Towards Conscious-like Behavior in Computer Game Characters

Raúl Arrabales, Agapito Ledezma, and Araceli Sanchis

Abstract—The main sources of inspiration for the design of more engaging synthetic characters are existing psychological models of human cognition. Usually, these models, and the associated Artificial Intelligence (AI) techniques, are based on partial aspects of the real complex systems involved in the generation of human-like behavior. Emotions, planning, learning, user modeling, set shifting, and attention mechanisms are some remarkable examples of features typically considered in isolation within classical AI control models. Artificial cognitive architectures aim at integrating many of these aspects together into effective control systems. However, the design of this sort of architectures is not straightforward. In this paper, we argue that current research efforts in the young field of Machine Consciousness (MC) could contribute to tackle complexity and provide a useful framework for the design of more appealing synthetic characters. This hypothesis is illustrated with the application of a novel consciousness-based cognitive architecture to the development of a First Person Shooter video game character.

I. INTRODUCTION

Developing video game synthetic characters that are able to produce human-like behavior is an open problem and a great challenge. In fact, the ultimate goal is to design characters that are indistinguishable from human players. In other words, agents able to pass the Turing test [1] (or more specifically, an adapted version of the Turing test designed for assessing the believability of video game characters [2]).

Classical AI character implementations, like *Pac-Man* ghosts or *Space Invaders* alien spacecrafts, were relatively simple to program, and little or none AI techniques had to be used. However, as video games have evolved into more complex virtual environments and elaborated scripts, they require much more realistic behaviors for their characters. Pre-programmed behaviors are valid to some extend in certain scenarios, but in realistic games consumers expect to find artificial opponents behaving like humans. When compared with other human players the behavior of artificial characters is usually considered disappointing.

Current AI characters can be intelligent in some sense; however, they cannot match the behavior produced by a human player (see for instance the results of the last 2K BotPrize contest [3] – a competition based on the Turing test). To date, playing with other humans is generally more realistic and engaging than playing with state of the art synthetic agents.

In the following we introduce the hypothesis that using cognitive architectures inspired on models of consciousness could be an effective approach to deal with the complexity of the problem. Additionally, we argue that the field of MC can also benefit from the experimentation with such architectures in the domain of computer games. First of all, we introduce and clarify the concepts of embodiment and situatedness as essential features of conscious agents. We also support that these properties are present in the domain of computer games.

A. Embodiment and situatedness in video games

Physical embodiment and real world situatedness are considered key factors in the production of consciousness. Some researchers argue that embodiment and situatedness are only possible in real physical agents [4]. However, we support that software agents, like video game bots, can be also embodied and situated (this position is also accepted for other consciousness-based agents, see for instance [5]).

Embodiment means to have a body equipped with sensors and actuators which enable structural coupling with the surroundings of the agent. Therefore, a video game agent is embodied in that sense (or 'virtually embodied' [6]), as it retrieves exteroceptive and proprioceptive sensing data through its software sensors, and can also perform actions using its software actuators (see Fig. 1). Situatedness refers to the causal interaction with the world. In the case of a video game character or bot, interaction with the simulated world affects the game environment, which in turn affects the agent. Furthermore, as human players can interact with the bot, even real world is causally affected by bot decisions (analogously, bot sensory input is causally affected by the actions performed in the real world by the human player).

In this work we focus specifically on multiplayer online games where artificial characters can be used. Examples of this type of video games are First Person Shooters (FPS), Role-Playing Games (RPG), and Massively Multiplayer Online Games (MMO). Common features of all these games are a virtual world where the games takes place, local and remote access through computer networks, and the possibility to either play with or against other humans or bots.

Designing intelligent control systems for video game bots is essentially equivalent to the task of designing intelligent control systems for autonomous robots. However, there exist some important differences:

- Robots, unlike bots, possess a physical body.
- Robots, unlike bots, interact directly with the physical world.
- Robots, unlike bots, usually have to deal with more noise and uncertainty.

We see these features as benefits rather than problems for the application of consciousness-based control architectures. In fact, the characteristics of computer games and other virtual environments are ideal for focusing on high level control, avoiding the usual problems of physical machinery. Examples of this approach in the field of Artificial General Intelligence (AGI) are [6, 7].

B. Cognitive architectures and video games

We believe that one of the main reasons why current AI bots are not able to perform as humans is because their control systems do not possess sufficient 'cognitive power'. That is to say, they are not able to integrate the whole set of cognitive skills that are usually identified in healthy human adults (see [8] for a comprehensive list of cognitive skills and a discussion about how could they be considered as functional advantages selected by evolution).

In general, what is missing is an effective combination of cognitive capabilities like attention, learning, depiction [9], set shifting, Theory of Mind (ToM, or the ability to attribute mental states to others [10]), planning, feelings (in the sense of higher order representations of emotions [11]), etc. While limited computational models of these skills could be individually implemented using existing AI techniques, the integration of all these functional components calls for the design of a flexible cognitive architecture. Before going into more detail about cognitive architectures, it is important to analyze what is the typical software architecture in video game bots and where exactly the cognitive skills mentioned above should be located.

Most of the implementations of AI bots are based on different techniques like finite state machines, fuzzy logic, artificial neural networks, decision trees, evolutionary algorithms, Bayesian programming, etc. Also, software architectures are used as a means to integrate several major components: character animation, low-level movement, teamwork coordination, combat skills, and action selection [12].

In this work, we focus on the high level control of bots, setting aside the aspects related with low-level control issues.

Examples of the latter are character's body animation, inverse kinematics, and execution of basic actions (like jumping or firing).

Usually, a higher layer, where the goals of the bot are defined, is implemented on top of other control subsystems, producing the final behavior of the bot (see Fig. 1). The generic higher control layer represents the top level of control. Is in this *Behavior Controller* where the final behavior of the bot is determined. While the *Movement Module* determines how the bot can move effectively from
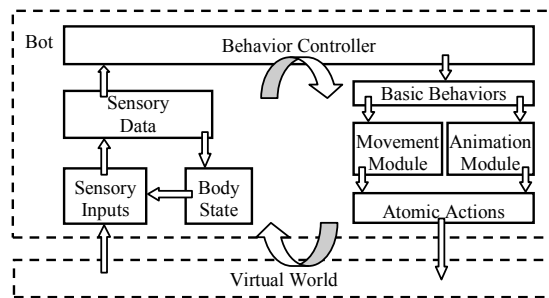

Fig. 1. Video game bot generic control architecture. The higher level of control (Behavior Controller) can enclose a subsumed cognitive control architecture that takes the processed sensory data as input and continuously generates a selection of basic behaviors.

one point to another, the Behavior Controller is in charge of deciding where to move. As a general rule, higher levels send abstract commands to lower levels, which are responsible for the concrete sequence of actions.

As bots are designed to achieve mission specific goals, a repertory of preprogrammed behaviors can be defined (see *Basic Behaviors* module in Fig. 1). In this case, it is the responsibility of higher control levels (the Behavior Controller in this case) to decide what basic behavior to activate at any given time. A number of approaches have been proposed about the design of Behavior Controllers. When bio-inspired approaches are adopted, this higher control layer is usually referred to as the cognitive control layer.

In the context of this work, when we refer to the cognitive architecture that controls the bot, we mean a system embedded in the general control architecture acting as Behavior Controller. Therefore, two levels of architectures can be distinguished: the general bot control architecture (as sketched in Fig. 1), and the cognitive control architecture (the main topic of the remainder of this paper), which is enclosed in the higher layer of the former.

State of the art multiplayer video games provide a rich environment where both humans and artificial bots can interact. In addition, unlike real world tasks using physical agents, video games provide an environment where physical sensorimotor noise and uncertainty do not represent an extra challenge. Consequently, these games have become a suitable and convenient platform for the research in artificial cognitive systems. Although the issue of reproducing human-like behavior in synthetic characters has been approached by many authors (see [13] for instance), most of the research efforts neglect consciousness as a possible source of inspiration for new developments. Nevertheless, complex cognitive architectures like Soar [14] have been integrated in video game bots control systems [15].

We believe the reason why consciousness has been largely ignored is the poor understanding of this phenomenon within the field of classical AI (in contrast to Strong AI or Artificial General Intelligence). The growing interest in the scientific study of consciousness during the

last two decades has motivated new research lines in artificial cognitive systems. Specifically, the emergence of the MC (or equally Artificial Consciousness) research community and associated computational models and implementations (see [16-18] for instance) calls for the application of these new models to the domain of computer games.

C. Machine Consciousness

MC is a young and multidisciplinary field of research concerned with the replication of consciousness in machines. This is indeed a vast area of research, where different subareas can be identified: design of machines showing conscious-like behaviors, implementation of cognitive capabilities associated with consciousness, design of human-consciousness inspired architectures, and creation of phenomenally conscious machines [19]. In this work we will focus on the first subarea: reproducing conscious-like behaviors. Delimiting the specific scope of this research line needs some clarification about the other related subareas.

The phenomenal aspect of consciousness – the subjective conscious experience, or the '*what is it like*' to be conscious [20] – is the most controversial issue in consciousness studies. In order to avoid this dimension of consciousness for the moment, it is useful to conceptually distinguish between two main aspects or dimensions of consciousness: phenomenal consciousness (P-Consciousness) and access consciousness (A-Consciousness) [21]. While P-Consciousness refers to subjective experience and qualia [22], A-Consciousness refers to the accessibility of mental contents for reasoning, volition, and verbal report. Whether or not P-Consciousness plays a functional role in humans is a controversial issue (and could be thoughtfully discussed elsewhere, see for instance [23-25] for different arguments about this issue). Nevertheless, in the context of this work we will focus exclusively on A-Consciousness, i.e. well-known functional features of consciousness and how they can be integrated in a cognitive architecture.

Observing how the human brain works from a computational perspective, we could identify a huge number of neuronal parallel processes which are not accessible to our conscious mind. Only a small selection of contents gain access to a unique sequential thread that we identify as the 'conscious thread of mind'. Determining how this access can be effectively managed and relevant content selected is one of the most important requirements in the design of consciousness-based cognitive architectures.

Main theories of consciousness attempt to provide an explanation for the mechanisms involved in the access to conscious contents. In order to explore the application of these mechanisms to artificial agents we have developed a cognitive architecture called CERA-CRANIUM [26, 27]. Current implementation of CERA-CRANIUM is based on the Global Workspace Theory (GWT) [28] and the Multiple Draft Model (MDM) [29].

In the remainder of this paper, we introduce the proposed computational model of consciousness (section II). Then, we describe a novel consciousness-based cognitive architecture (section III). Specifically, the hypothesis of creating conscious-like behavior in synthetic characters is discussed in detail, and illustrated with the application of the CERA-CRANIUM architecture to the development of a bot for the FPS game Unreal Tournament 2004 (UT2004) [30] (section IV). Finally, some conclusions are discussed and future research work is outlined (section V).

II. A COMPUTATIONAL MODEL OF CONSCIOUSNESS

A review of the main scientific theories of consciousness is out the scope of this paper (see [31] for such a review). As mentioned above, the proposed architecture is mainly based on the GWT and MDM. These theories provide rather metaphorical than technical descriptions of the main conscious processes in humans. Therefore, different computational models could be inspired by their principles. CERA-CRANIUM is one example; see [32, 33] for other AC implementations also based on the GWT but oriented to other problem domains.

According to the GWT [28], a large number of unconscious specialized processors provide data to a global workspace in which coherent information patterns are selected. Baars describes this workspace as a 'theater', where the spotlight represents the focus of conscious attention. In this model, all possible conscious contents are located in the scene (or working memory). The information obtained under the spotlight is distributed globally throughout the theater to all processors ('audience'). This is the mechanism by which conscious experience activates unconscious contexts. Behind scenes, contextual processors shape the events taking place in the spotlight and help to interpret future conscious events. Recent evidence obtained using brain imaging techniques seems to support that real neuronal processes associated with consciousness follow similar mechanisms [34].

The MDM provides a somewhat comparable metaphor [29]. Dennett describes an editorial review process, where coalitions of unconscious processors represent content drafts. These drafts evolve thanks to reiterative edition and review until a final version wins the competition and is 'published' as the official version, i.e. as the conscious content of the mind. Like in a newsroom, the information is continuously being rewritten and edited, modifying existing drafts to adapt them to the new data currently being acquired. The subject is not aware of any of these processes; he or she has access exclusively to a sequential, and already interpreted, version of what is happening ('the *movie* being projected inside your head' [35]). Most of the interim draft versions are deleted and only the winning narrative flow is fixed in memory. Additionally, contents in memory are also subject of constant revision.

These two views of consciousness have a common denominator that can be taken as a design guideline for a partial computational model of consciousness. Cognitive theories like the two described above, amongst others, share

the assumption that unity of self has its roots in non-unitary mechanisms. Specifically, it is argued that conscious contents emerge as a result of competition and collaboration between specialized processors (see for instance Minsky's society of mind [36] for another example along the lines of the two mentioned above).

Following the principle of competition and collaboration between specialized processors, Fig. 2 illustrates the basics of the framework that we have used to implement a computational model of A-Consciousness. Essentially, we have adopted the concept of blackboard [37], adapting and integrating the system into a control architecture able to support higher level cognitive functions associated with consciousness.

This generic computational model provides the underlying mechanisms for developing flexible perception processes as sensory data can be combined depending on active contexts. Our proposal of integration of this global workspace mechanism with a layered control architecture is described in the next section.

III. CERA-CRANIUM ARCHITECTURE

The CERA-CRANIUM cognitive architecture was originally designed as a test bed for computational models of consciousness and attention applied to robotics [26, 27]. Nevertheless, the original design was also intended to be easily adapted to different autonomous agent scenarios, including multiplayer games. In order to explore the application of MC models into the domain of video games we have developed an adapted version of the architecture which is able to control bots in the game Unreal Tournament 2004 by Epic Games, Inc. [30].

CERA-CRANIUM is implemented using two main software components: CERA is a control architecture structured in layers, and CRANIUM is a tool for the creation
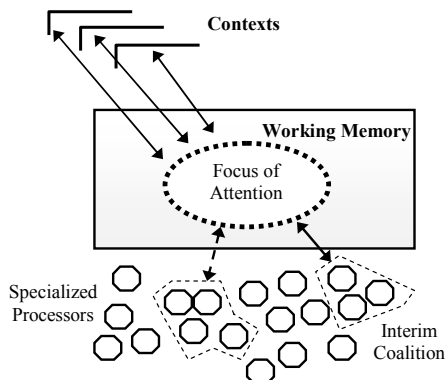

Fig. 2. Computational model of A-Consciousness. Interim coalitions of specialized processors compete for access to the focus of conscious attention. The application of contexts modulates the activation of processors, thus shaping the contents being formed in the working memory.

and management of high amounts of parallel processes in shared workspaces. CERA uses the services provided by CRANIUM for generating highly dynamic and adaptable perception processes orchestrated by a cognitive model of consciousness.

A. CERA

CERA is structured in three main layers and a software interface for sensors and actuators (see Fig. 3). CERA physical layer manages low-level representations of agent's sensors and actuators. Additionally, the physical layer performs sensory data preparation and preprocessing when required. Contextualization parameters like relative positions and timestamps are also calculated in the physical layer. CERA mission-specific layer produces elaborated sensorimotor content related to agent's missions. This process generally involves combining physical layer single contents into multimodal complex contents. Finally, the CERA core layer, the highest control level, implements a configurable cognitive model of consciousness based on a set of modules that perform cognitive functions.

The CERA sensory-motor services layer is an additional agent-dependent software interface which provides access to sensors and actuators.

B. CRANIUM

CRANIUM is a software component which provides services for the execution of thousands of asynchronous but coordinated concurrent processes. This software module is based on the Concurrency and Coordination Runtime (CCR) for the management of asynchronous input-output flow between processes [38]. Distributed execution across multiple computers is also possible thanks to the use of the Decentralized Software Services (DSS) library [39].

The CRANIUM Workspace is the system that provides a shared environment for execution (working memory). These workspaces can be seen as implementations of *pandemoniums* [29], where *demons* compete with each other for activation. Each of this specialized processor or demon is designed to perform a specific function taking specific types of data as input. CRANIUM also provides a mechanism to update the activation level of the processors. This value is usually calculated using contexts (see below), and as a heuristic measure of how much the output of a processor can contribute to the global solution.

C. CERA-CRANIUM perception flow

The mechanism for creating dynamic coalition of processors as explained in the former section (Fig. 2) is implemented in CERA-CRANIUM using the concepts of *single percepts*, *complex percepts*, and *mission percepts* [27]. Rather than having the processors themselves to be associated together, the architecture is based on the reiterated and incremental composition of sensory contents.
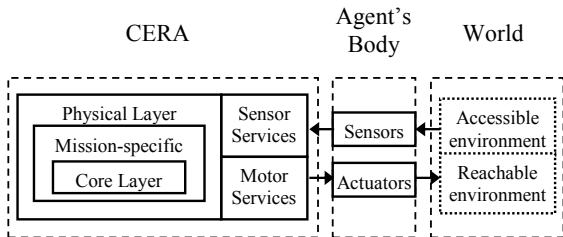
Fig. 3. CERA Cognitive Architecture. The layered design involves a subsumption relation: higher layers process more abstract content than lower layers. Sensor and motor services are enclosed in the so-called sensoy-motor services layer, which is agent dependent.
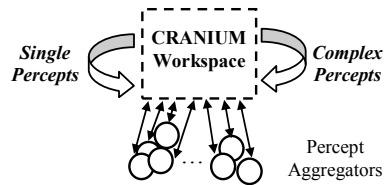

Fig. 4. CERA physical layer workspace. Percept aggregators are special cases of specialized processors that are able to build new complex percepts using other complex percepts and/or single percepts as input. See [27] for a description of other types of processors.

Single percepts are the simplest form of perceptual content. They are generated in the physical layer using sensor preprocessors that combine sensory data retrieved from the sensory-motor services layer (see Fig. 3) with additional contextual information. Basic contextual information present in all single percepts includes relative (or subjective) location of the source of the sensory data and specific time of data acquisition. Generally, all single percepts are monomodal.

As soon as single percepts are created they are submitted to a CRANIUM workspace in the CERA physical layer where they become globally available for all existing specialized processors (see Fig. 4). The processors in the physical layer are able to combine related single percepts to form multimodal complex percepts, which are in turn submitted again to the same workspace (some processors are able to use not only single percepts but also complex percepts to build new more elaborated complex percepts).

Complex percepts are also sent to a second CRANIUM workspace located in the CERA mission-specific layer, where an analogous process takes place. In this case, mission-specific processors receive complex percepts as input and generate mission percepts.

While single and complex percepts are representations tightly grounded but loosely meaningful in terms of the problem-domain description level, mission percepts contain high-level representations directly related to the goals of the agent. In other words, they can be used as significant contents for explicit reasoning (see next section for examples of this in the domain of FPS video games).

Although mission percepts could have been generated directly using just one workspace, CERA layered design calls for the decoupling of physical related and domain-specific representations. This approach permits us to change the physical layer according to the particular needs of the agent being controlled, without affecting the code of the mission-specific layer. Analogously, different mission-specific layers can be used with the same physical layer, i.e. same agent.

D. Workspace modulation

As explained above, workspaces allocated in CERA physical and mission-specific layers play the role of working memory. However, they are not intended to be passive temporal storing mechanisms, but active modulators of the production of higher level representations (complex and mission percepts). As depicted in Fig. 2, the focus of attention should be selected in terms of most active processors. In the case of CERA-CRANIUM, this means that a contextual bias has to be induced in the system in order to determine the level of activation of each processor, hence determining the specific content that is generated in the form of percepts.

One of the main tasks of the CERA core layer is to generate the modulation signal that affects the workspaces in physical and mission-specific layers. This bias-inducing mechanism closes the highest level feedback loop existing between the CERA core layer and the rest of the architecture (see Fig. 5). The core layer input (highest level perception) is shaped by its own output (workspace modulation signal).

Context commands are the most important type of workspace modulation commands. These context commands are able to specify monomodal and multimodal contexts (see [27, 40] for details). For example, a context command might refer to a particular relative location, thus causing percepts originated closer to that location to be more likely selected in the workspaces. This also means that any decision making performed in the core layer will be based mainly on information obtained from these percepts.

One of the main tasks of the CERA core layer is to continuously calculate a contextual region of interest within the perceived sensory space. The input for this process is the set of mission percepts received from the lower layer. These percepts represent the explicit (or functionally 'conscious') content of the agent. From the point of view of MDM, the core layer selects the winning version to be 'published'.

CERA layered design decouples higher cognitive abilities from domain-specific perception. For that reason cognitive skills like attention and set shifting are actually implemented in the core layer. Other cognitive abilities, like several forms of learning and ToM, are currently under development.
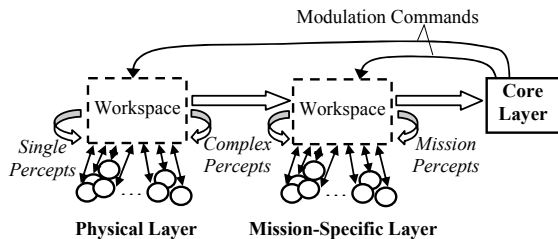
Fig. 5. Workspace modulation mechanism. The level of activation of specialized processors and the generated percepts is updated based on modulation commands sent from the core layer. Activation values determine which percepts are generated, which in turn affects the modulation commands triggered by the core layer.
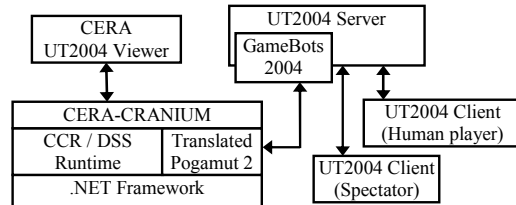

Fig. 6. Experimentation environment. The architecture sketched out corresponds to one CERA-CRANIUM controlled bot, one human spectator, and one human player. All depicted components can be distributed and interconnected through a computer network. The UT2004 server hosts the game and a number of clients can join the game. The clients used for participating or watching the game are copies of the UT2003 commercial game. Additionally, a small user interface (CERA UT2004 viewer) can be used to inspect some variables of the running cognitive architecture. Note that CERA-CRANIUM plays the role of the Behavior Controller described in Fig.1

IV. TOWARDS CONSCIOUS-LIKE UT2004 BOTS

We have developed an adapted version of the CERA-CRANIUM cognitive architecture with the aim to explore the application of MC implementations to the domain of multiplayer video games. We have used GameBots [41] and the Pogamut 2 platform [42] to interface with the UT2004 game server (as CERA-CRANIUM is implemented in .NET Framework and Pogamut 2 is written in the JAVA programming language, we have used IKVM [43] to generate a fully working .NET version of the Pogamut 2 Core library). Fig. 6 depicts the experimentation environment in which the tests have been performed.

The game server is configured to host a deathmatch game in which two or more bots can compete against each other. The goal of the deathmatch play mode is to kill as many other players as possible during the game.

A. CERA-CRANIUM Instantiation for UT2004

The following components had to be adapted / developed in order to build a working version of our cognitive architecture in the experimentation environment described above (all these components are enclosed within the CERA-CRANIUM box drawn in Fig. 6):

- CERA sensory-motor services layer.
- CERA physical layer specialized processors.
- CERA mission-specific layer specialized processors.
- CERA core layer goals and model state.

As we have decided to use the agent representation provided by Pogamut 2, a complete new version of the CERA sensory-motor services layer had to be written in order to deal with available sensory and motor primitives (see Fig. 3). This permits us to seamlessly connect Pogamut's high level primitives with the CERA physical layer. In other words, using the CERA-CRANIUM nomenclature, some Pogamut primitives are translated into CERA physical layer sensor readings, which in turn are used to generate single percepts. The sensory-motor services layer also manages the gap between Pogamut and CERA-CRANIUM semantics. For instance, some Pogamut sensor primitives are periodically invoked in order to query agent's

state, and only significant changes result in the generation of the corresponding CERA physical sensor readings. Notifications, like player death messages obtained from Pogamut listeners, are also converted into single percepts.

Once the sensor readings are sent to the physical layer, sensor preprocessors are the modules in charge of combining sensor readings with timestamps and relative location vectors in order to build single percepts. Timestamps are calculated based on CERA internal timer and relative locations are estimated using the so-called *J-indexes*. A *J-Index* is a representation based on geometrical vectors which contains data about the shape and position of a certain sensory stimulus [27]. A three-dimensional space has been considered in the current implementation of the CERA-CRANIUM bot (location parameters from Pogamut primitives are also propagated from the sensory-motor services layer with the aim to calculate the *J-indexes* in the physical layer).

Additionally, when sensory primitives from Pogamut are converted into single percepts, a differentiation is made between exteroceptive and proprioceptive sensing. For instance, a call to *memory.getAgentHealth()* in the CERA sensory-motor services layer will result in a proprioceptive single percept indicating the corresponding state of the agent. This helps in maintaining an updated model of self-state in the mission-specific layer.

Typical single percepts being generated include agent location percepts, weapons seen percepts, ammos seen percepts, etc. Thanks to the usage of the Pogamut library, the information represented in single percepts has a high level meaning; therefore, the processing needed in the generation of complex percepts is much lower than the required in physical robots scenarios, where sensor data preprocessing, fusion, and classification is required prior to any meaningful labeling [44].

As single percepts are generated and become available in the physical layer workspace, they can be combined using temporal and localization contexts. The main objective of

this process is to generate meaningful complex percepts that will be sent to the mission-specific layer. The state maintained in the mission-specific workspace is the representation used for the implementation of cognitive functions. In this sense, the mission-specific layer workspace could be considered as the explicit working memory of the cognitive bot. Is in this scenario where the different versions of the world perceived by the agent will compete for activation, and winning mission percepts will make their way up to the CERA core layer in order to participate in 'conscious' volition, i.e. be taken into account in the rule-based systems.

In the current implementation of the CERA-CRANIUM bot the following type of mission percepts are considered (other types of mission percepts and their associated generation process are under development): *Enemy Approaching*, *Enemy Running Away*, *Enemy Attacking*, *Myself in Trouble*, *Enemy Destroyed*, etc. These partial percepts of the world can be seen as possible options of what is really happening out there. The described contextualization mechanism (using percept positions and time as main criteria), is applied to the workspace, thus providing the required conditions to periodically build and select these percepts.

Like specialized areas of the human brain that are able to combine several afferent signals to determine the presence of certain conditions, CERA mission-specific specialized processors combine the single and complex percepts obtained from the game virtual world. For instance, the presence in the mission-specific workspace of the following single and complex percepts: *Being Damaged*, *See Enemy*, and *Enemy Approaching*, might trigger the activation of the *Enemy Attacking Detector* specialized processor. This will likely cause the generation of an *Enemy Attacking* complex percept.

The generation of a responsive and adaptive behavior is managed by the application of contexts and an emotions-inspired rule-based system [26]. The CERA core layer will issue context commands directed towards the regions (*J-indexes*) where the winning complex percepts are relatively located. Most basic responsive behaviors include turning to center the action in front of the agent when the associated emotion is neutral, coming close to events (mission percepts) evaluated as positive, and increasing distance to events evaluated as negative.

The 'emotional' evaluation of mission percepts is based on the set of goals established in the CERA core layer. In general, any percept indicating that progress is made to achieve a goal is considered positive. If a percept is considered as a drawback (like being attacked), it is assigned a negative 'emotional' value.

B. Evaluation

In order to assess the believability of the bot controlled by the CERA-CRANIUM architecture we have confronted it with a number of different UT2004 bots (see Table I).

Tests include deathmatch games against human players,

PLAYER	AVERAGE SCORE
Rule-Based System Bot [45]	19.2
Q-Learning Bot[45]	5.2
Q-Learning and Expert Systems Hybrid Bot [45]	11.3
CERA-CRANIUM Bot	18.3

The Average score column refers to the average number of kills during a sequence of ten deathmatch games in which all bots play against each other. Two preliminary deathmach games are used as learning session.

rule-based bots without learning capabilities, bots implementing Q-learning variants [45], and other bots like Amis, the bot that won the 2K BotPrize 2008 competition [46]. Nevertheless, assessing believability is a complex and fairly subjective matter. Testing learning capabilities or other performance indicators would be much easier. In that case, objective measures, like number of frags or deathmatch score, could be used (see Table I). In general, estimating to what extent human-like behavior is being achieved by an artificial agent is a challenge. Some interesting approaches have been proposed, but they are not of practical application in this domain [25, 47].

In order to estimate the presence of conscious-like behavior, we believe the best approach is to use adapted Turing tests. Along this line, the most suitable testing environment will be the forthcoming 2K BotPrize 2009 competition [3] in which the CERA-CRANIUM bot is expected to participate. The evaluation method we are following in preparation for the competition includes the usage of the mentioned objective measures with the aim to shape an average performing bot. Bots behaving too efficiently or too bad would unlikely be considered human-like (they are also less engaging). Tests being performed with the current implementation show that our bot makes poor decisions in many situations, most of the time due to the lack of complex learning abilities.

V. CONCLUSIONS AND FUTURE WORK

This research work illustrates the application of a consciousness-inspired cognitive architecture to the domain of video game bots. Although the presented CERA-CRANIUM bot is still under development and lacks some important cognitive features, like advanced learning algorithms and long term memory, we argue that current implementation fulfills the basics for producing conscious-like behavior. The proposed mechanism is able to generate a number of 'unconscious' versions (or *drafts* in the sense of MDM) of the perceived world and to select a winning 'conscious' version, which is adapted over time. The application of continuously updated context criteria helps stabilizing the bot's current view of 'what is happening' around him, thus producing a coherent stream of A-Consciousness sensory content. The result of the described perception system in combination with the rule-based system of the core layer is a coherent behavior that also helps to create an appearance of agency and unity.

Whether the framework being put forward in this paper is powerful enough to fool the judges of BotPrize adapted Turing test remains to be seen. Nevertheless, useful feedback for the design of more believable bots is undoubtedly being obtained from this work. The enhancement of the existing CERA-CRANIUM bot with episodic memory, different learning algorithms, and improved ToM modeling is expected to produce more human-like behaviors.

REFERENCES

[1] A. Turing, "Computing Machinery and Intelligence," *Mind,* 1950.

[2] D. Livingstone. (2006), Turing's test and believable AI in games. *Comput.Entertain. 4(1),* pp. 6.

[3] "The 2K BotPrize. Can computers play like people?" 2009. Available: http://botprize.org/.

[4] E. Prem. (1997), Epistemological aspects of embodied artificial intelligence. *Cybern. Syst. 28(6),* pp. 3.

[5] S. Franklin, "A consciousness based architecture for a functioning mind," in *Visions of Mind* D. Davis and P. A. Hershey, Eds. IDEA Group, Inc., 2005.

[6] B. Goertzel. Achieving advanced machine consciousness through integrative, virtually embodied artificial general intelligence. Proceedings of the Nokia Workshop on Machine Consciousness. 2008, pp. 19-21.

[7] B. Goertzel, C. Pennachin, N. Geisweiller, M. Looks, A. Senna, W. Silva, A. Heljakka and C. Lopes. An integrative methodology for teaching embodied non-linguistic agents, applied to virtual animals in second life. Proceedings of the First Artificial General Intelligence Conference, 2008.

[8] R. Arrabales, A. Ledezma and A. Sanchis. Establishing a roadmap and metrics for conscious machines development. Proceedings of the 8th IEEE International Conference on Cognitive Informatics, *to be published*.

[9] I. Aleksander and B. Dunmall, "Axioms and Tests for the Presence of Minimal Consciousness in Agents," *Journal of Consciousness Studies,* vol. 10, 2003.

[10] L. S. Vygotsky. (1980, November). *Mind in Society: The Development of Higher Psychological Processes.*

[11] A. R. Damasio, *The Feeling of what Happens: Body and Emotion in the Making of Consciousness.* London: Heinemann, 1999.

[12] P. Tozour, "First-person shooter AI architecture," in *AI Game Programming Wisdom* S. Rabin, Ed. Charles River Media, 2002, pp. 387-396.

[13] C. Bauckhage, B. Gorman, C. Thurau and M. Humphrys. 2007, Learning human behavior from analyzing activities in virtual environments. *MMI-Interaktiv 12,* pp. 3-17.

[14] A. Laird, J. E. Newell and P. S. Rosenbloom, "SOAR: Architecture for General Intelligence," *Artif. Intell.,* vol. 33, pp. 1-64, 1987.

[15] A. Lemke and L. Zilmer-Perdersen, "Virtual Evacuation Training using Interactive Cognitive Agents," *Master's Thesis. Technical University of Denmark,* 2008.

[16] O. Holland, *Machine Consciousness.* UK: Imprint Academic, 2003.

[17] A. Chella, R. Manzotti, *Artificial Consciousness.* UK: Imprint Academic, 2007.

[18] R. Arrabales and A. Sanchis (2008, 6/1). Applying machine consciousness models in autonomous situated agents. *Pattern Recognition Letters, 29(8),* pp. 1033-1038.

[19] D. Gamez. (2008, 9). Progress in machine consciousness. *Conscious. Cogn. 17(3),* pp. 887-910.

[20] T. Nagel, "What Is It Like To Be a Bat?" *The Philosophical Review,* vol. 83, pp. 435-450, 1974.

[21] N. Block, "On a Confusion about a Function of Consciousness," *Behav. Brain Sci.,* pp. 227-287, 1995.

[22] J. Levine, "Materialism and Qualia: The Explanatory Gap." *Pacific Philosophical Quarterly,* 1983.

[23] R. Arrabales, A. Ledezma and A. Sanchis, "Defining a practical approach to artificial qualia," in Knowledge, Understanding and Consciousness, Workshop at KIMAS 2009, *to be published*.

[24] D. C. Dennett. (1988), Quining qualia. *Consciousness in Modern Science.* Available: http://cogprints.org/254/

[25] A. Sloman, "Why some machines may need qualia and how they can have them: Including a demanding new turing test for robot philosophers." in *AAAI Fall Symposium on AI and Consciousness: Theoretical Foundations and Current Approaches,* 2007, pp. 9-16.

[26] R. Arrabales, A. Ledezma and A. Sanchis, "Modeling consciousness for autonomous robot exploration," in *2nd International Work-Conference on the Interplay between Natural and Artificial Computation, IWINAC 2007*. Lecture Notes in Computer Science Series, Vol. 4527. pp. 51-60. *2007.*

[27] R. Arrabales, A. Ledezma and A. Sanchis, "CERA-CRANIUM: A test bed for machine consciousness research," in *International Workshop on Machine Consciousness* 2009, Towards a Science of Consciousness 2009. Hong Kong, June 2009.

[28] B. J. Baars, *A Cognitive Theory of Consciousness.* Cambridge: Cambridge University Press, 1988.

[29] D. C. Dennett, *Consciousness Explained.* Boston: Little, Brown and Co, 1991.

[30] Epic Games, Inc., "Unreal Tournament 2004," 2009. Available: http://www.unrealtournament2003.com/ut2004/index.html.

[31] A. P. Atkinson, M. S. C. Thomas and A. Cleeremans, (2000), "Consciousness: Mapping the theoretical landscape," in *Trends in Cognitive Science,* 4(10), pp. 372-382.

[32] S. Franklin, A. Kelemen and L. McCauley, "IDA: A Cognitive Agent Architecture," *IEEE Conference on Systems, Man and Cybernetics,* vol. 3, pp. 2646-2651, 1998.

[33] M. Shanaham, "Consciousness, emotion, and imagination. A brain-inspired architecture for cognitive robotics," in *AISB Workshop Next Generation Approaches to Machine Consciousness,* 2005.

[34] B. J. Baars, "The conscious access hypothesis: Origins and recent evidence," *Trends in Cognitive Science,* 6(1), pp. 47-52, 2002.

[35] C. Koch, "The Movie in Your Head," *Scientific American Mind,* October, 2005.

[36] M. Minsky, *Society of Mind.* New York: Simon & Schuster, 1988.

[37] H. P. Nii, "Blackboard Application Systems, Blackboard Systems and a Knowledge Engineering Perspective," *AI Magazine,* vol. 7, pp. 82-107, 1986.

[38] J. Richter, "Concurrent Affairs: Concurrency and Coordination Runtime," *MSDN Magazine,* September 2006.

[39] H. F. Nielsen and G. Chrysanthakopoulos, "Decentralized software services protocol - DSSP," Microsoft Corporation, November, 2006.

[40] R. Arrabales, A. Ledezma and A. Sanchis, "A Cognitive Approach to Multimodal Attention," *Journal of Physical Agents,* vol. 3, pp. 53-64, 2009.

[41] G. A. Kaminka, M. M. Veloso, S. Schaffer, C. Sollitto, R. Adobbati, A. N. Marshall, A. Scholer and S. Tejada. (2002, GameBots: A flexible test bed for multiagent team research. *Commun ACM 45(1),* pp. 43-45.

[42] R. Kadlec, J. Gemrot, O. Burkert, M. Bída, J. Havlíček and C. Brom, "POGAMUT 2 - A platform for fast development of virtual agents' behavior," in *11th International Conference on Computer Games: AI, Animation, Mobile, Educational & Serious Games,* 2007.

[43] J. Frijters, "IKVM, an implementation of Java for Mono and the .NET Framework." 2009. Available: http://www.ikvm.net/.

[44] R. Arrabales, A. Ledezma and A. Sanchis, "A multimodal attention mechanism for autonomous mobile robotics," Journal of Physical Agents, 3(1), pp. 53-64. 2008.

[45] F. J. González López, "Diseño e Implementación de un Personaje Sintético Inteligente para un Videojuego de Acción en Primera Persona," *Proyecto Fin De Carrera. Departamento De Informática. Universidad Carlos III De Madrid,* 2009.

[46] M. Stolba, J. Gemrot and J. Simlovic, "AMIS. 2008 BotPrize Contest Best Bot, 2008. Available: https://artemis.ms.mff.cuni.cz/pogamut/tiki-index.php?page=Botprize%202008%20winning%20bot.

[47] S. Harnad. (1994), Levels of functional equivalence in reverse bioengineering: The darwinian turing test for artificial life. *Artif. Life,* 1(3), pp. 293-301.