



Artificial Intelligence approaches for the generation and assessment of believable human-like behaviour in virtual characters



Joan Marc Llargues Asensio^a, Juan Peralta^a, Raul Arrabales^b, Manuel Gonzalez Bedia^{c,*}, Paulo Cortez^d, Antonio Lopez Peña^a

^a Universidad Autnoma de Barcelona, Edifici O, Campus UAB 08193 Bellaterra, Barcelona, Spain

^b Conscious Machines Lab, Madrid, Spain

^c Universidad de Zaragoza, Calle Pedro Cerbuna, 12, 50009 Zaragoza, Spain

^d ALGORITMI Research Centre, University of Minho, Guimaraes, Portugal

ARTICLE INFO

Article history:

Available online 23 May 2014

Keywords:

Turing test
Human-like behaviour
Believability
Non-player characters
Cognitive architectures
Genetic algorithm
Artificial neural networks

ABSTRACT

Having artificial agents to autonomously produce human-like behaviour is one of the most ambitious original goals of Artificial Intelligence (AI) and remains an open problem nowadays. The imitation game originally proposed by Turing constitute a very effective method to prove the indistinguishability of an artificial agent. The behaviour of an agent is said to be indistinguishable from that of a human when observers (the so-called judges in the Turing test) cannot tell apart humans and non-human agents. Different environments, testing protocols, scopes and problem domains can be established to develop limited versions or variants of the original Turing test. In this paper we use a specific version of the Turing test, based on the international BotPrize competition, built in a First-Person Shooter video game, where both human players and non-player characters interact in complex virtual environments. Based on our past experience both in the BotPrize competition and other robotics and computer game AI applications we have developed three new more advanced controllers for believable agents: two based on a combination of the CERA–CRANIUM and SOAR cognitive architectures and other based on ADANN, a system for the automatic evolution and adaptation of artificial neural networks. These two new agents have been put to the test jointly with CCBot3, the winner of BotPrize 2010 competition (Arrabales et al., 2012), and have showed a significant improvement in the humanness ratio. Additionally, we have confronted all these bots to both First-person believability assessment (BotPrize original judging protocol) and Third-person believability assessment, demonstrating that the active involvement of the judge has a great impact in the recognition of human-like behaviour.

© 2014 Elsevier Ltd. All rights reserved.

1. Introduction

The design and implementation of believable artificial agents, truly indistinguishable from humans, remains an open problem. This challenge has been typically addressed from two interrelated perspectives within cognitive science. On one hand, psychological models of human cognition try to explain how human behaviour is produced. On the other hand, computational models implemented in artificial agents try to replicate to some extent human-like behaviour. In this work, we focus exclusively in the sensorimotor behavioural dimension, setting aside any concerns

related to the physical appearance of the artificial agents or their verbal report capabilities.

The imitation game proposed by Turing is the paradigmatic test for believability. However, current state of the art in cognitive and computer sciences has not reached the degree of development in which this test could be considered truly achievable. Therefore, a number of different variations of the original Turing test have been proposed, usually limited Turing tests with relaxed constraints and more specific problem domains. In this paper, we focus in a specific limited version of the Turing test designed for virtual characters and based in a First-Person Shooter (FPS) video game.

From the point of view of the scientific research on human cognition, video game characters are an interesting case of artificial agents because they are easy to implement using the video game industry state of the art tools and their virtual environments can become quite complex, simulating a great variety of contexts and

* Corresponding author. Tel.: +34 93581 4090.

E-mail addresses: fireball787b@gmail.com (J.M. Llargues Asensio), jperalta@cvc.ub.es (J. Peralta), raul@conscious-robots.com (R. Arrabales), mgbedia@unizar.es (M.G. Bedia), pcortez@dsi.uminho.pt (P. Cortez), antonio@cvc.uab.es (A.L. Peña).

ambient conditions. Furthermore, interaction with real world and with human players is also seamlessly integrated in real-time, as video games are designed to facilitate the prompt interaction between human players and non-player characters (NPC).

While old game character implementations (for instance, Pac-Man ghosts or Space Invaders alien spacecrafts) were based on really simple pre-programmed and scripted behaviours, modern AAA video games are developed to simulate real complex environments and they require engaging, realistic and believable human-like behaviour for their NPCs. Although scripted behaviours might still be acceptable for some specific scenarios, AAA game consumers expect to find synthetic characters at the same level of behavioural realism and unpredictability as evoked by the visual experience of the game.

Generally, human-like behaviour is difficult to both define and test. In fact, the Turing test paradigm stills apply to this problem because no better alternatives have been found to characterise human behaviour. In the realm of computer games, this elusive characterisation might, in principle, be seen easier to define. For instance, human players usually consider disappointing the behaviour of artificial characters for two main reasons (Nareyek, 2004): they are either too intelligent, rational and accurate to be human, or on the contrary, they are too silly. Therefore, the challenge is to find that blurred medium level that characterises human player behaviour.

From the point of view of cognitive science, human-level intelligence and human-like behaviour can be considered as produced by several interrelated psychological processes, ranging from basic activation processes like primary motivations to complex high level cognitive processes such as set shifting and imitation learning. The current knowledge we have about these processes can be used to inspire the design of artificial cognitive architectures. In this paper, we present three different approaches to this sort of inspiration and put them to the test in an adapted version of the Turing test based in a video game (Hingston, 2009). Additionally, we assess the believability (or “humanness”) of these bots using two different assessing methods: First-person and Third-person judges.

The remainder of this paper is structured as follows. In the next section we discuss the problems of assessing believability and describe the testing protocols we have used in this research. In Section 3 we present the different approaches to the design of believable agents, followed in Section 4 by a description of the implementations that we have developed for the believability experiments. Finally, experimental results are presented in Section 5 and discussed in Section 7.

2. Testing for believability in video games

Testing for human-like behaviour is not straightforward as different observers usually pay attention to different aspects (Arrabales et al., 2012). Therefore the task of judging the believability of a video game character can be approached from the perspective of inter-subjective assessment. In this context there is a key factor to take into account: the possible differences between First-person and Third-person observation. Togelius et al. (2012) argue that believability is better assessed from a Third-person perspective rather than a First-person perspective, i.e. where the assessor is not a participant in the game. As described below, the BotPrize testing protocol forces all human judges to take active part in the game and perform the assessing task as First-person observers. One of the main contributions of this work is to compare the believability results of the same bots both using the First-person perspective of the BotPrize environment and the Third-person perspective using recorded video from the very same

testing sessions. In other words, believability for each bot is assessed using two different methods but using the same game play data.

2.1. The BotPrize testing protocol

The first method that we have used in order to assess the believability of our bots is the international BotPrize competition testing environment (Hingston, 2009). The BotPrize challenge (held yearly since 2008) was originally conceived as a Turing test for First-Person video game bots (NPCs). In the classical Turing test accurate verbal report and conversational skills are the key factors, however in the domain of FPS bots these aspects are neglected, focusing the assessment completely in observed non-verbal sensory-motor skills.

BotPrize environment is based in the video game “Unreal Tournament 2004” by Epic Games, a First-Person Shooter set in a fictional future with futuristic weapons. The objective of the game (deathmatch mode) is to kill as many opponents as possible without being killed by the other players. Both artificial bots and human players connect to the game server by means of a local area network or over the Internet.

Different judging schemes were used in early editions of the BotPrize competition. In this work, we use the latest scheme adopted in 2010 (Hingston et al., 2010). In this scheme a judging gun (the “Link Gun”) is included in the game. All players, humans and NPCs spawn with a Link Gun with infinite ammo. Although the primary and alternate fire modes of the judging gun look and sound the same to all observers, they have completely different meanings and effects: the primary firing mode is meant to issue a vote for a bot (artificial player or NPC) and the alternate firing mode is meant to issue a vote for a human player.

If a human player shoots a bot or another human using the primary firing mode of the Link Gun, then the bot or the other human player obtains a bot vote. Analogously, if a human player shoots a bot or another human using the alternate firing mode of the Link Gun, the other player (bot or human) obtains a human vote. At the end the humanity percentage will be the human votes divided by the total number of votes received.

During our tests we allow the players to judge any other players as many times as desired. Using the judging gun the game play is transformed from a pure deathmatch game into a hybrid game in which both judging and killing/surviving aspects have to be taking into account simultaneously. It is important to remark that all players (humans and artificial) have access to the judging gun. Therefore, the designer of a bot also has to take decisions on how and when the bot will use the Link Gun, as this usage will also be part of the observed behaviour.

Apart from the judging gun, the rest of the weapons function as usual. However, the damage produced by these weapons is reduced by a 60%, thus giving human players more chances to observe the other players before being riddled under enemy fire.

In order to obtain a significant amount of judging data and reduce the bias that a given map would introduce, different testing sessions in different maps are organised using a centralised game server that runs the BotPrize mod. Each session lasts for 15 min and different maps and scenarios are used each time. Anonymity of players is guaranteed using random player names and random player skins (clothes and body appearance) that changes from one session to the next.

The number of human players and bots is balanced, having a similar number of human judges and artificial characters. All human players are meant to be judges, but they also compete for the highest score (that they obtain both from judging and from killing and not being killed).

The BotPrize testing protocol is a First-person observation approach, as judges are not allowed to use the spectator mode of the game, and they are also subject of attacks and votes (Link Gun shoots) from other players. As a Third-person test we have used the method described in the following section.

2.2. The Third-person observer testing protocol

As Third-person believability assessment we use the same game data recorded by the game video server during the sessions of the First-person experimentation. However, we generate independent video footage for each player from a virtual Third-person high-angle “chasing camera”. In other words, for each player in the game (bots and humans) a Third-person camera keeps a high-angle view of the action and the corresponding video is stored.

Video recordings obtained using the former method can be later used for offline observation. In this case, the judging protocol can be based on non-player human observers who watch the videos and try to tell apart humans and bots. Specific video sequences are selected for judging purposes out of the total available video files. This selection is performed based on the content presented in the film. Basically, scenes just containing a single character moving from one place to another are discarded, and we have decided to focus and select those situations in which two or more characters are engaged in some kind of interaction.

This testing method is in principle much more demanding, and humanness ratios are expected to be much lower, as human observers can pay all their attention to the action in the video, not needing to worry about being killed or achieving a high score in the game. While the judge in the First-person approach have to divide attention in two different tasks (play and judge), the judge in the Third-person approach only needs to focus on judging. Similarly, motivational aspects also differ in the two approaches.

3. Different approaches to the design of believable characters

A number of different approaches can be used to address the problem of believable behaviour generation. We can distinguish between two main types of approaches in the design of artificial agent controllers. In one hand, controllers can be built and trained based on data obtained by logs of human behavioural data. On the other hand, controllers can be designed based on models of human cognition. While the former exploit the statistical structure of actions in typical human behaviour, the latter focus on psychological models of the human being. In this work, we take a more cognitive stance, considering the problem of behaviour generation as a S–O–R (Stimulus–Organism–Response) problem (Ertmer and Newby, 1993), as we consider that classical behaviourist S–R models cannot cope with the complexity of believable behaviour. However, S–R models cannot be neglected in specific problem domain scenarios, like computer video game characters, as controllers based in the replay of human trace data have shown good results (Schrum et al., 2011).

Although replaying human trace data can provide good results, this approach is not applicable to environments where no previous human data is available, for instance in new maps of a video game, or in games where the content is generated real time (Yannakakis and Togelius, 2011). S–O–R models are neither free of theoretical and practical problems. In the experimental settings for believability tests stimuli and responses can be well defined and their scope limited. However, the organism component, i.e., the very design of the agent controller, can be addressed in different ways. The problem of modelling O in a S–O–R framework can be considered as the problem of choosing a model of the mind that allows the agent to show perception and action capabilities. This model could be either

implicit (subsymbolic) or explicit (symbolic). In other words, stimuli can be associated to actions using implicit rules (as in artificial neural networks) or using explicit rules (as in cognitive architectures). In the following subsections we discuss the details of these approaches and how they can be combined into hybrid models. Exploring the differences between these approaches is interesting to understand the role of the symbol grounding problem in the domain of agent believability (Anderson, 2003).

3.1. The cognitive approach

The cognitive modelling approach is based on the imitation of the human psychological processes that are considered the cause of the human observable behaviour. From this point of view, in order to generate human-like behaviour in a machine, the AI engineer tries to simulate information processing models inspired by psychological models of human cognition. These models usually offer plausible explanations about how different aspect of cognition work in humans and how they interact with each other and with the environment.

Artificial cognitive systems aim at integrating both activation processes, like emotion and motivation, and several cognitive processes, like attention, memory, learning, and even language production and understanding. Ultimately, cognitive systems are expected to become effective and adaptive control systems for a situated agent. Although a number of popular models exist in the field of cognitive psychology for each cognitive process, the real challenge is to integrate them all into a functional control system able to develop human-level intelligence and human-like behaviour.

Classical AI techniques and models are usually based on very specific and partial explanations of human cognition. For instance, reinforcement learning algorithms focus on the self-adaptation of a system to achieve one or more goals (Szepesvri, 2010), however when we deal with the problem of human-like behaviour we need to take a more global approach, where the actual definition of the goals of the system might be the real challenge.

In this work, we describe two bots, codenamed CCBot and CCBotSOAR, both based on artificial cognitive architectures. As explained below, these control systems deal with the problem of integrating and coordinating multiple cognitive processes in order to generate human-like behaviour.

3.2. The subsymbolic approach

Typically, cognitive systems basically deal with symbolic representations of the world. However, at a lower level, taking the inspiration not from the psychological processes but from their neurobiological substrate, the artificial neural networks (ANN) approach Schrum et al. (2012) focuses on imitating the adaptive and pattern recognition capabilities of the nervous system. The human body uses automatic processes purely based on sensory-motor skills for prototypical reactions that need to be executed promptly without losing any time to deeply process the information. For example, when we touch something hot and we burn our hand, the reaction is a quick withdrawal. If this kind of human behaviour is transported to the game, we can make bots have reactions according to each sensation that comes from the environment. Although the reaction can be pre-programmed, there are so many different combination of input information coming to the individual that taking all them into account and all the possible combinations is too difficult. ANN can be taught to act when sensory (input) information is received, for that task, first it has to be defined in detail which sensory information we want to process (inputs) and which actions the bot can perform (outputs).

3.3. Hybrid approaches

Humans cope both with symbolic and subsymbolic representations. While language and conscious thought are the paradigmatic examples of explicit and symbolic information processing, we can easily reckon that a great part of human behaviour is also generated by implicit and automatic processes purely based on sensory-motor skills. Therefore, we think hybrid approaches that can successfully integrate symbolic and subsymbolic approaches are a promising research line towards the generation of human-like behaviour. However, in this deathmatch-mode video game environment where judging is based on short-term assessments both cognitive approaches and pure ANN approaches are expected to reach similar believability ratios. In more complex Turing test environments, where both verbal report and sensory-motor skills are expected to be integrated, hybrid approaches are required in order to generate believable behaviours.

In the next section we describe the implementations we have designed and built, indicating how we aim at integrating implicit and explicit information processing.

4. NPC controller implementations

In this section we describe the three different NPC controllers that we have designed, built and confronted to each other for experimentation.

4.1. CCBot

The Conscious-Robots bot, also known as CCBot2, has been designed and implemented based on the cognitive architecture CERA–CRANIUM, which takes inspiration from several cognitive theories of cognition and human consciousness (Arrabales et al., 2012). CCBot2 is essentially a control architecture that implements some key aspects from the field of Machine Consciousness (MC) research. Basically, the cognitive approach to MC addresses the integration of different cognitive processes into an integrated thread of conscious experience. As described below, in the case of CCBot2 we focus on the integration of basic cognitive processes such as attention and short-term memory. Additionally, activation processes like motivation and emotion are modelled to a lesser extend in this implementation.

While classical agent control architectures generally focus in next action selection, CERA–CRANIUM also take decisions about what should be the content of agents conscious perception.

There exist a number of cognitive architectures that can be potentially used for the control of video game characters (Samsonovich, 2010) for a review of main cognitive architectures used in robotics, intelligent software agents and other related applications. For instance, SOAR (Laird et al., 1987) and ACT-R (Anderson, 1993) are well-known cognitive architectures that can be used in this domain. In fact, we have also used a combination of CERA–CRANIUM and SOAR in the bot CCBotSOAR (see Section 4.2).

Pursuing the goal of human-like behaviour is directly related with the design of cognitive architectures because human higher cognitive processes themselves are believed to be the origin of the behavioural patterns that characterise humans. In this work we propose and demonstrate the use of new trends within the field of cognitive modelling. In the case of CERA–CRANIUM we use consciousness as the main inspiration towards the design for more human-like cognitive systems.

In this section we describe the particular way in which CCBot processes the sensory-motor information and generates sequences of adaptive human-like actions. We also argue that having a

mechanism to select the conscious contents of the agent's simulated mind is required in order to successfully generate human-like behaviour.

Action selection in CERA–CRANIUM is based on a competitive selection process that establishes, at any given time, a limited set of percepts considered the conscious content of the agent. The architecture CERA–CRANIUM consists of two main components: CERA and CRANIUM. The former is a control architecture of four layers (see Fig. 1) and the latter is a platform able to run and manage a large amount of parallel specialized processors.

CERA's top layers manage more abstract meaning while lower layers deal with raw sensory data from the simulated sensors. In CCBot we have used an implementation with the following layers: sensory-motor services layer, physical layer, mission-specific layer, and core layer.

CRANIUM is an implementation of a blackboard system (Nii, 1994). On one hand, CRANIUM is in charge of managing and dispatching the execution of specialized processors; on the other hand, it has a shared working memory space “the CRANIUM workspace” where the processors can interchange information. Each of the processors is designed to perform a specific function on certain types of data.

There are two instances of CRANIUM located in the physical and mission-specific layers. These two CRANIUM instances simulate the working-memory mechanism of the bot. Contents in working memory are iteratively elaborated by specialized processors (see Fig. 2).

CERA's sensory-motor services layer is an interface between the architecture and the sensors and actuators of the bot. This layer includes the required services to retrieve sensory data readings and to send motor commands to the actuators. In the case of CCBot series, this layer is an adaptation of the Pogamut functions to the CERA–CRANIUM architecture (all the bots described in this paper use Pogamut 3 library functions (Gemrot et al., 2009) to interface with Unreal Tournament 2004 video game).

In the physical layer we manage low-level representations of the sensory data and effectors commands. Typically, this layer includes processors that create more abstract representations out of the sensory information and processors that decompose high-level commands into atomic actions, which can be sent directly to the sensory-motor services layer.

Mission-specific layer produces and manages elaborated sensory-motor content related to bot's assigned missions. The information from the physical layer is combined into more meaningful contents related to the goals of the agent and processed to execute the corresponding mission behaviours.

The top layer is the so-called core layer, where the mechanisms associated with the cognitive functions are regulated. The cognitive control modules enclosed in this layer are in charge of regulating the way the lower layers work, that is, how the processors interact with each other and what it is the most relevant content to be processed at any given time. In particular, CCBot2 included in this layer only an attentional mechanism, that prioritizes those contents closer to the current active sensory-motor context of the bot.

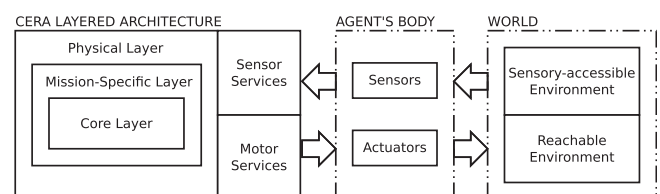


Fig. 1. CERA–CRANIUM Architecture.

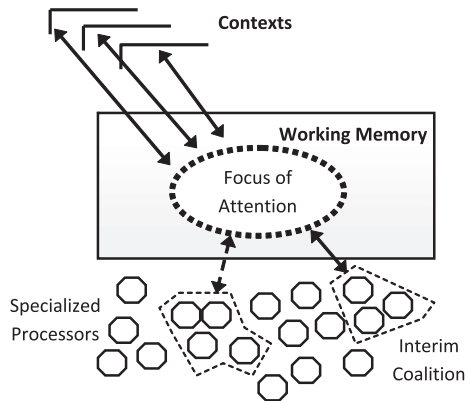


Fig. 2. CRANIUM and working memory mechanism.

CRANIUM includes a mechanism to select which processors are assigned more priority for execution. Although in principle all processors are treated as equals and all of them are supposed to run asynchronously and concurrently, limitations on computational resources have to be taken into account. Consequently, there is an implicit competition for activation amongst all the processors. The level of priority attained by a processor does not only affect its available execution time, but the specific information it might generate and submit to the workspace. This mechanism can be seen as a winner-takes-all algorithm, where the most activated signals are the ones most likely processed. From this point of view, CRANIUM is a particular implementation of a “pandemonium”, where daemons “or specialized processors” compete with each other for activation (Dennett, 1993). The activation of each processor is calculated based on a heuristic estimation of how much it can contribute to information processing in current active sensory-motor context. The concrete parameters used for this estimation are established by the CERA core layer. As a general rule, CRANIUM workspace operation is constantly modulated by commands sent from the CERA core layer.

The two workspaces we use (see Fig. 3) are connected with each other through CERA sensory-motor information flows and share selected contents generated by the processors, specifically those with the highest activation level. So the outcome of the workspaces can be seen as a filter where only the signals that contribute more to the global solution pass from one layer to the other. The workspace in the physical layer contains all the processors in charge of processing all data coming from the sensor services and also all processors which decompose the actions into basic commands, atomic actions, for the effectors. The workspace located in the mission layer is populated with higher-level specialized processors, which process the most activated information generated in the physical layer and the signals produced in the workspace itself. All this processed information generated by the specialized processors are integrated into data packages called single percepts, complex percepts, and mission percepts, depending on the sort of information they contain. The single percepts are those which include atomic information, usually sensory signals, while complex

percepts include more elaborated and multimodal information. Finally, mission percepts contain information related with goals (and they are only produced in the mission layer).

There are two flows of information in CERA-CRANIUM. One is the bottom-up flow, where the information from the sensors, the percepts, are processed and combined in order to obtain abstract representations of the environment and the state of the agent. The bottom-up flow can be referred as to the perception flow. The top-down information flow is concerned with the generation of adaptive behaviours oriented to achieve the agent's goals.

The whole set of specialized processors used in the workspaces cannot be described here due to space constraints. CCBot implementation is described in detail in Arrabales et al. (2012).

A relatively simple task like shooting an enemy might involve the interaction of seven different specialized processors (Arrabales et al., 2012). For instance, there is a “Player Novelty Detector” in the physical layer that detects when a new enemy appears in the field of view and submit that specific information to the working memory. We have also implemented another processor called “Select Enemy To Shoot” that operates in the mission layer and selects an enemy amongst all players within the field of view.

4.2. CCBotSOAR

The CCBotSOAR controller is based on the CERA-CRANIUM architecture which has been improved with two additional mechanisms that allow: (i) the self-assessment of performance in the achievement of targets (“longtime memory”), (ii) the monitoring of the effects on behaviour of the emotional state of the bot (“feelings”), (iii) the adjustment of the pursued objectives to the context (“situatedness”) and (iv) the focus level as a measure of the game progresses. In Vernon et al. (2007) a comparative assessment of the most relevant cognitive architectures is made from which it can be noticed that the only characteristic that the Global Workspace architecture (Baars, 1997) –on which CERA-CRANIUM has its foundations– is missing is *adaptation*. Driven by this fact, we have integrated a new module in CERA-CRANIUM that enables the addition of this attribute. By means of a SOAR module (Laird et al., 1987) integrated in CERA-CRANIUM, the system can deploy a learning and memory based decision-making mechanism based on the bot's experiences that are implemented as “artificial emotions”. In the following paragraphs, we show an explanation of each of the new features of CCBotSOAR.

4.2.1. Long-time memory

SOAR includes a reinforcement learning system that alters the selection of operators with a reward system. Through several cycles, the bot selects and executes actions while calculating in each of the steps which is the best action according to a reinforcement function. In our implementation, the reinforcement function is grounded on the idea of emotional “appraisal” (based on the theories of Arnold (1960) and Schachter and Singer (1962)) and its positive (*action attraction*) and negative variation (*action repulsion*). A more detailed explanation about the emotional model is shown below. Given the emotional appraisal in certain state, the way in which the learning operates is following the learning algorithm SARSA (Laird et al., 1987), which is more appropriate than other more common ones –such as Q-learning algorithm– for very dynamic environments like the one in the Unreal Tournament. The learning mechanism (i.e. selection of operators) is completely provided by the architecture SOAR.

4.2.2. Emotional state

According to the classical theories (Arnold, 1960 and Schachter and Singer, 1962), we could assume the emotions to be an entity

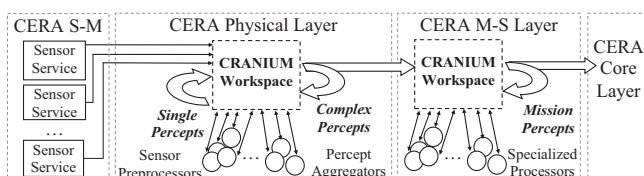


Fig. 3. CERA-CRANIUM bottom-up flow.

product of two aspects: (1) a generic response of the autonomous system (“arraisal”) and (2) the cognitive evolution of this disturbance (“appraisal”). From these ideas, we consider bots as having an emotional module comprising the following components:

- **Internal state:** The agent’s internal state will be composed of motivations connected with the *drives* (“essential needs”) of the bot that will drive it to act in a certain way. In our model we have implemented:
 - *Motivations:* Aggressiveness (tendency to kill enemies), Illness (need for improved health) and Inability (need for ammunition).
 - *Basic drives:* The initial and ideal value of all drives is zero. In this initial setting we will consider drives as satisfied since there is not an associated need. We model two types: health pursuit (high >100, low <40, medium, between both limits) and ammunition search (high >100, low <50, between both limits).
 - *Actions* (aiming to satisfy the drive): collect first-aid kit, kill enemy, collect ammunition, etc.
- **External state:** It is a binary state (yes/no) recording proximity of certain elements of the Unreal Tournament environment that can interact with the bot, more specifically, if the bot finds (i) first-aid kit, (ii) enemies or (ii) ammunition. The global intensity of the drive (attribute around which the bot action choice revolves) is calculated as the sum of two terms: the own intensity of the related drive and the value of the closest related external stimulus (i.e., if the bot is close to a first-aid kit, the drive relating to health will be increased). It is conventionally assumed that while all the drives values are inferior to a given threshold, the bot will not have a dominating motivation, and it will be in a state in which all its needs are satisfied.
- **Bot’s mood and appraisal function:** We define the bot’s mood as the “degree of satisfaction of its needs” in such a way that the mood will reach its maximum when all the bot’s drives are satisfied. The mood is modelled as a function of (1) the agent’s drives D_i and (2) the pondered importance of each drive (applying weights w_i) with respect to a reference ideal mood (M_{id}) which intends to reflect the bot’s personality.

$$Mood = M_{id} - \sum w_i \cdot D_i$$

As the drive’s values increase/decrease, the agent mood changes. If the variation in the mood ($\Delta Mood$) exceeds a threshold, the corresponding “appraisal” will be triggered. The behaviour between two thresholds (denoted by L_h and L_s) is defined as follows: (1) Positive appraisal (If $\Delta Mood > L_h$) and Negative appraisal (If $\Delta Mood < L_s$) that set the start of certain behaviour.

4.2.3. Situatedness

Frequently, videogamers recognise the so-called “hot points” of a multiplayer game, i.e., they will know in which areas of the environment they are more likely to be killed (e.g., because it is a frequent path for many players, because it does not have many places where to hide, because there is a specific condition of the scenery such as quicksand, etc.). As a new modification, a map of the environment is included in CCBotSoar where the hazard of navigation points (NavPoints in the Unreal Tournament maps) close to conflict situations is portrayed. Through a classification in five levels of danger (*null, low, medium, high, fatal*) according to the number of deaths caused, CCBotSoar will have available this “hazard mental map”, analogously to the one players as a consequence of their experience in the game or their terrain knowledge.

4.2.4. Focus level

We have included a new modification in the architecture related to the lost of focus level accordingly to the results in the game. This aspect is related to situations when the enemies seem unbeatable (apathy) or when players think they are (euphoria). In both situations, the bot’s behaviour is programmed to be more erratic and less focus on the game. In order to model this effect, we introduce a weighting factor which is a function of the difference in targets up to the end of the game. This parameter –a focus tempering coefficient– moderates the values of the emotions shown through the game.

4.2.5. Integration of new features developed

In the original behaviour of CERA–CRANIUM, the execution cycle is approximately four cycles per second and the architecture would select an action of each type amongst those proposed by the workspace in the physical layer, which would then be all performed. In the CCBotSOAR architecture, this pattern is modified in such a way that every time a cycle is executed, a SOAR operator is assigned to each of the proposed actions, and only one is selected by means of the rules engine. The global action selected in the figure results from: (1) SOAR’s decision system and its reinforcement learning, and (2) a global factor that gathers elements of emotional character which have been previously described. In order to be clearer, Fig. 4 shows how the information flow is modified by the additional module.

When trying to set the parameters in each of the sections (persistence of long-term memory in the learning process, ideal and threshold mood values, map of situatedness, focus tempering coefficient, etc.) real values obtained from games with expert players have been used.

4.3. ADANN bot

When working with ANN, finding an adequate ANN model is a key issue. Different studies have dealt with the design of an ANN from two different points of view.

- **Topology:** number of hidden layers, hidden nodes in each layer, etc.; and
- **Connection weights:** values for each connection in an ANN.

In this specific domain the topology is given by the problem. After a detailed study of the inputs (current health, armour, damage, weapons, etc.) and outputs (movement, rotate, dodge, jump, crouch, etc.) from the game that should be used, carried us to the final decision of using the same architecture for each ANN. Related to the estimation of the connection weights, it is well known that learning algorithms like backpropagation usually got stuck in a local minimum [Riedmiller et al. \(1993\)](#). Moreover, it was not possible to obtain learning patterns from the game to train the ANN models because of the restrictions derived from working with real time computer games. [Whitley and Hanson \(1989\)](#) proposed the

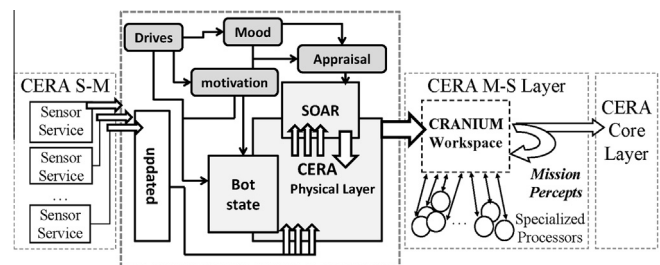


Fig. 4. CCBotSOAR bottom-up flow.

use of evolutionary computation to search for appropriate connection weights and avoiding the local minimum problem by means of a global search Rocha et al. (2007). The process of obtaining a candidate model will be split into two steps. First, several random initialized individuals (ANN) of a first generation are obtained. We make them combat using the input information that comes from the game and let them apply actions (outputs), while we measure how good they are (fitness value). This fitness value is measured as the number of deaths caused by the bot divided by the number of times it has died. Every six minutes we apply genetic operators (i.e., gaussian mutation) to obtain a new generation. The whole process (shown in Fig. 5) is repeated until a maximum number of generations (i.e., 100) is reached.

At the end of the training process we will have a good NPC which will be introduced into the game (test) with other human and non-human players as it will be explained below in detail.

5. Experimental results

In the following we summarise the results we have obtained confronting our bot controllers to both the First-person and Third-person believability assessments.

5.1. Structure of the “user study”

User studies cover a wide range of activities designed to obtain information on the interactions between users and specific products. We next summarise the basic principles of the *user study* that we have used to measure the differences confronting our bot's controllers to both the First-person and Third-person believability assessments. User tests involve analysing carefully the users behaviour as they interact with a product or system in some stages:

Objective:

Our tests try to find out which of the developed controllers is “more human”.

Methodology:

The methods and techniques we use in order to obtain this information were *first-person* (participants votes) and *Third-person* (votes on the recorded videos) interactions.

Design of the tasks and settings:

Details of the First-person and Third-person tasks are provided below:

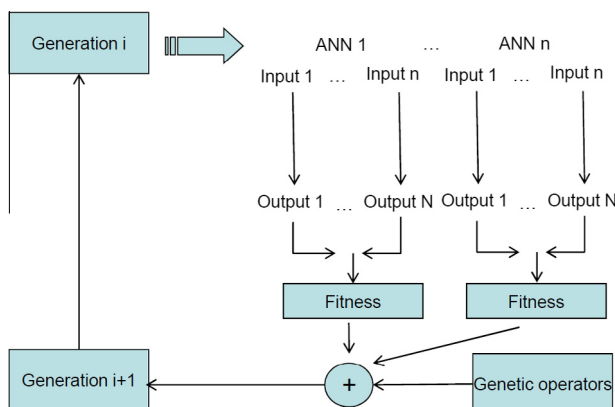


Fig. 5. ADANN schema.

- First-person approach: (i) 20 matches (4 maps of 15 min each) during 5 sessions (5 different days) and (ii) 6 judges (three human players and three NPCs). Different human judges participated in different matches.
- Third-person approach: (i) 10 video clips (1 min each) where all the bots and human players are included (they are not all present in all the videos, a weighted selection has been considered), (ii) the 10 video clips were purposefully selected to comply with the experiment requirements (i.e., must include actions with two or more players interacting) from a total of 35 h of recorded video, and (iii) 12 subjects were selected for the evaluation who had no prior knowledge in videogames and had not participated in the First-person experiment.

A vote is considered positive when they guess correctly and negative when they fail or do not know what to answer.

Test environment:

The BotPrize competition environment <http://human-machine.unizar.es/> and recorded videos of the games.

Results and Analysis:

The differences in believability assessment will be measured using a First-person approach and Third-person approach. These measurements will be expressed in terms of percentage of success vs. votes.

We follow by explaining in detail the stages of the experiment.

5.2. First-person assessment

As described above, we used the BotPrize competition environment and testing protocol as First-person observer method for assessing believability. We ran a total of 20 matches conducted during 5 sessions of 1 h each. Matches last for 15 min and 4 different maps have been used per session, counterbalancing level maps across the sessions. Therefore, human players (judges) were asked to play (and judge) for 1 h (4 maps of 15 min each) in 5 different days, with a period of one week between consecutive sessions. The whole testing procedure took place during 5 weeks and the selected human players were always the same. There were the same number of NPC and human players, three human and three NPC.

Human judges agreed voluntarily to participate in this study. They neither had previous experience in the design or programming of a NPC, nor any expertise in Artificial Intelligence. However, they were selected because they have intermediate experience with FPS video games (none of the judges was a novice or advanced player).

As it can be observed in Table 1, after the five judging sessions (session 1 to 5) of the three different approaches presented in this paper against human players, ADANN system outperforms the other approaches followed very near by CCBotSOAR.

In order to have another measure of believability we have also used a Third-person observer method based on recorded data from the BotPrize sessions described in the former section.

The 20 matches from the First-person assessment experiment were recorded in the central video game server. These deathmatch

Table 1
First-person humanity assessment results (in percentage).

Bots	S1	S2	S3	S4	S5	Average
CCBot	19.59	19.84	16.81	20.14	28.70	21.02
ADANN	17.34	37.04	31.73	30.26	47.23	32.72
CCBotSOAR	21.18	27.06	35.80	38.82	40.61	32.69

game recordings were used to generate 35 h of video that included a Third-person view of each player (20 matches, 15 min per match, 3 bots and 4 human players per match).

A set of 10 video clips with a duration of 1 min each were selected as a representative sample of the whole visual data collected from the video game server. These clips were extracted based on the following criteria: they include a sequence of actions in which two or more players are interacting (most of video footage time corresponds to periods in which the player observed is just moving from one place in the map to another, or not interacting with any other player), all bots and human players are included in the clips, and clips correspond to different maps or different parts of the maps.

This representative sample of 10 video clips were made available to a set of human judges using a mobile application called Knowxel.¹ Knowxel is a multipurpose crowdsourcing platform oriented to mobile devices (see Amato et al., 2013, 2014). This platform was specifically developed for mobile devices in order to exploit the strengths of such devices; namely: (i) massivity, (ii) ubiquity and (iii) embedded sensors. The combined use of mobile platforms and the crowdsourcing model allows to tackle from the simplest to the most complex tasks. Knowxel allows to conduct a video-based poll experiment where the users vote at the end of each video clip. Possible votes were: (a) I think the player is a human, (b) I think the player is a bot, (c) I cannot tell whether it is a human or a bot. Fig. 6 shows an example of the interface of the Android app that we have used in order to present the videos to the Third-person human judges.

In this Third-person assessment process we asked for voluntary participation to university students with no former experience in video game NPC development or Artificial Intelligence. However, all of them had previous experience playing FPS video games, and of course none of them participated in the First-person assessment experiment.

Table 2 represents the difference between the believability assessment using the First-person approach (BotPrize) and Third-person approach (video observation).

Comparing the results of the two different assessment methods we can see that the Third-person assessment is much more demanding. Using the Third-person assessment average humanness ratio for humans and bots are 68,67 and 22,83 respectively (see Fig. 7) (see next section for a detailed discussion).

6. Discussion

As we have shown, three different approaches were designed and compared using two different measurements methods. We can observe that although ANN approach obtains better results in First-person experimentation, in general, cognitive approach CCBotSOAR is the best option. In Fig. 8 we compare the results of the two assessment methods for the three bots. Consistently the First-person method offers a higher humanness ratio for all bots. Applying the Student's T-test for the paired samples corresponding to the two assessment methods we obtain a $P = 0,038 < 0,05$, thus confirming that we cannot consider the First-person and Third-person methods as equivalent.

While the bot solely based in the CERA–CRANIUM architecture lacks any learning or long-term adaptation mechanism, the two bots with better results implement different mechanism of adaptation. We believe the reason why these two bots have similar performance in terms of believability lies in the learning/adaptation mechanisms.

CCBotSOAR implements a reinforcement learning mechanism as described in Section 4.2. (see “long time memory” subsection),

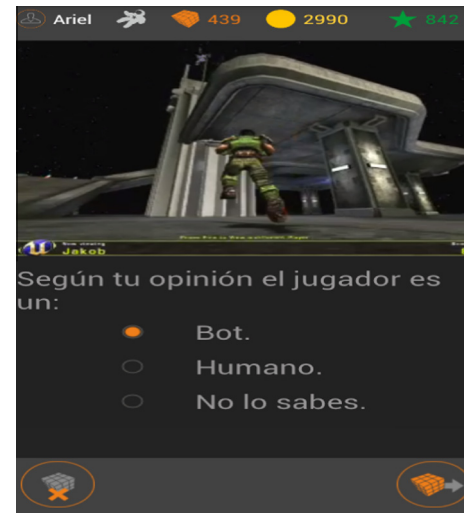


Fig. 6. Snapshot of the Knowxel mobile application used in the Third-person judging process.

which makes it more adaptable to the dynamics of the interaction with other players. Similarly, ADANN also adapts well to the dynamics of the interaction with other players, however, in this case the mechanism used is not an explicit learning algorithm, but a genetic algorithm optimisation. It can be argued that the fitness function in the genetic algorithm in ADANNBot is equivalent in terms of adaptation results to the reinforcement learning algorithm implemented in CCBotSOAR.

The believability assessment methods applied in this work are indeed fully behavioural tests, as they are inspired by the Turing test. Pure behavioural approaches are pragmatic as they do not take into account the internal states or mechanisms of the agents. In other words, Stimulus–Response (S–R) models are the references to understand and evaluate the subjects being tested. When we confront artificial agents to a full Turing test this should not be a problem, as we assume human-like internal processes exist (even though we do not study them directly) and the resulting behaviour is the result of the interaction between the stimuli and these internal psychological processes. However, when we are dealing with limited versions of the Turing test, where some behaviours might well be pre-programmed, we might want to also inspect the inner workings of the agents. By inspecting and analysing the inner processes of the agents we are adopting a Stimulus–Organism–Response (S–O–R) stance. Additionally, by “looking inside” the agents we are making sure no deceiving strategies have been implemented in order to fool the judges.

Human judges are usually influenced by the personification effect (Arrabales et al., 2012) and they tend to assign psychological states, personality and intentions to simple behaviours that may be just simple pre-programmed routines. Although this issue is not a main concern in the present work, as we exclusively focus on the perceived humanness, more comprehensive S–O–R oriented tests might be developed based on Machine Consciousness testing

Table 2
Comparison of First-person and Third-person results.

Bots	3rd Person	1st Person	Average
CCBot	17.39	21.02	19.20
ADANN	25.00	32.72	28.86
CCBotSOAR	26.09	32.69	29.39

¹ www.knowxel.com

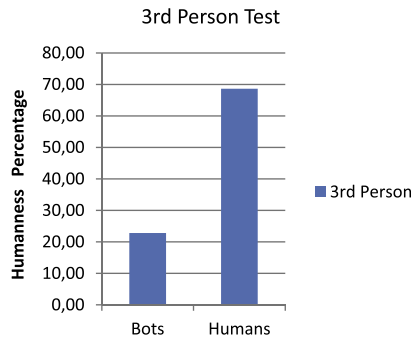


Fig. 7. Third person humanity assessment results (in percentage).

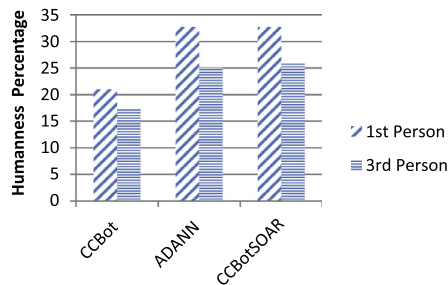


Fig. 8. First Person versus Third Person humanity assessment results (in percentage).

approaches like the *ConsScale* cognitive development measure (Arrabales et al., 2012).

7. Conclusions

As introduced in this paper, the automatic generation of human-like behaviour is an enormous challenge, even when addressed in the constrained domain of a video game without verbal interaction. We have explored the application of different control architectures and also new ways to address the problem of assessment. The results obtained clearly indicate that the Third-person approach to assessment is much more demanding in this context. Therefore, we plan to perform more extensive testing using this approach. As explained in the paper, when we design cognitive architectures we model different cognitive processes and try to integrate them effectively so the resulting synergies translate into a more human-like behaviour. We believe that the results obtained in the experimentation are consistent with the implementation and integration of the cognitive processes we have focused on: attention, memory and the activation processes like emotion.

Although good results have been obtained, for example, by ADANN, still the training and testing process are still separated. Due to this, although a good NPC is obtained during the training process, once this bot is taken from the training sandbox to the testing one, it will not learn more while it is playing. Then, a possible future work would be to make use of non-supervised learning process and ANN so there would be only one sandbox, being it used for training and testing at the same time. On the other hand, every time the bot is taken to a new game, it would be able to keep on learning from new players.

In order to achieve better humanness results we plan to integrate higher level cognitive processes, mainly learning mechanisms such as reinforcement learning. In fact, as we have seen, CCBotSOAR clearly surpasses CCBot thanks to this feature.

Acknowledgements

The research reported here has been in part supported by the project TIN2011-24660, funded by the Spanish Ministry of Science and Innovation, and the project FCT-13-7848, funded by the Spanish Foundation for Science and Technology <http://human-machine.unizar.es/>. This work is also supported by the Spanish MICINN projects TRA2011-29454-C03-01 and TIN2011-25606. The authors also want to thank Philip Hingston for providing the BotPrize competition testing environment.

References

- Amato, J., Sappa, A. D., Fornés, A., Lumbreras, F., Lladós, J. (2013). Divide and conquer: atomizing and parallelizing a task in a mobile crowdsourcing platform. In *Proceedings of the 2nd ACM international workshop on crowdsourcing for multimedia, CrowdMM '13, 2013* (pp. 21–22).
- Amato, A., Lumbreras, F., Sappa, A. D. (2014). A general-purpose crowdsourcing platform for mobile devices. In *Proceedings of the 9th international conference on computer vision theory and applications*.
- Anderson, J. R. (1993). *Rules of the mind*. New Jersey: Lawrence Erlbaum Associates Inc.
- Anderson, M. L. (2003). Embodied cognition: A field guide. *Artificial Intelligence*, 149(1), 91–130.
- Arnold, M. (1960). *Emotions and Personality*. Columbia University Press.
- Arrabales, R., Ledezma, A., & Sanchis, A. (2012). ConsScale fps: Cognitive integration for improved believability in computer game bots. In P. Hingston (Ed.), *Believable bots* (pp. 193–214). Berlin Heidelberg: Springer.
- Arrabales, R., Muñoz, J., Ledezma, A., & Sanchis, A. (2012). A machine consciousness approach to the design of human-like bots. In P. Hingston (Ed.), *Believable bots* (pp. 171–191). Berlin Heidelberg: Springer.
- Baars, B. J. (1997). In the theatre of consciousness: Global workspace theory, a rigorous scientific theory of consciousness. *Journal of Consciousness Studies*, 4(4), 292–309.
- Dennett, D. (1993). *Consciousness explained, penguin science*. Penguin Adult.
- Ertmer, P. A., & Newby, T. J. (1993). Behaviorism, cognitivism, constructivism: Comparing critical features from an instructional design perspective. *Performance Improvement Quarterly*, 6(4), 50–72.
- Gemrot, J., Kadlec, R., Bda, M., Burkert, O., Pbil, R., Havlcek, J., et al. (2009). Pogamut 3 can assist developers in building ai (not only) for their videogame agents. In F. Dignum, J. Bradshaw, B. Silverman, & W. Doesburg (Eds.), *agents for games and simulations, of lecture notes in computer science* (Vol. 5920, pp. 1–15). Berlin Heidelberg: Springer.
- Hingston, P. (2009). A turing test for computer game bots, computational intelligence and AI in games. *IEEE Transactions on*, 1(3), 169–186.
- Hingston, P. (2010). A new design for a turing test for bots. In *IEEE symposium on computational intelligence and games (CIG), 2010* (pp. 345–350). <http://human-machine.unizar.es/>.
- Laird, J. E., Newell, A., & Rosenbloom, P. S. (1987). Soar: An architecture for general intelligence. *Artificial Intelligence*, 33(1), 1–64. [http://dx.doi.org/10.1016/0004-3702\(87\)90050-6](http://dx.doi.org/10.1016/0004-3702(87)90050-6). <http://www.sciencedirect.com/science/article/pii/0004370287900506>.
- Nareyek, A. (2004). Ai in computer games. *Queue*, 1(10), 58–65.
- Nii, H. P. (1994). Blackboard systems at the architecture level. *Expert Systems with Applications*, 7(1), 43–54.
- Riedmiller, M., Braun, H. (1993). A direct adaptive method for faster backpropagation learning: The rprop algorithm. In *IEEE international conference on neural networks, 1993. Vol. 1* (pp. 586–591). <http://dx.doi.org/10.1109/ICNN.1993.298623>.
- Rocha, M., Cortez, P., & Neves, J. (2007). Evolution of neural networks for classification and regression. *Neurocomputing*, 70(1618), 2809–2816.
- Samsonovich, A. V. (2010). Toward a unified catalog of implemented cognitive architectures. In *Proceedings of the 2010 conference on biologically inspired cognitive architectures 2010: Proceedings of the first annual meeting of the bica society* (pp. 195–244). Amsterdam, The Netherlands: IOS Press.
- Schachter, S., & Singer, J. (1962). Cognitive, social, and physiological determinants of emotional state. *Psychological Review*, 69(5), 379–399.
- Schrum, J., Karpov, I., Miikkulainen, R. (2011). Ut2: Human-like behavior via neuroevolution of combat behavior and replay of human traces. In *Computational intelligence and games (CIG), 2011* (pp. 329–336). <http://dx.doi.org/10.1109/CIG.2011.6032024>.
- Schrum, J., Karpov, I. V., & Miikkulainen, R. (2012). Humanlike combat behavior via multiobjective neuroevolution. In P. F. Hingston (Ed.), *Believable bots* (pp. 119–150). Berlin Heidelberg: Springer.
- Szepesvri, C. (2010). Algorithms for reinforcement learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 4(1), 1–103.
- Togelius, J., Yannakakis, G., Karakovsky, S., & Shaker, N. (2012). Assessing believability. In P. Hingston (Ed.), *Believable bots* (pp. 219–233). Berlin Heidelberg: Springer.
- Vernon, D., Metta, G., & Sandini, G. (2007). A survey of artificial cognitive systems: Implications for the autonomous development of mental capabilities in

- computational agents. *IEEE Transactions on Evolutionary Computation*, 11(2), 151–180. <http://dx.doi.org/10.1109/TEVC.2006.890274>.
- Whitley, D., & Hanson, T. (1989). Optimizing neural networks using faster, more accurate genetic search. In *Proceedings of the third international conference on genetic algorithms* (pp. 391–396). Morgan Kaufmann Publishers Inc..
- Yannakakis, G., & Togelius, J. (2011). Experience-driven procedural content generation. *IEEE Transactions on Affective Computing*, 2(3), 147–161. <http://dx.doi.org/10.1109/T-AFFC.2011.6>.