A tool to design interactive characters based on embodied cognition

Joan Llobera and Ronan Boulic, *Senior Member, IEEE*

Abstract — **Creating interactive characters is one of the most challenging tasks of videogame design. In order to facilitate such an endeavour, we introduce a decisional and behaviour synthesis architecture integrated in the game engine Unity3D. A distinguishing feature of this architecture is that it embraces embodied cognition principles and uses them as implementation requirements. From these we derive an architecture which is based on a novel combination of previously proposed systems, together with some simplifications.**
We also argue that the architecture proposed has properties – modularity, scalability, stability- which can be beneficial for its practical industrial adoption, particularly in the context of the recent improvement in machine learning techniques. Artificial intelligence is a quite technical domain, and we believe such a tool can facilitate interactive character creation by creative minds in industrial applications.

Index Terms— **Virtual reality, artificial intelligence, decision-making, character modelling**

I. INTRODUCTION

CREATING interactive characters is, arguably, one of the most challenging tasks of videogame design and implementation. The arrival to the consumer market of virtual reality displays is also steering the demand of immersive virtual environments (IVE) with interactive characters that are adapted to the specificities of such environments. IVE experiences are, typically, embodied [1,2]**.** This has several implications regarding user expectations and design principles, which makes even more difficult the creation of appropriate interactive characters.

In this paper we introduce a behaviour-based artificial intelligence architecture compatible with principles of embodied cognition, and usable in the context of industrial content production. The aim of this work is, first, to facilitate the creation of interactive characters by non-technical people.

In addition, we also believe that embracing principles widely accepted in embodied cognition can facilitate the adoption of machine learning techniques and cognitive models in the design of interactive characters.

The document is organized as follows. In section II we outline general embodied cognition principles and characterize the related work in the fields of behavioural intelligence and interactive character design. In section III we introduce the proposed architecture, the visual interface that allows monitoring and editing characters using it and the design choices guiding the integration within Unity3D [3], a commonly used game engine. In section IV we present the results. Section V discusses advantages and drawbacks of using such a system when compared with designing interactive characters with more common approaches to interactive character design, and the extent the designed system realizes the principles of embodied cognition previously outlined.

II. RELATED WORK

A. Principles of embodied cognition

Embodied cognition assumes that we think not only with our minds, but also with our bodies (and our brains in our bodies), and that this fact is far from innocuous [4]. At a behavioural and decisional level, embodied cognition can easily be opposed with more traditional rationalistic approaches, still largely predominant in the field of artificial intelligence [5]. This paradigm has largely been explored and, at some extent, replicated in artificial systems, in the fields of virtual reality and robotics. In Virtual Reality, the rendering from a first-person perspective, integrating head and body movements, has enabled the experimental study of out-of-body experiences [2], [6]. In robotics, the implementation of sensorimotor loops has fostered the development of different cognitive architectures based on the *enactive* paradigm [4]. Although there is a clear acknowledgement that embodied cognition is a paradigm with solid foundations, it is still not certain how higher order cognition develops from the enactive paradigm. Nehaniv et al. [7] summarize it by stating that *For human cognition, the enactive paradigm is yet to develop a detailed understanding of higher-order cognition*.

There are, however, embodied cognition principles that describe what seem to be natural properties of the human cognition. One of such properties is the fact that our behaviour seems to be associated with intentions [8], in the sense of purposes or goals, and that such characteristic is very important, particularly for social interaction [9], [10]. Some animals also seem to use intention-based communication principles (see, for example [11]) and, in principle, both robots and interactive characters can be designed to do so.

SL. Address: 381, Consell de Cent Street, 08009 Barcelona, Spain (e-mail: joan@timepath.io). Dr Ronan Boulic is the head of the Immersive Interaction Group at EPFL. Address: EPFL-IC-IIG, Station 14 CH-1015 Lausanne, Switzerland (e-mail: ronan.boulic@epfl.ch)

In a communicative context, Relevance Theory [12] outlines two cognitive principles which have met a relatively large consensus in the way it depicts intention-based communication. For instance, the first principle states that *human cognition tends to be geared towards the maximization of the cognitive effect,* which is a relevance measure. This implies that, given a certain context, performing an action will be done in such a way that it maximizes the cognitive knowledge it provides, given a certain amount of information shown in the action.

This general vision seems compatible with the depiction of a cognitive system supported by a massive interrelation of cognitive modules embodied in the brain and its dialogue with the environment [13]. Consistent with the previous depiction, there also exist developmental theories claiming that our cognitive development is nurtured from the interaction of these intentional actions and their interaction with the world [10], [14]. From a biological perspective, *embodied* is largely –and often tacitly- assumed to mean instantiated in distributed, parallel excitatory and inhibitory networks that co-exist and interact recursively. For example, in the brain there are two different kinds of neuronal networks. On one side interneurons are inhibitory, short-distance and abundant connections, and which significantly contribute to the encoding of relevant information in spatio-temporal spike dynamics [15]. Opposite to this, pyramidal neurons are long range excitatory connections Each of these networks constitute separate yet complementary mechanisms of brain activity and regulation [16]. It is also worth highlighting that the dialogue between excitatory and inhibitory networks in a balanced but dynamic interrelation is a general biological regulation mechanism which goes largely beyond the cognitive part of such systems. This picture also seems consistent with biological learning and adaptation mechanisms. For example, brain plasticity is made possible by long term potentiation (LTP) and depression (LTD), which affect and sculpt neural firing statistics, among other brain plasticity mechanisms (changes in the topology of connections, neurogenesis, etc.).

These principles are not exclusively human: other living creatures have been shown to have it. These principles are neither exclusive to embodied cognition. We believe, though, that their combination is representative of embodied cognition as it is described across disciplines. Indeed, although this outline of general principles of embodied cognition is, at first glance, quite brief, it already gives us with few principles which are desirable for designing an agent which has to show behavioural intelligence:

1- Low level abstract perceptions (space, time, change) are built from sensorimotor loops through enactive mechanisms.
2- Cognitive agents have goals
3- Maximization of relevance measures can give a cognitive agent pragmatic behaviour skills
4- Massive modularity is a good way to build an embodied cognitive agent that can learn to behave from its interaction with its environment.
5- These modules are regulated through recursive dynamic networks of excitatory and inhibitory signals.

6- LTP and LTD are biological learning mechanisms that a) are slower than the excitatory and inhibitory signals, b) occur massively in all the modules and c) can regulate and modify notably the overall behaviour of the cognitive system.

To what extent can an artificial synthetic agent implement some or all of these principles?

B. Decision systems

To create interactive characters that appear to have some form of social intelligence, we compare two existing decision systems: Behaviour Trees and Behaviour Networks.

To create apparent behavioural intelligence in videogames, the first choice seems to be Behaviour Trees. Behaviour Trees are generally used for fast-action games, such as first person shooters. There is a plethora of variations, but the essential idea is that an agent makes decisions based on decision trees, with simple rules (pass, logical if, random, etc.) for a parent node to choose its child or sibling. The leaf nodes trigger some form of behaviour.

The main practical advantage of Behaviour Trees is that they are easy to elaborate. To a certain extent, they can also be engineered to do quite sophisticated behaviour, the typical scenario being the creation of characters that behave like soldiers in a battlefield.

Extensions and variations of Behaviour Trees are numerous in the literature. See, for example, an extension to process emotional factors [17], and a parameterized version of Behaviour Trees [18]. There are also several implementations available online, either with free open source license, or as more complex commercial packages with a well-designed user interface, such as the Behaviour Designer Unity3D asset [19].

Behaviour Networks have a different user base adoption, but are interesting from an embodied cognition perspective. Behaviour Networks maximize utility on the domain of Goals. However, these Goals can also be situation-dependent, i.e., the importance of a Goal can change dynamically according to a perception-based relevance measure. In addition, Behaviour Networks encapsulate behaviour in Skills. These are not simply action triggers: the selection of a Skill depends on Preconditions and available Resources, and a Skill can trigger an action at different intensities. In addition, different intensity of decisions could easily be mapped to different motor behaviour. For example, a navigation skill could decide to trigger a run or a walk behaviour, depending on a set of Preconditions.

C. Behaviour synthesis

The synthesis of behaviour in automated interactive characters requires the combination of different animations. There exist different methods for behaviour synthesis, altough in practice commercial interactive contents are based, most often, on hierarchical state machines. This is the case, for example, for Unity3D's Mecanim system [20], as well as for the Unreal's Persona animation system [21] .

Alternative systems also exist. For example, ADAPT [22] is an architecture for behaviour synthesis that, instead of using a hierarchical representation to combine animation layers, uses

what are called *choreographers*, i.e., generators or transformers of movements, organised in a dataflow graph, which specifies how the different motions should be combined. In this way, walking and reaching animations can be combined, first, and then a later step can adjust the pose to fit with physical constraints. In this sense, the structure is not a hierarchical one, but rather introduces constraints on the order in which movements are processed. ADAPT has been shown useful, for example, for the synthesis of interactive stories [22], [23].

It is also worth mentioning that the previous frameworks for behaviour synthesis can further benefit from higher level representations, such as commonly done for path planning. Kallman and Kapadia [24] provide an updated review of different methods for these methods.

D. Machine learning in interactive character design

Machine learning has also been used in embodied approaches to interactive character design. Gillies [25], [26] showed how machine learning can be used for interactive character design. Kastanis and Slater [27] showed how reinforcement learning could be used by an automated interactive character to learn social rules, like interpersonal distances, from the interaction with a human participant embodied in a virtual body.

Other methods, originated in the field of robotics, and thus essentially embodied, have shown that example-based learning can be used to learn to perform a task successfully. For example, [28], [29] show how machine learning in perception-action loops achieve optimal real-time behaviour synthesis in relatively simple and local domains. However, performing complicated tasks that require coordinated actions is a challenge that seems hard to solve due to the large dimensionality [30].

Integrative approaches also exist. In particular, significant efforts to emulate the full developmental process of competence acquisition in linguistic domains have shown that combining the development of action, conceptualization and social interaction can produce sophisticated interactive behaviour in robots, taking only natural interactions as input [31]. These can be based on networks of Self-Organizing-Maps [32], using techniques which allow for little or no design capacity from the content creator, and therefore with results that, upon comparison with the videogame literature, are closer to player modelling [33] rather than to behaviour synthesis efforts in a production context.

III. THE ARCHITECTURE

A. A user-centred approach

We define Simple Behaviour Networks (SBN) as a simplification of Extended Behaviour Networks (EBN) [34]. From a user perspective, the use of such an architecture involves two steps: first, defining Perceptions, Actions and Skills encapsulating different abilities of the interactive character (talking, shooting, etc.) and, second, giving a set of Goals to the character in order to motivate the use of these Skills within the interactive gameplay.

The first step requires a relatively technical process, while the second is actually very fast, with little or no configuration needed, once the first step is completed. However, the second step (the definition of goals and their importance) requires fast prototype iteration to be able to adjust the overall experience that results from multi-agent interactions.

We combine SBN with the ADAPT system for behaviour synthesis. The main reason to do this is that the ADAPT system does not impose a hierarchical structure, as opposed to hierarchical state machines typically built in commercial game engines. This greatly facilitates the definition of Skills.

Taking a closer look, we can also notice that the choreographer introduced in ADAPT is, from a topological perspective, a partial order processing structure on the time domain. The choreographer has a role similar to the hierarchical layering of state machines characteristic of Unity3D's Mecanim and Unreal's Persona. Mapping SBN Skills to ADAPT choreographers allows having a completely distributed and non-hierarchical architecture for decision and behaviour synthesis.

B. Formal definition

An SBN is a set of Perceptions, Resources, Goals, Skills, and Parameters, where:

A Perception p is a fuzzy proposition, i.e., a proposition with a truth value between 0 and 1 obtained from an evaluation function.

Fuzzy propositions can be combined through the operators *and* and *not* [35]. Given two propositions a and b, with fuzzy truth values τ_a and τ_b, the unary operator *not* '¬' is defined as:

$$\neg(a) \triangleq 1 - \tau_a$$

where τ_a is the fuzzy truth value of proposition a. The binary operator *and* '∧' is defined as:

$$a \wedge b \triangleq \min(\tau_a, \tau_b)$$

A Resource *res* is a proposition with a real positive truth value τ_{res}. It also has a fuzzy activation threshold of θ_{res}. It is used to represent the amount of energy, body parts, bullets, or any other physical object needed to perform an action.

A Goal g is a tuple *(GCon, i, RCon)*, where:
- *GCon* is the Goal condition, defined as a combination of Perceptions (using *and* and *not*)
- *i* is the importance of the Goal
- *RCon* is the relevance condition, i.e., the situation-dependent and dynamic importance of the Goal, defined as combinations of Perceptions (using *and* and *not*)

A Skill is a tuple *(Pre, b, d, Post, Res, a)* where:
- *Pre* is a combination of Perceptions
- *b* is the behaviour performed once the Skill is selected for execution.
- *d* is the decidedness of execution (between 0 and 1)
- *Post* is a set of tuples *(Eff, ex)*, where *Eff* is a fuzzy proposition and *ex* is the likeliness of achieving the expected effect
- *Res* is a set of tuples (r*es*, τ_u), where r*es* is a resource, and τ_u is a positive real value, representing the amount of *res*

expected to be used by the execution of the Skill
- a is the activation of the Skill

The Parameters are the weighting factors that characterize how Skills activate one another, and provide values that have been worked well in our past experience:
- γ is the activation weight from one module to another
- δ is the inhibition weight from one module to another
- β is the inertia of a module, i.e., its own influence across different cycles
- θ is a positive activation threshold, with θ_{max} being a fuzzy value (i.e., between 0 and 1) standing as its maximal value
- $\Delta\theta$ is the threshold decay, across activations, and smaller than θ_{max}.

The activation a of a certain Skill S_k at time t is related with its own inertia β and how its expected effects *Eff*, or the expected effect of its successor modules, match the Goal Condition *Gcon* of a given Goal g.

Concretely, given a Goal g, consider the set of N Skills $\{S_1, \ldots, S_N\}$ defined as the skill k and its successors. We select the subset of Skills $\{S_j\}$, denoted by index j, as the Skills whose expected effect Eff_j, part of its *Post* conditions, matches the Goal Condition *GCon* of Goal g. Eff_j, has a likelihood of ex_j.

Under the previous definitions, each Goal g contributes to the activation of Skill k through the equation:

$$a_k^t \triangleq \beta \, a_k^{t-1} + \sum_{\forall g} \text{Sel} \, {}_{max}^{abs} \left(\gamma \, f(i, r^t) \, ex_j, -\delta \, f(i, r^t) ex_j, in_\gamma, in_\delta \right)$$

Where i and r^t are the static importance and the relevance at time t of Goal g, $f(i, r^t)$ is a triangular norm, and $\text{Sel} \, {}_{max}^{abs}$ is a four input function which selects the input whose absolute value is maximum.

in_γ, in_δ are inertia terms, from successor and conflictor Skills. For each Goal g, if k is a successor Skill and e the executability of a Skill (i.e. the truth value associated to its precondition *Pre*):

$$in_\gamma \triangleq \gamma \, \sigma(a_k^{t-1}) ex_j (1 - e_{succ})$$

And 0 otherwise. Similarly, for each Goal g, if k is a conflictor Skill, we define:

$$in_\delta \triangleq -\delta \, \sigma(a_k^{t-1}) ex_j \, e_{conf}$$

And 0 otherwise. σ is a sigmoid function, for which Goetz [36] showed that, in general, it leads to stable behaviour in recurrent networks of Skills.

In other terms: the activation will follow an inertial path, but its increase or decrease will depend, for each Goal, on whether its successor Skills can help or prevent achieving that Goal. The value of the *in* inertia terms will depend upon the update of the successor and conflictor Skills in previous iterations.

It is important to consider that in Sel the terms are selected depending on which absolute value is greater, but the value used conserves the sign. This is important to make sure a conflict between two opposite Goals will always consistently activate a Skill, as opposed to cancelling each other's contribution to Skill activation, something that could potentially create flip-back-and-forth undesired dynamics.

We can now define the complete decision algorithm, for a given Skill S_k:

1. Calculate the activation a and the executability e of the module
2. Calculate the execution-value $h(a, e)$, which can be eventually any monotonic function, but in SBN is simply the product.
3. For each resource *res* used by Skill k, starting with the previous unavailable resource:
   (a) Check if the execution value exceeds the activation threshold of the resource *res*, this is $h(a, e) \geq \theta_{res}$
   (b) Check if the amount of resource units still available are enough, this is, if: $\tau_u \geq \tau_{res}$. If so, bind the resources, i.e., increase the number of used units by the expected number of units the behaviour will use.
4. If both tests succeeded,
   (a) Execute the corresponding behaviour
   (b) Reset the activation threshold of all the resources used, (reduce the amount of bound resource and start again)
5. Otherwise, reduce θ by $\Delta\theta$ and start again.

The decidedness is a simple parameterization of the behaviour based on previous action performing. Further details can be found in [34]

C. User interface design

The use of novel artificial intelligence tools for content creation purposes requires the development of specific user interfaces [37]. In this case, the goal of the user interface is to facilitate the design of SBN by non-technical users. For this purpose, we have adopted a visual layout that allows defining Perceptions, Resources, Goals and Skills with simple click and drag mouse made for such design. Two properties of SBN have motivated the design choices.

First, the fact that Perceptions are both inputs and outputs: in Skills they are part of propositions defining Preconditions and Expected Effects; in Goals they are part of Relevance Conditions and Goal Conditions. Since we want such connections to be established in the user interface by clicking and dragging between Perceptions, Goals and Skills, we choose to display , Perceptions as large horizontal bars (see figure 1). This has the following immediate practical advantages:

1. It is easy to visualize their truth value,
2. It greatly facilitates visualizing how this truth value is used as an input value (Preconditions in Skills, and Relevance Conditions in Goals).
3. It also facilitates visualizing how a change in the truth value of a given Perception also activate certain Skills (through their Expected Conditions) and Goals (through their Goal Conditions).

Second, the fact that in practice a given action will be performed if a cascading sequence of Skills is likely to achieve a certain Goal Condition with a high relevance state. For example, if the Goal Condition is to pick an object, a navigation Skill will be selected before the reaching Skill is triggered. It is therefore crucial to visualize clearly how Goals and Skills are selected in relation to the dynamic changes in the Perceptions' truth values.
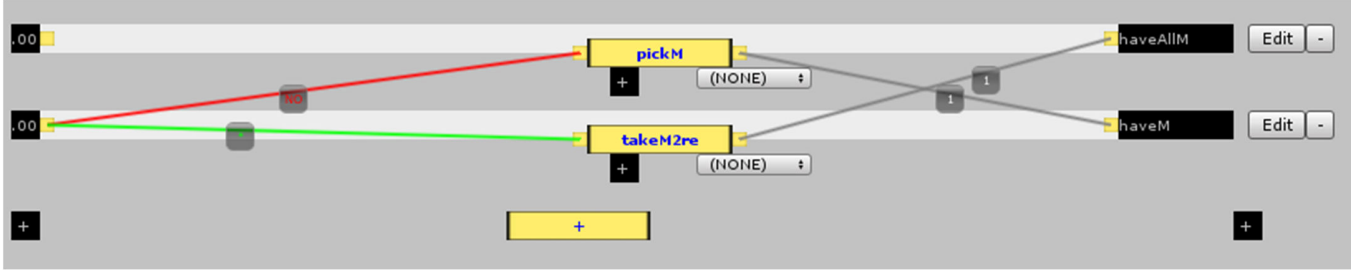
Figure 1:Skills can be defined with simple click and drag mouse input. For example, the box on top can be read as "If not haveM doing: pick M has effect haveM with likeliness 1". The lower one can be read as: "If haveM then doing: take M to Region has effect haveAllM with likeliness 1". Negating preconditions and effects, as well as changing the likeliness of the expected effects can also be done exclusively with mouse input. Goals work in a similar way.
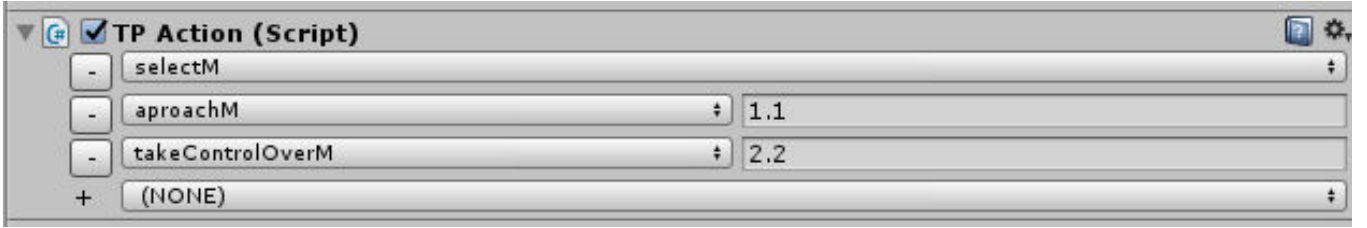

Figure 2: The behaviour triggered by a Skill can be defined through a drop-down menu, to select user-defined method (as are the perceptions)

Showing Skills in the middle of the horizontal Perception bars (see figure 1) allows visualizing the dynamic relations between Preconditions of a given Skill and Expected Effects of other Skills. Furthermore, in this user interface the activation values of the different Skills determine their horizontal position, and the decidedness of execution is also shown. This allows visualizing all the values of a Skill at a glance.

In addition to the two properties of SBN mentioned, the design of the user interface has been driven by its practical usage with mouse input. The definition of behaviours and Perceptions (see figure 2) can be done through dropdown menus, and adjusted introducing simple parameters. To facilitate visual inspection, Goals are visualized in a way very similar to Skills (see figure 1). The user can connect the Skills and Goals to add more preconditions or expected effects. Additional Perceptions, Resources, Goals and Skills can be added through the "+" boxes. Finally, the Resources used for each skill can also be defined through the dropdown menu below each Skill.

A significant advantage of such a user interface is that all the runtime parameters of the SBN can be visualized at execution time, together with the behaviour of the agent. This greatly facilitates the detection of differences between the desired behaviour, and the actual behaviour of the agent, at execution time (see figure 3).

In figure 1, we can see 2 Perceptions and 2 Skills defined for an agent that picks meteorites which fall from the sky and drops them in a given region. At execution time, the first Perception, *haveM*, is true when the agent has a meteorite. The second, *haveAllM* is never true, and it acts as a Goal Condition, endlessly pursued. This is the only motivation of the agent to behave. The first skill, *pickM* has as precondition *not haveM*, and an effect *haveM,* with a likeliness of achieving this effect that has value 1.

At execution time (see figure 3), the Perceptions are quantified, and the user can see their values in the black boxes, as well as in the horizontal bars. These Perceptions in turn determine the Goals' Relevance conditions *RCon*. The user is able to see these values easily within the Goal boxes, as well as the horizontal position of the Goal box. The relevant Goals further activate the Skills to achieve those, whose activation value is shown in the horizontal position of the Skill boxes. The determinedness with which the behaviour is executed is also shown, next to the name of the Skill.

D. Game engine integration

We take advantage of the reflexive capacities of C# to allow defining the Perceptions through drop-down menus, as shown in figure 2. The methods available in the dropdown menus can also be extended through scripting. In such a way, a programmer can define complex Perception and Action functions used to implement the Skills. On a second step, these can be used by the person designing the experience through simple drop-down menus, thus decoupling the sometimes complex process of developing Perception or Action functionality from the fast prototyping process of interactive behaviour design. In Unity3D, GameObjects are the essential building blocks of the virtual world. By defining SBN components as GameObjects, we allow these parts to be stored and easily combined to create new Perceptions, Resources, Goals and Skills (see figure 4). Once an SBN template is defined from existing Action and Perception methods, an agent using SBN can be created by simply selecting a body and a SBN template. At execution time, the SBN is instantiated in the agent, and the relations between Perceptions, Resources, Goals and Skills determine its behaviour.
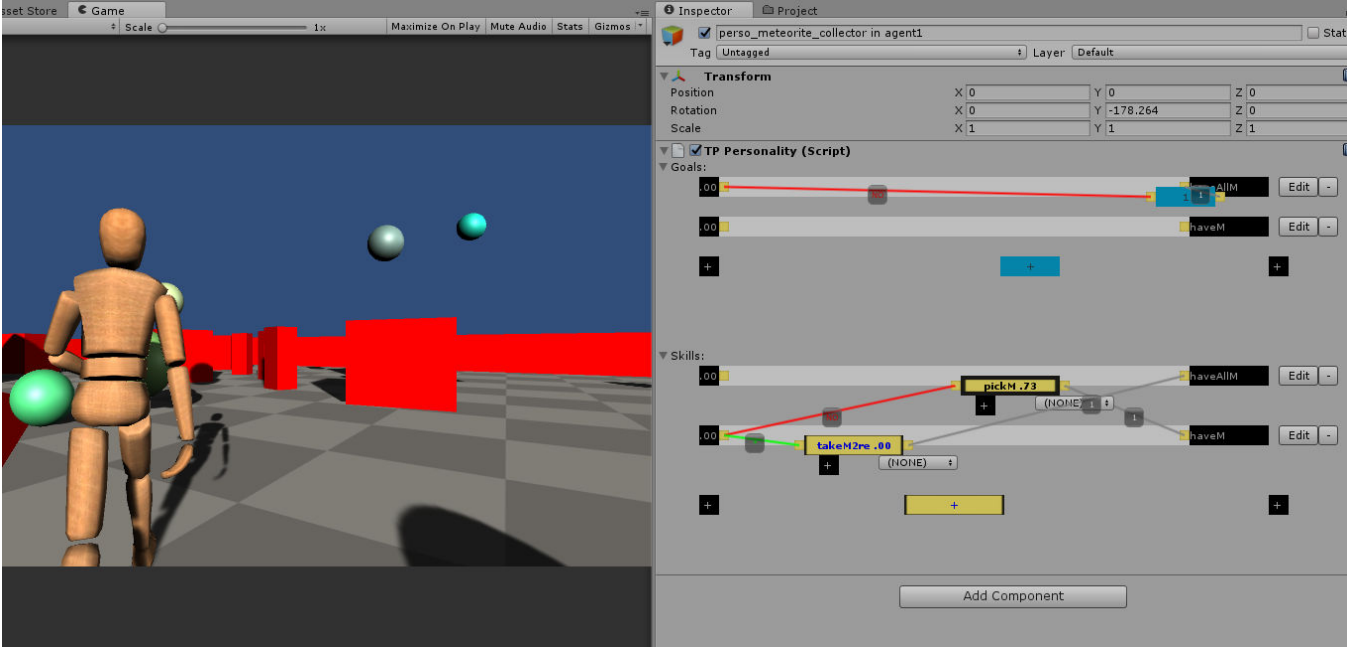
Figure 3: The user interface allows visualizing at execution time all the real-time parameters of an SBN, together with the behaviour of the agent.

The user interface shows the real-time evolution of the SBN and facilitates the identification of bottlenecks or mistakes in the design, thus contributing to adjusting the SBN in order to match the desired behaviour. In the supplementary material a video demonstrates the efficiency with which a novel SBN can be created.

IV. RESULTS

The current dominant practice in videogame character design is based on the combination of hierarchical state machines for behaviour synthesis with behaviour trees for decision-making. As an alternative, we propose SBN combined with ADAPT.

The usage of this software package involves two parts: the scripting of Perception and Action methods, and the creation of SBN using such components in networks of Goals, Skills and Resources.

Similarly to existing implementations of behaviour trees, this software package can be used within a game engine, they can be defined through a graphical user interface, and offer the possibility to cut, copy and paste, as well as dragging and dropping Perceptions, Resources, Goals and Skills to combine existing SBN into new ones.

We took extensive advantage of the functionalities of the Unity3D Application Programming Interface (API) to facilitate user adoption. Admittedly, the design choices we made aimed to give facilities to define SBN with point and click actions, already integrated with ADAPT, and be able to assemble instances of SBN from combining already existing Perceptions and Skills. The only programming skills needed are to define novel Perception and Action methods.

We have fixed the Parameters to these values:

- The activation γ takes a value of 1.5
- The inhibition δ takes a value of 1.5
- The inertia β takes a value of 0.2
- The threshold decay $\Delta\theta$ takes the value of 0.02, at 10% of θ_{max}.

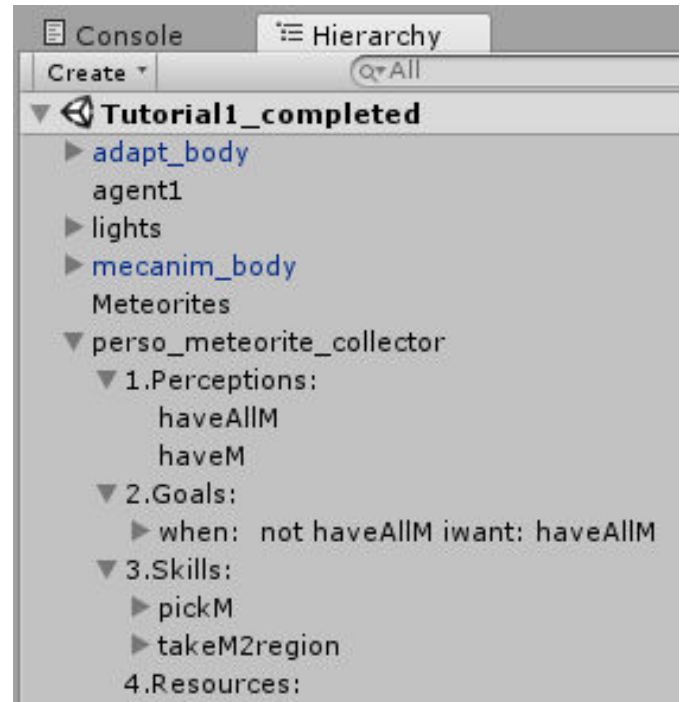The maximal threshold value θ_{max} is 0.2

Figure 4: The SBN is shown in the Hierarchical scene description, allowing for easy cut, copy and paste of components

The sigmoid function is defined with the following values:

$$\sigma(x) = \frac{1}{1 + e^{5(0.55-x)}}$$

We believe in practice such values work well, and there is no apparent reason to change them. For example, the 10% rule makes sure an iteration cycle is completed without too much computational load, but allowing for a desirable nuance in the decision-making process when several Skills are interacting to determine their activation weights.

In practice, running on a DELL PRECISION M3800 portable work station, without using any graphical processing acceleration for AI decision-making, a dozen of characters implementing simple examples of SBN with ADAPT embodiment could run simultaneously at interactive frame rates (>25 fps).

A trained user can create a novel SBN in less than 2 minutes (see video). This simplifies significantly the process of interactive character creation, and allows focusing the effort in the experience design and the implementation of good-quality animations for the agents.

V. DISCUSSION

A. Design choices

The combination of SBN and ADAPT allows creating interactive characters with methods and processes that seem adapted to a structured way of organizing content production. They also allow easy task separation. SBN, although based on EBN, introduce two architecture simplifications. First, we have fixated the Parameters. This has been done to further clarify the exact nature of the decision-making algorithm and release the burden of tweaking these from the experience designer.

Second, the use of conjunction, disjunction and negation operators to define relevance conditions for Skills [34] has been reduced in SBN to only the conjunction and negation operators. The main reason for this choice is that the use of behaviour networks is far from trivial. Reducing the amount of design options and parameters is helpful to focus on the more delicate aspects of the design process, such as appropriate balance between Skills, and introducing intensity weights in action selection. In addition, such features are not, or rarely, used in practice. For example, [34] never uses disjunctions, and in our experience with such architecture to prototype virtual reality content, the need for it has never appeared. Last but not least, removing the disjunction operator makes the definition of Skills and Goals more similar, which greatly facilitates the development of a visual layout such as the one introduced. It also facilitates learning to use such a system.

We have also chosen to not include certain features and extensions of EBN. For example, [38] expands EBN to include emotional factors, and [39] incorporates probabilistic decision biases characteristic of human decisions. These added further complexity in the design process, which we tried to avoid.

B. Principles of embodied cognition

The combination of SBN with ADAPT choreographers also facilitates the exploration of algorithmic versions of the embodied cognition principles previously outlined. This is particularly true if we compare it with traditional industrial methods such as Behaviour Trees and hierarchical state machines. We can now review the principles of embodied cognition to see how SBN combined with ADAPT fits with these principles:

1- Sensorimotor loops in SBN can make embodied cognitive agents deal practically with space, time, and changes, and even play football [40]. In addition, the sensorimotor loops inherent to SBN can be combined with machine learning methods to infer topological space and other representations through enactive mechanisms. This means that work such as described in [7], [41] can be encapsulated in a small set of SBN Perceptions and Skills.

2- As Maes [42] showed, *Situated agents can have goals.* Behaviour Networks are defined from a network of Goals and Perceptions, which in turn activate or not a network of Skills (there described as competence modules). This it at stark contrast with Behavior Trees.

3- Relevance measures appear naturally when an SBN agents' situation, or environment, changes [40]. In this context, relevance is formally defined as a situation-dependent importance measure of a Goal, given a certain situation. The SBN decision procedure is based on a utility-maximization mechanism on the domain defined by Goals. This seems quite aligned with the principle outlined by Relevance theory, although admittedly in a quite more primitive form.

4- Not more than 20 or 30 competence modules are needed to make a robot have the apparent will to play football, in the limitations of its body [34].

5- A distinguishing feature of Extended Behaviour Networks [34] is that the definition of Resources, in addition to Perceptions, introduces *de facto* a parallel inhibitory signalling pathway, complementary to the network of Skills and Perceptions. An additional advantage is that such connections are done with connectors that correspond to fuzzy propositions, and therefore allow certain understanding or interpretation of the designed connectivity.

6- Although to our knowledge it has not been systematically explored, the use of machine learning techniques can naturally be integrated in behaviour networks. Concretely, the probabilistic value of firing an output of a Skill is linked to an expected effect, which in a learning scenario is a probabilistic relation between a stimuli and a response, and it can easily integrate simple or complex statistical learning techniques through this reinforcement loop.

Overall, we believe that the combined use of SBN with ADAPT choreographers allows encapsulating different cognitive skills and learning processes in a way that can greatly facilitate their future integration in modular libraries, and further facilitate their adoption for the creation of smart interactive characters.

C. Future work

Further research would also be desirable to integrate contributions from different areas for automated skill synthesis, or automated connectivity optimization. Regarding automated connectivity optimization, Reinforcement learning methods appear as an immediate candidate for the adjustment of the expected effects likelihood values within Skills.

The idea of automated skill synthesis, although to our knowledge largely unexplored, seems consistent with emergent views on the encoding of behaviour in the motor cortex [43], particularly with the idea that motor synthesis is encoded in a set of skills, which synthesize behaviour based on domain competence, instead of a hierarchical or sequential view. The Combination of SBN with ADAPT choreographers should in principle allow creating Skills from raw data processed with modern machine learning techniques. However, the declarative nature of the connections and Skills in SBN would still allow maintaining a clear understanding of the domain which is encoded in each Skill, thus facilitating the design of the characters.

It would also be interesting to take advantage of the modular organization of Skills and implement one or several Skills from an existing implementation of Spoken Dialog Systems (SDS). As reported by Hirschberg et al. [44], although SDS now work fairly well in limited domains, where the topics of the interaction are known in advance and where the words people are likely to use can be predetermined, they are not yet very successful in open-domain interaction, where users may talk about anything at all. In such open-ended conversations, we could take advantage of the fact that SBN agents have explicitly stated situation-dependant Goals, and we could use these to simulate inferences about the Goals of other people in the conversation (within the space of the agents representations), bringing conversational decisions slightly closer to the two principles of Relevance Theory [12].

D. Practical usage

ADAPT has been used in different case scenarios for creating interactive content, and we believe SBN combined with ADAPT can further help creating those. This raises a more difficult, and practical, question: to what extent is this tool useful? After all, the divide between academic research in game AI and industrial production of game AI is still large [45]. First person shooters work very well, and create audiences of millions of enthusiasts. At least part of the merit must come from the behavioural intelligence shown by the interactive characters.

We believe that, in practice, it is a design choice. It depends on what an agents' Goals and Skills are, and how such Goals become apparent when interacting with human end-users. For an agent with simple motivations, it might be more effective to define a hierarchical tree of decisions and hierarchical state machines for behaviour synthesis. However, to design agents with more nuanced, situation-dependent motivations, or to integrate particular machine learning techniques for skill encapsulation, it might be beneficial to consider the use of SBN. Nowadays, the synthesis of behavioural intelligence is

relatively easy using consumer laptops and tools which are now available online, either for free or at a very modest price. However, the idea of using such tools for the synthesis of interactive behaviour seems surprisingly unexplored in the behavioural academic literature. We therefore make the result of our efforts publicly available [46] to facilitate the adoption by actual content creators within their endeavours.

REFERENCES

[1]   M. Slater, "Place illusion and plausibility can lead to realistic behaviour in immersive virtual environments," *Philosophical Transactions of the Royal Society of London. Series B, Biological Sciences*, vol. 364, no. 1535, pp. 3549–57, 2009.

[2]   B. Lenggenhager, T. Tadi, T. Metzinger, and O. Blanke, "Video ergo sum: manipulating bodily self-consciousness," *Science (New York, N.Y.)*, vol. 317, no. 5841, pp. 1096–9, 2007.

[3]   "Unity3D." [Online]. Available: www.unity3d.com.

[4]   E. Rosch, F. Varela, and E. Thompson, *The embodied mind. Cognitive Science and Human Experience*. The MIT Press, 1991.

[5]   D. C. Parkes and M. P. Wellman, "Economic reasoning and artificial intelligence," *Science*, vol. 349, no. 6245, pp. 267–272, 2015.

[6]   H. H. Ehrsson, "The experimental induction of out-of-body experiences," *Science (New York, N.Y.)*, vol. 317, no. 5841, pp. 1048–1048, 2007.

[7]   C. L. Nehaniv *et al.*, "Interaction and experience in enactive intelligence and humanoid robotics," in *2013 IEEE Symposium on Artificial Life (ALife)*, 2013, pp. 148–155.

[8]   Jacob, Pierre, "Intentionality," *The Stanford Encyclopedia of Philosophy*, 2014.

[9]   M. A. Halliday, *Spoken and written language*. Oxford University Press, 1989.

[10] M. Tomasello, *Constructing a language: A usage-based theory of language acquisition*. Harvard Univ Press, 2005.

[11] T. Bugnyar, S. A. Reber, and C. Buckner, "Ravens attribute visual access to unseen competitors," *Nature Communications*, vol. 7, 2016.

[12] D. Sperber and D. Wilson, "Relevance Theory," *Handbook of Pragmatics. Oxford: Blackwell*, pp. 607–632, 2004.

[13] D. Sperber, *Explaining culture: A naturalistic approach*. Blackwell Publishers, 1996.

[14] L. S. Vygotsky, *Thought and language*, vol. Rev. MIT Press, 1986.

[15] C. Assisi, M. Stopfer, and M. Bazhenov, "Using the structure of inhibitory networks to unravel mechanisms of spatiotemporal patterning," *Neuron*, vol. 69, no. 2, pp. 373–386, 2011.

[16] G. Buzsáki, *Rhythms of the Brain*. Oxford University Press, USA, 2006.

[17] A. Johansson and P. Dell'Acqua, "Emotional behavior trees," in *Computational Intelligence and Games (CIG), 2012 IEEE Conference on*, 2012, pp. 355–362.

[18] A. Shoulson, F. M. Garcia, M. Jones, R. Mead, and N. I. Badler, "Parameterizing Behavior Trees," in *Proceedings of the 4th International Conference on Motion in Games*, Berlin, Heidelberg, 2011, pp. 144–155.

[19] "Behaviour Designer." [Online]. Available: https://www.assetstore.unity3d.com/en/#!/content/15277.

[20] "Unity3D Mecanim State Machines." [Online]. Available: https://docs.unity3d.com/Manual/StateMachineBasics.html.

[21] "Unreal engine." [Online]. Available: https://www.unrealengine.com/.

[22] A. Shoulson, N. Marshak, M. Kapadia, N. Badler, and others, "ADAPT: The Agent Developmentand Prototyping Testbed," *Visualization and Computer Graphics, IEEE Transactions on*, vol. 20, no. 7, pp. 1035–1047, 2014.

[23] C. A. Dominguez, Y. Ichimura, and M. Kapadia, "Automated interactive narrative synthesis using dramatic theory," in *Proceedings of the 8th ACM SIGGRAPH Conference on Motion in Games*, 2015, pp. 103–112.

[24] M. Kallmann and M. Kapadia, "Geometric and Discrete Path Planning for Interactive Virtual Worlds," *Synthesis Lectures on Visual Computing: Computer Graphics, Animation, Computational Photography, and Imaging*, vol. 8, no. 1, pp. 1–201, 2016.

[25] M. Gillies, H. Brenton, and A. Kleinsmith, "Embodied design of full bodied interaction with virtual humans," in *Proceedings of the 2nd International Workshop on Movement and Computing*, 2015, pp. 1–8.

[26] M. Gillies and A. Kleinsmith, "Non-representational interaction design," in *Contemporary Sensorimotor Theory*, Springer, 2014, pp. 201–208.

[27] I. Kastanis and M. Slater, "Reinforcement learning utilizes proxemics: An avatar learns to manipulate the position of people in immersive virtual reality," *ACM Transactions on Applied Perception (TAP)*, vol. 9, no. 1, pp. 3:1–3:15, 2012.

[28] K. P. Körding and D. M. Wolpert, "Bayesian integration in sensorimotor learning," *Nature*, vol. 427, no. 6971, pp. 244–7, 2004.

[29] J. Kober and J. Peters, "Policy search for motor primitives in robotics," *Machine Learning*, pp. 1–33, 2009.

[30] D. M. Wolpert, "Probabilistic models in human sensorimotor control," *Human movement science*, vol. 26, no. 4, pp. 511–524, 2007.

[31] C. Lyon *et al.*, "Embodied Language Learning and Cognitive Bootstrapping: Methods and Design Principles," *International Journal of Advanced Robotic Systems*, vol. 13, p. 105, 2016.

[32] A. F. Morse, J. De Greeff, T. Belpeame, and A. Cangelosi, "Epigenetic robotics architecture (ERA)," *IEEE Transactions on Autonomous Mental Development*, vol. 2, no. 4, pp. 325–339, 2010.

[33] A. Drachen, A. Canossa, and G. N. Yannakakis, "Player modeling using self-organization in Tomb Raider:

Underworld," in *2009 IEEE symposium on computational intelligence and games*, 2009, pp. 1–8.

[34] K. Dorer, "Extended behavior networks for behavior selection in dynamic and continuous domains," presented at the ECAI Workshop on Agents in Dynamic and Real-time Environments, 2004.

[35] L. A. Zadeh, "Fuzzy sets," *Fuzzy Sets, Fuzzy Logic, and Fuzzy Systems: Selected Papers*, 1995.

[36] P. S. Goetz, "Attractors in recurrent behavior networks," PhD Dissertation, State University of New York at Buffalo Buffalo, NY, USA, 1998.

[37] S. Poulakos, M. Kapadia, A. Schüpfer, F. Zünd, R. W. Sumner, and M. Gross, "Towards an Accessible Interface for Story World Building," in *Eleventh Artificial Intelligence and Interactive Digital Entertainment Conference*, 2015.

[38] A. Johansson and P. D. Acqua, "Affective States in Behavior Networks," *Intelligent Computer Graphics 2009*, vol. 240, pp. 19–39, 2009.

[39] K. Dorer, "Modeling Human Decision Making Using Extended Behavior Networks," *RoboCup 2009: Robot Soccer World Cup XIII*, pp. 81–91, 2010.

[40] K. Dorer, "Behavior Networks for Continuous Domains using Situation-Dependent Motivations," presented at the International Joint Conference on Artificial Intelligence, 1999.

[41] M. Hoffmann, "Minimally cognitive robotics: body schema, forward models, and sensorimotor contingencies in a quadruped machine," in *Contemporary Sensorimotor Theory*, Springer, 2014, pp. 209–233.

[42] P. Maes, "Situated agents can have goals," *Robotics and Autonomous Systems*, vol. 6, no. 1–2, pp. 49–70, 1990.

[43] M. S. Graziano, "Ethological action maps: a paradigm shift for the motor cortex," *Trends in cognitive sciences*, vol. 20, no. 2, pp. 121–132, 2016.

[44] J. Hirschberg and C. D. Manning, "Advances in natural language processing," *Science*, vol. 349, no. 6245, pp. 261–266, 2015.

[45] G. N. Yannakakis and J. Togelius, "A panorama of artificial and computational intelligence in games," *IEEE Transactions on Computational Intelligence and AI in Games*, vol. 7, no. 4, pp. 317–335, 2015.

[46] "Behavior Networks Designer." [Online]. Available: http://u3d.as/K5s.
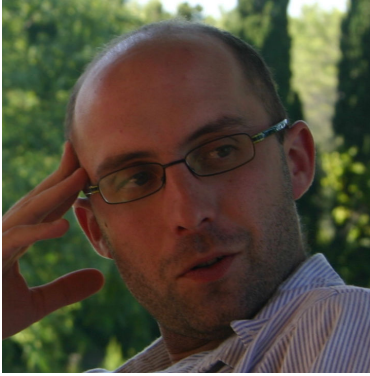
Joan Llobera is a Senior researcher at the i2cat foundation (www.i2cat.net). He is also the Founder of Timepath (www.timepath.io), a startup devoted to the creation of smart interactive characters for virtual reality experiences. He obtained a double diploma in Electrical engineering (Universitat Politècnica de Catalunya –UPC- and Télécom Paris) and holds 2 Masters, one in cognitive sciences (EHESS) and another in software (UPC). In 2012 he obtained a PhD at the Universitat de Barcelona on the topic of Stories in Virtual Reality. Between 2011 and 2015 he worked as a postdoc researcher at the Ecole Polytechnique Fédérale de Lausanne (EPFL), between the Laboratory of Cognitive Neuroscience and the Immersive Interaction Group. His mains fields of interest are Virtual Reality, Cognitive Sciences, and Content Creation.

Ronan Boulic (Senior Member 2005) received the Ph.D. degree in computer science from University of Rennes, France, in 1986 and the Habilitation degree in computer science from University of Grenoble, France, in 1995. He is currently a Senior Scientist with École Polytechnique Fédérale de Lausanne, Switzerland. He leads the Immersive Interaction Research Group. He has co-authored over 141 refereed publications, including 42 in ISI-indexed journals, and contributed to ten books. Dr. Boulic is a Senior Member of the Association for Computing Machinery and a Member of Eurographics. He has served on over 88 program committees of key conferences in computer graphics, computer animation, and virtual Reality.. His main fields of interest are 3D interactions, motion capture, modeling, and synthesis for virtual humans and robots, Virtual Reality, Embodiment