Keywords*—Motivational System, Goal-directed Behaviors, Motivations, Emotions, Cognitive Architectures, Cognitive System Toolkit (CST).**

I. INTRODUCTION

Many efforts have been employed to understand how motivations and emotions can be useful for the construction of intelligent systems. Inspired originally on the earlier studies on general intelligence in the late 70's, the field of "Cognitive Architectures" (CAs) arose to designate essential structures and processes of a general computational model inspired by cognition and behavior, becoming an important sub-field of *Cognitive Computing*. Over the last 40 years, a large number of CAs has been proposed. In 2010, Samsonovich [1] conducted a broad study resulting in a comparative table, presenting a comprehensive review of the most important implemented CAs in the literature. More recently, Kotseruba [2] performed a detailed analysis of how this field of research developed in the last 40 years. There are many flavors of CAs, with different aspects of cognitive modeling being used in their implementation. In this work, we focus on CAs presenting some sort of motivational or emotional capability.

The study of motivational or emotional capabilities inserts itself under the broader field of behavior systems. Strictly speaking, if we want to classify any kind of behavior (performed by animals or machines), possibly we will discover that they can be in one of the following classes: random, reactive or goal-directed behaviors.

Random behaviors are those which do not depend, or do not receive influence from anything else. Reactive behaviors are behaviors that depend on some kind of input, to which the system reacts. Usually, this input is some kind of sensory input, which is transformed internally into an output by the system. The output of a reactive system is basically a deterministic function of its inputs, and possibly some sort of internal variables of the system. Both random and reactive behaviors are typical in machines and other non-animated objects. The understanding of the third category is a little bit trickier. Goal-directed behaviors are typical in living systems, particularly animals. The idea of a goal-directed behavior is that this behavior is not just a random behavior or a reaction to something else. The idea of a goal-directed behavior is that there is a finality in that behavior, a goal that must the reached, a purpose to be accomplished. There is a deep discussion in philosophy regarding this issue. This discussion starts in Aristotle and his notion of final cause, passing through subsequent discussions on teleology, teleonomy and finally reaching cybernetics: the science of control. It is important to differentiate goal-directed behaviors from reactive behaviors. A reactive behavior does not have an envisioned future state to achieve. It simply reacts to the inputs, without the requirement of what should come in the future. Goal-directed systems, on the contrary, have an envisioned future to reach. System outputs are in some sense committed with this expected future.

The research on motivational systems is inserted in this study of goal-directed behavior. There are simple kinds of goal-directed behaviors. Any closed-loop control system, like a thermostat, performs a sort of goal-directed behavior. When a thermostat determines the control signals to a cooler system, it is not simply reacting to an input, but they are changing the environment in order for a goal (e.g. a reference temperature) to be reached. And due to the inherent feedback of a closed-loop system, the environment slowly converges to this temperature. But goal-directed systems can be much more complex than a simple thermostat. The definition of a *goal* might be as simple as a precise future state to be reached, or involve desired properties for a still unknown future state, where any feasible future state meeting these properties might be acceptable. According to Hull [3], goal-directed behavior is explained in terms of *needs* intrinsic to living beings, which drive their behavior at the environment. Using this idea, many researchers, as e.g. Toates [4] and [5] proposed biologically-inspired motivated behavior systems using the idea of internal needs as a source of motivation. These internal needs are called "Drives" and perform important functions in animals such as: i) motivate them to select the best decision to suppress physiologic and social needs. ii) influence the emotive state passing activation energy to the emotive processes. iii) provide a learning context to animals to enhance and create new skills.

Many well known cognitive architectures (e.g. SOAR, CLARION, LIDA, CERA-CRANIUM, MicroPsi, etc.) rely on some sort of motivation and/or emotion representation in their internal modules. In order to gain some sort of goal-directed behavior, motivational or emotional mechanisms are not simply a hype, but a powerful tool in the construction of action-selection mechanisms. These capabilities are surely expected in any modern cognitive architecture.

In this work, we propose a Motivational System for the CST Toolkit [6], a Java-based toolkit for the construction of CAs being developed by our research group, which relies on many concepts which are common to other CAs, but in its current stage of development still does not have motivational capabilities. In order to propose this Motivational System, we studied different implementations of motivational systems in different cognitive architectures, and the different available theories for motivation and emotions in the field of cognitive modeling.

II. MOTIVATIONS AND EMOTIONS FROM A COGNITIVE MODELING PERSPECTIVE

The concept of motivational behavior in CAs has its inspiration in studies about human motivation, realized by Hull [3] and Maslow [7] in Cognitive Psychology. According to Hulls theory of behavior [3], when a motor action is a pre-requisite to optimize the probability of survival of either an individual or a species, we say that there is a state of needness. This need *motivates* or *drives* the associated motor action. So, Hull defines a Drive as being a variable used to characterize a need. Drives are used as a measurement of a desirable future state which a creature must reach, to survive. In a living being, a drive might be related to the many needs, such as the need for food, for water, for air, the need to avoid injury, to maintain an optimal temperature, the need to rest, to sleep, to mate, etc. In an artificial agent, drives are associated with the desirable behaviors we want the agent to manifest. They are involved with a desirable future state for the agent. So, a drive can be seen as the measurement of the agents success in achieving a purpose. Also, a behavior which is performed to satisfy a drive is said to be a motivated behavior.

The notion of drive is very important for understanding another critical cognitive capability: emotions. There is an intrinsic relationship between motivations and emotions. The concept of emotions came from cognitive psychology and philosophy, as an alternative way to address the problem of behavior generation [8]. There is no consensus about what emotions really are. Different approaches have different views for what they are and how to model them. For example, Ortony, Clore & Collins [9] understand emotions as “valenced reactions to events, agents, or objects, with their particular nature being determined by the way in which the eliciting situation is construed”. Sloman [10], in turn, understands emotions as internal “alarms” which give a momentary emphasis to certain groups of signals. Damasio [11] distinguish between “emotions”, which affect the body and “feelings”, which are a cognitive introspection of emotion. Other authors have completely different views about what emotions are. For example, to Canamero [12], emotions work like “amplifiers” for motivations, working as homeostatic processes related to physiological variables.

III. THE CLARION MOTIVATIONAL SUBSYSTEM

CLARION [13], [14] is a cognitive architecture composed of distinct sub-systems, each of them processing information of two different kinds: explicit and implicit. Explicit information is usually symbolic, and performed by a rule-based system. Implicit information is usually sub-symbolic, and performed by means of a neural network. Clarion is composed by 4 main subsystems: action-centered subsystem (ACS), non-action-centered subsystem (NACS), motivational subsystem (MS) and meta-cognitive subsystem (MCS). Each sub-system is responsible for performing a specific function in the architecture. ACS is responsible for controlling agents' actions. NACS is responsible for maintaining the agent's general knowledge, whether explicit or implicit. The MS sub-system is responsible for promoting underlying motivations for the agent's operations. This means that the MS provides impulses and feedbacks to verify if the results of these operations were satisfactory or not. MCS is responsible for monitoring, directing and modifying the operations of all other sub-systems, mainly the ACS subsystem.

According to Sun [14] the motivational sub-system (MS) is responsible for supporting decision-making within CLARION by providing the infrastructure for “goals” and “drives” . The MS basic infrastructure includes drives at the bottom-level (implicit) and goals at the top-level (explicit). The drives are directly related to the needs of the cognitive agent. These needs are the key factors for the agent to survive in an environment. These fundamental needs are called “Primary Drives”. Primary drives can be classified into two types, representing physiological and social needs. For example, drives such as: thirst, hunger, sleep, avoid physical damage and reproduction are drives related to physiological needs (referred as “Low-Level Primary Drives”). In contrast, honor, autonomy, curiosity and fairness are drives related to social needs (classified as “High-Level Primary Drives”). Additionally, CLARION presents another kind of drive, derived from the combination of other drives. These drives are classified as “Secondary Drives”. Sun [14], [15] analyze secondary drives as more mutable and usually acquired through a primary drive satisfaction process. Additionally, secondary drives might be learned through the process of conditioning, or by receiving external instructions.

As previously stated, drives help in making decisions within CLARION. Let's imagine a thirst drive in a cognitive agent. When the need for water increases significantly, the drive intensity raises. The consequence is to set new “goals”, in order to suppress the intensity of the thirst drive. Some drives may be more important than others, and therefore they are organized hierarchically. Drives at the highest hierarchy are most likely to be satisfied at any given time. An example is the self-preservation drive, while compared to a reproduction drive. An agent will not be motivated to reproduce if this behavior puts the agent in a dangerous situation. In this case, the auto-preservation drive is said to be higher in the hierarchy than the reproduction drive. One important factor is the accessibility of a drive. Something which is required for a drive must first be available before the drive can be satisfied. For example, if an agent requires water, but it has not yet found it, the agent must continue his search for water first, before trying to drink. In this way, the need for water will directly determine the set of goals of a cognitive agent. Even if a need is stronger than others, the motivational subsystem might try to satisfy another need if a required object for the first need is not available [15].

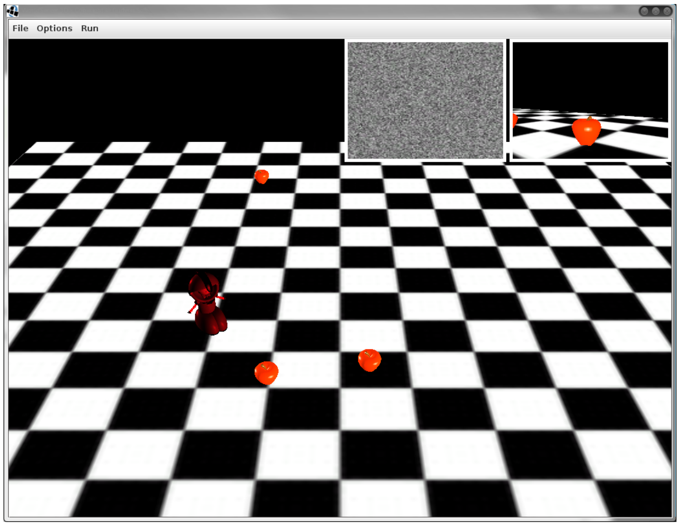


Fig. 1. The WorldServer 3D Application.

According to Sun [14], goals are directly generated from drives. There are two ways of setting a goal in CLARION. The first one is through what Sun calls“Balance-of-interests”. This method proposes a competition among goals. Each drive votes in multiple goals with a different numerical value (values varying between 0 and 9) indicating the priority for selecting a certain goal. The goal with the highest number of votes is selected. The second method proposed by Sun, called “Winner-takes-all”, selects a goal based on the highest-intensity drive. This means that the chosen goal will be the one satisfying the winner drive.

IV. THE COGNITIVE SYSTEM TOOLKIT AND THE WORLD SERVER 3D VIRTUAL ENVIRONMENT

The MS we are developing is designed as a subsystem of the CST toolkit [6]. CST is a computational toolkit for the construction of general purpose CAs being developed in the Java language at the University of Campinas. The toolkit relies on a set of core concepts which were described in [6]. Among these concepts, the notion of a *codelet* is of utmost importance. This notion was introduced originally by Hofstadter [16], and further enhanced by Franklin [17]. Codelets are pieces of code that are executed continually and cyclically, being responsible to perform any cognitive activity in artificial agents. Codelets are the basic building blocks used to develop the mind structure of a cognitive agent, implementing subsystems for many cognitive functions like attention, perception, emotions, learning and others. CST's codelets are quite similar to those used in the LIDA cognitive architecture [17]. A second important core concept in CST is the notion of *memory objects*, which are conceptual (and computational) entities representing a cognitive agent's many kinds of memories. In CST, memory objects are a kind of wrapper to any data which is supposed to be used by a codelet. They are used as input and output variables, for long-term or short-term use. More details on the CST project can be found in [6], [18].

To test and evaluate our MS, we used the *WorldServer3D* application (WS3D), a virtual environment developed by our research group at Unicamp. Figure 1 provides a screenshot of the WS3D interface. WS3D allows the simulation of a virtual environment where the user is able to test different agent minds to an artificial creature, a character which has a series of tasks in the virtual environment. The creature “lives” in a 3D virtual world, where it can find food (apples and nuts) and valuables (jewels of different colors). An agent can sense these objects and perform a set of actions on them. In the case of jewels, an agent can get them (put in its bag) or bury them (hide it from an opponent). In the case of nuts and apples, an agent can get them (put in bag), bury them, or eat them. Once an object appear at the environment, it stays there until they are picked, or eaten (food). The difference between apples and nuts is that apples become rotten after some time (looses its ability to supply energy), but nuts do not. A leaflet provides combinations of jewel colors which can be exchanged by points. The goal of the creature is to maximize its points while maintaining its energy balance. For that, it needs to evaluate its leaflet and try to pick jewels which maximize its income of points, while maintaining its energy balance. Creatures can bury apples, nuts or jewels, to avoid that other creatures pick them, if they are competing in the virtual environment. Knowing the exact position where they were buried, the creature can return later and pick the object by first unburying them. This could be a strategy if their bags are full. If the creature's energy level reaches the value 0, the creature dies. WS3D implements the communication among creatures and their minds using TCP/IP. Thus, it is possible to run the virtual environment in a machine and the minds in the same or in other machines, if the required computational effort is too intense.

V. THE CST MOTIVATIONAL SUBSYSTEM

Our Motivational System is constructed on top of many ideas collected from the literature, from where we derived our proposal. One important background theory we are relying on is the theory on *Subsumption Architecture*. The *Subsumption Architecture* is a generic name for a family of computational architectures used in intelligent control (particularly in robotics), developed by Rodney Brooks in the 90's, which gave rise to the whole *Behavior-based Robotics* research field [19].

Some authors [20]–[22] proposed a *Dynamic Subsumption* scheme, in which there is no fixed dominant input in a suppression node, but this dominance can be changed dynamically in time, according to specific situations. Dynamical subsumption is the standard way in which CST merges multiple behaviors affecting the same actuators. Our Motivational Subsystem is a variation of this scheme, using Hull's [3] ideas of *needs* and *drives*. Considering that a *drive* is a kind of a measure of the intensity of a *need*, this drive is used to evaluate the relative importance of a specific behavior it is attached to. So, many alternative behaviors are evaluated in parallel, according to the agent's many needs and the dynamic subsumption mechanism selects the behavior related to the most urgent need. It is implicit in this view that the behavior affected by the drive might cause a reduction in the corresponding need.

Figure 2 illustrates how drives are used to motivate a behavior. We use CST's concepts of codelets and memory objects in order to compose the motivational system. Codelets are represented as rounded boxes and memory objects as circles within dotted rounded boxes (the different memories in the system - see [6] for a more elaborate description of

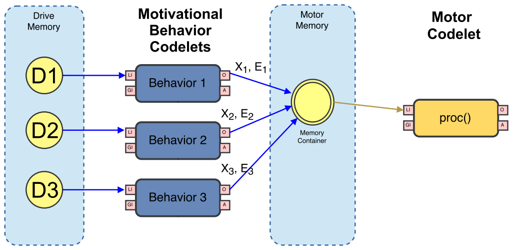


Fig. 2. How Motivated Behaviors are driven.

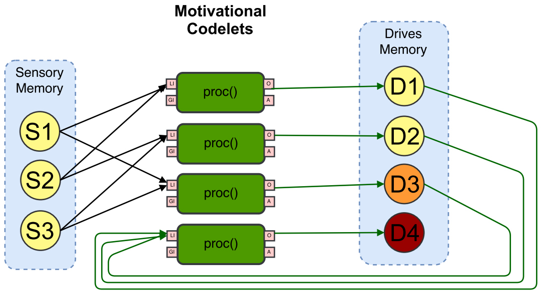


Fig. 3. Drives are generated by Motivational Codelets.

codelets and memory objects). The idea is that many behaviors provide different motor prescriptions for the same actuator X. The different drives D1, D2 and D3 will be used to produce three different memory objects $\{X_i, E_i\}$, each one holding a pair with X_i being a motor signal and E_i being an evaluator for the memory object. The Memory Container in the Motor Memory will be used to choose which motor signal X_i will be conveyed to the Motor Codelet (the one with biggest E_i).

In our model, drives are memory objects generated by *Motivational Codelets* in two ways: i) directly from sensory variables or ii) derived from other existing drives in the agent. In the first case, drives are said to be of a primary type. In the second case, they are said to be of a secondary type, being necessarily considered *High-level Secondary Drives*. According to 14], [23 the *High-level Secondary Drives* are more changeable, and can be acquired to satisfy the primary drives through of "conditioning" or from external instructions.

In figure 3 we show Motivational Codelets generating a set of drives. Drives D1 and D2 are Low-level Primary Drives, D3 is a High-level Primary Drive and D4 is a High-level Secondary Drive.

Different cognitive architectures [14], [24] propose that there might be a functional distinction between high-level and low-level primary drives, and also high-level secondary drives. If all the drives are functionally exactly the same, there might be situations in which the system might give preference to e.g. a social drive, like "social acceptance" instead to a physiologic drive like hunger. This is acceptable in a normal situation, but if both drives are in a state of "urgency", in which the system have a risk to collapse (as e.g. in a state of minimum energy), it is clear that the hunger drive must have some sort of priority, due to the potential harmful consequences to the system (if the system shuts down due to lack of energy, the social "urgency" has no meaning at all). To allow a special treatment in situations of "emergency", we introduced a special mechanism in our model of motivations. This special mechanism works as follows. Besides the *activity level*, or intensity, already associated to a drive, we include now more two parameters: *urgency threshold* and *priority*. The *urgency threshold* is a limit value for the *activity level* that, up reaching this value, the drive is considered to be in a state of urgency. In this state of urgency, instead of using the *activity level* to select among different drives, the system must use a fixed *priority* assigned to the drive, which will warranty that the correct drive will win among motivated behaviors.

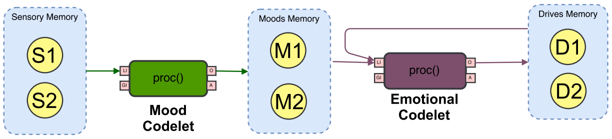


Fig. 4. Emotional Codelets might assign a cognitive distortion to drives.

Secondary drives require a further elaboration, as they are directly dependent of other drives. To compute the activity level of a secondary drive, Motivational Codelets perform a weighted average of their input drives, modulated by a *relevance* factor. This *relevance* attribute is a numeric variable ranging from 0 to 1, which is multiplied by the activity level of the input drive to compute the secondary drive activity level.

Finally, our motivational subsumption system includes the effect of emotions. For that, we will be relying on Canamero's model of emotions [12] where emotions are viewed as cognitive distortions on the map of drives, changing for a given time the relative intensity of the different drives on the system, amplifying some drives and decreasing the intensity of others. For that purpose, we include another parameter in our conception of drive: *emotional distortion*, which is a value (positive or negative) which is summed with the *activity level* of a drive, in order to prioritize the motivated behaviors. In our architecture, the definition of this *emotional distortion* is assigned to Emotional Codelets, as illustrated in figure 4.

Emotional codelets are modulated by *Moods*. Moods are a kind of emotional state which is determined by *Mood Codelets*, based on sensory data acquired from the environment. Depending on being in a *normal* mood, or on an alternate mood like *sleepy*, *worried*, *terrified*, *in love*, etc., the emotional codelet might determine a cognitive distortion to the drives landscape, making that a different priority is used to select motivated behavior.

Definition 1: A *drive* d is defined as a tuple $d = \{A, \;, \;, p\}$ where:

> A is the *activity level* representing the intensity of the drive, $A \in [0, 1]$
> is the *urgency threshold* such that if $A >$ then the drive is in a state or urgency, $\in [0, 1]$.
> is the *emotional distortion* which should be added to A in order to compute the intensity of the drive, $\in [\;1, 1]$ and $0 \;\; (A + \;) \;\; 1$.
> p is the *priority*, $p \in [0, 0.5]$, which is used to assign a fixed priority while in urgency mode.

Now, based on this definition of *drive* we can understand how the calculation of the *Eval* value in the Memory Objects which are generated by the Motivational Behavioral Codelets should be computed:

Definition 2: The calculation of the *Eval* parameter of a Memory Object generated by a Motivational Codelet, should adopt the following criteria:

$$Eval = \begin{cases} 0{:}5 + p & \text{if } A > \quad , \\ (A + \quad){=}2 & \text{if } A \quad\quad . \end{cases} \tag{1}$$

If we follow definition 2 we will see that while in normal mode, Eval will be a number between 0 and 0.5, while if in urgency mode, Eval will be a number between 0.5 and 1. Using this convention, we have a warranty that while in a state of urgency, the drive with the biggest priority will always be selected by the dynamic subsumption mechanism. This is depicted in figure 5.

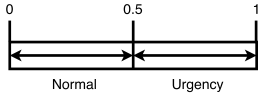


Fig. 5. Value of Eval while in Normal or Urgency Mode.

VI. DEVELOPMENT AND EXPERIMENTS

Our MS was developed and included as a `motivational` package into CST source code. The code for the following classes were included: *Drive*, *Mood*, *MotivationalCodelet* and *MoodCodelet*. We developed unit and integration tests of these classes, verifying the behavior of each class separately and together. Performance tests are still missing, though.

A. Experiments

To verify the efficiency and effectiveness of our motivational package, two applications were developed, controlling intelligent agents in the WS3D virtual environment. The first one is a reference application, using a purely reactive behavior. The second one is an application embedding a motivational system using our motivational package. Both applications relates to the same scenario, consisting of playing a game where a creature must capture a set of jewels prescribed by a leaflet, while keeping their energy level above zero. At the beginning of each simulation, the environment generates three leaflets for each creature, and a fixed amount of jewels (of different types) is randomly distributed at the environment. Jewels have a finite set of colors: blue, magenta, white, red, green and yellow. The total number of jewels for each leaflet is 9.

The creature's energy level can range from a maximum of 1000 to a minimum of 0, when the creature dies. This energy level decreases by steps of 50 throughout the simulation time. Both applications are able to perform the following behaviors:

Random Movement: responsible for performing random movements.

Collision avoidance: responsible for avoiding collisions in walls present in the simulation environment.

Capture a Jewel: responsible for capturing a jewel while next to it.

Eat Food: responsible for eating an apple or nut, while next to it.

Bury a Jewel or Food: responsible for burying a jewel or food while next to it.

Go to the Closest Jewel: responsible for moving the agent to the nearest jewel.

Go to the Closest Food: responsible for moving the agent to the nearest food.

Even though these behaviors are present in both applications, their implementation is different in each of them.

The simulation time for both the reactive and the motivational controller is 15 minutes. The experiment was run five times for each controller application, implying in 5 different scenarios in terms of jewels and food.

B. The Reactive and Motivational Controller Applications

The following Motivational Codelets were developed for the motivational controller: Hunger, Ambition, Boredom and Danger Avoidance. They are responsible for generating and maintaining drives for specific behaviors in the creature. The hunger drive incites the creature in looking for food. This drive is triggered by the creature's energy level. If it is too low, this drive is high. As soon as the creature encounters food and eats it, its energy increases and the drive decreases. The Ambition drive causes the creature to explore the environment looking for its leaflets jewels. Unlike the hunger drive, the ambition drive tends to always get higher, representing that the closest it is from its final goal (the overall leaflet), its ambtition becomes higher. However, as soon as the creature completes its leaflets, the ambition drive's activation decreases to zero. The Boredom drive increases according to whether the creature stays for a long time (20 seconds) in the same position in the environment. Generally, this occurs when the creature does not find the objects it wants, and consequently, it rotates on its own axis in the same position. This drive encourages a random movement behavior, such that the creature might go to a different position and possibly be able to detect objects which cannot be detected from the current position, possibly "hidden" behind walls. Finally, the Danger avoidance drive prompts a creature to avoid collisions with environmentally unreliable objects. If a creature, in its way to a different position is risking to collide with an object (wall, food, or jewelry), the activation of this drive increases significantly and encourages different actions according to the object in front of it. For example, if the creature is risking to collide with a jewel that is not in its leaflets or a food that is in the way while going somewhere, then the creature buries them. In the case of buried food, the agent has the ability to store the positions where it buried them, and later, if necessary, unbury and eat it.

The creature's visual sense also affects the many drives. When the creature perceives a food in its visual field, the hunger drive activation grows by 20%. The same happens with the drive of ambition when the creature perceives a jewel in its leaflets, which causes an increase of 20% in its activation. When the agent perceives walls in their visual field, the danger avoidance drive increases by 0.05.

Table I prescribes the many parameters used for these drives, together with the used heuristics for each of them.

The reactive controller exhibits the same behaviors used in the motivational controller. However, the reactive controller uses a classical Subsumption architecture, as proposed by [25] for decision making. According to the Subsumption architecture, behaviors are structured into layers of preference, such that a higher level behavior inhibits a lower level one. In

TABLE I. DRIVES' PARAMETERS

Drive	Priority	Urgency	Formula
Hunger	0.3	0.8	$A = 0{:}95 * \max(F_d; F_d * (1 + F_s))$
Ambition	0.2	0.9	$A = 0{:}95 * \max(A_d; A_d * (1 + J_s))$
Boredom	0.4	0.8	$A = T_{sp}$=20000
Danger	0.45	0.8	$A = \begin{cases} 0{:}8 + B_s & if\, B_d <= 40 \\ B_s & if\, B_d > 40 \end{cases}$

where A is Activation, F_d is Food deficit, F_s is Food stimulus, A_d is Ambition deficit, J_s is Jewel stimulus, T_{sp} is Time in same position(ms), B_s is Block stimulus and B_d is Block distance

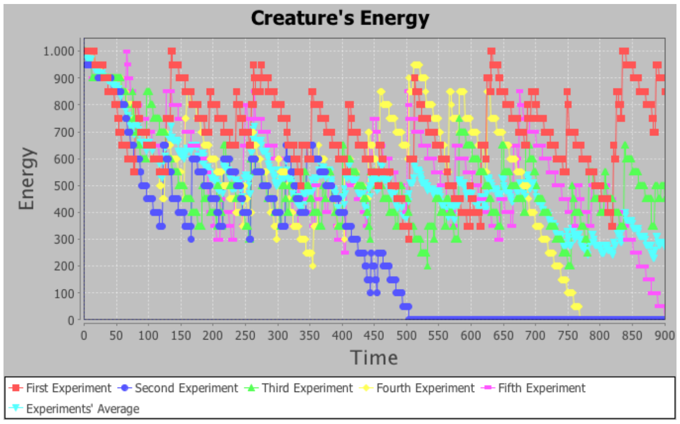


Fig. 6. The Creature's Energy spent along of the time by Reactive Controller.

the reactive controller, when a jewel or a food enters the field of vision, both the "go to the nearest jewel" and "go to the nearest food" behaviors are activated. However, since "going to the nearest jewel" is more relevant than "going to the nearest food", the last one is inhibited by the former. The "random movement" behavior inhibits "going to the nearest jewel" and "going to the nearest food". "Collision Avoidance" inhibits "going to the nearest jewel", "going to the nearest food" and "random movement". And finally, "going to the nearest jewel" inhibits "going to the nearest food". There is only one situation when "go to the nearest food" inhibits the "go nearest jewelry" behavior. This occurs when the creature's energy is less than or equal to 40%. "Eating food" and "Capture jewel" will only be active if the creature is close enough to food or jewel (less than 60 pixels). More details on the reactive controller can be found in the application source code, according to the link in section VIII.

VII. RESULTS AND DATA ANALYSIS

Figures 6 and 7 provide a summary on how the creatures energy has changed for the reactive and the motivational controller, along the five experiments for each case.

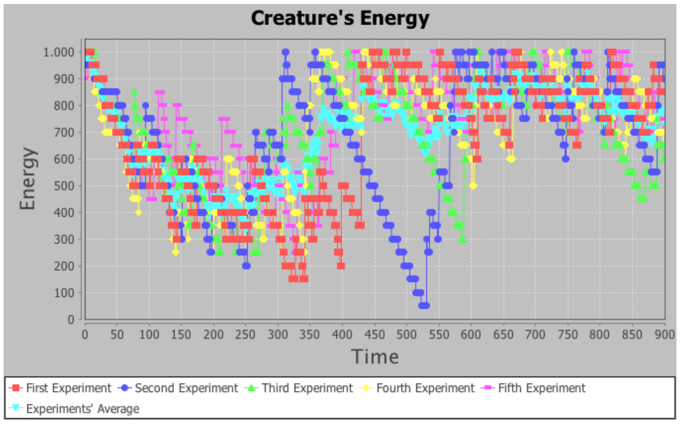


Fig. 7. The Creature's Energy spent along of the time by Motivational Controller.

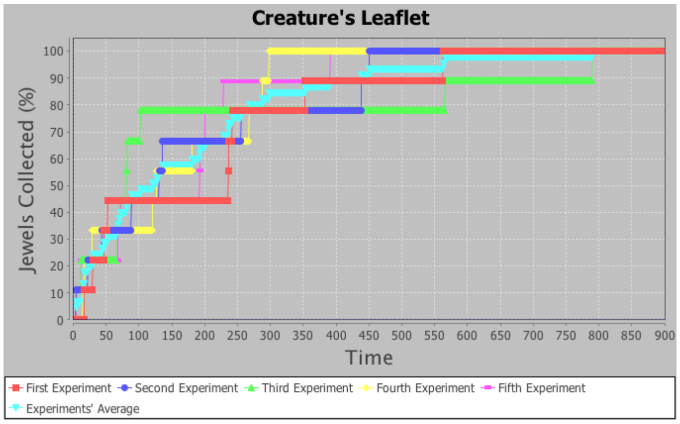


Fig. 8. Percentage of Jewels Captured from Reactive Controller.

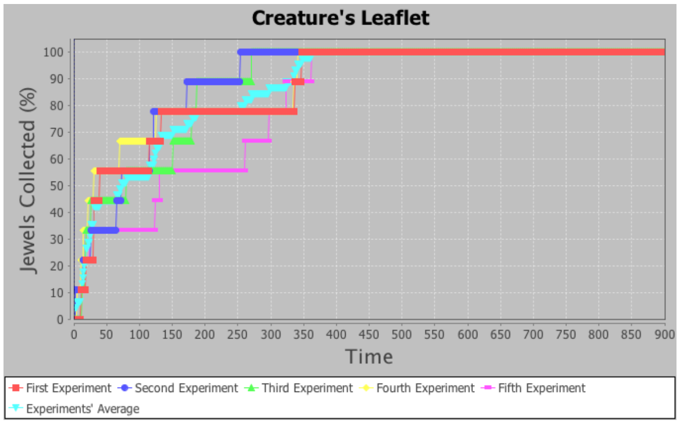


Fig. 9. Percentage of Jewels Captured from Motivational Controller.

The motivational controller never reached its minimum energy. Even though the reactive controller also demonstrated a good performance, regarding energy balance, in three experiments (Second, Fourth, and Fifth) the creature totally lost its energy. If we look at the average of both graphics, the motivational controller controlled the creature's energy more effectively than the reactive controller, because for most of the execution time the creature's energy remained above 50%. In contrast, the reactive controller allowed the energy level to be below 55% for the most of the experiment time. The experiment that obtained the best result for the reactive controller from the point of view of energy expenditure was the first, where the creature's energy decreased at a worst case in 30% of the total (at 504 seconds). The worst result was the second experiment where the creature's energy reached 0% at 504 seconds. For the motivational controller, the best result was in the fifth experiment, where the critical energy point was 25% at 326 seconds. The worst experiment was the second, where the critical point reached 5% energy at 523 seconds.

Figures 8 and 9 show the percentage of leaflet jewels captured by the agent during the experiment time, for the five experiments.

As we can see from the graphics above, both the reactive

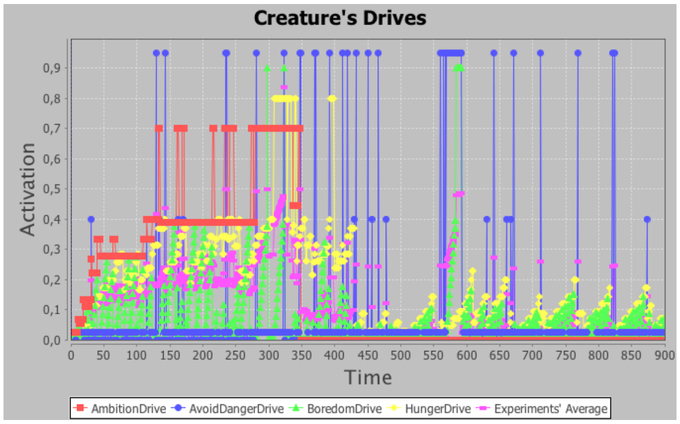
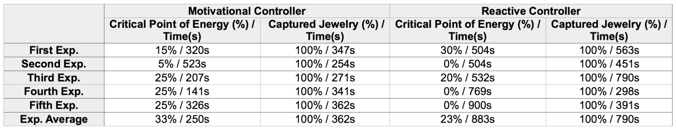


Fig. 10. Drives activation of Motivational Controller.



	Motivational Controller		Reactive Controller	
	Critical Point of Energy (%) / Time(s)	Captured Jewelry (%) / Time(s)	Critical Point of Energy (%) / Time(s)	Captured Jewelry (%) / Time(s)
First Exp.	15% / 320s	100% / 347s	30% / 504s	100% / 563s
Second Exp.	5% / 523s	100% / 254s	0% / 504s	100% / 451s
Third Exp.	25% / 207s	100% / 271s	20% / 532s	100% / 790s
Fourth Exp.	25% / 141s	100% / 341s	0% / 769s	100% / 298s
Fifth Exp.	25% / 326s	100% / 362s	0% / 900s	100% / 391s
Exp. Average	33% / 250s	100% / 362s	23% / 883s	100% / 790s

Fig. 11. Comparison Between the Motivational and Reactive Controllers.

and the motivational controller were able to finish their goal in capturing jewels in all the experiments. But the motivational controller was able to finish its capture earlier than the reactive controller. In the average time series, the motivational controller finished his leaflets at 362 seconds while in the average reactive time series, it was able to conclude only at 790 seconds. In addition, in the only experiment where the reactive controller was faster in concluding its leaflet, (the fourth experiment), the creature ran out of energy at 769 seconds. The worst result of the reactive controller was in the third experiment, where the creature completed its leaflets only at 790 seconds. In this experiment, though, the creature was able to maintain its energy level above zero. The best result for the motivational controller was in the second experiment, where it was able to conclude the leaflet at 254 seconds. Its worst experiment was the fifth experiment, finalizing the leaflets at 362 seconds. Figure 10 provides a comparison of all these tests. Figure 11 shows the activation of the many different drives along the five experiments, for the motivational controller.

VIII. CONCLUSION

In this work we proposed a motivational system to be integrated in CST, the Cognitive Systems toolkit. This motivational system was tested in a simple experiment which provided good results, outperforming a purely reactive one. Our motivational system is inspired in the MS system in Clarion, but is different in many ways. Even though the Clarion MS is able to differentiate between low level and high level primary drives, allowing a mechanism were priority is assigned to a class of drivers over others, our implementation, using the priority and urgency threshold parameters provides a better mechanism, where a whole pyramid of needs (as e.g. in Maslow's theory) can be modeled (while in Clarion only 2 levels are possible). Our mechanism also allows an emotional distortion to be applied. Nevertheless, these advantages might require further testings in order to be fully evaluated.

The following links provide access to the source code of the most recent version of all the software used in this project.

https://github.com/CST-Group/cst.git.
https://github.com/CST-Group/MotivationalSystemWithWorldServer3D.git.
https://github.com/CST-Group/ReactiveSystemWithWorldServer3D.git.
http://cst.fee.unicamp.br/.

REFERENCES

[1] A. V. Samsonovich, "Toward a unified catalog of implemented cognitive architectures.," *BICA*, vol. 221, pp. 195–244, 2010.

[2] I. Kotseruba, O. J. A. Gonzalez, and J. K. Tsotsos, "A review of 40 years of cognitive architecture research: Focus on perception, attention, learning and applications," *arXiv preprint arXiv:1610.08602*, 2016.

[3] C. L. Hull, *Principles of behavior: An introduction to behavior theory.* Appleton-Century, 1943.

[4] F. M. Toates, *Motivational systems*. No. 4 in Problems in the Behavioral Sciences, Cambridge University Press, 1986.

[5] C. Breazeal *et al.*, "A motivational system for regulating human-robot interaction," in *Aaai/iaai*, pp. 54–61, 1998.

[6] A. L. Paraense, K. Raizer, S. M. de Paula, E. Rohmer, and R. R. Gudwin, "The cognitive systems toolkit and the cst reference cognitive architecture," *Biologically Inspired Cognitive Architectures*, vol. 17, pp. 32–48, 2016.

[7] A. H. Maslow, "A theory of human motivation.," *Psychological review*, vol. 50, no. 4, p. 370, 1943.

[8] R. W. Picard and R. Picard, *Affective computing*, vol. 252. MIT press Cambridge, 1997.

[9] A. Ortony, G. L. Clore, and A. Collins, *The cognitive structure of emotions*. Cambridge university press, 1990.

[10] A. Sloman, "Damasio, descartes, alarms and meta-management," in *Systems, Man, and Cybernetics, 1998. 1998 IEEE International Conference on*, vol. 3, pp. 2652–2657, IEEE, 1998.

[11] A. R. Damasio, *Descartes error: Emotion, Reason and the Human Brain*. G.P. Putnam Sons, New York, 1994.

[12] D. Cañamero, "A hormonal model of emotions for behavior control," *VUB AI-Lab Memo*, vol. 2006, 1997.

[13] R. Sun, "Cognitive architectures and multi-agent social simulation," in *Pacific Rim International Workshop on Multi-Agents*, pp. 7–21, Springer, 2005.

[14] R. Sun, "Motivational representations within a computational cognitive architecture," *Cognitive Computation*, vol. 1, no. 1, pp. 91–103, 2009.

[15] R. Sun, "A tutorial on clarion 5.0," *Cognitive Science Department, Rensselaer Polytechnic Institute*, 2003.

[16] D. R. Hofstadter, M. Mitchell, *et al.*, "The copycat project: A model of mental fluidity and analogy-making," *Advances in connectionist and neural computation theory*, vol. 2, no. 31-112, pp. 29–30, 1994.

[17] S. Franklin, T. Madl, S. Strain, U. Faghihi, D. Dong, S. Kugele, J. Snaider, P. Agrawal, and S. Chen, "A lida cognitive model tutorial," *Biologically Inspired Cognitive Architectures*, vol. 16, pp. 105–130, 2016.

[18] K. Raizer, A. L. Paraense, and R. R. Gudwin, "A cognitive architecture with incremental levels of machine consciousness inspired by cognitive neuroscience," *International Journal of Machine Consciousness*, vol. 4, no. 02, pp. 335–352, 2012.

[19] R. C. Arkin, *Behavior-based robotics*. MIT press, 1998.

[20] H. Nakashima and I. Noda, "Dynamic subsumption architecture for programming intelligent agents.," in *icmas*, pp. 190–197, Citeseer, 1998.

[21] Y. Hamadi, S. Jabbour, and L. Saïs, "Learning for dynamic subsumption," *International Journal on Artificial Intelligence Tools*, vol. 19, no. 04, pp. 511–529, 2010.

[22] F. W. Heckel and G. M. Youngblood, "Multi-agent coordination using dynamic behavior-based subsumption.," in *AIIDE*, 2010.

[23] R. Buck, *Human motivation and emotion*. John Wiley & Sons, 1988.

[24] R. J. McCall, *Fundamental motivation and perception for a systems-level cognitive architecture*. PhD thesis, THE UNIVERSITY OF MEMPHIS, 2014.

[25] R. Brooks, "A robust layered control system for a mobile robot," *IEEE journal on robotics and automation*, vol. 2, no. 1, pp. 14–23, 1986.