

Emergent Representations and Reasoning in Adaptive Agents.

Joost Broekens and Doug DeGroot

LIACS, Leiden University, Niels Bohrweg 1, 2333CA, Leiden, The Netherlands
broekens@liacs.nl, degroot@acm.org

Abstract

Classically, cognition assumes that the underlying mechanisms of thinking are based on symbol manipulation processes. This assumption has several drawbacks, such as the issue of where symbols (representations) actually come from. To overcome this drawback, the interactivist approach proposes that representations emerge from the continuous interaction with the environment and subsequent anticipation of such interaction. This approach provides understanding of the nature of knowledge representation and reasoning in adaptive agents. We have used the interactivist approach as basis for an interaction-based computational model. Our model is embedded in an adaptive agent whose task it is to find food in a maze. Feedback about the agent's success is given through a reinforcement signal that marks the current interactions. With this computational model, we show the actual emergence of representation as well as the emergence of what we call 'simple reasoning'. Finally, we discuss the relation between our approach and emergent artificial consciousness.

1. Introduction.

It is important to understand the nature of knowledge representation and reasoning in adaptive agents. This understanding is needed to efficiently solve problems related to the 'frame problem', to generalization versus specialization of knowledge, and to information retrieval from memory. In this paper we show how representations and a simple form of reasoning emerge from a computational model we have developed based on interactivist assumptions. Our model is embedded into a software agent that has to find its way through a maze in order to find food. Although the task is simple - i.e. being able to find food - we believe that the emergence of representation and reasoning shows how our model, as well as the interactivist approach, helps to understand the nature of knowledge representation and reasoning in adaptive agents. Additionally our computational model is easy to understand and implement. The remainder of this introduction explains the interactivist concepts that are relevant to the experimental results presented in this paper

as obtained with our current computational model. The next section describes our computational model related to the interactivist approach, followed by a section on the experimental results. We end this paper with an interpretation of the results and a discussion relating our approach to emergent artificial consciousness as well to two recent theories of consciousness, the global workspace theory [1] and the dynamic core model [6]. Finally, we discuss several directions for further research.

1.1. Interactivism and cognition.

Classical cognition assumes that the underlying mechanisms of thinking are based on symbol manipulation processes (see, e.g., [8]). One of the problems with this account of cognition is the symbol-grounding problem, where do symbols come from? This becomes specifically important in light of the goals and plans of agents. Who sets the goal, what is a plan, when is a goal detected as being achieved and how is a goal represented? It has been argued that answering these questions within the symbol manipulative paradigm results in infinite regress and cannot do away with the need for an external observer to imbue meaning upon the symbols [3]. Interactivist approaches to cognition (including some robotic approaches [4]) attempt to solve this problem by moving away from symbols as basis for representations. These approaches adopt the more naturalist view that continuous interaction between the agent and the environment is the basis for representations [2,3]. More precisely, this approach attempts to explain representations (and a large amount of other adaptive and intelligent phenomena) in terms of emergence of hierarchical levels of interaction complexity [2]. At the lowest levels, representations emerge based on potential interactions with the environment, while at the highest levels complex cognitive phenomena like consciousness emerge [2].

1.2. Interactivist concepts.

Several interactivist concepts are particularly important for the interpretation of our experimental results. These concepts are; interaction-potentiality (henceforth IP) because this is the building block of representations;

representations, because we argue that at least one emerges in our computational model; and reasoning, because we argue that a simple form also emerges in our computational model.

1.2.1. Interaction-potentialities. Any complex organism must solve the problem of action selection [3]. Action - or more precisely the potential to do something rather than something else - is a basic characteristic of all living organisms. Being able to make a selection presupposes the existence of multiple possibilities. These are called interaction-potentialities. This can be clarified with a concrete example. When a bee encounters two different flowers, it will choose to go to one of the flowers first. Being able to choose assumes that the bee has (at least) two potential actions that appear relevant in the current environmental context (e.g., go to flower one, and, go to flower two). These two actions are prepared by IPs, reflecting the potential but not yet effectuated status of the actions. Note that choice and selection are not meant as deliberative processes, but more so as the potential to act in one way rather than another as a result of interaction with the environment.

1.2.2. Representations. Although IPs can themselves directly prepare actions, they can also prepare other IPs, creating a hierarchy of growing complexity. This complex hierarchy is learned based on the continuous interaction with the environment, resulting in an interaction-based model of the environment. Characteristic of such a model is that it does not symbolically encode objects in the environment, but it encodes a complex web of potential (of potential of potential etc...) interactions with objects in the environment.

Two important elements of representations can be explained in terms of this web of IPs: content and truth-value [3]. Content refers to the fact that a representation is about something rather than something else, for example, this is an apple and not a green balloon. The set of prepared IPs is the current content of the representation. An apple allows for eating and cannot be deflated, while a balloon is inedible and can be deflated. Such differences define the content for a specific representation.

Truth-value refers to the fact that an environment can contain, for example, an apple or not. This is reflected by the fact that some IPs might be prepared, but when the actual action is performed, the environment reacts differently. There is a discrepancy between the potentialities and the actual outcome thus the representation was not true in its entirety. For example, if I bite in a balloon there is a bang, which would definitely not happen when biting an apple. Ergo, there was no apple.

When the outcome of a prepared IP differs from the actual outcome, there is a system detectable error [2]. This

automatic detection of discrepancy between the real environment and the predicted environment results in a destabilization of the active IPs, while a lack of discrepancy results in stabilization of the active IPs [3]. Thus an interactivist agent maintains a dynamic model of the environment by acting in that environment, which is computationally shown in [9].

1.2.3. Reasoning. Such a dynamic interactive model does not possess a separate 'reasoning module' interpreting the information present in the model. Reasoning in the context of interactivism is an emergent property of the model resulting from biases towards one interaction outcome rather than another. If the system prepares to act in two different ways, but for some reason one of these ways appears 'more attractive', then the system will eventually prepare the more attractive action and thereby make a choice. The question is how to actually bias this preparation. This can be achieved using reinforcement. A positive reinforcement signal is attached to the active IPs that prepared the beneficial action (beneficial, e.g., because the action favors survival of the agent). This results in a higher probability of the action being prepared again than before the reinforcement. In case of a negative reinforcement this effect is reversed. This bias could then be propagated through the web of related IPs resulting in an overall bias towards the situation that prepared for the beneficial action. In other words, decisions are made to arrive at a beneficial situation.

1.3. Hypotheses.

To verify that reasoning can be an emergent property of an interactivist representational system, we implement a computational model based on interactivist concepts and embed this model into a software agent. We first replicate the emergence of representation as shown in [9] (although we use a different computational model), and then evaluate whether the agent shows any emergent behavior that could be characterized as 'reasoning'. We define reasoning minimally, as "any observable behavior that reflects a beneficial decision between at least two possibilities that is neither explicable due to chance nor without representation".

2. Computational paradigm.

In this section we describe our overall approach and some of the important design decisions for the computational model we have used in our experiment. The underlying assumption is that the computational model considers both the agent as well as the environment of the agent as its external environment. This means that, for the model, there is no conceptual difference between action of

the agent and stimulus from the agent's environment. These are both called stimuli. This reflects the underlying idea that action and stimulus are two sides of the same coin.

2.1. Nodes and connections represent interaction.

The computational model consists of mainly a directed graph (Figure 1). A node encodes either a stimulus or an interaction. An interaction is a sequence of either two stimuli (e.g., node 'D-1' encoding for the sequence of stimuli 'D' and '1') or one interaction and one stimulus (e.g., node '(D-1)-D' encoding for '(D-1)' and 'D'). A node is called *primary* if it encodes for a stimulus and *secondary* if it encodes for an interaction. A secondary node has two incoming edges. The first, s_1 , connects from a primary or secondary node, and the second, s_2 , connects from a primary node. In Figure 1 the left incoming edges represent s_1 and the right incoming edges represent s_2 . Observe, for example in Figure 1c, the difference between node 'D-1' with a primary s_1 and '(D-1)-D' with a secondary s_1 . Every node has zero or more outgoing edges that are always connected to secondary nodes.

The model's behavior is explained next. A node in the graph either is active at time t (dark-gray nodes), was active at time $t-1$ (light-gray nodes) or is not active (white nodes). The graph is empty at initialization. Whenever a stimulus is perceived at time t , the node encoding for that stimulus is activated. If no node exists, the model automatically adds a new primary node to the graph (e.g. node 'D' in Figure 1a). Primary nodes and stimuli have a one-to-one relationship. Every primary node contains a copy of one unique stimulus, allowing the node to be activated based on a simple comparison function.

Every node that was active at $t-1$ is an s_1 (e.g., node 'D' in Figure 1b). The node that is active at time t is an s_2 (e.g., node '1' in Figure 1b). All secondary nodes connected to s_1 and s_2 are activated (e.g., nodes '1-D' and '(D-1)-D' in Figure 1c). If a secondary node should encode for the interaction between s_1 and s_2 but does not exist, the model adds this node to the graph and then activates it (e.g., node 'D-1' in Figure 1b). If an appropriate primary or secondary node exists, no new node is created (e.g., the dotted node 'D' in Figure 1c). Secondary nodes are logical nodes and do not contain copies of stimuli. In our model a secondary node is an IP. Such a node can be prepared by an s_1 without actually being active. When prepared, it predicts a potential next stimulus. This prediction can be confirmed by an s_2 , which is a node activated by a stimulus. Once active, the secondary node prepares other secondary nodes, enabling a growing hierarchy of interaction complexity. Due to the continuous flow of stimuli into the model, secondary nodes are created at increasing hierarchical levels (Figure 1c and d). To limit

the computational complexity of our experiment we have limited the maximum level.

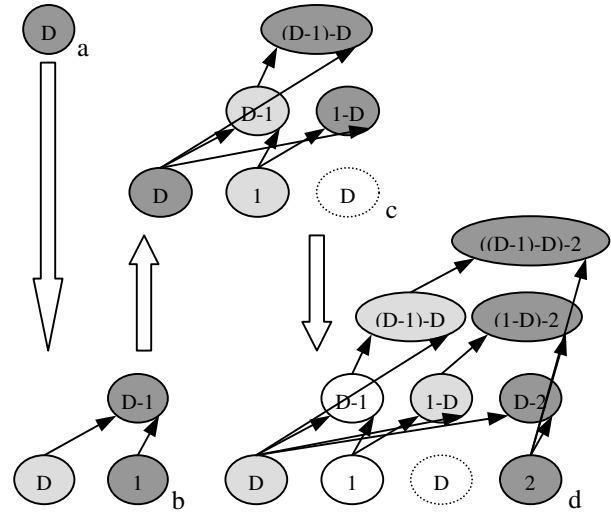


Figure 1a-d. Interaction hierarchy resulting from the sequence of stimuli 'D1D2' presented to an initially empty model at time $t=0$ to 3. 'D' is the action 'down', '1' and '2' are different cells in the maze. For further explanation, see text.

2.2. Model-learning and task-learning: exposure and reinforcement.

If a stabilization and destabilization mechanism is added to the computational model, it is capable of learning an adaptive model of the environment. This mechanism is implemented using a usage-measure, $f(v)$. If a node is activated, the v of that node is increased by 1. The function $f(v)$ calculates the usage of the node based on that node's v relative to the summed v s over all nodes. If v drops below a certain threshold the node is deleted (including all its dependencies). Consequently, consistent interaction with any part of the environment results in a stable sub-graph of secondary nodes. Inconsistent interaction results in the destabilization of the involved nodes and eventually in the deletion of these nodes. Apart from some difficulties related to 'forgetting' (like throwing away a useful, but seldom used node), this approach works well and is an intuitive and automatic mechanism for learning and forgetting.

A second way of learning - important for decision-making - is task-learning. Task-learning is not possible by pure exposure to stimuli, since a priori preferences do not exist in our model. We implement a form of reinforcement to bias IPs. This should be enough for emergent reasoning, as previously hypothesized. The reinforcement mechanism is as follows. First, when the agent acts in a way that favors survival, all currently active nodes are marked with a 'computational somatic marker' (see [5] for

a detailed account of somatic markers). This is the initial reinforcement. Second, upon activation of a node x , the markers $\mu_1 \dots \mu_n$ that are attached to the n hierarchically higher IPs that are prepared by node x , are used to update the marker v of node x . The effect $\mu_1 \dots \mu_n$ have on v , is relative to the usage $v_1 \dots v_n$ of the n prepared hierarchically higher IPs. For example in Figure 1d, if node 'D' is active, it prepares 'D-1' and 'D-2'. If these prepared nodes have usage 1 and 3 respectively, and both received a marker at some previous time step, the marker of node 'D' will be influenced 3 times more by the marker of node 'D-2' than by the marker of node 'D-1'. Markers are thus propagated back through the IP hierarchy only when the nodes to which they are attached are prepared. This propagation reflects the probabilistic properties of the environment since $v_1 \dots v_n$ actually is the learned distribution of the conditional occurrence of the n hierarchically higher IPs under the assumption that x is active. Third, the markers of all prepared IPs are categorized by s_2 , and then summed (in Figure 1d, if 'D' would be active, '1' and '2' are s_2 s resulting in two categories). Now, only one s_2 is randomly selected using these category-markers as distribution. The category with the highest marker therefore has the highest chance of being chosen. The result is a probabilistic winner-take-all selection of s_2 , which is the actual biasing mechanism. This learning approach enables credit-assignment that is local, not global, reflecting natural reinforcement principles of current activity.

3. Maze learning experiment.

In order to test our model, and the interactivist hypothesis of emergent decision making, we have developed a simple interactive environment in the form of a 'lava' maze, in which an adaptive agent is able to move up, down, left and right. Three different mazes are used. The first maze (Figure 2a-e) contains a starting location and two (variable) food locations, which the agent should learn to find. When arriving at a food location, the agent receives a positive reinforcement and is replaced at a starting location. Lava is a location with a negative reinforcement. The agent is thus able to walk through the lava but it is encouraged not to do so. The maze is constructed such that it allows the detection of decision-making; it contains a branch, leading to both the food locations. The second maze (Figure 2f) additionally contains a second (variable) starting location. The third maze (Figure 2g) is explained later.

3.1. Stimulus-action pairs: IPs.

In our experiment two types of stimuli exist: agent stimuli and environmental stimuli. Agent stimuli are up,

down, left and right and are called *actions*, while environmental stimuli are arbitrary names of the locations the agent visits, called *e-stimuli*. However, please note that our computational model considers actions and e-stimuli to be the same. The model proposes actions and the agent reacts by executing them. Every action at time t results in one e-stimulus at time $t+1$. Thus, the most primitive IPs consist of action-e-stimulus and e-stimulus-action pairs, while more complex IPs are built hierarchically upon these. The proposal of actions is biased based on the markers attached to the IPs, as described above. In the discussion we describe a more complex computational model that differs from this one and that may result in more complex, adaptive and intelligent behavior.

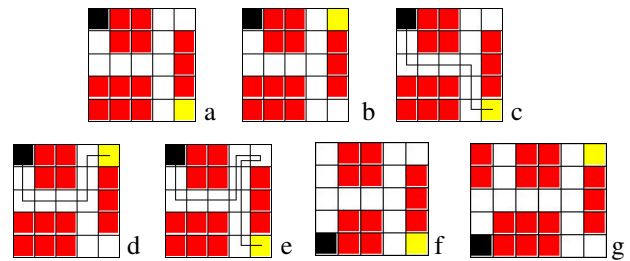


Figure 2a-g. Maze layout showing the two possible food and starting locations. Lava is red (dark gray), food is yellow (light gray) and the agent is black and at its starting location. See text for explanation.

3.2. Experiment setup.

In this paper we report on the results of six experiments. Every experiment is run 30 times. Every run consists of 100 trials. On every trial the agent attempts to find the food. To find a food location, the agent is allowed to use a maximum of 1000 moves per trial after which the agent is replaced at its starting location. The number of moves (interactions) needed to arrive at the food location is recorded per trial and averaged over the 30 runs, creating a learning curve (Figure 3a-f).

In the first experiment the agent learns the optimal route to the first food location (Figure 2c, 3a). In the second experiment - using the trained model from the first experiment - the food is moved to the second location. The agent first unlearns the first food location and learns the second food location (Figure 2d, 3b). In the third experiment, the agent starts with an empty model, and the food location is switched every trial (Figure 2c-e, 3c). This allows us to evaluate if the agent is able to learn both food locations. In the fourth experiment, the agent again starts with an empty model and the starting position of the agent is switched every trial but the food location is kept constant (Figure 2f, 3d). This is done to verify the ability of the model to cope with a varied starting position. In the fifth experiment the model is first trained to recognize the

route to the food location, just like in the first step. Second, after learning the route to the food position, the starting position of the agent is changed to a newly created part of the maze (Figure 2g, 3e). This test allows us to investigate the ability of the model to reuse existing knowledge while adapting to a new situation. The sixth experiment is an integrated test using both a varying food location and a varying starting location (using the maze in Figure 2f). Both the starting location and the food location is randomly changed, effectively creating a situation in which the 4 combinations of starting and food location can occur. Please note that to conclude convergence this experiment needs 200 trials per run (Figure 3f).

3.3. Results.

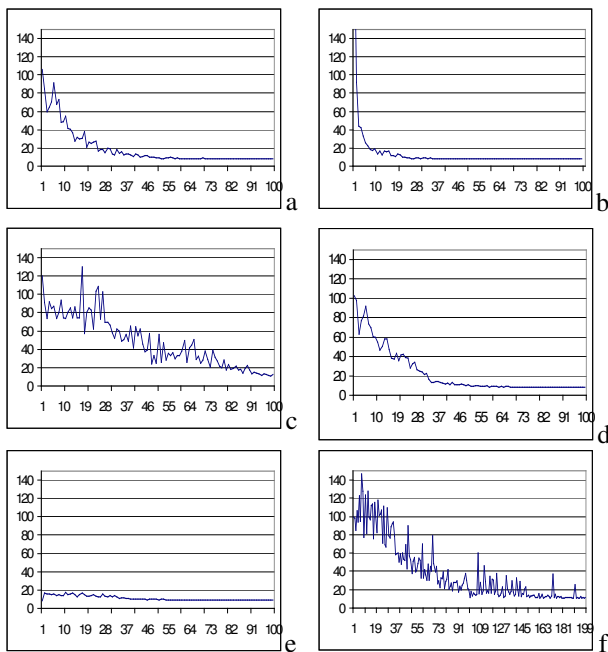


Figure 3a-f. Average learning curves (over 30 runs). Trials are on the horizontal axe; steps needed to get to the food are on the vertical axe.

Four main results have been observed. First, the agent learns to use the white squares only (and not walk on lava). When sidetracked, it learns to get back on these white squares as quickly as possible. Second, the agent learns the food location (Figure 3a), and is able to unlearn this location when the food is moved to the second location (Figure 3b). Third, after replacing the food to the second location, the agent first appears to be exploring its environment. Exploration in our case results from random movements when there is no strong bias towards a certain action. After an exploration period, the agent finds the new food location, and by repetition learns the optimal route to this new location. Interestingly, the optimal route to the new food location is learned significantly faster than to the old food location, even though the agent has more

difficulty finding the new location the first couple of times as a result of unlearning the old location. This phenomenon is shown by the high-starting but steep learning curve in Figure 3b (new food location) compared to the lower-starting but gradual curve in Figure 3a (old food location). This suggests the use of available knowledge about the lava and the first part of the route that is common to both food locations. This suggestion is supported by the result from experiment 5. Learning the optimal route to a known food location after being replaced to a newly created part of the maze is very easy. The flat learning curve shows direct reuse of the previously learned route and convergence to the optimal solution is quicker (Figure 3e). The fourth result is that the agent finds both food locations and is able to optimize the route to both locations (Figure 3c). In other words, the first time it passes the crossing it tries, for example, ‘up’, and when food is not found at the end of the route, it moves back and tries ‘down’. Even randomly changing both food and starting location proved to be converging to the optimal solution (Figure 3f).

4. Discussion.

In this section we show, based on the results, why we believe that our model develops emergent representation and a simple form of emergent reasoning. Furthermore, we discuss the relation between our computational approach and emergent artificial consciousness by relating our approach to two theories of consciousness, the global workspace theory [1] and the dynamic core model [6].

4.1. Using symbols but being non-symbolic.

Our use of ‘stimulus encoding’ might be associated with a symbolic approach. We use symbols as a basic abstraction level for the stimuli. However, we argue that using a symbol for a stimulus is not the same as using a symbolic approach to representation. Representations that are important for the agent are, e.g., ‘crossing’, ‘food’ and ‘no-food-here’. The symbols that are used for the stimuli do not encode this kind of information. These symbols are better viewed as environmental discriminators, used to encode that *up* does not equal *down*. Although the agent reacts in a specific way to *up* or *down*, *up* and *down* do not have any special status for the computational model. For the model these are exactly the same as, for example, *location_1* or *location_2* in the maze. The agent learns what it can do with a ‘crossing’, without explicitly representing a ‘crossing’. The meaning of a stimulus is mostly defined by the context (consisting of already active IPs), not by the symbol used to encode the stimulus. This context emerges in the model under the influence of stabilization, destabilization and reinforcement.

4.2. Emergent representations.

From a behavioral point of view the agent appears to learn a representation of the route to get to a food location. This is shown by the agent taking the optimal route to the food. The agent also learns a representation of 'path'. It knows how to stay on it and how to get back on it when sidetracked. These results suggest that representations of the environment emerge in our model. However, in our specific case, we could argue that this is a cognitive interpretation of behavior that can also be explained by pure reactive behavior initiated by the location the agent is at. Any location is unique, so the agent can easily learn what the single next action is to avoid getting a negative reinforcement. Following a route or using a path is just the concatenation of these reactive behaviors. Therefore, we do not consider the optimal route or 'path' per se an emergent representation.

A more potent candidate for the emergence of representation in our model is shown by the ability to try one food location and when nothing is found, choose the second food location when arriving at the crossing for the second time. To do this, the agent must have knowledge of a potential food location, a representation of food. The agent also has a fairly stable representation of 'food is not here', since it is able to walk back the same path it took and then choose a different action. This ability is shown even when both starting location and food location are randomly varied. Regardless of the complexity of the emerged representation (arguably it is not that complex), it does have two important elements of representations: content (food-up or food-down), and truth-value (food not up, but down). A second candidate for an emergent representation is shown by the ability of the agent to quickly find the food location when positioned in a newly created part of the maze. The model develops a representation of the old maze, usable in a new situation. Although our computational model and analysis is different, we interpret these results as a replication of the emergence of representation in interactivist models [9].

4.3. Emergent reasoning.

Although reasoning is often understood as a process that is much more intelligent than the one presented in the definition earlier in this paper, our definition is a necessary condition for reasoning. Reasoning essentially is about making an informed choice. This ability is not programmed into the model, which is empty at start-up. The ability to make an informed choice is shown by the selection of a different interaction the second time the agent passes the crossing. This selection depends on the representation of the non-location of the food.

Importantly, there is no reasoning algorithm that interprets this representation. Instead, the ability to make this informed choice is an intrinsic property of the biased IPs in the model. The selection mechanism responsible for this choice is based - as mentioned earlier - on the probabilistic winner-take-all competition over all currently prepared interactions.

4.4. Varied starting location.

The effect of a varying starting location on the difficulty of the learning task is minimal, assumed that parts of the optimal route to the food are common to both starting locations. Continuous switching of the starting location during one run results in a slightly slower convergence to the optimal route than no switching (compare Figure 3d and Figure 3a). Learning the optimal route to the food location from a second starting location - using a model that has been trained on the route from the first starting location - converges faster (compare Figure 3e and Figure 3a). Additionally it is constantly close to the optimal route during convergence (Figure 3e). This can be explained as follows. Continuously varying the starting location allows the agent to build up the common parts of the route without loss. Just the starting-location-specific parts of the route require more effort, which explains the slightly slower convergence. The second scenario - learning the optimal route to the food location from a second starting location - results in explorative behavior at first, until a known location is encountered. Subsequently, the agent uses its knowledge of the environment to find the food. This allows the model to learn multiple routes to the food quite efficiently, even when positioned in a completely unknown part of the environment. One potential problem might be that reuse of an older route hampers learning of a completely new but shorter route. This might happen when, for example, the new starting location is closer to the old starting location but the total route using the old starting location as waypoint is longer than a direct route to the food. In general, however, learning to cope with a varied starting location is a less difficult task to learn than a varied food location. Our experiments confirm this, as shown by the much steeper learning curve for the continuously varied starting location (Figure 3d) than the continuously varied food location (Figure 3c).

4.5. Artificial Consciousness: a huge step.

Conscious machines are thought to have advanced adaptive and intelligent capabilities. This might allow them to function in complex and uncertain environments. Additionally these machines could give important insights into the workings of natural consciousness. In the rest of

this discussion we argue that several characteristics of an enhanced (hypothesized) version of our computational model can be related to characteristics of two important models of natural consciousness. This might indicate that a computational model based on interactivist assumptions is an interesting direction for research related to artificial consciousness.

4.5.1. Configuration of the mental state. dynamic emergent structure. Our computational model is quite powerful. It develops based on zero knowledge and does not need a separate learning phase. Additionally, because only a subset of the model - consisting of activated and prepared IPs - is used at one moment, the model's computational complexity depends on the maximum level of the IP hierarchy, not on the total size of the model.

However, to discuss artificial consciousness we have to make several assumptions. First, we assume that an instance of the computational model with a rich developmental history exists (i.e., it is filled with a large hierarchy of IPs). Second, the model is able to process multiple stimuli at the same time. Third, the activity of IPs does not decay in a binary fashion (i.e., active or not), but gradually. Forth, the model does not run in lock-step with the environment. See the future work section for a short discussion regarding these assumptions.

Imagine an agent who has just been activated. Due to the continuous flow of stimuli into the model, secondary nodes (IPs) are prepared and activated at increasing hierarchical levels (compare Figure 1). At any moment in time interactions are selected, resulting in some behavior, creating new stimuli, which on their turn prepare and activate IPs. This results in a model that is 'configured' to be in a state that consists of a hierarchy of prepared and active IPs. This state is dynamic, depending on interactions with the environment; it is a dynamic emergent structure in the model. Such a dynamic emergent structure sheds light on one of the functions of consciousness: an active interpretive filter of stimuli. The interpretation *is* the context created by the active (prepared, decaying) IPs. This structure can be viewed as the 'mental state' of the agent or the content of consciousness. This emergent structure is also a computational analogy for the subjective nature of conscious experience and the common-sense notion that qualia are grounded in the flow of interactions with the environment. This model also predicts that phenomenal- and access-consciousness are more common than monitoring- and self-consciousness. The former two would depend on the existence of an information processing architecture that is able to host an emergent hierarchical state while the latter two would depend, additionally, on higher-level cognitive abilities. This model of artificial consciousness shows several

similarities with two important theories of natural consciousness.

4.5.2. Global Workspace Theory. Bernard Baars [1] has proposed the global workspace theory. The main characteristic of this theory is that brain processes (e.g. thoughts, images, associations, etc...) compete with each other to emerge onto the 'stage of consciousness', metaphorically described as a theatre stage. Unconscious processes try to get on the stage, through the recruitment of enough support from the audience (which is a collection of specialized unconscious processes). The audience listens to all processes on the stage, and to those trying to get on the stage. On stage, a process can influence the agent more directly because of the large impact of the information it broadcasts to the audience. Processes on the stage also have access to the information from the audience. Consciousness is thus the result of bottom-up and top-down competition between respectively unconscious and conscious processes.

Our hypothesized model works much the same way. First, stimuli (including internally generated stimuli, since the model does not run in lock-step with the environment) enter the model at low abstraction. When fitting the current context this flow of stimuli prepares and activates hierarchically higher IPs. Because these IPs are hierarchically high, they have a large amount of support from past interaction with the environment. This support consists of a whole hierarchy of IPs and has a large impact on the interpretation of future stimuli. This can be interpreted as a top-down force. Therefore, the context created by the active (prepared, decaying) IPs is an analogue for the theatre stage, in particular the hierarchically higher IPs that are part of the context. Second, IPs compete with others at all levels, because interaction with the environment restricts activation of certain IPs but not of others. This interaction-based competition is a strong bottom-up force.

4.5.3. Thalamo-Cortical loop. Edelman and Tononi [6] have proposed the dynamic core (DC) model. In this model stimuli are perceived and sent to the thalamus (the sensory relay station of the brain). The thalamus sends these stimuli to the cortex.

An important characteristic of the DC model is that reentry of perceived stimuli is achieved by a feedback loop from the cortex to the thalamus. This continuous process of reinterpretation of stimuli through a thalamo-cortical loop generates a dynamic core of brain activity that continuously interprets newly perceived stimuli. Architecturally the enhanced hypothesized model described in this discussion reflects these assumptions.

First, stimuli are perceived (thalamus) and prepare IPs (the cortex). Active IPs are stimuli for hierarchically higher IPs (reflecting the reentry of cortically processed

stimuli into the thalamus). Second, our dynamic state configuration is compatible with the concept of a dynamic core. Continuous stimulus interpretation shapes the dynamic core, which as a result is a continuously changing but relatively stable pattern, much like our hypothesized dynamic emergent structure. Last, both our model and the DC model view memory as a system property - not as a module or localizable function - in a way that is strongly analogous. Our model 'remembers' by gradually rebuilding the interaction hierarchy that characterizes a situation. This process of rebuilding is based on continuous interaction with the environment (or imagined interaction with the environment). This interaction activates hierarchically higher IPs, configuring the mental state. In the DC model, memory results from a process of continual recategorization. Recategorization is a result of reentry of cortically processed stimuli that blend with new stimuli, shaping the DC. Thalamo-cortical loop systems (or operationally similar ones) have been put forward as basis for artificial consciousness [7].

5. Conclusion

The interactivist approach to cognition is useful for the understanding of learning and problem solving in adaptive agents. Two predictions of the approach - the emergence of representation and the emergence of reasoning - have been verified experimentally using a computational model that is based strictly on interactivist principles. Lastly, by relating our computational approach to two theories of consciousness, we have shown why a computational model based on interactivist assumptions is an interesting direction for research aimed at artificial consciousness.

6. Future Work

Although promising, the computational model currently has important shortcomings if it is to be even a remote candidate approach to artificial consciousness. The overall problem with these shortcomings is that the solutions seem to dramatically increase computational complexity, since all involve more parallelism. We do not detail this problem here, but just point out several shortcomings of the current model.

Lock step: As mentioned in the discussion about consciousness, we have to assume the model is able to 'run' out of lock-step with its environment. This is necessary, because cognition is an autonomous process of the brain, influenced but not prescribed by stimuli. If the model is in lock step, an agent that uses it will for example have a tendency to wait for incoming events, which is clearly undesirable behavior.

One event per time-unit: Currently, only one event at the time can be fed into the model. This is a serious

drawback since, among other things, this results in the inability to form IPs that are constructed of a combination of events occurring at the same time. Thus, our model is very good at sequence learning, but not at all at learning Hebbian associations of events occurring together.

No decay: To be able to think about something that has not been encountered (e.g. for creative thinking and *deductive* reasoning), our model needs the ability to keep old IPs active for some time (even if these IPs proved to be not applicable to the environment). Currently, if the situation is such that a completely new event enters the model, the entire mental state (active hierarchy of IPs) is lost, since there are no nodes at all for this new event. This is a serious problem for the adaptive power of our model, and is biologically and psychologically implausible. Gradual decay of activity might solve this problem, as well as simplify reinforcement. The reinforcement signal (marker) does not need to be propagated back, but can simply be attached to IPs that are still active.

7. References

- [1] B.J. Baars. "In the Theatre of Consciousness, Global Workspace Theory, A Rigorous Scientific Theory of Consciousness." *Journal of Consciousness Studies*, 4 (4), 1997, pp. 292-309.
- [2] M.H. Bickhard. "Levels of representationality." *JETAI*, 10 (2), 1998, pp. 179-215.
- [3] M.H. Bickhard. "Motivation and Emotion: An Interactive Process Model." in *The Caldron of Consciousness*, R.D. Ellis and N. Newton (eds.), New York: John Benjamins, 2000, pp. 161-178.
- [4] R.A. Brooks and L.A. Stein. "Building Brains for Bodies." in *Autonomous Robots* (1), Boston: Kluwer Academic Publishers, 1994, pp. 7-25.
- [5] A.R. Damasio. *Descartes' error: Emotion, reason, and the human brain*. New York: G.P. Putnam, 1994.
- [6] G. Edelman and G.A. Tononi. *Universe of Consciousness: How Matter Becomes Imagination*, New York: Basic Books, 2000.
- [7] S. Jones. "On the Evolution of Artificial Consciousness." *Proc. of the Sixth European Conference on Artificial Life, ECAL*, 2001, pp. 427-431.
- [8] B.C. Smith. "Computation" in *The MIT encyclopedia of the cognitive sciences*, R. A. Wilson and F. C. Keil (eds.), Cambridge, MA: The MIT Press, 1999, pp. 558-560.
- [9] G. Stojanov and A. Kulakov. Interactivist approach to representation in epigenetic agents. *Proc. Third International Workshop on Epigenetic Robotics: Modeling Cognitive Development in Robotic Systems*, 2003, pp. 123-130.