A Neuronal Global Workspace for Human-like Control of a Computer Game Character

Zafeirios Fountas, David Gamez, Andreas K. Fidjeland

***Abstract*—This paper describes a system that uses a global workspace architecture implemented in spiking neurons to control an avatar within the Unreal Tournament 2004 (UT2004) computer game. This system is designed to display human-like behaviour within UT2004, which provides a good environment for comparing human and embodied AI behaviour without the cost and difficulty of full humanoid robots. Using a biologically-inspired approach, the architecture is loosely based on theories about the high level control circuits in the brain, and it is the first neural implementation of a global workspace that is embodied in a dynamic real time environment. At its current stage of development the system can navigate through UT2004 and shoot opponents. We are currently completing the implementation and testing in preparation for the human-like bot competition at CIG 2011 in September.**

I. Introduction

In 1950 Turing [1] proposed that an imitation game could be used to answer the question of whether a machine can think. In this game, a human judge engages in conversation with a human and a machine and has to decide which is human. The conversation is conducted over a text-only channel, such as a computer console interface, to ensure that the voice or body of the machine does not influence the judge's decision. While Turing posed his original test as a way of answering the question of whether machines can think, it is now more typically seen as a way of evaluating the extent to which machines have achieved human-level intelligence.

Since the Turing Test was proposed, there has been a substantial amount of work on programs (chatterbots) that attempt to pass the test by imitating the conversational behaviour of humans. Many of these compete in the annual Loebner Prize competition [2]. None of these programs have passed the Turing Test so far, and they often rely on conversational tricks for their apparent realism — for example, parrying the interlocutor with a question when they cannot process or understand the input.[1]

One reason why chatterbots have failed to pass the Turing Test is that they are not embodied in the world. For example, a chatterbot can only respond to the question "Describe what you see on your left" if the programmers of the chatterbot put in the answer when they created the program. This lack of embodiment also leads to a constant need for dissimulation on the part of the chatterbot — when asked about its last illness, it has to make something up. There is also an emerging consensus within the AI community that many of the things that we do with our bodies, such as learning a new motor action, require and exhibit intelligence, and most of AI research now focuses on intelligent actions in the world, rather than conversation.

This emphasis on embodied artificial intelligence has led to a number of variations on the Turing Test. In addition to its bronze medal for human-like conversation, the Loebner Prize also has a gold medal for a system that can imitate the body and voice of a human. Harnad [6] has put forward a T3 version of the Turing Test, in which a system has to be indistinguishable from a human in its external behaviour for a lifetime, and a T4 version of the Turing Test in which a machine has to be indistinguishable in both external and internal function. The key challenge with these more advanced Turing tests is that it takes a tremendous amount of effort to produce and control a convincing robot body with similar degrees of freedom to the human body. Although some progress has been made with the production of robots that closely mimic the external appearance of humans [7], and people are starting to build robots that are closely modelled on the internal structure of the human body [8], these robots are available to very few researchers and the problems of controlling such robots in a human-like manner are only just starting to be addressed.

A more realistic way of testing human-like intelligence within an embodied setting has been proposed by Hingston [9], who created the BotPrize [10] competition, which evaluates the ability of software programs (or bots) to provide human-like control of avatars within the Unreal Tournament 2004 (UT2004) computer game environment. In this competition human judges play the game and rate the humanness of other avatars, which might be controlled by a human or a bot, with the most human-like bot winning the competition. If a bot became indistinguishable from the human players, then it might be considered to have human-level intelligence within the UT2004 environment according to Turing [1]. This type of work also has practical applications since it is more challenging and rewarding to play computer games against opponents that display human-like behaviour. Human-like bot behaviour can also be applied to agent-based simulations in economics and the social sciences, and it can be used in

Zafeirios Fountas is with Imperial College London, Department of Computing, 180 Queen's Gate, London SW7 2AZ, UK (email: zafeirios.fountas09@imperial.ac.uk).

[1]Attempts are still being made to develop systems with the full background knowledge that would be required to carry out human-like conversation. The Cyc project [3] is developing a large database of common sense knowledge and hopes to use logical reasoning on the facts stored in this database to provide human-like reasoning. The Wolfram—Alpha system [4] stores expert knowledge in a form that enables answers to arbitrary questions in a domain to be computed. Wolfram [5] hypothesizes that all natural processes can be viewed as computations, and claims that it may eventually become possible to store and compute all possible knowledge in this way.

film special effects to model the behaviour of crowds and background characters in a natural manner.

This paper describes a system that we are developing to produce human-like control of an avatar within the UT2004 environment. Our working hypothesis is that biologically-inspired neural mechanisms could be an effective way of constructing a system with human-like behaviour. This approach enables us to learn from the neural circuits that produce our behaviour, and the models developed by biologically-inspired robotics can contribute to the neuroscientific understanding of the brain. The high level coordination and control of our system is carried out using a global workspace architecture [11], which is a popular theory about conscious control in the brain that explains how a serial procession of conscious states could be produced by interactions between multiple parallel processes. Several attempts have been made to show that the brain contains a global workspace [12], and a number of brain-inspired models have been developed to explain how a global workspace could be implemented in biological neurons. Programmatic implementations of the global workspace architecture have been used to control a naval dispatching system [13], and it was used in a winning entry of a previous BotPrize competition [14]. The key difference between our system and previous neural and programmatic implementations is that our system is both embodied in a complex real time environment *and* based on biologically-inspired circuits of spiking neurons, using CUDA hardware acceleration on graphics cards to simulate networks with tens of thousands of neurons and millions of connections in real time.

The first part of this paper starts with background information about the global workspace architecture and previous implementations using neural networks and computer programs (Section II). Next, the architecture of our system is described, including the structure of the global workspace, the planned and implemented parallel processors, and the mechanisms for extracting sensory data from UT2004 and passing motor data from the network to the avatar (Section III). Section IV sets out the implementation of our system, and the paper concludes with a discussion and plans for future work.

II. Background

A. Global Workspace

Global workspace architecture was first put forward by Baars [11] as a model of consciousness, and it is a way in which separate processes working in parallel can be coordinated to produce a single stream of control. The basic idea is that a number of separate parallel processes compete to place their information in the global workspace. The winning information is then broadcast to all of the other processes, which initiates another cycle of competition and broadcast (Figure 1). Different types of processes are specialized for different tasks, such as analyzing perceptual information or carrying out actions, and processes can also form coalitions that work towards a common goal. These mechanisms enable global workspace theory to account for

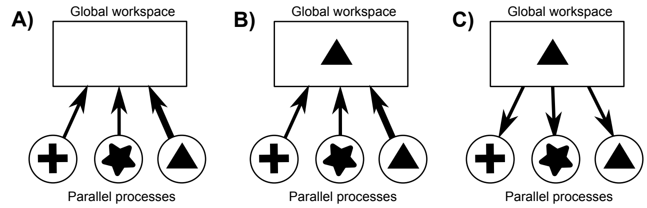

Fig. 1. Operation of the global workspace architecture. A) Three parallel processes compete to place their information in the global workspace; B) The process on the right wins and its information is copied into the global workspace; C) Information in the global workspace is broadcast back to the parallel processes and the cycle of competition and broadcast begins again.

the ability of consciousness to handle novel situations, its serial procession of states and the transition of information between consciousness and unconsciousness.

A number of neural models of global workspace architecture have been built to investigate how it might be implemented and operate in the brain. Dehaene et al. created a neural simulation to study how a global workspace and specialised processes interact during the Stroop task and the attentional blink [15], [16], [17], and a related model was constructed by Zylberberg et al. [18]. A brain-inspired cognitive architecture based on global workspace theory has been developed by Shanahan [19], which controlled a simulated wheeled robot, and Shanahan [20] built a global workspace model using simulated spiking neurons, which showed how a biologically plausible implementation of the global workspace architecture could move through a serial progression of stable states. More recently Shanahan [21] hypothesized that neural oscillation could be the mechanism that implements a global workspace in the brain, and he is developing neural and oscillator models of how this could work.

The global workspace architecture has also proved to be a successful way of controlling complex software systems. Franklin's [13] IDA naval dispatching system was created to assign sailors to new billets at the end of their tour of duty. This task involves natural language conversation, interaction with databases, adherence to Navy policy and checks on job requirements, costs and sailors' job satisfaction. These functions are carried out using a large number of codelets that are specialised for different tasks and organised using a global workspace architecture. A version of IDA that is capable of learning (LIDA) has also been developed [22]. Another successful software system based on a global workspace architecture is CERA-CRANIUM [14], which won the BotPrize in 2010 and the human-like bots competition at the 2011 IEEE Congress on Evolutionary Computation. CERA-CRANIUM uses two global workspaces: the first receives simple percepts, aggregates them into more complex percepts, and then passes them on to the second workspace. In the second workspace the complex percepts are used to generate mission percepts, which set the high level goals of the system.

While programmatic implementations of global workspace

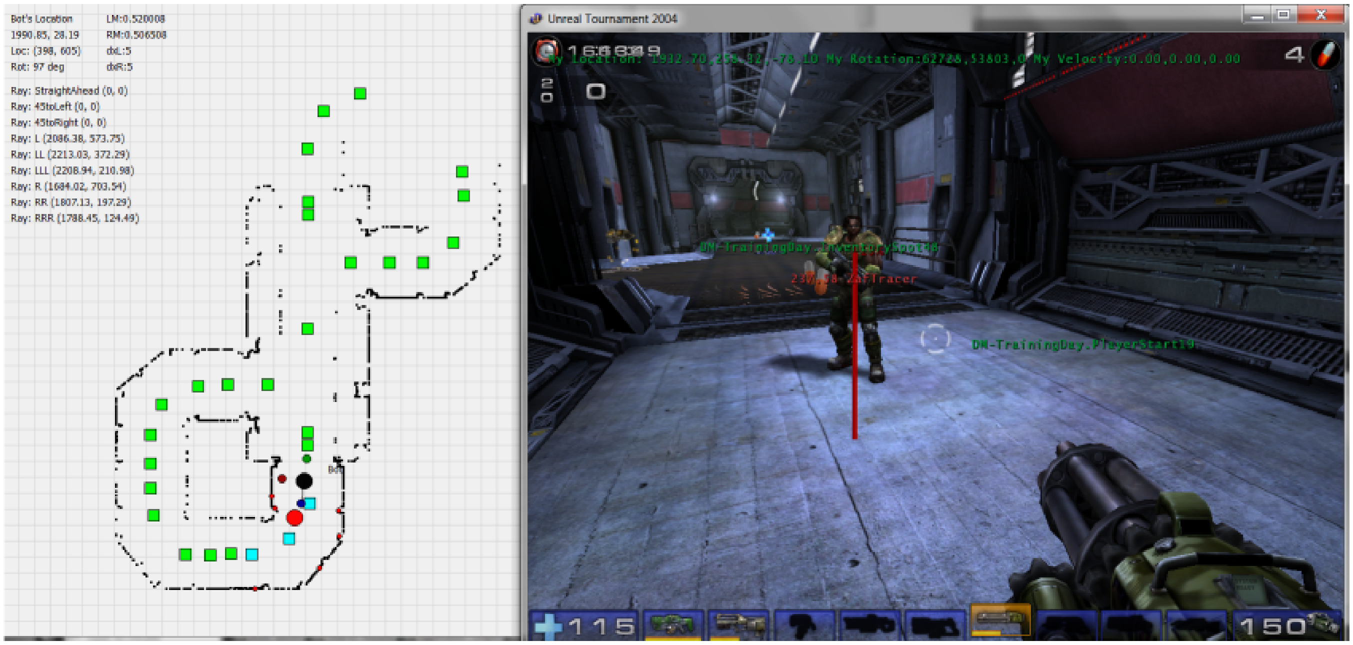

Fig. 2. Game environment. Left: map showing the data available to the bot (black) including other players (red), navigation points both in view (cyan) and out of view (green), as well as wall locations gathered from ray tracing. Right: the in-game view showing an external view of the avatar.

architecture have been used to solve challenging problems in real time, with the exception of [19], the neural implementations have typically been disembodied models that are designed to show how a global workspace architecture might be implemented in the brain. The aim of the research described in this paper is to close the gap between the successful programmatic implementations and the models that have been developed in neuroscience: using spiking neurons to implement a global workspace that controls a system within a challenging real time environment. Addressing this type of problem with a neural architecture will also help to clarify how a global workspace might be implemented in the brain.

B. *Spiking Neural Networks and Robotics*

The work described in this paper takes place against the background of previous research on the control of robots using biologically-inspired neural networks. There has been a substantial amount of research in this area, with a particularly good example being the work of Krichmar et al. [23], who used a network of 90,000 rate-based neuronal units and 1.4 million synaptic connections to control a wheeled robot. This network included models of the visual system and hippocampus, and used learning to solve the water maze task. Work that has been done on the control of robots using spiking neural networks includes a network with 18,000 neurons and 700,000 connections developed by Gamez [24], which controlled the eye movements of a virtual robot and used a learnt association between motor output and visual input to reason about future actions. There is also the work of Bouganis and Shanahan [25], who developed a spiking network that autonomously learnt to control a robot arm after an initial period of motor babbling, and there has been some research on the evolution of spiking neural networks to control robots [26]. As far as we are aware, none of the previous work in this area has used a global workspace implemented in spiking neurons to control a computer game character.

III. Neuronal Global Workspace

This section describes the global workspace that we are developing to control avatars within the UT2004 environment, while Section IV covers the implementation of this architecture in spiking neurons. The system currently includes two behavioural modules for wall-following (III-B) and shooting (III-C), with the ability to switch between behaviours. We are also in the process of developing other behavioural modules (III-D) which will endow the system with the ability to deal with a larger range of situations.

A. *Sensory input and output*

The bots operate in a 3D game environment, but instead of the rich visual interface provided to human players, the bots access information about the game in a somewhat abstract form. To begin with the bot has direct access to its location, direction and velocity, as well as access to the location of all currently visible items, players and navigation points. This data is provided in an absolute 3D coordinate frame, as shown in Figure 2. Second, the spatial structure of the immediate environment of the bot is available as range-finder data. The bot can decide where to cast rays, and it gets back a distance measurement of the point of intersection with a wall or other object, which enables it to build up a map of its environment at an appropriate spatial resolution. A third type of data are the scalars that describe different states of the avatar, such as crouching, standing, jumping and shooting.

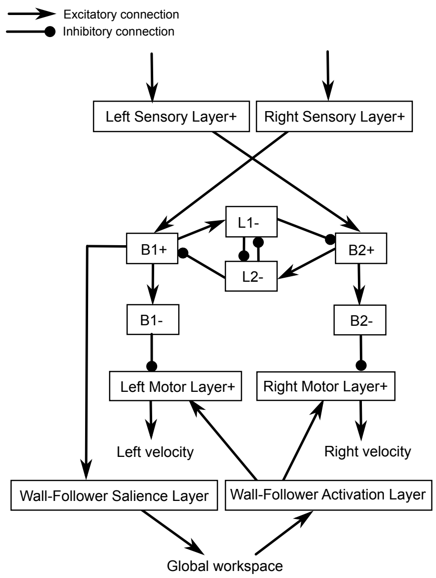

Fig. 3. Wall-following module

The bot can also access interoceptic data as scalar measures of overall health, adrenaline and armour.

To use this data within our system it is necessary to convert the scalar values containing information about the UT2004 environment into patterns of spikes that are fed into the neural network. To begin with, four pre-processing steps are applied to the input data. First, information about an enemy player or other objects is converted into a direction and distance measurement that is relative to the bot's location. Second, information about pain is processed as the rate of decrease of health plus the number of hit points. Third, the velocity of the bot is calculated, and finally, a series of predefined cast rays are converted into a simulated range-finder sensor. After the above conversions, each piece of information is transformed into a number in the range 0 to 1 that is multiplied by a weight. The result λ is a mean firing rate (spikes/millisecond), which is used to deliver spikes governed by a Poisson distribution with rate λ to the neuron layers representing the corresponding sensory data.

The avatar is controlled using commands that control movement, rotation, shooting and jumping. To make the system easier to control, the bot is treated as a differential driven robot and the mean firing rates of two layers of neurons are used to set the speed of two virtual motors. The location of the next point that the MOVE command will use is then found using a differential kinematic model. SHOOT and JUMP commands are sent to the bot if the mean firing rates of the two corresponding output layers exceeds a threshold.

B. Wall-following module

The wall-following module enables the bot to avoid obstacles and follow walls, and it is based on the Braitenberg vehicle type 3b [27]. Figure 3 shows the neural architecture of this module in which the activation layer equally excites

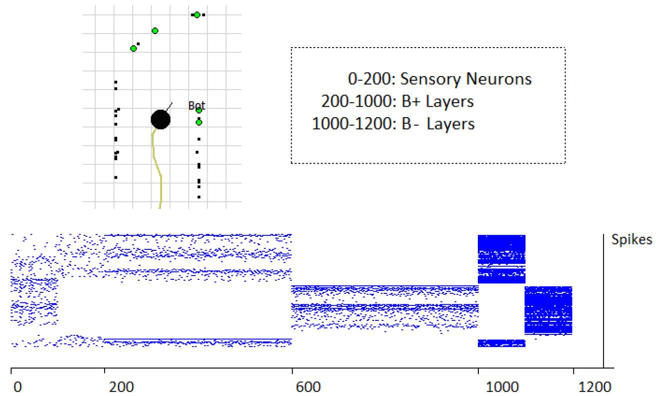

Fig. 4. Spiking within the wall-following module

the two motor layers, while B1 and B2 inhibit them. The B layers are connected with the two excitatory sensory layers as well as with two layers that cause mutual inhibition. This arrangement forms a winner-takes-all mechanism that can handle situations in which both sensors receive similar inputs. The layers that represent scalar values, such as sensory-motor layers, L-, B-, activation and salience layers, consist of 100 neurons each, while the B+ layers contain 400 neurons each. The activation of these layers during wall-following is shown in Figure 4. Alternation between the two B+ layers of neurons indicates changes in direction and the result of the winner-takes-all mechanism.

When only this module is used the bot wanders through the environment avoiding walls, as shown in Figure 5. In typical operation a goal-directed module might control the avatar, with the wall-following module taking control when the avatar comes into close proximity with walls or when there is no strong goal active in the system.

C. Shooting module

The shooting module causes the bot to fire at enemies when they are nearby. A neuron population vector representing the direction of the enemy is extracted from a sensory part of the global workspace, and it is converted by the shooting module into a shooting direction as well as a neuronal activation pattern that represents the salience of this behaviour. When this pattern gains access to the workspace it is broadcast back to all of the connected modules and excites the activation layer. If the mean firing rate of the activation layer exceeds a threshold, the bot will start shooting in the selected direction.

D. Other behavioural modules

We are working on the development of other modules that will contribute to the bot's behaviour under different circumstances. These modules will use information in different parts of the global workspace to control their activation and operation, and they will compete to broadcast their information back to the global workspace. A selection of planned modules follows.

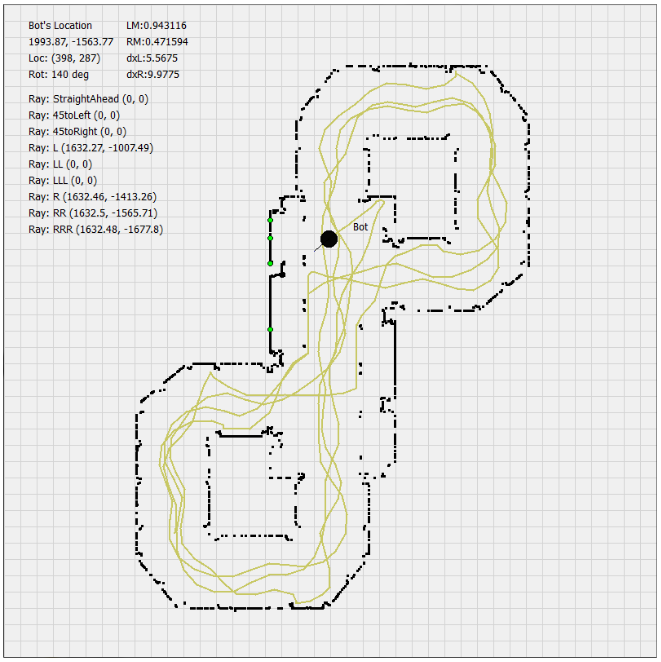

Fig. 5. Wall-following through the environment. The figure shows a path the bot takes with only the wall-following module activated.

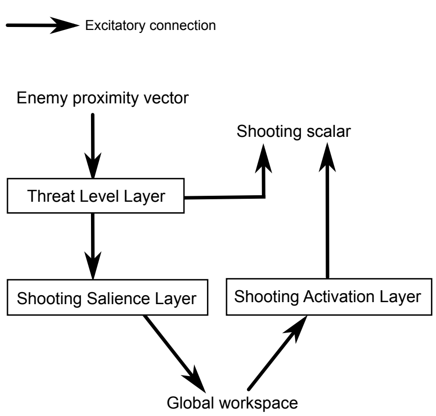

Fig. 6. Shooting module

a) Chasing: The bot will follow enemies that try to flee due to falling health. It will activate when the bot's own health is good and a close enemy is moving away. The module will drive the desired direction of movement towards the fleeing player and maintain firing.

b) Fleeing: The bot has to ensure its own survival, which in some cases entails fleeing. The fleeing module will activate when enemy proximity is strong and health is low. When active, it will drive the facing and movement directions to move the bot away from the source of immediate danger.

c) Recuperating: To ensure its survival the bot must replenish its health after receiving damage. The recuperating module will activate when health is low and the location of health vials is known. When active, this module will drive the direction of movement towards observed health vials.

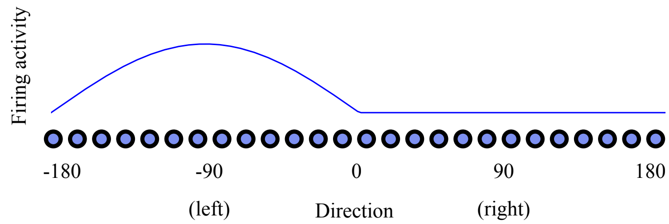

Fig. 7. Encoding of spatial data using an egocentric one-dimensional population code. A vector of neurons represent a range of spatial directions with respect to the current direction of view. The figure shows how firing activity in a neuron population could encode the presence of a wall to the left, where firing activity is inversely proportional to distance.

E. Global workspace

The global workspace facilitates cooperation and competition between the behavioural modules and ensures that only a single module controls the motor output at any point in time. Based on the workspace model in [20] the architecture consists of four layers of neurons linked with topographic excitatory connections (to preserve the patterns of activation) and diffuse inhibitory connections. The information in each layer is identical and the four layers are used as a kind of working memory that enables information to reverberate while the sensory and behavioural modules compete to broadcast their information. The layers in the workspace are divided into areas representing four different types of globally broadcast information: 1) desired bodily state; 2) proprioceptive data, i.e. the actual body movements; 3) exteroceptive data, i.e. the state of the observable part of the environment; and 4) interoceptive date, i.e. the internal state of the bot (Figure 8).

Each datum in the workspace is represented by a population of neurons. Data representing locations in the bot's environment are represented using an egocentric one-dimensional neuron population vector in which each neuron represents a direction in a 360° view around the bot, with the directions expressed relative to the current view direction, and the salience indicated by the firing rate. For example, Figure 7 shows a possibly encoding for wall proximity for a bot with a wall on its left side. Other workspace data represent scalar values (for example, whether firing is taking place or the current health level) using a rate code, in which the overall activity of a neuron encodes the value.

IV. IMPLEMENTATION

The current implementation is a working bot that interfaces with the UT2004 game environment and implements the wall-following and shooting behaviours using spiking neural networks run on the NeMo simulator. The system is capable of moving around the environment and shooting players that are in close proximity. An overview of the system is given in Figure 9.

A. Neuronal populations

The neuronal modules are implemented using a simulated neural substrate consisting of Izhikevich model neurons [28].

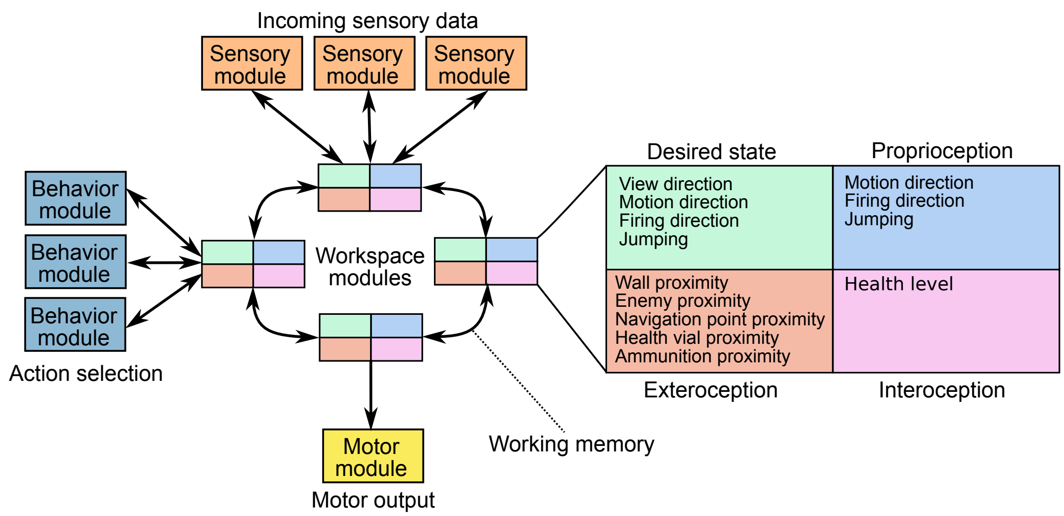

Fig. 8. Neuronal global workspace architecture

This phenomenological neuron model accurately reproduces the spiking response of neurons without modelling the ionic flows in detail. It is also computationally cheap and suitable for large-scale network implementations [29], [30].

The Izhikevich model consists of a two-dimensional system of ordinary differential equations defined by

$$\dot{v} = 0.04v^2 + 5v + 140 - u + I \tag{1}$$

$$\dot{u} = a(bv - u) \tag{2}$$

with an after-spike resetting

$$\text{if } v \geq 30 \text{ mV, then} \begin{cases} v \leftarrow c \\ u \leftarrow u + d \end{cases}$$

where v represents the membrane potential and u the membrane recovery variable, which provides post-potential negative feedback to v. The parameter a describes the time scale of the recovery variable, b describes its sensitivity to sub-threshold fluctuations, c gives the after-spike reset value of the membrane potential, and d describes the after-spike reset of the recovery variable.

The neurons in the network communicate by means of *spikes*, which are discrete signals generated when a neuron fires. The dynamics of synaptic transmission can be modelled with various levels of biological fidelity, which can account for the time-varying effect of a spike on the postsynaptic neuron and for dynamic variations in the synaptic efficacy. In this work we use synapses with a static synaptic weight and with spikes modelled as Dirac pulses. The term I in Equation 1 represents the combined current from spike arrivals from all presynaptic neurons, which are summed every simulation cycle.

The neuron model parameters (a–d) can be set to reproduce the behaviour of different types of neurons, which vary in terms of their response characteristics. We use two primary types of neurons, excitatory and inhibitory, which have some correspondence with biological neuron types. The firing of excitatory neurons tends to excite postsynaptic neurons, i.e. make them more likely to fire. The firing of inhibitory neurons tends to inhibit postsynaptic neurons, i.e. make them less likely to fire. Excitation and inhibition is modelled by positive and negative connection weights, and the excitatory and inhibitory neurons have different response characteristics. Functionally, the excitatory neurons in our networks represent the percepts in the global workspace and the modules feeding into it. Inhibitory neurons implement winner-takes-all mechanisms.

B. Spiking Neural Network Simulation

The neural network in our system is simulated using the NeMo spiking neural network simulation environment [31] developed by some of the authors of the current paper. NeMo is a high-performance simulator which runs parallel simulations on graphics processing units (GPUs). The simulator is primarily intended for computational neuroscience research, and general-purpose GPUs are used as the implementation medium due to the high level of parallelism and very high memory bandwidth. The use of GPUs makes the simulator eminently suitable for implementing neurally-based cognitive game characters in first-person shooters and similar games since the required parallel hardware is typically already present on the player's computer.

The state of the neural network is updated in a discrete-time simulation with a 1ms time step. The controlling application provides input spikes from sensors and converts output spikes to motor signals at each time step. The simulation consists of two main phases: neuron state update and spike delivery. The neuron state update is trivially parallel — a perfect match for the GPU — whereas the spike delivery requires communication between different threads. The spike delivery is therefore the limiting factor for the performance. On a high-end consumer-grade GPU, NeMo can deliver in excess of one billion spikes per wall clock second. The

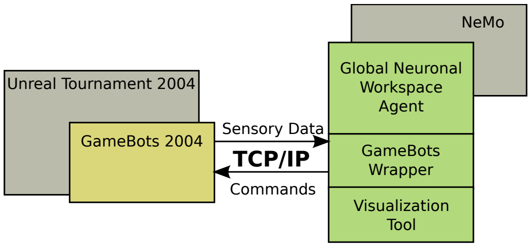

Fig. 9. System overview

existing behavioural modules are implemented with less than 2000 neurons with a fairly low level of connectivity, so the proposed network with a much larger number of modules should be able to comfortably run in real time.

C. UT2004 interface

Our system uses sockets to communicate with a GameBots 2004 server [32], which interfaces with UT2004 (Figure 9). In the case of a bot connection, GameBots sends synchronous and asynchronous messages containing information about the state of the world, and it receives commands that control the avatar in its environment. To facilitate this communication we have created a syntactic analyser that parses the GameBots messages and updates the attributes of the corresponding objects that form the system's representation of the world. The system then processes the gathered information and outputs the desired actions. This output is translated into commands that are transmitted back to the GameBots server. The time between two complete communication loops is independent of the time step of the internal neural simulation which is constant and set to 1 ms.

V. Discussion and Future Work

The system described in this paper is currently being developed for the human-like bot competition at CIG 2011 in September, and a substantial part of our future work will be the completion and testing of the system. One key performance criteria will be the system's ability to compete effectively within the UT2004 environment, since if it gets killed immediately, it will have little chance to display its human-like behaviour. We are also developing ways of measuring the humanness of an avatar's behaviour that will be used to evaluate and improve our bot. A measure of human-like behaviour can also be directly applied in the BotPrize competition, which has a prize for the bot that makes the best judgement about which of the other avatars is controlled by a human.

The most straightforward measure of behaviour within the UT2004 environment is the paths that are followed by the avatars during the course of play. While humans smoothly navigate around UT2004 using the rich visual interface, avatars guided by range-finder data are likely to exhibit erratic movement behaviour and get stuck on walls and corners. Avatars controlled by bots might also fail to explore larger areas of the game environment — for example, a wall-following bot might endlessly circle a block or column. It is also likely that human players will move towards salient features of their environment, such as doors, staircases or objects. To evaluate the degree to which our system exhibits human-like behaviour we are planning to gather path data from human players in the UT2004 environment (possibly log data from competitions) and compile statistics on the distribution of the distance of avatars from walls and other obstacles, the degree to which their path intersects with complex features of the environment, such as doors, stairs and corners, and the amount of the environment that is explored. These statistics will be used to evaluate the degree to which the movement behaviour of a bot is human-like within the UT2004 environment. It might also be possible to extend this approach to measure the humanness of more complex behaviours, such as avoidance or combat.

In the longer term we are planning to extend and fine tune the sensory and action modules to improve the system's human-like behaviour, and we would also like to use this system to explore how neural synchronization could implement a global workspace, along the lines proposed by Shanahan [21]. In our current system, the broadcast information is the activation within a population of global workspace neurons; in the neural synchronization approach the globally broadcast information is information that is passed within a population of synchronized neurons. A neural synchronization approach would open up the possibility of a more biologically realistic and dynamic definition of the global workspace, and it would also be possible to connect our work with the extensive literature on the link between neural synchronization and consciousness [33].

It is an open question whether spiking neural networks will prove to be an advantageous or even an adequate way of producing human-like behaviour within the UT2004 environment. On the one hand, spiking neural networks in the human brain can solve this problem, and so it appears to be possible, at least in principle, to build a network that functions in this way. On the other hand, the networks that we are currently capable of constructing are many orders of magnitude smaller than the human brain and we are only just starting to learn how to engineer spiking neural networks — it may take many years before this approach can produce the behavioural complexity of even the simplest animals. In the longer term, the spiking neural network approach has the advantage that it can both take inspiration from the brain and help us to understand the brain; in the shorter term the advantages and disadvantages of this approach will become apparent when our system competes against bots based on other principles in September.

VI. Conclusions

This paper has described a system that we are developing to display human-like behaviour within the UT2004 environment. The system uses a global workspace architecture implemented in spiking neurons to control the avatar, and the neural network that we have developed is already capable of navigating through the UT2004 environment and shooting opponents. We are currently part way through the implementation and testing of the global workspace, which will provide integration and coordination between the different sensory and motor modules, and we have plans to implement further sensory and behavioural modules before the human-like bot competition at CIG 2011 in September.

ACKNOWLEDGEMENTS

This work was supported by EPSRC grant EP/F033516/1.

REFERENCES

[1] A. Turing, "Computing machinery and intelligence," *Mind*, vol. 59, pp. 433–460, 1950.

[2] H. Loebner, "Loebner prize website." [Online]. Available: http://www.loebner.net/Prizef/loebner-prize.html

[3] M. Witbrock, C. Matuszek, A. Brusseau, R. Kahlert, C. Fraser, and D. Lenat, "Knowledge begets knowledge: Steps towards assisted knowledge acquisition in Cyc," in *Proc. AAAI 2005 Spring Symposium on Knowledge Collection from Volunteer Contributors*, 2005.

[4] S. Wolfram, "Wolfram—alpha website." [Online]. Available: http://www.wolframalpha.com

[5] ——, *A new kind of science*. Champaign, Ill.: Wolfram Media; London : Turnaround, 2002.

[6] S. Harnad, "Levels of functional equivalence in reverse bioengineering: The darwinian Turing test for artificial life," *Artificial Life*, vol. 1, no. 3, pp. 293–301, 1994.

[7] H. Ishiguro, "Studies on humanlike robots - humanoid, android and geminoid," in *Proc. Int. Conf. Simulation, Modeling, and Programming for Autonomous Robots (SIMPAR)*, vol. 5325. Springer, 2008, pp. 2–3.

[8] H. Marques, M. Jäntsch, S. Wittmeier, C. Alessandro, O. Holland, A. Diamond, M. Lungarella, and R. Knight, "ECCE1: the first of a series of anthropomimetic musculoskelal upper torsos," in *Proc. Humanoids*, 2010.

[9] P. Hingston, "A Turing test for computer game bots," *IEEE Transactions on Computational Intelligence and AI In Games*, vol. 1, no. 3, pp. 169–186, 2009.

[10] ——, "Botprize website." [Online]. Available: http://botprize.org

[11] B. J. Baars, *A cognitive theory of consciousness*. Cambridge England; New York: Cambridge University Press, 1988.

[12] S. B. Cho, B. J. Baars, and J. Newman, "A neural global workspace model for conscious attention," *Neural Networks*, vol. 10, no. 7, pp. 1195–1206, 1997.

[13] S. Franklin, "IDA — a conscious artifact?" *Journal of Consciousness Studies*, vol. 10, no. 4–5, pp. 47–66, 2003.

[14] R. Arrabales, A. Ledezma, and A. Sanchis, "Towards conscious-like behavior in computer game characters," in *Proc. Int. Conf. Computational Intelligence and Games*. IEEE Press, 2009, pp. 217–224.

[15] S. Dehaene, M. Kerszberg, and J. P. Changeux, "A neuronal model of a global workspace in effortful cognitive tasks," *Proc. Nat'l Academy Science USA*, vol. 95, no. 24, pp. 14 529–34, 1998.

[16] S. Dehaene, C. Sergent, and J. P. Changeux, "A neuronal network model linking subjective reports and objective physiological data during conscious perception," *Proc. Nat'l Academy Science USA*, vol. 100, no. 14, pp. 8520–5, 2003.

[17] S. Dehaene and J. P. Changeux, "Ongoing spontaneous activity controls access to consciousness: a neuronal model for inattentional blindness," *PLoS Biol*, vol. 3, no. 5, p. e141, 2005.

[18] A. Zylberberg, D. Fernndez Slezak, P. R. Roelfsema, S. Dehaene, and M. Sigman, "The brain's router: A cortical network model of serial processing in the primate brain," *PLoS Comput Biol*, vol. 6, no. 4, p. e1000765, April 2010.

[19] M. P. Shanahan, "A cognitive architecture that combines internal simulation with a global workspace," *Consciousness and Cognition*, vol. 15, pp. 433–449, 2006.

[20] ——, "A spiking neuron model of cortical broadcast and competition," *Conscious and Cognition*, vol. 17, no. 1, pp. 288–303, 2008.

[21] ——, *Embodiment and the inner life: cognition and consciousness in the space of possible minds*. Oxford: Oxford University Press, 2010.

[22] U. Ramamurthy and S. Franklin, "Self system in a model of cognition," in *AISB Symposium on Machine Consciousness*, 2011, pp. 51–4.

[23] J. L. Krichmar, D. A. Nitz, J. A. Gally, and G. M. Edelman, "Characterizing functional hippocampal pathways in a brain-based device as it solves a spatial memory task," *Proc Natl Acad Sci U S A*, vol. 102, no. 6, pp. 2111–6, 2005.

[24] D. Gamez, "Information integration based predictions about the conscious states of a spiking neural network," *Consciousness and Cognition*, vol. 19, no. 1, pp. 294–310, 2010.

[25] A. Bouganis and M. Shanahan, "Training a spiking neural network to control a 4-DoF robotic arm based on spike timing-dependent plasticity," in *Proc. IEEE Int. Joint Conference on Neural Networks (IJCNN)*, 2010, pp. 4104–4111.

[26] H. Hagras, A. Pounds-Cornish, M. Colley, V. Callaghan, and G. Clarke, "Evolving spiking neural network controllers for autonomous robots," in *Proc. ICRA*, vol. 5, 2004, pp. 4620–4626.

[27] V. Braitenberg, *Vehicles: Experiments in synthetic psychology*. Cambridge, MA: MIT Press, 1984.

[28] E. M. Izhikevich, "Simple model of spiking neurons," *IEEE Trans. Neural Networks*, vol. 14, pp. 1569–1572, 2003.

[29] R. Ananthanarayanan, S. K. Esser, H. D. Simon, and D. S. Modha, "The cat is out of the bag: cortical simulations with 10^9 neurons, 10^{13} synapses," in *Proc. Conf. High Performance Computing Networking, Storage and Analysis*. New York, NY, USA: ACM, 2009, pp. 1–12.

[30] E. Izhikevich and G. Edelman, "Large-scale model of mammalian thalamocortical systems." *Proc. Nat'l Academy Science USA*, 2008.

[31] A. K. Fidjeland and M. P. Shanahan, "Accelerated simulation of spiking neural networks using GPUs," in *Proc. IEEE Int. Joint Conference on Neural Networks (IJCNN)*, 2010.

[32] R. Adobbati, A. N. Marshall, A. Scholer, S. Tejada, G. Kaminka, S. Schaffer, and C. Sollitto, *Gamebots: A 3D virtual world test-bed for multi-agent research*, 2001, vol. 45, pp. 47–52.

[33] F. Varela, J. P. Lachaux, E. Rodriguez, and J. Martinerie, "The brainweb: phase synchronization and large-scale integration," *Nat Rev Neurosci*, vol. 2, no. 4, pp. 229–39, 2001.