The Extended Ramp Model:
A Biomimetic Model of Behaviour Arbitration for Lightweight Cognitive Architectures

Swen E. Gaudl[a], Joanna J. Bryson[b]

[a]*Falmouth University, Metamakers Institute, Penryn Campus, Treliever Road, Penryn TR10 9FE, UK*
[b]*Department of Computer Science, University of Bath, Claverton Down, Bath, BA2 7AY, UK*

Abstract

In this article, we present an idea for a more intuitive, low-cost, adjustable mechanism for behaviour control and management. One focus of current development in virtual agents, robotics and digital games is on increasingly complex and realistic systems that more accurately simulate intelligence found in Nature. This development introduces a multitude of control parameters creating high computational costs. The resulting complexity limits the applicability of AI systems. One solution to this problem it to focus on smaller, more manageable, and flexible systems which can be simultaneously created, instantiated, and controlled. Here we introduce a biologically inspired systems-engineering approach for enriching behaviour arbitration with a low computational overhead. We focus on an easy way to control the maintenance, inhibition and alternation of high-level behaviours (goals) in cases where static priorities are undesirable. The models we consider here are biomimetic, based on neuro-cognitive research findings from dopaminic cells responsible for controlling goal switching and maintenance in the mammalian brain. The most promising model we find is applicable to selection problems with multiple conflicting goals. It utilizes a ramp function to control the execution and inhibition of behaviours more accurately than previous mechanisms, allowing an additional layer of control on existing behaviour prioritization systems.

Keywords: Reactive Planning, Action Selection, BBAI, POSH, BOD, Bio-Inspiration, Modelling, Behaviour Design, Curiosity
2010 MSC: 00-01, 99-00

1. Introduction: Behavior Arbitration and Lightweight Cognitive Architectures

The mechanism described in this article addresses the issue of responsive and flexible action selection for behaviour-based AI [1, 2] or similar modular approaches to cognitive systems, where the system contains a set of potentially conflicting rival goals. We focus on intelligence for limited-performance systems, e.g. subsystems in a larger system which cannot simultaneously exploit the same resource. These systems face resource constraints such that they are not able to or not intended to use a fully-fledged cognitive architecture such as SOAR [3] or ACT-R [4]. Examples of resource restrictions include limited CPU cycles, low power consumption or restricted memory size. For both clarity on the type of problem we face and inspiration for its solution we look to nature. Arbitrating conflicting goals is an essential skill in animal behaviour as it heavily influences the fitness of an entity.

Take for example an antelope at a waterhole. The antelope lives in a hostile environment and water is a valuable resource because it is both needed and sparse. Inside the waterhole there is a group of crocodiles waiting for an animal to get close enough to be eaten. As the antelope needs to drink it has now two highly prominent and conflicting goals—survival by avoiding the predator and survival by drinking from the waterhole. If the antelope is not able to solve that situation by selecting one it would simply die in front of the waterhole.

Consider also the situation when the antelope is drinking and a predator emerges close to it. It is already pursuing an important goal—drinking to sustain living—yet it needs to make a decision as quickly as possible to escape the predator without hesitating or reverting back to the drinking behaviour, at least in the near term. Resource constraints in technology can be different, but still create conundrums.

Looking more closely at digital games, we find similar problems of control in a very different context. It is quite common to allow the AI only to occupy a small number of cycles per frame as most of the resources are needed for visual representations. It is arguable whether this is a correct choice, but it can be expected to be a given fact in most commercial products. Including a full-fledged cognitive architecture to control multiple cognitive agents in such an environment is in most cases not desirable as the cognitive architecture requires both more runtime resources and more work to design. The commercial game Assassins Creed Unity for example, where the player explores Paris during the French Revolution, features hundreds of agents moving around the city. In addition, designing the cognitive agents themselves is more time consuming than writing scripted agents, and extra time spent on the design the player might never encounter is not seen as worthwhile for the game

Email addresses: `swen.gaudl@gmail.com` (Swen E. Gaudl),
`jjb@bath.ac.uk` (Joanna J. Bryson)
URL: `http://swen.fairrats.eu` (Swen E. Gaudl)

developer. Game agents have in most cases the specific task to follow a specific designed role they are assigned to by the author. This creates a crafted experience similar to an actor in a theatre play. The agent is not to act entirely freely. In addition, once a game is conceived there is often considerable economic motivation to bring it to market quickly. Consequently, the main interest of game AI designers and engineers is to have flexible, modular tools for creating template agents, and then to modify those to create the desired character outcomes.

In robotic applications the need for easily-modifiable, lightweight but robust behaviour is driven not only by military research or humanoid robotics but also by commercial applications such as crop analysis and land scouting done by unmanned aerial systems (UAS). Again, in these applications, a large fully cognitive system is rarely needed but a lightweight, flexible and modular architecture to control basic functionality is essential. The AI for such systems is only responsible for a limited but important subspace of the problem at hand, e.g. to secure the safety of the robot and its surrounding operating environment. These underlying basic functionalities need to be fast and reliable requiring only minimal input. Brooks presents a solution to that problem with the SUBSUMPTION architecture [5] which allows the development of reactive agents utilising their embodiment. The benefit of reactive agents becomes visible when they utilise their embodiment, it can act as a form of memory or state, this provided a breakthrough in managing constraint resources [6]. In nature, a solution to faster action sequences can be found when habitual behaviours are employed which are learnt, fast, robust and require nearly no cognition.

The work presented here is motivated by an analysis of existing agent architectures and agent modelling environments for autonomous agents in digital games [7] and robotics. Existing cognitive approaches such as SOAR, ACT-R, LIDA [8] and CRAM [9] are extremely powerful, allowing the creation of sophisticated agents. However, due to the high complexity and steep learning curve they are seldom used outside of academic demonstrations and simplified problem spaces. Even where they are used, they are used primarily in communities strongly linked to an academic environment, such as military war games. When full cognitive reasoners or large expert systems are not needed or applicable, lightweight architectures and models such as BEHAVIOR-ORIENTED DESIGN (BOD) [10], Agile Behaviour Design [11] and BEHAVIOR TREE (BT) [12] or purpose specific architectures such as Pogamut [13], A Behavior Language (ABL) [14], and FAtiMA Modular [15] can be used. Purpose-specific architectures offer an optimized workflow for specific settings, reducing development time. Lightweight systems due to their lower additional computational cost and lower learning curve are generally more favored in non-academic application. These systems have to date been used most widely in the computer games industry, a substantial part of the contemporary economy that takes in more money than more traditional entertainment such as the film industry. Game AI requires agents that are able to act in real-time, can be instantiated quickly and leave the impression of human-like or animal-like intelligence. Lightweight cognitive architectures may be used either for individual agents or for swarms of shallow agents in a variety of digital environments (not only games), as well as for small autonomous robots such as Aibo or Roomba, or even substantial numbers of swarming robots [16, 17]. Due to the flexible nature of the applied approaches the resulting system can be tailored towards a specific scenario, reducing the computational cost drastically. This contrasts with most cognitive architectures which are intended as general problem solvers applicable to a wide range of problems. To better facilitate developers and researchers using lightweight architectures, and to enrich their action selection and behaviour arbitration mechanisms we have been examining bio-mimetic models. We developed a mechanism based on a accumulative ramp function [18], similar to the ramp activation found in the mammalian brain cells responsible for goal switching. The model is based on a system of dopaminergic cells in the Basal Ganglia of the mammalian brain [19, 20]. Our model is designed to be applicable to a broad range of systems. In keeping with lightweight architectures, it has a low computational overhead, making it highly versatile. The final model allows for an easy way to control the maintenance, inhibition and switching of high level behaviours in cases where static linear goal structures or predefined behaviour switching is undesirable for the action selection mechanism.

Action selection is of course a crucial part to embodied cognition, and even lightweight models such as those discussed here are of interest to philosophers of mind. First, evolution also favors lower costs, so sometimes insights into nature can be gathered from the experience of engineering, despite the differences in implementation between massively parallel biology and essentially sequential silicon [21]. Even models of consciousness do not necessarily need to be implemented at the neural level [22, 23]. Second, lightweight architectures operate at a level of abstraction more similar to most philosophy of mind, so are more easily comparable. Finally, more agile and accessible AI development allows more exploration of theories and their consequences. This allows philosophers and others in the social sciences and humanities another way to inform their intuitions about the consequences of their models [24]. Here we emphasize creating lightweight yet cognitively-motivated models to observe and perceive the world in sufficient detail to react or behave naturally. Our approach uses a functional representation to model phenomena which are expressed in a similar way in the neuronal structures of the mammalian brain. We care about and focus on the outcome of modules—both the actual brain structures and our functional modules—and aim at representing the functional outcome accurately rather than the actual underlying structure.

As computers get more powerful a shift towards more closely representing the underlying structures may be possible, although nature of course has had billions of years to create powerful structures such as the brain, and its integral host our body. In digital agents and robotics most of the organic structure evolved by nature is not needed to create similar outcomes so we focus on creating modules which are functionally similar instead of mechanistically similar.

The rest of the article is organized as follows. In the next section we describe our motivation for biomimetic modeling and introduce the extended ramp model and the ideas behind it. This is followed by illustrating an example domain where our approach can be tested and evaluated, in the same setting that was used by an already-published approach—Flexible Latching [25]. After presenting our ideas we discuss our results and intended future work, including a discussion of evaluation strategies which concludes the article.

2. Bio-Mimetic Models

In this section we discuss the bio-mimetic model and its implications. We start off by presenting our motivation for applying bio-mimetic concepts to action selection schemes and present our ramp-function arbitration mechanism. We take as a starting point for our research Flexible Latching [25], see Figure 1. Flexible Latching introduces a simple latch which reduces dithering—a rapid switching between goals which can mean that more time is spent in transition than in useful pursuit and consumption of single goals. Dithering between different behaviours is quite common in nature when multiple goals collide, expressed and observable as hesitation or thrashing. Unlatched behaviours may execute once immediately after their trigger condition is met, but stop quickly thereafter. The latch however acts similarly to a hysteresis function.

For the sake of an illustration imagine following example: a leaking canister which loses water over time. As soon as a lower threshold, say δ, is reached the canister is filled up to that threshold. It is easy to imagine that if you only refill up to δ whenever the water is below the threshold the time between each re-fill is relatively short.
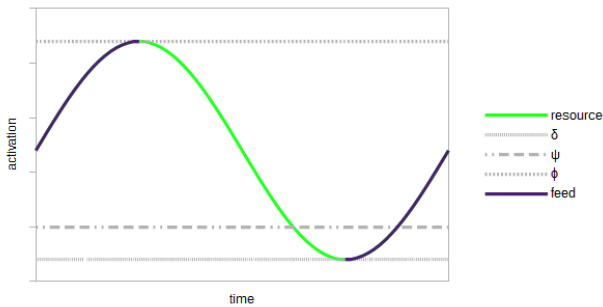

Figure 1: A Flexible Latch using two thresholds—δ and ϕ—to control dithering. In a simple latch, a goal can take control from when activation reaches the lower boundary δ until is reaches the upper boundary ϕ. Reaching ϕ, the goal is inhibited until activation falls below δ again. A flexible latch adds a third threshold, ψ, above which a latch is recomputed if the agent is interrupted. [25] find the best threshold for ψ to be $\psi = \delta$.

A strict latch now adds another threshold ϕ on top. Whenever the water now reaches δ you spend all your actions to refill the water level until it reaches ϕ. The exception is that if a higher priority goal triggers, it may control action selection, but when it is satisfied, control is returned to the latched behaviour. The inclusion of the strict latch allows extra time between ϕ and δ which can be spent on alternative actions. Flexible Latching

extends the strict latching by acknowledging interruptions and re-evaluating whether the current goal should still be pursued afterwards. [25] show that this is still more efficient, as the agents do not pursue goals that are no longer urgent nor currently convenient after the interruption.

Taking now a look at other selection processes inspired by nature, neural networks (NN) are the most prominent. Using a neural network it is possible to learn and solve selection tasks for problems where an algorithmic description of the problem is not known or costly. The NN is able to approach a general solution only by providing it with a set of specific, known input and solution pairs to adapt itself towards the solution space. However, for NNs the overall action selection or computational process is not transparent, thus tweaking a Neural Network to perform in a certain way is difficult. Additionally, NNs are normally trained to solve problems where sufficient and non-sparse training data is available rather than problems like action selection or real-time behavioural control in changing or untrained situations that offer no or sparse a priori training data. An example for a commercial game using a neural network is Black&White by Lionhead which used a neural net for training a few aspects of the player's pets intelligence. Recent success in Deep Networks and reinforcement learning in the games domain shows great promise to apply such techniques in special cases where the search space is either very limited or training data exists. For example the Atari general game playing environment is a popular testbed for neural network research [26] as it offers a vast variety of examples. However, approaches such as AlphaGo [27] or a General Atatri player using Deep Networks [28] require immense training time and data as well as a low-resolution, low-dimensional input, features which are increasingly difficult in real-time applications to complex environments.

For this work let us take a closer look at a single neuron model first as it reveals some interesting mechanisms which can be exploited in other contexts as well. There exists a variety of activation functions for neuronal models. Those include the spike or Dirac function used in Spiking Neural Networks (SNN), the sigmoid which has a fixed output range between zero and one, and the ramp function which combines a monotonic increased activation and an instant activation drop. This last forms the basis of the model presented here. Biomimetic models like NNs are an important asset of the computer science toolset. They present good and scalable solutions for addressing complex issues. We found that the ramp function is favored for goal arbitration [29], as we review in the following section. This finding motivates our present approach as we believe it to be a simple and elegant mechanism for augmentation.

3. The Extended Ramp Goal Model (Ergo)

Here we describe our biologically-inspired action selection mechanism, the Extended Ramp Goal model (Ergo), starting from its biological inspiration. Current research suggests, that dopaminergic cells in the Basal Ganglia of the mammalian brain are likely to be responsible for the maintenance and switching
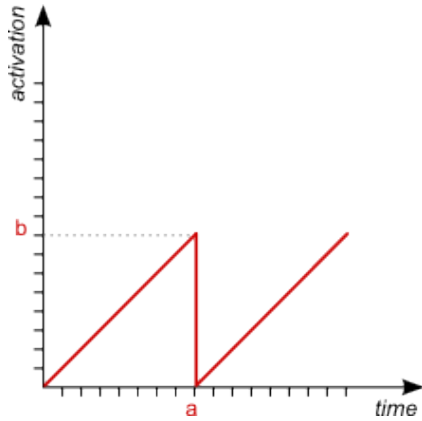
Figure 2: A single ramp function used for inhibiting a behaviour. A behaviour controlled by a ramp is only inhibited when another behaviour gains a higher activation or once the goal is reached. Once a behaviour completes it instantly drops activation. The behaviour completes its goal at time a, stopping at activation level b and dropping to zero activation.

of goals and thus behaviours [19, 20]. Clusters of those cells exhibit a ramp-like activation behaviour when in pursuit of a goal, which we will use as a basis for our current model, see Figure 2. The two important features of our ramp are the linear, strictly monotonic climbing of the activation and the instant activation drop exhibited when reaching the success criteria for the goal. These properties provide a predictable yet flexible mechanism.
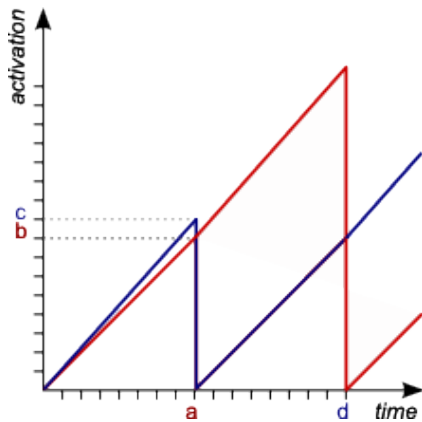

Figure 3: Two behaviours using ramps for action selection. From time t = 0 to t = a, the first behaviour is active having a higher activation due to the larger inclination. This is visible by the difference in activations b and c. At time t = a, the success criterion for the first behaviour is met and the activation drops resulting in a switch of the active behaviour to the second one. The second behaviour reaches its goal at time t = d allowing other behaviours to take over again.

We believe that the linear monotonic inclination minimizes the direct competition between behaviours. In our approach, we focus specifically on lightweight goal maintenance and switching. The approach is designed to give an active behaviour a higher inclination than all other behaviours which was inspired by [29]. Redish argues that the active ventral striatal anticipatory ramp cells (or often called goal cells in the literature) increase their firing pattern upon approaching a goal. The remaining ramp cells do not increase their firing pattern during that time. We abstract this cluster firing behaviour in our model and represent it as a larger inclination gain for the active cell and a lower one for non-active cells. As the climbing behaviour of the ramp is strictly monotonic and the inclination of active behaviours is larger, the difference between the active behaviour and the inactive ones grows, see Figure 3. Our representation of the different activation levels, Figure 3, is also backed up by the literature [20] as the anterior cingulate cortex (ACC) seems to change strategies based on evaluating possible reward levels.

Our approach using different inclinations makes it harder to dither between highly prioritized goals even when loosing focus for short period of time. The instant drop in activation automatically inhibits a behaviour from being executed again. Thus, the need for stopping behaviour dithering is reduced due to the higher level of the newly active behaviour which is a key element of our model. If multiple behaviours have an equally large activation and are primed for activation, one has to be chosen randomly. But this is only the case at the start of a new agent. Multiple behaviours could start with the same base activation and they are at that point all inactive in which case one has to be chosen. In nature this situation rarely occurs, one example would be being born and having to completely re-evaluate how to gain oxygen and nutrition. For our experiments, we initialized the resources of each agent using a random distribution within the bounds of δ and ϕ. This minimizes not only possible direct conflicts at startup time, it also provides more activity and liveliness to the simulation, which we discuss in the next section.

The latch is modelled using the internal motivational states of an agent, for example hunger or thirst. In contrast, the ramp uses an internal hidden tracker of its activation, a signal for urgent execution of the integrated behaviour, and a signal for when the behaviour achieved its goal. In this respect, the extended ramp model can also be used as a model of curiosity [30]. The model properties, such as the monotic inclination and the rapid activation drop, allow for modelling diverse curiosity such as the later described exploration. However, Ergo goes beyond that by offering a way to model specific curiosity [31], such as pursuing a specific goal until it is reached. This is possible as it is harder to disrupt an active behaviour due to the higher inclination and the constant gain. By augmenting specific behaviours the designer has the possibility to create curious drives for an agent. The signals our model listens to are dealt with asynchronously reducing the chance that behaviours block each other or the agent. Ergo's integration does not introduce such additional conflicts by waiting for a trigger signal. This allows us to run the computation of the ramp and its augmentation even on large sets of behaviours in parallel. In the basic system just described, the ramp is able to deal well with competing behaviours but it is not able to react quickly in a natural, believable way in urgent, changing situations.

To handle those rapid challenges we extend the original concept of the ramp. We introduce an exponential inclination gain whenever a behaviour needs to execute because of eminent danger or some other large risk. This extension defines our new mechanism, the Extended Ramp Goal (Ergo) model. The exponential gain in activation still takes a certain amount of time be-

fore triggering a behaviour switch to the behaviour which needs to execute. The resulting behaviour is similar to the phenomena observable in the mammalian brain, where it also takes a small amount of time for activation to spread between within a region before a task switch is triggered. This spreading activation before a behaviour switch can be found in situations when a change in dopamine levels happens or when a change and alignment in neuronal firing patterns happens. The time span between the trigger and the execution of the behaviour however is small.

Most related work [32, 33] apply the ramp function in the context of neural networks for goal maintenance. In contrast to those approaches, ours is the first attempt to apply a ramp-like criteria directly to a lightweight behaviour arbitration process without using a neural network to control the maintenance. Using Ergo to augment action should work without additional hand-tuning of its parameters, however, as the model can be interpreted as a multi-layer feed forward network which is possible by only looking at the ergo parameters. Having now a set of identical neural nodes it is possible to use any parameter optimisation technique or reinforcement learning to optimise the agent. However, as each node reflects altering a specific behaviour the network of ergo nodes can also be hand-tuned, which was a development focus. The next section presents an example domain in which we tested Ergo.

To summarize: Current research on the Basal Ganglia suggests that the goal maintenance in the mammalian brain is controlled by dopaminergic cells which in turn exhibit a ramp-like activation. Here we propose a new mechanism, Ergo, which extends the application of the ramp from neural networks into lighter-weight, more abstract action selection systems. We also modify the function so it can respond to urgent situations. An additional benefit of the model is that it also be used to model curiosity. In the next section we describe a test domain where multiple conflicting goals can arise for an action selection mechanisms. Natural agents from single cell paramecia to human beings face this situation constantly, for example should you continue reading this article or go write one of your own? Additional information on the agents' behaviours is presented to allow for a better understanding of the domain and the possible actions of an agent.

4. Example Domain

To develop and test our model, we chose Behavior Oriented Design (BOD) as our test-bed lightweight architecture. BOD allows the description of cognitive agents utilizing the parallel-rooted slipstack hierarchical (POSH) dynamic planning language. POSH includes a linear goal structure where each goal has a fixed priority with respect to the others, although each goal can be inhibited either by having unmet preconditions or through a system of scheduling. One reason POSH is well suited for our experiments is because it has already been fitted with a modification to this structure to allow more biologically plausible action selection. This mechanism is Flexible Latching [25] described earlier in Section 3. As a simulation environment we use the MASON agent-based modeling platform (Luke, et al.
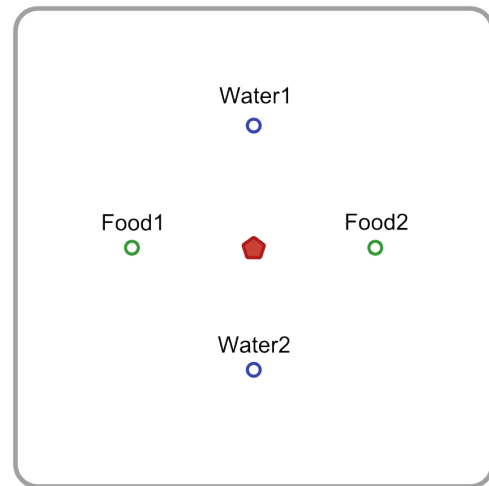

Figure 4: A representation of the simulation environment for testing the goal ramp model in a Mason agent simulation. The environment contains two food and two water sources equidistant from the center. All agents spawn at the center of the simulation at time t = 0.

2003) because of it offers a well-defined approach to agent simulation and provides an easy-to-use Java interface designed for agent based-modelling. The simulation environment is identical to Sim1 used by [25]. It contains two resource types, water and food, equidistant from the center of a map. All agents spawn at the center of the map, see Figure 4.

Agents can travel a fixed distance in any direction for every tick of the system clock. If they reach the edge of the map they enter at the opposite side again allowing them to travel from the left food source to the right food source by going left. Due to the layout of the map there is no benefit to either travel over the map edges or staying within the boundaries as the distances are exactly the same. It is also noteworthy that an agent cannot block a path, resource, or another agent in any way, which would be possible in nature but introduces unnecessarily complicated dynamics for the task at hand. Each agent constantly uses a certain amount of energy and water to survive, simulating natural metabolic costs and presenting the problem of self-sustenance. The amount of energy needed does not change during the simulation even if an agent does not move. If an agent's accumulated store of one of the two resources it needs drops to zero, then the agent dies. All agents are initialized within a lower boundary δ and upper boundary ϕ for the two resources. Whenever an agent is feeding from one the resources it gains energy. The gain is set to be larger than the consumption otherwise the agent would have no chance of surviving. For our setting the gain is set to ten times the consumption per tick. To allow the agent to track when it urgently needs to feed on a resource we make its intelligence sensitive to when its units of a specific resource drop below δ—an artificial threshold we use to model hunger.

Whenever the units reach the upper bound ϕ, the agent is programmed to detect that it has satisfied the need for that resource, so that it may distribute its time across other of its goals. The shortest path between one food and water resource requires an agent to spend approximately 10 units of both resources
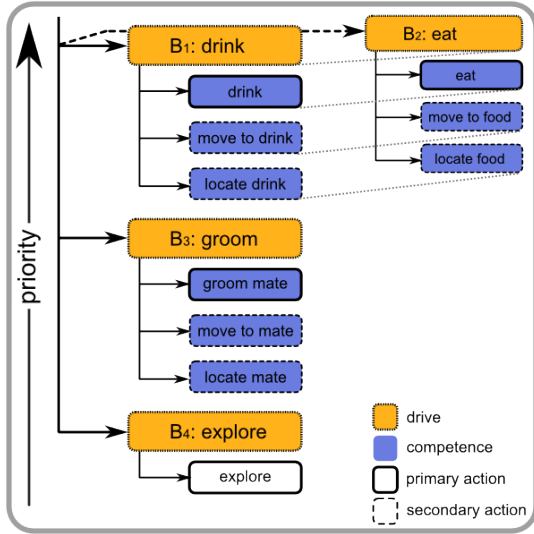
5

Figure 5: condensed view of a drive collection which specifies the behaviour of one of the agents in the simulation. The action plan contains four different behaviour drives which are prioritized from top to bottom. Drive B_1 and B_2 have the same priority, meaning they are equally important in general and their priority must be arbitrated in some sensible manner so both can be achieved.

which is the amount it can gain from feeding for one tick. For the present experimental setting we do not deplete the resources so the agent only needs to take care of consuming enough food and water to stay alive and pursue its other goals. A more realistic setting might include patchy, degenerating resources, but we are not introducing this level of complexity to our preliminary experiments. At any rate, optimal foraging theory is fairly well understood [34]. What we are researching here are lightweight mechanisms for goal arbitration. In Figure 5 a simplified version of the POSH action plan is shown which is used for all agents. Each agent has four drives which are prioritized based on each drive's position in the action plan. Each drive would satisfy a specific goal of the agent, for example in case of behaviour B_1 it is the need to drink. In POSH those goals are specified by internal or external senses, in this case the sense wants to drink. There is a special case which is behaviour B_4— the lowest-priority drive. The lowest drive should always be able to execute as it is treated as a default or fallback. If no drive can be executed the plan terminates and the agent stops and terminates as well. The behaviours B_1 and B_2 have equal priority indicating they are equally important for the agent— both are required for its survival. At this point we introduce our biomimetic augmentations to ensure that both drives are met in an efficient way, with neither dithering nor neglect.

5. Behavioural Outcomes

In this article we present a new biomimetic model—Ergo—based on research on the Basal Ganglia for augmenting existing action selection mechanisms [18]. The test-bed has been used to evaluate the implementation of the model against a comparable approach—Flexible Latching. This allows us to compare whether a new perspective on behaviour arbitration might be more intuitive and flexible than existing approaches. For our experiments, all behaviours use the same configuration of the extended ramp with the same inclination gain. Future work should involve optimizing the inclination gain based on initial priorities of the behaviours allowing a more fine grained approach to scheduling the arbitration process. To judge the quality of a well performing augmentation we introduce following evaluation criteria:

- time the agent remains alive

- time left for other behaviours

- robustness in face of noise and interruptions

- programmability

We ran a set of 50 independent trials per parameter to analyse their influence on the effectiveness of the augmentation and re-ran all simulation with the Flexible Latching model to have a direct comparison on the same system. We first analysed how well both approaches—Flexible Latching and Ergo—are able to handle non hostile environments.

In non-hostile environments both models perform well. In all trials for this setting, the latched and Ergo augmented agents remain alive being able to explore, feed, and groom. The Flexible Latch performed more consummatory actions in terms of the total amount of all performed actions. This is because the latch allows the agent to accumulate up to a certain amount of resources needed to survive and then spend the remaining time on lower priority behaviours. As grooming is harder to achieve than exploration, due to requiring a partner to groom, the latched behaviour is executing a large amount of exploration moves. In contrast to this the Ergo agent—augmented with the extended ramp—is trying to spend the extra time on all behaviours evenly. This resulted in reducing the amount of exploration moves drastically, while increasing the amount of grooming. As grooming is an activity with is only possible under conditions, yet the agent still tries to pursue it as much as possible not concentrating on other activities it can also be seen as a behaviour employing specific curiosity [31]. The agent searches for partners to groom, moves towards on of them and then focuses on that activity. Under less hostile conditions the Flexible Latch is performing equal or better than Ergo in the total amount of actions. Ergo however only responds to agent signals such as goal reached or urgent—minimizing the interdependence between agent and selection mechanism— which affects the complexity and robustness of the software design. It does not optimize free time as efficiently as the scenario-optimized Flexible Latch. We currently have not specified problem dependent parameters in Ergo to allow for a better integration into generic action selection mechanisms. Adjusting Ergo after integration of its vanilla state would provide a significant performance boost but needs extra time which we on purpose wanted to minimize. Moving however from non-hostile environments towards an increasingly hostile and noisy setting, and given the low amount of resources the agent can store, the results change drastically. As we introduce disruptions and interrupts into the agents' environment, the static latch agent has a large action

count drop, even by introducing only one interruption per behaviour. The Flexible Latch agent outperforms the agent with a static latch, verifying the findings of [25], yet it is now far less effective than Ergo. The Flexible Latch with a single interruption already drops a third of its consummatory actions. In contrast, the increased noise and hostility of the environment makes the advantages of Ergo visible. Continuing the pursuit of a set goal when faced with distractions is an important feature of employing specific curiosity as it allows the agent to concentrate its attention on the task in front of it. The whole range of executable behaviours scales in a way that allows more agents to stay alive, active and lively—demonstrating the whole range of possible behaviours. The Ergo action activity only drops by a tenth compared to the noise-free setting. The more interruptions we introduce the bigger the impact on the death rate of the latched agents whereas a significantly higher percentage of the Ergo augmented agents remain alive. Setting the environment to more hostile settings shows that the ramp adapts the agent quite well to the harsher conditions and benefits from the exponential influence of the urgency on the inhibition process. The exhibited behaviour is similar to focusing on a given task or under stress. This means that the focus is harder to lose and distractions normally get ignored to guarantee that this essential task the animal is focusing on can be achieved. A technical discussion of the model and a statistical analysis can be found in [18], however, further work is needed to better analyse the influence of noise and interruptions on Ergo augmented behaviour and verify our findings. A focus will also be on identifying the impact inclination gain and the exponential modifier for the gain on the robustness of generic lightweight agents.

6. Conclusion

Our current results indicate that the ramp function outperforms the Flexible Latching in certain scenarios even when it is not hand-adapted for the specific problem. It offers some unique features which offer additional interesting potential, such as an approach to modelling curiosity. The ramp acts only upon a small set of signals and needs less fine-tuning to perform well. It has an easy-to-understand visual representation of the maintenance and inhibition process presenting what we believe to be an intuitive approach. Future work is needed on the influence of finer grained prioritization and the influence of the ramps specific inclination gain for particular behaviours. This would allow us to better understand if Ergo can scale well to a variety of different problems. Additionally, future work is needed to see if an augmented agent with multiple Ergo instances can be automatically tuned through an interpretation of Ergo nodes as a multi-layer neural network.

More generally we have discussed the ability of lightweight cognitive architectures and their importance in a variety of domains. Due to their lower computational overhead and high practical applicability it is our belief that research on advancing those architectures deserves more attention. This research could provide fruitful results for example for the games industry by expanding beyond the current capabilities of agent design

and architectures, but also to robotics, and scientific and philosophical simulations. To deepen our understanding of challenges in those domains we are working on a project to apply our lightweight augmentation to the domain of digital games and are aiming to experiment with perceived behaviour selection and how different users compare selection processes in virtual agents. For this we developed a prototypical smartphone game for android phones. To return to our earlier discussion of the relationship between research in lightweight architectures and the philosophy of mind, we hope that our discussion has engaged the interest of the reader and made our point for us. Architectures like those presented here, including the currently-popular spreading-activation architectures such as the Global Workspace Theory (GWT) [35, 36] and LIDA cannot account for all of action selection [25]. We know that many details of action selection are handled by other, simpler neurological mechanisms in real primates, and computer science gives us a good reason why: combinatorics [21]. But goal selection—the focus of interest—is a differentiable sub-part of action selection overall, one that requires competition between all of what may well be a limited set of contenders. We hope the work presented here will be useful in a large variety of applications, including perhaps better understanding Nature itself.

7. Acknowledgements

[1] Brooks R. A robust layered control system for a mobile robot. Robotics and Automation, IEEE Journal of 1986;2(1):14–23.

[2] Bryson JJ. Cross-paradigm analysis of autonomous agent architecture. Journal of Experimental and Theoretical Artificial Intelligence 2000;12(2):165–90.

[3] Laird JE, Newell A, Rosenbloom PS. Soar: An architecture for general intelligence. Artif Intell 1987;33(1):1–64.

[4] Anderson JR, Bothell D, Byrne MD, Douglass S, Lebiere C, Qin Y. An integrated theory of the mind. Psychological review 2004;111(4):1036.

[5] Brooks RA. Intelligence without representation. Artificial Intelligence 1991;47(13):139–59.

[6] Nolfi S. Power and the limits of reactive agents. Neurocomputing 2002;42(1):119 –45.

[7] Grow A, Gaudl SE, Gomes PF, Mateas M, Wardrip-Fruin N. A methodology for requirements analysis of ai architecture authoring tools. In: Foundations of Digital Games 2014. Society for the Advancement of the Science of Digital Games; 2014,.

[8] D'Mello SK, Franklin S, Ramamurthy U, Baars BJ. A cognitive science based machine learning architecture. In: AAAI Spring Symposium: Between a Rock and a Hard Place: Cognitive Science Principles Meet AI-Hard Problems. AAAI; 2006, p. 40–5.

[9] Beetz M, Mösenlechner L, Tenorth M, Rühr T. Cram – a cognitive robot abstract machine. In: 5th International Conference on Cognitive Systems (CogSys 2012). 2012,.

[10] Bryson JJ. Intelligence by design: Principles of modularity and coordination for engineering complex adaptive agents. Ph.D. thesis; MIT, Department of EECS; Cambridge, MA; 2001. AI Technical Report 2001-003.

[11] Gaudl SE. Building robust real-time game ai: Simplifying & automating integral process steps in multi-platform design. Ph.D. thesis; Department of Computer Science, University of Bath; 2016.

[12] Champandard AJ. Understanding behavior trees. http://aigamedev.com/open/article/bt-overview/; 2007. last visited: 3. June 2017.

[13] Gemrot J, Kadlec R, Bída M, Burkert O, Píbil R, Havlíček J, et al. Pogamut 3 can assist developers in building ai (not only) for their videogame

agents. In: Agents for Games and Simulations. No. 5920 in LNCS; Springer; 2009, p. 1–15.

[14] Mateas M, Stern A. A behavior language for story-based believable agents. Intelligent Systems, IEEE 2002;17(4):39–47.

[15] Dias J, Mascarenhas S, Paiva A. Fatima modular: Towards an agent architecture with a generic appraisal framework. In: Emotion Modeling. Springer; 2014, p. 44–56.

[16] Chaimowicz L, Kumar V. Aerial shepherds: Coordination among uavs and swarms of robots. In: Distributed Autonomous Robotic Systems 6. Springer; 2007, p. 243–52.

[17] Rubenstein M, Cornejo A, Nagpal R. Programmable self-assembly in a thousand-robot swarm. Science 2014;345(6198):795–9.

[18] Gaudl SE, Bryson JJ. Extended ramp goal module: Low-cost behaviour arbitration for real-time controllers based on biological models of dopamine cells. In: Computational Intelligence and Games (CIG), 2014 IEEE Conference on. IEEE; 2014, p. 1–8.

[19] Cools R. Chemical neuromodulation of goal-directed behavior. In: [37]; chap. Search, Goals, and the Brain; 2012, p. 111–25.

[20] Brown JW, Nee DE. Executive control of cognitive search. In: [37]; chap. Search, Goals, and the Brain; 2012, p. 69–80.

[21] Bryson JJ. Modular representations of cognitive phenomena in AI, psychology and neuroscience. In: Davis DN, editor. Visions of Mind: Architectures for Cognition and Affect. Idea Group; 2005, p. 66–89.

[22] Franklin S, Baars BJ, Ramamurthy U. Robots need conscious perception: A reply to aleksander and haikonen. In: NEWSLETTER ON PHILOSOPHY AND COMPUTERS; vol. 09. APA; 2009, p. 13–5.

[23] Bryson JJ. A role for consciousness in action selection. International Journal of Machine Consciousness 2012;04(02):471–82.

[24] Whitehouse H, Kahn K, Hochberg ME, Bryson JJ. The role for simulations in theory construction for the social sciences: Case studies concerning Divergent Modes of Religiosity. Religion, Brain & Behavior 2012;2(3):182–201.

[25] Rohlfshagen P, Bryson JJ. Flexible latching: A biologically-inspired mechanism for improving the management of homeostatic goals. Cognitive Computation 2010;2(3):230–41.

[26] Bellemare MG, Naddaf Y, Veness J, Bowling M. The arcade learning environment: An evaluation platform for general agents. Journal of Artificial Intelligence Research 2013;47:253–79.

[27] Silver D, Huang A, Maddison CJ, Guez A, Sifre L, van den Driessche G, et al. Mastering the game of go with deep neural networks and tree search. Nature 2016;529(7587):484–9.

[28] Mnih V, Kavukcuoglu K, Silver D, Graves A, Antonoglou I, Wierstra D, et al. Playing atari with deep reinforcement learning. CoRR 2013;abs/1312.5602.

[29] Redish AD. Search processes and hippocampus. In: [37]; chap. Search, Goals, and the Brain; 2012, p. 81–96.

[30] Berlyne DE. Curiosity and exploration. Science 1966;153(3731):25–33.

[31] Loewenstein G. The psychology of curiosity: A review and reinterpretation. Psychological Bulletin 1994;:75–98.

[32] Stewart TC, Bekolay T, Eliasmith C. Learning to select actions with spiking neurons in the basal ganglia. Frontiers in Neuroscience 2012;6(2):1–14.

[33] Velsquez JD. Modeling emotion-based decision-making. 1998.

[34] Stephens D, Couzin ID, Giraldeau L. Ecological and behavioral approaches to search behavior. In: Cognitive Search: Evolution, Algorithms and the Brain. MIT Press; 2012, p. 25–46.

[35] Baars BJ. The conscious access hypothesis: origins and recent evidence. Trends in cognitive sciences 2002;6(1):47–52.

[36] Shanahan M. A cognitive architecture that combines internal simulation with a global workspace. Consciousness and cognition 2006;15(2):433–49.

[37] Todd P, Hills T, Robbins T. Cognitive Search: Evolution, Algorithms, and the Brain. Strüngmann Forum reports; University Press Group Limited; 2012. ISBN 9780262018098.