

Improved Animal-Like Maintenance of Homeostatic Goals via Flexible Latching*

Philipp Rohlfshagen

School of Computer Science, University of Birmingham, Birmingham, B15 2TT, United Kingdom

Joanna J. Bryson

Department of Computer Science, University of Bath, Bath, BA2 7AY, United Kingdom
The Konrad Lorenz Institute for Evolution and Cognition Research, A-3422 Altenberg, Austria

Abstract

Artificial cognitive systems sometimes neglect the impact on action selection of natural durative-state mechanisms like emotions and drives. These chemically-regulated motivation systems assist natural action selection through temporarily focusing an agent's behavioural attention on particular problems. This can improve efficiency by avoiding dithering, but taken to extremes can be inefficient in ways that seem cognitively improbable for mammal-level intelligence. This article demonstrates a flexible latching method that provides appraisal-based sensitivity to interruption, allowing reassessment of the current focus of attention. This drastically improves efficiency in handling multiple competing goals at the cost of a surprisingly small amount of extra cognitive complexity.

Introduction

The term "action selection" might imply cognition, but this is merely due to anthropomorphic labelling. If we take *cognition* to be a process requiring time (probably a form of search), and *action selection* to be any mechanism for determining the present course of action, then much of action selection is really non-cognitive. Our action choices are limited both by evolution and individual experience; many of them may be essentially reflexive. Such limiting is necessary if action selection is to be achieved in a timely manner (Simon 1972; Chapman 1987; Gigerenzer & Todd 1999). However, there is no question that humans and other species do engage in cognition in some contexts. This paper proposes and examines one such context: the arbitration between different goals.

In the simulation of animal behaviour, artificial agents are usually designed to interact with their surroundings, including other agents, and to carry out some set of tasks. In particular, it is often the case that agents are required to ensure their survival. Besides more dramatic aspects, e.g. fending off predators, this requires agents to locate, approach and consume sources of energy. Furthermore, there are often

additional tasks that need to be carried out, such as maintaining a social network through grooming group-mates. All these behaviours require both time and energy, and it follows that agents possessing more efficient behaviour management should, in general, fare better than other agents with less efficient behaviour selection.

In the work presented here, our agents must not only ensure that consummatory behaviours (i.e. feeding) have a positive energy intake (that is, the energy spent carrying out a particular action must be less than the energy acquired) but they must also have the ability to store excess energy in order to pursue auxiliary behaviours. These auxiliary behaviours include social behaviours, which are also motivated by survival-oriented drives similar to those for feeding, as socialising promotes long-term survival by facilitating group living (Dunbar 1993; Korstjens, Verhoeckx, & Dunbar 2006).

The context of our cognitive task is a comparative study of three different mechanisms for allowing agents to satisfy their needs. We focus in particular on a potential inefficiency that may occur when an agent attempts to acquire excess satisfaction. We propose that if an agent is interrupted at any stage during this period, a choice needs to be made whether or not to continue with the current activity or whether to attend to other, possibly more relevant behaviours. Persistence avoids the inefficiency of *dithering* between two goals, while some degree of flexibility avoids the inefficiency of pursuing a goal excessively. We assume biological motivation systems have evolved to manage this trade-off. Here we do not attempt a perfect model of such a system. Instead we present and evaluate a simple control mechanism that achieves this at a minimal cognitive cost. We use a basic latching system augmented with the ability to detect potentially relevant interruptions. This triggers a reevaluation of priorities already present in the agents' action-selection system. Although quite simple, this system is related to at least one theory of conscious attention (Norman & Shallice 1986). We return to this last point in the discussion.

Method

In this section we first describe the agent architecture we use to test our system. The results are general, but the architecture is described for clarity. We then describe the latching mechanisms we have implemented and the testing scenarios.

*Research funded by the British Engineering and Physical Sciences Research Council (EPSRC) under the Adaptive and Interactive Behaviour of Animal & Computational Systems programme, grant number GR/S79299/01.
Copyright © 2008, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

Following sections present our results and discussion.

Basic Action Selection

The agents are specified using the behaviour-oriented design (BOD) methodology (Bryson & Stein 2001), a system that produces layered or hybrid agents consisting of (a) modules that specify details of their behaviour and (b) dynamic plans that specify agent-wide, cross-modular prioritisation. Action selection is carried out using parallel-rooted, ordered slip-stack hierarchical (POSH) dynamic plans (Bryson 2003).

We chose BOD as a fairly simple example of an architectural consensus achieved in the late 1990s for real-time situated systems: that AI is best constructed using a combination of modularity, for providing intelligent primitives, and structured hierarchical plans, for encoding priorities (Hexmoor, Horswill, & Kortenkamp 1997; Kortenkamp, Bonasso, & Murphy 1998; Bryson 2000). Even mainstream cognitive architectures such as Soar and ACT-R can be described in this way (Laird & Rosenbloom 1996). Although somewhat neglected in academia in the last decade due to an emphasis on learned action selection, similar architectural threads can be seen in contemporary games AI programming with the adoption of behaviour trees to extend the expressivity of finite state machines. The details of the structured action selection system are unimportant to the mechanism presented in this paper. All that is assumed is:

1. some mechanism for storing temporary values of long-term state (e.g. learning),
2. some mechanism of expressing a variety of goals and their associated actions, and
3. the notion of a trigger or precondition as part of the mechanism for choosing between goals and actions.

A single POSH plan was used to specify the priorities of all agents tested here. The plan, shown in Figure , assumes four basic behaviours: B_1 to B_4 . The top level of the plan hierarchy (the *drive collection*) is checked at every time step, and the first element whose trigger is true is executed. All but behaviour B_4 further contain a sub-plan (called a *competence*) also containing elements each with their own trigger. Competences maintain decision memory and control behaviour until they either terminate, pass control to another child competence, or the main drive collection takes control back for a higher-priority problem.

The first two behaviours, which are of the highest priority, fulfill consummatory needs, such as eating or drinking, the neglect of which would cause the agent to die. Behaviours B_3 and B_4 are of lower priority and are only considered for potential execution if B_1 and B_2 are not triggered. Behaviour B_3 represents grooming (but could equally well be mating) which requires two agents to interact with one another. A final behaviour (B_4) serves as a default behaviour and should always be triggerable (exploring in this case).

The primary focus of this investigation is on behaviours B_3 and B_4 . Lower priority behaviours may only be executed if all higher priority behaviours are managed efficiently and although these behaviours are non-essential to the survival

of the agent as an individual, they may significantly impact the survival of the species as a whole: consummatory behaviours are of highest priority as they ensure the survival of the agent. Behaviours such as breeding, on the other hand, are necessarily of lower priority, despite their significance to the group of agents as a whole. It is therefore paramount that these higher-level behaviours are managed efficiently enough to allow agents to pursue other behaviours as well.

Each behaviour is composed of numerous elements, some of which may be classified as *secondary actions*. In the case of feeding, the secondary actions would be ‘locating food source’ and ‘move towards food source’. The *primary action* would correspond to ‘eat’ and it is that action that should be executed most frequently to ensure an efficient execution of the behaviour. *Dithering*, the rapid switching between goals, for example, results in helper actions being performed excessively. Each behaviour B_i has one such primary action which will be denoted as B_i^α . In summary, it is not only desirable to be able to follow all behaviours, but, at the same time, the execution of primary actions should be maximised, and that of helper actions minimised. The frequency at which primary actions are executed is thus at the centre of our investigation.

Agents and State

Each behaviour B_i is associated with a module, which includes a single-valued internal state E_i . Here, for the sake of clarity and without loss of generality, we use the concept of energy to denote the internal state of the agent: each behaviour B_i has a current level of energy E_i . The agents live in a toroidal, discrete-time world with dimensions of 600×600 pixels. At every time-step, all energy states E_i are decreased by e_i^- and, given a behaviour is vital to the agent’s survival, death is imminent once $E_i \leq 0$. For each behaviour, we define a threshold δ_i such that B_i is triggered once $E_i < \delta_i$. Once B_i is triggered, the agent executes the actions associated with that particular behaviour. The behaviours B_1 and B_2 , for example, correspond to consummatory activities (eating or drinking): the agent first locates an energy source, moves towards the energy source (at a speed of 2 pixels/time step) and consumes the source once in close proximity. This consumption raises the agent’s internal state by e_i^+ . Clearly we must ensure that $e_i^+ \gg e_i^-$, \forall_i as otherwise an agent would never be able to satisfy a need (and in the case of essential behaviours, the agent would eventually die). We have chosen the same values for all behaviours: $e^+ = 1.1$ and $e^- = 0.1$ and hence drop the subscript from here on. This gives a net energy gain of $e^\pm = 1$.

All lower-priority behaviours (i.e. B_3 and B_4) may only be executed if B_1 and B_2 are satisfied. What it means for a behaviour to be “satisfied” depends upon the implementation of the agents’ action selection — the basis of this paper, described next.

Conditions

We use three different action selection mechanisms and evaluate their impact on the energy management of the agent: unlatched, strict latch and flexible latch.

```

(
(SDC life (goal (s_one_step (s_succeed 0)))
(drives
((dead (trigger((s_is_dead 0))) a_stay_dead))
((drink (trigger((s_wants_drink))) c-drink) (eat (trigger((s_wants_food))) c-eat))
((groom (trigger((s_wants_to_groom))) c-groom))
((explore (trigger((s_succeed))) a_explore))))

(C c-groom (goal ((s_succeed 0)))
(elements
((has-no-target (trigger((s_has_groom_target 0))) a_pick_groom_target))
((not-near-target (trigger((s_is_near_groom_target 0))) a_move_to_groom_target))
((default-groom (trigger((s_succeed))) a_groom_with_target))))

(C c-eat (goal ((s_succeed 0)))
(elements
((has-no-food (trigger((s_has_food 0))) a_pick_food))
((not-near-target (trigger((s_is_near_food_target 0))) a_move_to_food))
((default-feeding (trigger((s_succeed))) a_eat))))

(C c-drink (goal ((s_succeed 0)))
(elements
((has-no-drink (trigger((s_has_drink 0))) a_pick_drink))
((not-near-target (trigger((s_is_near_drink_target 0))) a_move_to_drink))
((default-feeding (trigger((s_is_near_drink_target))) a_drink))))
)

```

Figure 1: The POSH plan that determines priorities for the agents: the *drive collection* (SDC) is called at every time step and its elements checked in order: $\{B_1=\text{eat}, B_2=\text{drink}\}, \{B_3=\text{groom}\}, \{B_4=\text{explore}\}$. The first element whose trigger is true is executed. Equal priority elements (i.e. B_1 and B_4) are checked in random order. Primitive actions start with ‘a.’, primitive senses with ‘s.’, subplans start with ‘c.’.

Unlatched As mentioned in the previous section, a behaviour B_i is triggered if $E_i < \delta_i$. In this basic, unlatched, model, the drive terminates as soon as $E_i \geq \delta_i$ and the time spent at the energy source is expected to be relatively short (although this is not necessarily true in the case of multiple equal-priority behaviours). Furthermore, no excess energy is stored and the behaviour is triggered again very shortly after it is satisfied¹. When there are multiple such behaviours, the agent will continue to oscillate between them (*dithering*). Even if there is only a single top-priority behaviour, the agent will spend its entire time in close proximity to the energy source as the acquired energy is always insufficient to pursue anything else.

Strict latch In the latched models, the agent only terminates the drive once $E_i \geq \phi_i$ where $\phi_i \geq \delta_i$. Now, the agent has an energy reserve of $(\phi_i - \delta_i)/e^-$ time steps before the behaviour is triggered again. If all high-priority drives are latched in this way and the latch is sufficiently large (see next section), the agent is able to eventually follow lower-priority drives. This form of latching is very inefficient, however, if the agent inhabits a world where unexpected interruptions may occur: if an agent is almost finished with one activity but gets interrupted, the agent will continue to pursue this activity independent of other, lower priority needs. For example, an agent that is grooming and whose partner has left, might pursue another partner for five minutes when only an-

other five seconds of grooming would satiate it. This is true even if $E_i = \phi_i - \epsilon$ where $\epsilon \ll \phi_i - \delta_i$ and hence this form of latching is referred to as strict.

Flexible latch If the agent is able to detect interruptions, the interruption could trigger a decision that determines its subsequent activities. This decision could be conscious, but here we simply relax the latching by using yet another threshold, ψ_i , that is situated in-between the previously two established ones, $\delta_i \leq \psi_i \leq \phi_i$. This gives rise to three different scenarios: the interruption occurs when

- $E_i < \delta_i$, the drive remains ‘unsatisfied’
- $\delta_i < E_i < \psi_i$, the drive still remains ‘unsatisfied’
- $\psi_i < E_i < \phi_i$, the drive is, at least temporarily, ‘satisfied’

Here we consider two types of interrupts. The first type occurs when the source of satisfaction is depleted or otherwise removed (e.g., an agent loses his current grooming partner). The second type of interrupt is caused by higher priority drives that are triggered.

Threshold Selection

The previous section has discussed numerous different thresholds that require initialisation and the choice of parameters is crucial to the outcome of the simulation. First, it should be noted that the flexible latch is simply a generalisation of the strict latch, which in turn is a generalisation of the unlatched technique:

$$\text{Flexible latch} \quad \delta \leq \psi \leq \phi$$

¹The theoretical maximum possible excess energy in this case given the values of e^+ and e^- is 0.9 which will last for 9 time steps.

$$\begin{aligned} \text{Strict latch} & \quad \delta \leq \psi = \phi \\ \text{Unlatched} & \quad \delta = \psi = \phi \end{aligned}$$

In this investigation,, we have two primary points of interest, which are closely related: survival and efficiency. The survival of the agent crucially depends on the choice of δ . Efficiency, on the other hand, refers to the agent’s ability to pursue all its behaviours, not just high-priority ones, and depends on the choice of ϕ . In order for an agent to survive, any vital behaviour must be triggered such that the agent has enough energy to approach the energy source (locating an energy source can be done in a single time-step and is subsequently excluded from the following discussion):

$$\delta_i \geq \mathbb{E}_i^r \quad (1)$$

where \mathbb{E}_i^r is the energy required to reach the source: $(d_{max}/d_{mov}) \times e^-$, where d_{mov} is the distance an agent can move in a single time step and d_{max} is the maximum possible distance an agent can travel². If there are n equally vital behaviours, δ_i has to be adjusted accordingly:

$$\delta_i \geq \sum_{j=1}^{n-1} (\mathbb{E}_j^r + \mathbb{E}_j^c) + \mathbb{E}_j^r \quad (2)$$

where \mathbb{E}_i^c is the energy required to raise the energy level to the appropriate level:

$$\mathbb{E}_i^c = \frac{\psi_i - E_i}{e^\pm} \quad (3)$$

The value of ϕ , on the other hand, has to be set such that enough energy is stored to pursue all vital needs:

$$\phi_i \geq \delta_i + \sum_{j=1}^n (\mathbb{E}_j^r + \mathbb{E}_j^c) \quad (4)$$

Any excess energy is subsequently devoted to the other, lower-priority behaviours. This choice of ϕ_i necessarily affects \mathbb{E}_c as now more time is spent at the energy source (a difference of $\phi_i - \delta_i$). Interruptions drastically alter \mathbb{E}_c and the energy required to satisfy a latched behaviour given m interruptions is simply:

$$\mathbb{E}_i^c = \sum_{j=1}^m (\mathbb{E}_{ij}^r + \mathbb{E}_{ij}^c) \quad (5)$$

At each interruption, the agent should, in theory, decide whether it is worth pursuing the currently executed behaviour (i.e. if there is a positive or negative energy ratio). Usually there is insufficient knowledge available to make an informed decision due of the complexity or indeterminacy of the environment. Nature selects for agents with appropriate or at least adequate thresholds; here, in our particular simulations, we test a range of values for ψ .

²The theoretical maximum in this case is simply $\sqrt{width^2 \times height^2} \approx 424$ and it would take the agent a maximum of $424/2=212$ time steps to reach the target, consuming $212 \times 0.1 = 21.2$ units of energy.

Experiment and Simulation Details

Our experiments are organised into two sets. The first set uses *sim1*, a very well defined setup that allows a great degree of control over all aspects investigated, particularly frequency of interruption (see Figure 2a). The second set use *sim2* (Figure2b), a more realistic simulator where interruptions are caused by the dynamics of the environment itself.

In both simulations, there are 5 identical agents. Furthermore, *sim1* positions the energy sources such that they are maximum distance from one another³. In this simulation, we exactly control the number of interruptions an agent is exposed to throughout the execution of a single behaviour. Once an agent is interrupted, it is forced to consider an alternative energy source (it is not allowed to remain at the current one).

The second simulation is more realistic and is used to verify the results obtained from the first set of experiments. In *sim2*, energy sources are scattered randomly across the world. Each energy source has a certain load that depletes as an agent consumes it. Once depleted, the energy source vanishes, but, at the same time, a new energy source appears elsewhere in the world. The load of any energy source has a maximum of 50 units and depletes by 2 units if consumed. All energy sources gain 1 units per time step.

The experiments are executed over 15 distinct trials. Each trial executes the simulation for 5000 time steps. All internal states are initialised such that $E_i = \delta_i$, thus all behaviours are triggered immediately once the simulation begins. At each time step, the agent may execute a single action. The results are simply the number of times each primary action has been executed, averaged over all agents and trials. In all cases, a two-tailed t-test is used to test for significance with a confidence of 99.995%. We chose the same threshold settings across all behaviours and again, we drop the subscripts from here on. Furthermore, we set $\delta = 200$ in all experiments, giving an agent sufficient energy for $200/e^- = 2000$ time steps before the behaviour is triggered again.

Results

In this section, we will first present the results from the controlled environments, *sim 1*, followed by a comparison to the results obtained in the more realistic settings, *sim 2*.

Controlled Environment: *Sim1*

The first experiment compares the unlatched version with the strictly latched one. The results are shown in Table 1. The data confirms that in the unlatched case, dithering prevents the agent from pursuing any of the lower priority behaviours. The latch effectively solves this problem, although only if the latch is sufficiently large. A latch of size 10 does increase the activity of behaviours B_1 and B_2 but does not allow for the lower-priority behaviours B_3 and B_4 to be executed. Once the latch increases in size, so does the activity of the lower-priority behaviours. This result is not surprising and the data indicates that larger latches are indeed preferential.

³The simulation is toroidal and agents are able to move, for example, from the far left to the far right in one move.

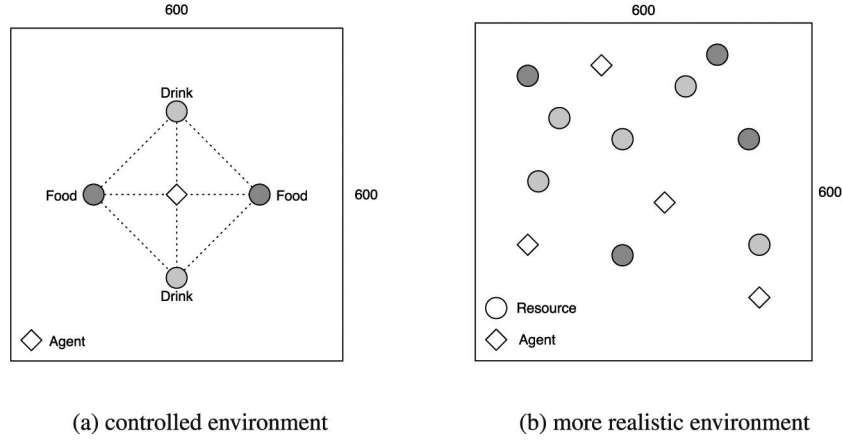


Figure 2: The two simulation environments used to test the overall efficiency of the agents: a completely controlled scenario (a) where energy sources are maximum distance apart, all agents are initially grouped at the centre and interruptions are externally induced, and a more realistic scenario (b) where agents and energy sources are placed randomly.

action	unlatched $\phi = \delta$	latched			significance		
		$\phi = \delta$	10	50	100	0-10	10-50
B_1^α	443	452	478	494	*	*	*
B_2^α	443	452	479	498	*	*	*
B_3^α	0	0	454	468		*	
B_4^α	0	0	1414	2037		*	*
total	886	903	2824	3498			

Table 1: Comparing latched and unlatched behaviours: a latch, $\phi - \delta \in \{10, 50, 100\}$, if sufficiently large, allows the agent to pursue the lower-priority behaviours B_3 and B_4 .

action	10			50			100			significance				
	1	3	5	1	3	5	1	3	5	0-1	0-3	0-5	1-3	3-5
B_1^α	458	442	420	478	481	462	519	504	508	*		*	*	
B_2^α	454	441	429	474	481	455	521	512	519	*		*		
B_3^α	0	0	0	277	1	0	468	421	1		*	*	*	*
B_4^α	0	0	0	95	0	0	1119	57	0	*	*	*	*	*
total	912	882	850	1324	962	917	2627	1493	1028					

Table 2: The performance of the agents given $\phi - \delta \in \{10, 50, 100\}$ and interruptions of frequency 1, 3 or 5. This is compared to the case without interruptions (0) as shown in table 1.

The next experiment investigates the apparent inefficiency of strict latching once an agent is confronted with interruptions. The data for this experiment is summarised in Table 2. Even in the case of a single interruption, the frequency of consummatory actions executed drops significantly. The right-most column in the table compares the performance of a latch of size 100 with 0, 1, 3 and 5 interruptions and the differences for the lower-priority actions are almost always significant.

The final experiment using *sim1* determines the performance of the flexible latch using the same settings as in the experiment before. Here, different values for the intermediate threshold ψ are tested. The value of ψ is denoted as the percentage of the latch itself. If, for example, $\delta = 100$ and

$\phi = 120$, a value of 25% would indicate $\psi = 105$. The results are shown in Table and a setting of $\psi = \delta$ seems most successful. However, as shown in Table 3, the differences are usually not significant. Nevertheless, such a setting is preferential as it allows to simply the flexible latch by effectively eliminating ψ altogether. Comparing the flexible latch to the strict latch shows a significant improvement in at least one action for any number of interruptions tested (compare Table 2 with Table ; significance is indicated in the rightmost column of Table 3).

Figure 3 shows graphically how the ability to detect interruptions improves the agent's overall efficiency. The graph plots the number of time steps spent executing the actions of interest given different frequencies of interruption. Further-

action	1				3				5			
	0%	25%	50%	75%	0%	25%	50%	75%	0%	25%	50%	75%
B_1^α	499	491	489	501	490	491	496	496	482	487	482	495
B_2^α	492	490	496	503	483	487	491	496	488	485	493	497
B_3^α	481	476	479	481	475	479	469	455	474	470	462	437
B_4^α	1723	1689	1528	1312	1458	1342	1059	651	1222	1150	880	495
total	3195	3146	2991	2797	2906	2799	2516	2098	2666	2592	2318	1923

	1			3			5			significance		
	0-25	25-50	50-75	0-25	25-50	50-75	0-25	25-50	50-75	1-1	3-3	5-5
B_1^α										*		*
B_2^α										*	*	*
B_3^α											*	*
B_4^α			*		*	*		*	*	*	*	*

Table 3: The performance of the agents using $\psi = \delta + p(\phi - \delta)$ where $p \in \{0, 0.25, 0.5, 0.75\}$, $\delta = 200$, $\phi = 300$ and frequency of interruptions equal to 1, 3 and 5. The right-most column compares the strictly and flexibly latched implementation for the different frequencies of interruptions. The statistical significances for this table is shown also.

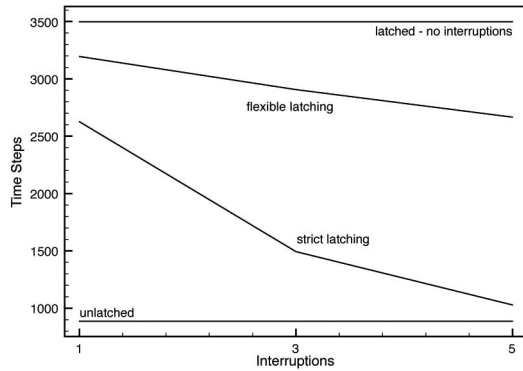


Figure 3: A graphical comparison of strict and flexible latching ($\sum_{i=1}^4 B_i^\alpha$). The top and bottom lines refer to the latched but uninterrupted and unlatched cases and are shown as reference values.

more, as a reference value, the unlatched and uninterrupted latched cases are also shown. As is evident, the performance of the strict latch degrades very quickly while the flexible latch degrades linearly.

Random Environment: *Sim2*

The previous results showed that in *sim1*, latching is necessary to allow an agent to execute lower-priority behaviours, and that it is best to abort a latched behaviour immediately upon interruption. We now examine these results in a system with a more “natural” setup using *sim2*, where the timing and frequency of interruption depends on the dynamics of the environment itself.

Table 4 compares all three implementation on *sim2*. The overall results are similar to before although there are some striking differences. Now, a latch of size 10 is sufficient to generate at least some frequency of execution for behaviours

B_3 and B_4 and there is no difference whether the agent is able to detect interruptions. Once the size of the latch increases, flexibility creates a significant difference for behaviour B_4 but not B_3 .

Discussion and Conclusion

The results for *sim1* show the utility of latching in the condition where there is a significant cost of switching between goals, and of flexible latching when there are large latches and frequent interruptions. The results for *sim2* show that when goal opportunities are more randomly and frequently available, there is little selective pressure for additional mechanisms.

We have presented a relatively simple way to introduce flexible latching into an autonomous system and presented an initial analysis of how to set appropriate thresholds. The completely unlatched condition may seem unrealistic, but several well-known “reactive” architectures have added latching only as an afterthought, and then to be handled with rather inelegant exception mechanisms (Rosenblatt & Payton 1989; Connell 1990). Others assume latching can be handled by intelligent planning (Bonasso *et al.* 1997), which is a rather high cognitive load. In general, reasoning about time and distant rewards is difficult for strictly symbolic systems (Ainslie 2005). We therefore support and elaborate the theory that evolved latch parameters create effectively an implicit time budget, as has been proposed by Dunbar (Dunbar 1993; Korstjens, Verhoeckx, & Dunbar 2006).

There have been surprisingly few recent attempts to propose general-purpose architecture features for homeostatic control, those that do tend to represent hormone levels in detail (Vargas *et al.* 2005). Gadanho (1999) has a similar perspective to our work⁴, using emotions to control the

⁴and indeed shared a graduate supervisor, John Hallam, with our second author.

action	unlatched	strict latched			flexible latched			significance		
	0	10	50	100	10	50	100	10-10	50-50	100-100
B_1^α	451	454	470	500	454	466	468			*
B_2^α	452	454	475	490	455	466	469		*	*
B_3^α	0	178	365	452	154	423	471			
B_4^α	0	71	264	689	22	704	1289		*	*
total	903	1156	1574	2131	1084	2058	2697			
dead	0	0	0	0	0	0	0			

Table 4: Comparing the unlatched, strictly and flexibly latched implementations in *sim2* using latch sizes of $\phi - \delta \in \{10, 50, 100\}$ and $\psi = \phi$.

temporal expression of behaviour. However, she focuses on modelling specific emotions and their impact on reinforcement learning systems, rather than a clear study of control. The mechanism demonstrated here is simple to implement and incorporate into standard module-based agent architectures. Also, she uses rising levels of emotions as the *source* of interrupt, rather than dealing with inefficiencies due to interruptions generated by the world.

Interestingly, at least two established models of consciousness are similar to our new model of flexibly-latched drives. Norman & Shallice (1986) describe consciousness as a higher-cost attentional system which is brought on line whenever the more basic, reliable, low-cost action-sequencing mechanism is unable to proceed. More recently, Shanahan (2005) proposes a model of mutually-inhibiting motives in a global workspace. We do not agree with Shanahan that such models can account for all of action selection (see e.g. the Tyrrell 1994 critique of Maes 1991). However, his model is similar to what we propose here for arbitration between certain types of high-level tasks.

This draws attention to an important limit of our work. Although we have shown massive efficiency improvements, time still increases linearly with the number of interruptions. Further, some forms of interruptions themselves will increase with the number of potential behaviours, particularly those that are generated by the action-selection mechanism itself as higher priorities trigger. What this implies is that agents should have a limited number of high-level motivations which are contested this way. In other work, we have suggested that the psychological entities called ‘drives’ or ‘emotions’ may be seen as each correlating to and regulating one such high-level need (Bryson 2008).

Of course, a simple system of eliciting drive levels and (possibly) weighing them against expected costs does not explain all the phenomena ordinarily associated with the term *consciousness*. That term is a repository of aggregated folk-psychological theories for aspects of behaviour ranging from perceptual experience through self-concept and on to the soul (Hobson, Pace-Schott, & Stickgold 2000; Dennett 2001). What we do note here is a control-level utility to adding a minimally cognitive mechanism to an otherwise reactive action-selection system.

Acknowledgements

Hagen Lehmann first identified that strict latching seemed inappropriate for modelling monkey behaviour, and pro-

vided the initial behaviour code that was extended for this paper. This research was conducted by PR based on a suggested fix by JJB. It was funded by the British EPSRC AIBACS programme, grant number GR/S79299/01.

References

- Ainslie, G. 2005. Emotion as a motivated behavior. In *Agents that Want and Like: Motivational and Emotional Roots of Cognition and Action (AISB 2005)*. Hertsfordshire: The Society for the Study of Artificial Intelligence and the Simulation of Behaviour.
- Bonasso, R. P.; Firby, R. J.; Gat, E.; Kortenkamp, D.; Miller, D. P.; and Slack, M. G. 1997. Experiences with an architecture for intelligent, reactive agents. *Journal of Experimental and Theoretical Artificial Intelligence* 9(2/3):237–256.
- Bryson, J. J., and Stein, L. A. 2001. Modularity and design in reactive intelligence. In *Proceedings of the 17th International Joint Conference on Artificial Intelligence*, 1115–1120. Seattle: Morgan Kaufmann.
- Bryson, J. J. 2000. Cross-paradigm analysis of autonomous agent architecture. *Journal of Experimental and Theoretical Artificial Intelligence* 12(2):165–190.
- Bryson, J. J. 2003. Action selection and individuation in agent based modelling. In Sallach, D. L., and Macal, C., eds., *Proceedings of Agent 2003: Challenges in Social Simulation*, 317–330. Argonne, IL: Argonne National Laboratory.
- Bryson, J. J. 2008. The impact of durative state on action selection. In Horswill, I.; Hudlicka, E.; Lisetti, C.; and Velasquez, J., eds., *Proceedings of the AAAI Spring Symposium on Emotion, Personality, and Social Behavior*, 2–9.
- Chapman, D. 1987. Planning for conjunctive goals. *Artificial Intelligence* 32:333–378.
- Connell, J. H. 1990. *Minimalist Mobile Robotics: A Colony-Style Architecture for a Mobile Robot*. Cambridge, MA: Academic Press. also MIT TR-1151.
- Dennett, D. C. 2001. Are we explaining consciousness yet? *Cognition* 79:221–237.
- Dunbar, R. I. M. 1993. Coevolution of neocortical size, group size and language in humans. *Behavioral and Brain Sciences* 16(4):681–735.

- Gadanh, S. C. 1999. *Reinforcement Learning in Autonomous Robots: An Empirical Investigation of the Role of Emotions*. Ph.D. Dissertation, University of Edinburgh.
- Gigerenzer, G., and Todd, P. M., eds. 1999. *Simple Heuristics that Make Us Smart*. Oxford University Press.
- Hexmoor, H.; Horswill, I.; and Kortenkamp, D. 1997. Special issue: Software architectures for hardware agents. *Journal of Experimental and Theoretical Artificial Intelligence* 9(2/3).
- Hobson, J. A.; Pace-Schott, E. F.; and Stickgold, R. 2000. Dreaming and the brain: Toward a cognitive neuroscience of conscious states. *Behavioral and Brain Sciences* 23(6):793–842.
- Korstjens, A. H.; Verhoeckx, I. L.; and Dunbar, R. I. M. 2006. Time as a constraint on group size in spider monkeys. *Behavioral Ecology and Sociobiology* 60(5):683–694.
- Kortenkamp, D.; Bonasso, R. P.; and Murphy, R., eds. 1998. *Artificial Intelligence and Mobile Robots: Case Studies of Successful Robot Systems*. Cambridge, MA: MIT Press.
- Laird, J. E., and Rosenbloom, P. S. 1996. The evolution of the Soar cognitive architecture. In Steier, D. M., and Mitchell, T. M., eds., *Mind Matters*. Erlbaum.
- Maes, P. 1991. The agent network architecture (ANA). *SIGART Bulletin* 2(4):115–120.
- Norman, D. A., and Shallice, T. 1986. Attention to action: Willed and automatic control of behavior. In Davidson, R.; Schwartz, G.; and Shapiro, D., eds., *Consciousness and Self Regulation: Advances in Research and Theory*, volume 4. New York: Plenum. 1–18.
- Rosenblatt, K., and Payton, D. 1989. A fine-grained alternative to the subsumption architecture for mobile robot control. In *Proceedings of the IEEE/INNS International Joint Conference on Neural Networks*. Montreal: Springer.
- Shanahan, M. P. 2005. Global access, embodiment, and the conscious subject. *Journal of Consciousness Studies* 12(12):46–66.
- Simon, H. A. 1972. Theories of bounded rationality. In Radner, C. B., and Radner, R., eds., *Decision and Organization*. Amsterdam: North Holland. 161–176.
- Tyrrell, T. 1994. An evaluation of Maes's bottom-up mechanism for behavior selection. *Adaptive Behavior* 2(4):307–348.
- Vargas, P.; Moioli, R.; de Castro, L. N.; Timmis, J.; Neal, M.; and Von Zuben, F. J. 2005. Artificial Homeostatic System: A Novel Approach. *Eighth European Conference of Advances in Artificial Life, ECAL* 754–764.