Biologically Inspired Cognitive Architectures II:
Papers from the AAAI Fall Symposium (FS-09-01)

Assessing and Characterizing the Cognitive Power of Machine Consciousness Implementations

Raúl Arrabales, Agapito Ledezma and Araceli Sanchis

Carlos III University of Madrid
Avda. Universidad, 30.
28911. Leganés. Spain.
rarrabal@inf.uc3m.es

Abstract

Many aspects can be taken into account in order to assess the power and potential of a cognitive architecture. In this paper we argue that *ConsScale*, a cognitive scale inspired on the development of consciousness, can be used to characterize and evaluate cognitive architectures from the point of view of the effective integration of their cognitive functionalities. Additionally, a graphical characterization of the cognitive power of artificial agents is proposed as a helpful tool for the analysis and comparison of Machine Consciousness implementations. This is illustrated with the application of the scale to a particular problem domain in the context of video game synthetic bots.

Introduction

All artificial cognitive systems usually share at least one property: they are inspired on biological organisms. However, the specific inspiring models and the particular way in which they are implemented may differ greatly from one implementation to another. Consequently, it is not straightforward to characterize the cognitive capabilities of an artificial architecture in such a way that it can be put in a general context, i.e. compared with other implementations based on different principles.

The definition of a generic framework for the evaluation and characterization of the cognitive capability of an artificial agent can be beneficial not only for the comparative analysis of existing implementations, but also for the planning of a roadmap for future implementations.

ConsScale is a proposal intended to define such a framework using architectural and behavioral criteria (Arrabales et al., 2009a, 2009b). While most of the existing consciousness metrics proposals are based on "low level" information integration measures, see for instance Tononi's Φ measure (Tononi, 2004) or Seth's causal density measure (Seth, 2005), *ConsScale* is in contrast based on higher level functional aspects of the system. This does not mean that we disregard information integration as a key property of conscious systems; in fact, we aim to characterize how effectively information integration and inter-function synergies can contribute to the generation of conscious-like behaviors.

The main conceptual tool that we use for the characterization of the cognitive power of an artificial creature is the definition of an ordered list of levels associated with consciousness.

Levels of Cognitive Power

ConsScale levels are defined using both architectural and functional constraints. In this paper we will focus mainly on the functional or cognitive capabilities for the discussion on the assessment of the global cognitive power of a creature. See (Arrabales et al. 2009b) for a deeper analysis of architectural criteria.

Although a total of 13 levels are defined in *ConsScale* (from level -1 to level 11, including level 0), here we will consider only the most common 9 levels (2-*Reactive*, 3-*Adaptive*, 4-*Attentional*, 5-*Executive*, 6-*Emotional*, 7-*Self-conscious*, 8-*Empathic*, 9-*Social*, and 10-*Human-like*). Table 1 summarizes the cognitive skills required in these levels. See (Arrabales et al., 2009a) for more details. Note that agents can only qualify as level n *if and only if* all lower levels are also satisfied. In other words, all levels subsume lower ones. Nevertheless, the scale can also rate "anomalous" implementations not following the proposed level ordering.

From the point of view of behavior, each level defines a set of generic cognitive skills (*CS*) that must be satisfied. Therefore, in order to apply the scale to a real world problem these CS need to be grounded to behavioral tests that could be evaluated by third-person observation.

Table 1. ConsScale levels 2 to 10.

L_i	Cognitive Skills (CS)
2	$CS_{2,1}$: Fixed reactive responses ("reflexes").
3	$CS_{3,1}$: Autonomous acquisition of new adaptive reactive responses. $CS_{3,2}$: Usage of proprioceptive sensing for embodied adaptive responses.
4	$CS_{4,1}$: Selection of relevant sensory information. $CS_{4,2}$: Selection of relevant motor information. $CS_{4,3}$: Selection of relevant memory information. $CS_{4,4}$: Evaluation (positive or negative) of selected objects or events. $CS_{4,5}$: Selection of what needs to be stored in memory. $CS_{4,6}$: Trial and error learning. Re-evaluation of selected objects or events. $CS_{4,7}$: Directed behavior toward specific targets like following or escape. $CS_{4,8}$: Evaluation of the performance in the achievement of a single goal. $CS_{4,9}$: Basic planning capability: calculation of next n sequential actions. $CS_{4,10}$: Depictive representations of percepts.
5	$CS_{5,1}$: Ability to move back and forth between multiple tasks. $CS_{5,2}$: Seeking of multiple goals. $CS_{5,3}$: Evaluation of the performance in the achievement of multiple goals. $CS_{5,4}$: Autonomous reinforcement learning (emotional learning). $CS_{5,5}$: Advanced planning capability considering all active goals.
6	$CS_{6,1}$: Self-status assessment (background emotions). $CS_{6,2}$: Background emotions cause effects in agent's body. $CS_{6,3}$: Representation of the effect of emotions in organism (feelings). $CS_{6,4}$: Ability to hold a precise and updated map of body schema. $CS_{6,5}$: Abstract learning (learned lessons generalization).
7	$CS_{7,1}$: Representation of the relation between self and perception. $CS_{7,2}$: Representation of the relation between self and action. $CS_{7,3}$: Representation of the relation between self and feelings. $CS_{7,4}$: Self-recognition capability. $CS_{7,5}$: Advance planning including the self as an actor in the plans. $CS_{7,6}$: Use of *imaginational* states in planning. $CS_{7,7}$: Learning of tool usage.
8	$CS_{8,1}$: Ability to model others as subjective selves. $CS_{8,2}$: Learning by imitation of a counterpart. $CS_{8,3}$: Ability to collaborate with others in the pursuit of a common goal. $CS_{8,4}$: Social planning (planning with socially aware plans). $CS_{8,5}$: Ability to make new tools.
9	$CS_{9,1}$: Ability to develop Machiavellian strategies like lying and cunning. $CS_{9,2}$: Social learning (learning of new Machiavellian strategies). $CS_{9,3}$: Advanced communication skills (accurate report of mental content). $CS_{9,4}$: Groups are able to develop a culture.
10	$CS_{10,1}$: Accurate verbal report. Advanced linguistic capabilities. $CS_{10,2}$: Ability to pass the Turing test. $CS_{10,3}$: Ability to modify and adapt the environment to agent's needs. $CS_{10,4}$: Groups are able to develop a civilization and advance culture and technology.

Characterizing the Global Cognitive Power

As we discuss in the next section, using the former definition of levels, a Machine Consciousness implementation could be studied and evaluated with the aim to find out which cognitive skills from the list are present. However, a real characterization of the global cognitive power of the implementation calls for the combination of the results of all levels. In other words, an integrative measure is required.

Two approaches to cognitive characterization are described in the following. The first one consists on the application of a quantitative score and has been already discussed in detail elsewhere (Arrabales et al., 2009a). The second one is a new proposal intended to enhance the cognitive profiling that *ConsScale* can offer. This second approach is based on a graphical representation of the cognitive profile.

ConsScale Quantitative Score

The *ConsScale* Quantitative Score (CQS) is another assessment tool associated with the scale. It is intended to provide a numerical value as an indication of the cognitive power of the implementation being evaluated. The CQS is calculated in three steps:

- **L_i (compliance with level i)**: provides a measurement (0.0 to 1.0) of the compliance with level i. Instead of a linear distribution, this measure follows an exponential curve as a means to represent the synergy between different skills in the same level, i.e. the greater is the number of *CS* already fulfilled in one level, the greater will be the contribution of additional skills.

- **CLS (Cumulative Level Score)**: combines all L_i measures into one single aggregated value (0.0 to 1.55). This score follows a logarithmic progression which prevents the final score to be distorted by the combined effect of large scores in higher levels with poor scores in lower levels (e.g. implementations good at levels 5 and 6 but with poor evaluations in lower levels should not be awarded very high scores).

- **CQS (*ConsScale* Quantitative Score)**: provides a single value (from 0 to 1000) that indicates the cumulative synergy produced by the integration of cognitive skills across all levels. CQS is designed as an exponential curve priming those implementations which follow the developmental path implicit in the *ConsScale* level ordering (see Fig. 1).

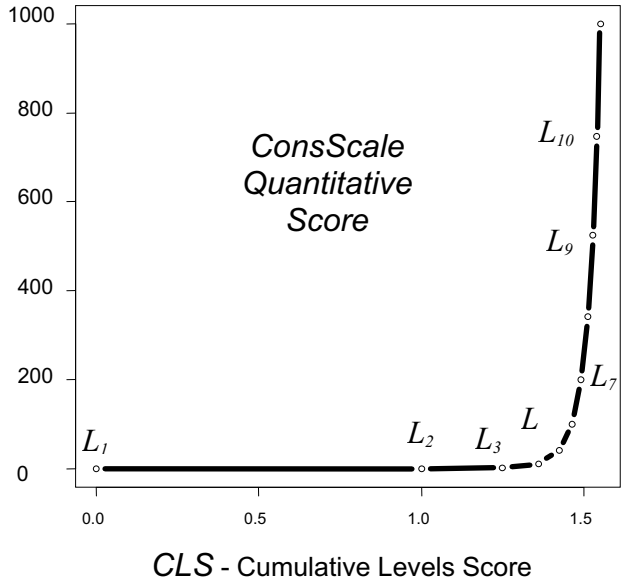

Figure 1. Possible CQS values as a function of CLS.

The mathematical procedure and details about the calculation of the CQS can be found in (Arrabales, et al., 2009a). Additionally, a CQS calculator is available at the *ConsScale* web site[1].

Graphical Cognitive Profiling

While having a single quantitative measure is useful for a quick characterization and evaluation, it lacks rich representation capabilities. For that reason, we propose the complementary use of graphical representations of the cognitive profiles.

Representing the cognitive profile of an agent in terms of *ConsScale* means to consider the particular L_i scores. Note that both CLS and CQS are one-dimensional parameters, calculated as a function of the multi-dimensional L_i; therefore L_i are the parameters to be used for a graphical representation that preserves the multi-dimensional richness of *ConsScale* levels definition.

For the sake of clarity, *ConsScale* levels -1 (*disembodied*), 0 (*isolated*), 1 (*decontrolled*), and 11 (*super-conscious*) have been excluded from the proposed graphical definition. They represent conceptual levels that complete the whole range of possible agents, but they would not provide additional meaning to the graphical profile representation.

We have decided to use radar charts as a compact and meaningful layout for the representation of the L_i values. See Fig. 2-4 for basic descriptions of the proposed graphical representation.

[1] http://www.conscious-robots.com/consscale/

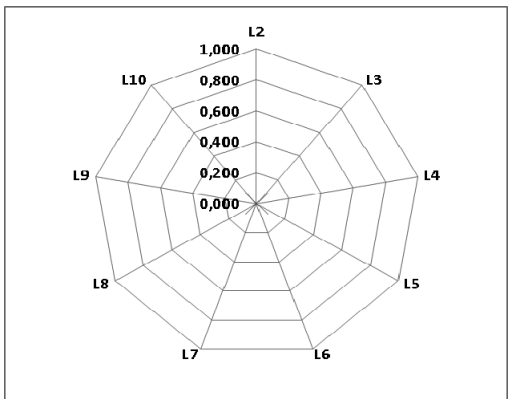

Figure 2. Empty *ConsScale* radar chart. Each axis represents the level of accomplishment in one *ConsScale* level. Possible values of each L_i axis range from 0.0 (no $CS_{i,x}$ is fulfilled) to 1.0 (all $CS_{i,x}$ are fulfilled). As all L_i are 0, this creature also has a CQS of 0.

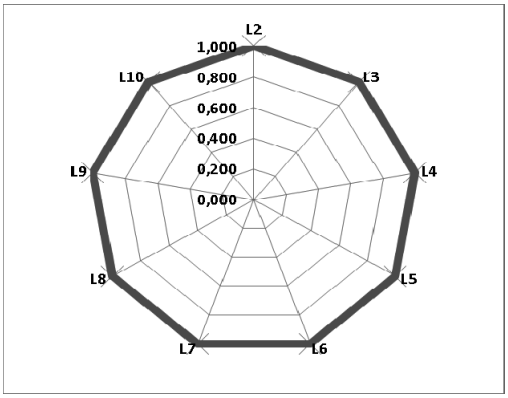

Figure 3. *ConsScale* radar chart representing a level 10 (*human-like*) creature. As levels 2 to 10 have an associated L_i value of 1, the corresponding CQS value is 745.74 (the maximum CQS value of 1000 is only achieved when L_{11} or *super-conscious* is also fulfilled).

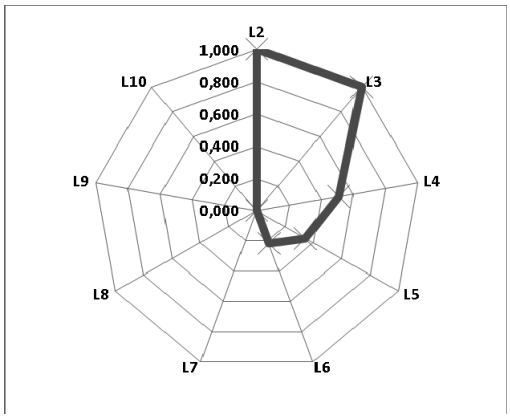

Figure 4. *ConsScale* radar chart representing a level 3 (adaptive) creature. Note that although this creature has features from higher levels it can only be considered as level 3. However its CQS (3,77) is higher than that of a pure level 3 creature (which would be 2,22).

A rough analysis of current cognitive architectures indicates that associated *ConsScale* radar charts would have good scores only in the upper right section of the chart.

Intuitively, we could think of the quest for human-level Machine Consciousness as a clock, where the hour hand represents the advance of the field. If we considered the area of the radar graph as an analog clock face, we could say that current Machine Consciousness implementations are located mostly within the first 3 hours. Future progression in the field is expected to create new artificial creatures whose cognitive profiles tend to fill the left half of the corresponding *ConsScale* radar graphs.

Practical Application

Defining generic levels of consciousness has the advantage of applicability to virtually any possible scenario. However, an instantiation process is required in order to practically assess the cognitive capability of Machine Consciousness implementations.

The set of cognitive skills described in table 1 refers to generic abilities. Therefore, they cannot be directly used for assessment. We call instantiation process to the set of tasks that need to be performed in order to apply the scale to a particular problem domain. Basically, behavior tests have to be designed in order to evaluate the presence of the cognitive skills considered in each level.

In the following we illustrate the *ConsScale* instantiation process using the domain of video game bots as example.

Problem Domain Definition: Video Game Intelligent Bots

In the context of *ConsScale*, a specific problem domain is defined in terms of:

- what the agent sensors can acquire (**objects** or percepts), and
- what the agent effectors can do (**actions** or verbs).

Essentially, we need to define an ontology of the target domain in such a way that generic cognitive skills associated to each *ConsScale* level can be translated into concrete testable behavior profiles.

In order to have a well defined problem domain we have constrained the scope of our experimentation environment: let us consider that the Machine Consciousness implementations to be evaluated are designed to be synthetic characters in a First-Person Shooter (FPS) video game. In this study we have focused on the particular case of Unreal Tournament 2004 (UT2004) game (Epic Games, 009). UT2004 bots being analyzed are specifically designed to compete against each other in a so-called deathmatch game. The main goal of the game is to kill as many other players as possible until either a maximum number of kills or a maximum game time is reached.

In this scenario the following objects are considered: players, ammo, weapons.

Analogously, the following actions have been identified: move, jump, run, turn, damage, fire, and sending chat messages.

The application of this ontology, based on objects and actions, permits us to redefine the *ConsScale* cognitive skills adapting them to the problem domain. Consequently, specific behavior profiles can be associated with these problem-specific cognitive skills, which in turn enable us to objectively evaluate the presence of the cognitive skills in the agent being studied.

Even though the defined ontology does not match with any possible internal representation that the agent might use or develop, it will be still valid for the behavioral evaluation based on third-person observations.

Using the FPS video game ontology that has been sketched out above, cognitive skills (CS) can now be instantiated (see table 1); additionally, associated behavioral profiles (BP) examples can be defined as follows (note that most of the higher level BP are not present in current state of the art implementations):

$CS_{2,1}$: Reflexes.

$BP_{2,1}$: Basic reflexes, as the ability to back up whenever the bot bumps into another bot or object.

$CS_{3,1}$: Ability to learn new simple behaviors adapted to the game.

$BP_{3,1}$: Basic behaviors that help the bot reaching better scores, like shooting other players when they are detected.

$CS_{3,2}$: Ability to use self state (health, ammo, etc.) to learn new adapted behaviors.

$BP_{3,2}$: Looking for health packs when health level is low or looking for ammo when needed.

$CS_{4,1}$: Ability to ignore sensory input not critical to current task.

$BP_{4,1}$: Ignoring detected ammo reloading kits when involved in a firefight and no more ammo is needed.

$CS_{4,2}$: Ability to discard actions not suitable for current situation.

$BP_{4,2}$: Actions like firing to walls when running away from an enemy are considered useless and avoided.

$CS_{4,3}$: Ability to select what information worth remembering (accessed from memory).

$BP_{4,3}$: When the bot is in need of ammo, it access its memory to get the position of previously seen ammo packs, then it goes directly to pick up the closest one.

$CS_{4,4}$: Ability to evaluate other players as friends or enemies. Ability to evaluate the benefits obtained by different ammo or health packs.

$BP_{4,4}$: Bot does not attack friends. Healing and re-arming is performed quickly by selecting the best health and ammo packs.

$CS_{4,5}$: Ability to select what information should be stored in memory.

$BP_{4,5}$: The position of health or ammo packs that could be needed later are stored in memory. The bot goes directly to a remembered position when it needs a pack (see $BP_{4,3}$).

$CS_{4,6}$: Ability to learn from trial and error.

$BP_{4,6}$: The bot identifies other players as friends or enemies by trial and error. If a player currently considered as friend (see $BP_{4,4}$) starts attacking the bot, it is now considered as enemy and the corresponding adaptive behaviors are performed (running away or shooting).

$CS_{4,7}$: Ability to adapt behavior to specific targets.

$BP_{4,7}$: The bot shows directed and sustained behavior towards enemies, like following and shooting them or running away from them.

$CS_{4,8}$: Ability to evaluate own's performance in combat.

$BP_{4,8}$: Actions that are not contributing to the expected goal are discarded. For instance, running away behavior is changed by another when this behavior is not contributing to diminish damage.

$CS_{4,9}$: Basic ability to plan next movements.

$BP_{4,9}$: Bot shows a coherent sequence of actions planned in order to reach certain goal. For instance, leaving a firefight for re-arming and then going back to combat.

$CS_{4,10}$: Ability to keep a depictive representation of objects in the game, i.e. representation in a sensorimotor grounded manner (Aleksander and Dunmall, 2003).

$BP_{4,10}$: The bot is able to effectively locate objects and calculate relative positions despite of its changing body and sensor positions (see $BP_{4,7}$). Bot shows a good shooting accuracy.

$CS_{5,1}$: Ability to interleave between game tasks.

$CS_{5,2}$: Ability to pursue several game goals.

$CS_{5,3}$: Ability to evaluate performance in relation with the accomplishment of several game goals.

$BP_{5,1\text{-}3}$: Behaviors interrupted due to certain circumstances are later resumed (see $BP_{4,9}$). For instance, a firefight is eluded because the bot is in need of healing, after getting a health pack, the bot resumes the attack. Additionally, the bot estimates to what extent goals are being accomplished depending on strategies being used. Effective behaviors are repeated more frequently than behaviors that lead to poor results.

$CS_{5,4}$: Ability to learn based on game experience.

$BP_{5,4}$: Evaluation performed according to $CS_{5,3}$ is used to select most promising strategies (see $BP_{5,3}$). For instance, the bot learns to use most destructive weapons when they are available.

$CS_{5,5}$: Ability to plan actions taking into account all active game goals.

$BP_{5,5}$: Actions are effectively interleaved as required for the accomplishment of multiple active goals. For instance, trajectory is slightly modified whilst chasing and shooting an enemy in order to pick up some ammo packs available in the surroundings.

$CS_{6,1}$: Ability to assess global self-status as an actor in the game. This represents functional aspects of emotions.

$CS_{6,2}$: Ability to adapt control mechanism to current status.

$CS_{6,3}$: Ability to keep a representation of emotions as described in $CS_{6,1}$ (Damasio, 1999).

$BP_{6,1\text{-}3}$: The bot enters a particular state depending on self-status assessment. Global behavior is biased by this state; for instance, if health is very low and no health packs are available, the bot tends to behave as if it was scared, avoiding any risk.

$CS_{6,4}$: Ability to keep an accurate representation of player's body.

$BP_{6,4}$: The bot control its position, gesture and orientation effectively. For instance, it is able to coordinate its sensorimotor systems to run in one direction while shooting to another relative direction at the same time.

$CS_{6,5}$: Ability to learn abstract concepts related to the game.

$BP_{6,5}$: Intelligent decisions indicate that specific knowledge about the game has been learnt. For instance, the bot tends to attack lonely enemies and run away from groups of enemies.

$CS_{7,1}$: Ability to maintain a model of self and a second order representation of the relation between the self and perceived game action.

$CS_{7,2}$: Ability to maintain an analogous second order representation of the relation between the self and bot actions.

$CS_{7,3}$: Ability to maintain a second order representation of the relation between feelings and self.

$CS_{7,4}$: Ability to self-recognize as a player in the game.

$CS_{7,5}$: Ability to make plans including the model of self as an actor.

$CS_{7,6}$: Ability to imagine the outcome of planned actions in terms of self.

$BP_{7,1\text{-}6}$: The behavior of the bot indicates that a sense of self is present. Decisions are not taken just as a function of player state (health, ammo, etc.), but based on a rich model of self which constitutes the basis for Theory of Mind capabilities (Vygotsky, 1980). The bot is able to recognize itself and the consequences of its own actions. In other words, a sense of agency is developed. Possible behavior tests include mirror test derivates as discussed in (Haikonen, 2007). Behavior is also modulated by the ability of the bot to foresee (imagine) the emotional outcome of a planned action. Therefore, new behaviors appear as a result of advance planning mechanism including imagination. For instance, the bot develop new strategies to attack enemies that have not been learned using reinforcement but imagination.

$CS_{7,7}$: Ability to use existing game objects as tools (note that support for using weapons and vehicles is native in the game, so their usage cannot be regarded as a bot cognitive capability).

$BP_{7,7}$: The bot manages to use some object as a means to achieve its objectives. For instance, using a movable object, like a box or a barrel, as an improvised shield.

$CS_{8,1}$: Ability to model other players as intentional selves.

$BP_{8,1}$: As other players are identified and modeled as selves, their movements can be predicted. The bot put itself in the place of another player to predict next actions of an opponent. Then, the behavior of the bot is shaped not only according to present sensory data but also using oponent's predicted movements. For instance, the bot predicts the

possible escape path of an enemy and make the necessary moves to block it.

$CS_{8,2}$: Ability to learn from other players by imitation.

$BP_{8,2}$: As the bot can manage both the model of self and models of others, it can also establish analogies and learn strategies by observing other bots. For instance, the bot can acquire new attack strategies developed by human players participating in the same game.

$CS_{8,3}$: Ability to collaborate with other players to get better scores.

$CS_{8,4}$: Ability to make plans including the models of other players as actors in the plans (intersubjectivity).

$BP_{8,3\text{-}4}$: Social behaviors like forming groups that collaborate in firefights.

$CS_{8,5}$: Ability to build new tools than can be used to achieve game goals.

$BP_{8,5}$: The bot combines several objects in order to build a new compound object that can be used either for defense or attack. For instance, building an improvised barricade made of a number of objects arranged along a line.

$CS_{9,1}$: Ability to develop Machiavellian as part of the game play.

$CS_{9,2}$: Learning of new Machiavellian strategies.

$BP_{9,1\text{-}2}$: The bot is able to reason about opponents' Theory of Mind, i.e. "I know you know I know" (Lewis, 2003). Therefore, it shows social intelligent behaviors like preparing an ambuscade.

$CS_{9,3}$: Ability to report mental content.

$BP_{9,3}$: The bot uses game's inbuilt chat system to coherently report its inner mental state.

$CS_{9,4}$: Ability to form cultural groups.

$BP_{9,4}$: Behavioral profiles associated with culture would require more complex environments. However, clues of cultural organization might be observed in groups of organized bots.

$CS_{10,1}$: Ability to produce accurate verbal report.

$CS_{10,2}$: Ability to pass an FPS adapted version of Turing test.

$BP_{10,1\text{-}2}$: The bot will pass an adapted Turing test, like the one proposed in the BotPrize competition[2]. Also classical Turing tests using game chat could be passed.

$CS_{10,3}$: Ability to modify the environment to serve bot's needs.

$CS_{10,4}$: Ability to develop civilizations and technology.

$BP_{10,3\text{-}4}$: Like in $BP_{9,4}$ complex behaviors associated with these skills requires more complex environments.

Note that the order of CS within a level is not significant. However, as described above, some CS can be grouped and associated the same behavioral profiles.

Conclusions

Thanks to the domain-specific behavior profiles defined in the previous section, agents can be evaluated by third-person observation. Higher level BPs are difficult to develop and being identified in such a simple game domain. Specifically, levels 9 and 10 would require extremely complex environments, like real world, in order to be satisfactorily tested.

Even though the specified problem domain is relatively simple, inferring BP from observations might sometimes be misleading. As the human observer has a strong theory of mind capability, he or she would tend to attribute mental states to the bot even when they are actually not present. Therefore, specific testing protocols like the BotPrize competition are to be used in order to increase the probability of accurate assessments.

Another assessment problem is related to the development of agents. As learning occurs over time, the same agent will show different developmental stages over time, i.e., different *ConsScale* profiles over time. Therefore, the tools proposed in this paper can also be used to assess the learning progression towards human-like cognitive capabilities.

[2] http://botprize.org/

Acknowledgements

We wish to thank Trung Doan for his suggestion of using spider graphs. This research has been supported by the Spanish Ministry of Education under CICYT grant TRA2007-67374-C02-02.

References

Aleksander, I. and Dunmall, B. (2003), Axioms and Tests for the Presence of Minimal Consciousness in Agents, *Journal of Consciousness Studies*, **10**(4-5), pp. 7 – 18.

Arrabales, R., Ledezma, A., and Sanchis, A. (2009a). Establishing a Roadmap and Metrics for Conscious Machines Development. Proc. of the 8th IEEE Intl. Conf. on Cognitive Informatics, pp. 94-101.

Arrabales, R., Ledezma, A., and Sanchis, A. (2009b). ConsScale: A Pragmatic Scale for Measuring the Level of Consciousness in Artificial Agents. *Journal of Consciousness Studies*. In press.

Damasio, A. R. (1999), *The Feeling of What Happens*, Heinemann, London.

Epic Games, Inc., Unreal Tournament 2004, (2009).

Haikonen, P. O. A. (2007c), Reflections of Consciousness: The Mirror Test, *in Proceedings of the 2007 AAAI Fall Symposium on Consciousness,* pp. 67 – 71.

Lewis, M. (2003). The Emergence of Consciousness and Its Role in Human Development. *Annals of the New York Academy of Sciences*. **1001**(1), pp. 104 – 133.

Seth, A. K. (2005), Causal connectivity of evolved neural networks during behavior, *Network: Computation in Neural Systems,* **16(**1), pp. 35 – 54.

Tononi, G. (2004), An information integration theory of consciousness, *BMC Neuroscience*, **5(**1), pp. 42.

Vygotsky, L.S. (1980). *Mind in Society: The Development of Higher Psychological Processes*. Harvard University Press.