Flexible Latching: A Biologically-Inspired Mechanism for Improving the Management of Homeostatic Goals

Philipp Rohlfshagen · Joanna J. Bryson

Received: 28 August 2009 / Accepted: 17 June 2010

Abstract Controlling cognitive systems like domestic robots or intelligent assistive environments requires striking an appropriate balance between responsiveness and persistence. Basic goal arbitration is an essential element of low level *action selection* for cognitive systems, necessarily preceding even deliberate control in the direction of attention. In natural intelligence, chemically regulated motivation systems focus an agent's behavioural attention on one problem at a time. Such simple durative decision state can improve the efficiency of artificial action selection by avoiding dithering, but taken to extremes such systems can be inefficient and produce cognitively implausible results. This article describes and demonstrates an easy-to-implement, general-purpose latching method that allows for a balance between persistence and flexibility in the presence of interruptions. This appraisal-based system facilitates automatic reassessment of the current focus of attention by existing action-selection mechanisms. The proposed mechanism, *flexible latching*, drastically improves efficiency in handling multiple competing goals at the cost of a surprisingly small amount of additional code (or cognitive) complexity. We discuss implications of these results for understanding natural cognitive systems.

Keywords Action selection · Drives · Modularity · Cognitive architectures

Introduction

The term *action selection* might seem to imply cognition, but this is merely due to anthropomorphic labelling. If we take *cognition* to be a process requiring time (probably a form of on-line search; [42]), and *action selection* to be any mechanism for determining the present course of action [11], then much of action selection is really non-cognitive. Action choices in animals are limited by both evolution and individual skill learning; for adult animals many actions may be essentially reflexive [5, 7]. Such limiting is necessary if action selection is to be achieved in a timely manner [15, 21, 37]. However, there is no question that animals (including humans) do engage in cognition in some contexts. This article examines one such context: the arbitration between different goals. Even here, basic arbitration must necessarily be automatic. However, functional and efficient behaviour requires that the automated system can in some situations be interrupted and controlled cognitively [39]. Here, we present a way to efficiently facilitate this capacity in artificial cognitive systems.

Budgeting time and pursuing multiple conflicting goals is a key aspect of any cognitive system [17, 22]. In the

A preliminary version of this paper appeared in the AAAI Fall Symposium on Biologically Inspired Cognitive Architectures [P. Rohlfshagen, J.J. Bryson, Improved Animal-Like Maintenance of Homeostatic Goals via Flexible Latching, in: Proceedings of BICA-08, 2008]. This version contains results omitted from the workshop paper, as well as further and improved expositions.
The majority of this work was carried out when PR was in the Department of Computer Science, University of Bath, Bath, BA2 7AY, United Kingdom, and JJB was on sabbatical at The Konrad Lorenz Institute for Evolution and Cognition Research, Adolf Lorenz Gasse 2, A-3422 Altenberg, Austria.

P. Rohlfshagen
School of Computer Science, University of Birmingham,
Edgbaston, Birmingham B15 2TT, UK
e-mail: p.rohlfshagen@cs.bham.ac.uk

J. J. Bryson (✉)
Department of Computer Science, University of Bath,
Bath BA2 7AY, UK
e-mail: jjb@cs.bath.ac.uk

simulation of real-time animal-like intelligence considered in this paper, artificial agents must carry out a set of tasks, essential to their survival, while also interacting with dynamic surroundings, including other agents. Other-agent interactions in particular may include activities that are potentially essential to the species as a whole but not necessarily in the interest of the performing individual's viability. This characterisation might suggest rather dramatic activities, e.g. fending off attack, but it can also apply to ordinary duties. In some sense, the tasks that the system was originally designed to carry out (e.g. mating in nature, or perhaps tea making for an office robot) are of lower immediate priority than making certain that the system maintains working order, since working order (e.g. the ability to move and manipulate) is a precondition of any other activity. Nevertheless, it is clear that we require an agent to devote considerable time to the goals that motivated its construction. Such critical but non-urgent goals are common amongst animals, such as maintaining a social network, reproducing or keeping clean. 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 this article, we demonstrate our goal-arbitration system using a simple artificial life task environment. Our agents must ensure they have the ability to store excess energy in order to pursue auxiliary behaviours. We discovered the need for an improved arbitration mechanism during the course of research on the evolution of primate social structures, so our examples derive from these models. The immediately urgent goals concern feeding, while the ultimately important goals are social networking and exploration. Note that in nature such goals could also be considered survival oriented, since socialising promotes long-term survival by facilitating group living [17, 25]. However, their payoff is more diffuse—it is seldom knowable when additional goodwill or information gathered may become critical, in contrast to starvation which has clear endogenous indicators. Thus, we place essential behaviours at a high priority but design an action-selection mechanism to ensure they are executed as efficiently as possible.

In this article, we present a comparative study of three variants of a simple action-selection mechanism designed to improve the agent's capacity for goal arbitration. Our primary motivation is a potential inefficiency that may occur when an agent attempts to acquire a buffer of excess satisfaction before pursuing its next goal. We propose that if an agent is interrupted at any stage during this period, a choice needs to be made concerning whether to continue with the current goal or whether to attend to other, possibly more relevant behaviours. Persistence avoids the inefficiency of *dithering* between multiple goals. Dithering is inefficient because there is typically a significant start-up cost to pursuing new goals before consummatory actions can take place. However, some degree of flexibility avoids the inefficiency of pursuing a goal which is no longer urgent and has locally become excessively costly.

We look to biological motivation systems for inspiration because these have presumably evolved to manage this trade-off. However, here we do not attempt a perfect or neurological model of nature. Rather, our emphasis in this article is engineering. We present and evaluate a simple control mechanism that achieves the requisite level of flexibility at minimal cost. In fact, two types of costs are kept minimal: both the advance, coding-time costs for the agent's designers and the real-time, cognitive-processing costs for the agents. We use a basic latching system augmented with the ability to detect potentially relevant interruptions. This threshold-based addition triggers a reevaluation of priorities already present in the agents' overall action-selection system.

Methods

In this section, we first describe the particular agent architectures we use to test our new goal-arbitration system. Although we use a single system here, it is an example of a common type of action-selection system, and we describe the augmentation in general terms so that it may be applied to other systems as well. We then describe the specific goals to be manipulated in the experiments and define the metrics of success in terms of these. Next, we describe the various latching mechanisms we have implemented for comparison. Finally, we describe the testing scenarios, including the agents' operating environment, followed by the presentation and discussion of our results.

Basic Action Selection

The agents are specified using the behaviour-oriented design (BOD) methodology [12], a system that produces complete, complex agents consisting of (a) modules that specify details of their behaviour and (b) dynamic plans that specify agent-wide, cross-modular priorities. Actions are produced by the modules; action selection (where there is contention) is carried out using the parallel-rooted, ordered slip-stack hierarchical (POSH) dynamic plan system [10].

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 [8, 24, 26]. Even mainstream cognitive architectures such as Soar and ACT-R can be described in this way [28, 38]. Such approaches have been somewhat neglected in the academic literature in the last decade due to an emphasis on machine learning approaches to action selection. However, in applied human-like AI such as games programming and cognitive robotics, such modular, hand-coded approaches are still very much the norm [23, 31].

The details of the structured action-selection system are unimportant to the mechanism presented in this paper. All that is assumed is

- some mechanism for storing temporary values of long-term state (e.g. learning),
- some mechanism of expressing a variety of goals and their associated actions, and
- 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 the agents tested here. That is, all the agents have the same priorities and therefore the same dynamic plan, though of course their expressed behaviour will vary due to their environment and their previous experience. What differs between conditions in the experiments described below are only the action-selection mechanisms and the testing environments.

The plan, shown in Fig. 1, assumes four drives, in this case each with a corresponding behaviour: B_1–B_4. In POSH, the top level of a plan hierarchy (the *drive collection*) is checked on every cycle of the controller. Control is passed to the highest-priority drive element whose trigger (line-labels in Fig. 1) is true. All but behaviour B_4 further contain a sub-plan, in POSH called a *competence*. Competences also contain elements each with its own trigger, but these are plans for the purpose of pursuing a single goal, and as such require less sophisticated scheduling than the drive collection. Competences maintain decision memory and control behaviour until they either terminate, pass control to a child competence of their own, or the main drive collection takes control back for a higher-priority problem. Their execution is similar to teleo-reactive plans [32] or indeed to the generalised plans created by STRIPS [18].

The first two behaviours, which are of the highest (and equal) priority, fulfil consumption-related 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. It should be noted that these behaviours are of lower priority simply because behaviours B_1 and B_2 are essential to the agent's immediate survival. This does not imply, however, that lower-priority behaviours are not important. They could be critical to the agent's mission.

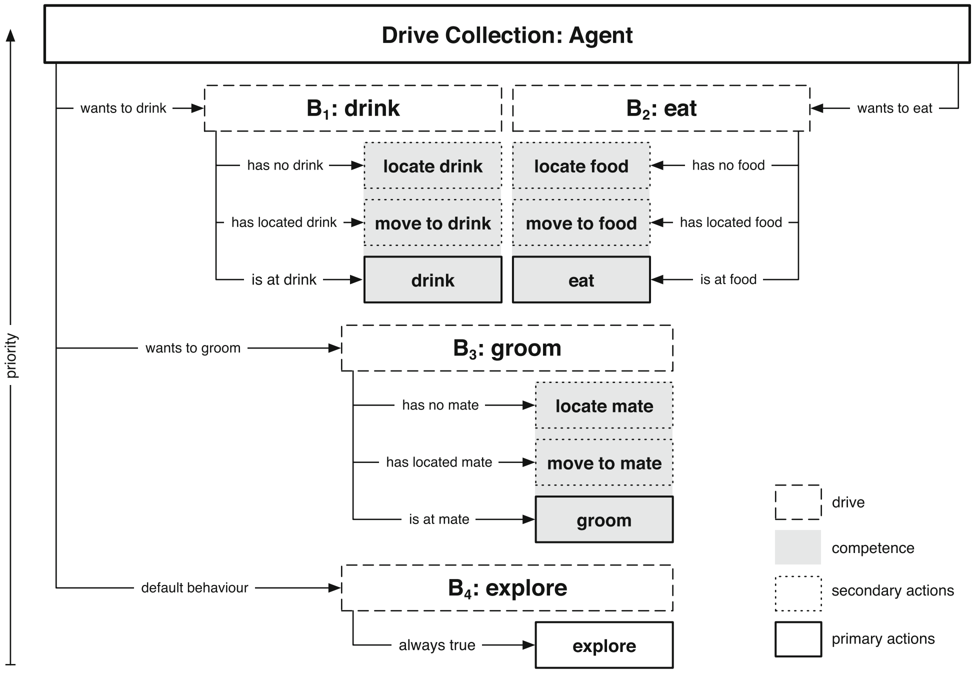


Fig. 1 The POSH plan that determines priorities for the agents: the *drive collection* is called at every program cycle and its elements checked in order: {B_1 = eat, B_2 = drink}, {B_3 = groom}, {B_4 = explore}. The highest-priority element whose trigger is true is executed. Equal-priority elements (e.g., B_1 and B_2) are checked in random order

Since our experimental environment represents primate social behaviour, these behaviours in fact relate to increasing the probability of longer life. As such, behaviour B_3 represents social networking through grooming, which requires two agents to interact with one another. The final behaviour (B_4) is exploration, possibly to find new food sources. In a POSH plan, the lowest-priority goal serves as a default behaviour and should always be triggerable. Thus, if an agent with this plan is efficiently arbitrating between goals, it should be able to spend most of its time exploring new space.

Metrics of Efficient Behaviour

The primary focus of our investigation is on behaviours B_3 and B_4. Lower-priority behaviours may only be executed if all higher-priority behaviours are managed efficiently. For artificial agents, the 'lower' behaviours are typically the ones that define and justify the agent's mission. Despite their significance, these behaviours are necessarily of lower priority than those that facilitate the survival of the agent so it can perform these tasks. It is therefore paramount that these higher-level behaviours are managed efficiently enough to allow agents to pursue other behaviours as well.

Each behaviour consists 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'. For all behaviours, executing the primary action with a high frequency relative to the secondary actions determines the degree of efficiency with which the behaviour is executed. *Dithering*, the rapid switching between goals, results in secondary actions being performed excessively in proportion to primary ones. In our example, each behaviour B_i has one such primary action which will be denoted as B_i^{α}. The frequency at which primary actions are executed determines the extent to which all behaviours may be executed and thus defines the metric of success at the centre of our investigation.

Agents and State

Each behaviour B_i is associated 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, world with dimensions of 600 × 600 pixels. Time is considered to be discrete and at every time step, all agents in the environment are updated simultaneously. In particular, at every time step, all energy states E_i are decreased by e_i^-. If a given 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 will execute the actions associated with that particular behaviour, if no higher-priority behaviour is also triggered. The behaviours B_1 and B_2 in our example correspond to sustenance 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). Here, we have chosen the same values for all behaviours: $e^+ = 1.1$ and $e^- = 0.1$ and hence drop the behaviour-dependent subscript i from here on. Since we are interested in the execution of lower-priority behaviours, an individual choice of energy gain/loss across the different behaviours would require the adjustment of the individual thresholds (which are tightly related to the net energy gain), unnecessarily complicating the model. Overall, this gives a net energy gain of $e^{\pm} = 1$ for any primary action.

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 article which we describe next.

Conditions

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

Unlatched

As mentioned in the previous section, a behaviour B_i is triggered if $E_i < \delta_i$. In the 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 depends strictly on $\delta_i - E_i$ which may vary depending on the number of equal-priority behaviours). Furthermore, no excess energy is stored, and the behaviour is triggered again very shortly after it is satisfied[1]. 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.

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

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-or-same priority needs. For example, an agent that is grooming and whose partner has left might pursue another partner for 5 min when only another 5 s of grooming would have satiated it. This is true even if $E_i = \phi_i - \varepsilon$ where $\varepsilon \ll \phi_i - \delta_i$ 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. Such a decision might 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 two different scenarios. If the interruption occurs when:

1. $\delta_i < E_i < \psi_i$, the drive remains 'unsatisfied'
2. $\psi_i < E_i < \phi_i$, then the drive is considered 'satisfied'

Note that for $\delta_i < E_i < \phi_i$, the status of any latch is path or history dependent—if E_i was more recently below δ the drive is now unsatisfied, if it was more recently satiated (about ϕ) than it is not. What is new for the flexible latch is that if an interruption occurs in the third scenario, where E_i had been below δ but has now been raised above ψ_i, this path dependency is dismissed.

Threshold Selection

The previous section has discussed 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:

Flexible latch	$\delta \leq \psi \leq \phi$
Strict latch	$\delta \leq \psi = \phi$
Unlatched	$\delta = \psi = \phi$

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 and ψ. 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):

$$\delta_i \geq E_i^r \tag{1}$$

where E_i^r is the energy required to reach the source: $(d_{\max}/d_{\text{mov}}) \times e^-$, 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 travel[2]. If there are n equally vital behaviours, δ_i has to be adjusted accordingly:

$$\delta_i \geq \sum_{j=1}^{n-1} \left(E_j^r + E_j^c \right) + E_n^r \tag{2}$$

where E_i^c is the energy required to raise the energy level to the appropriate level:

$$E_i^c = \frac{\delta_i - E_i}{e^{\pm}} \tag{3}$$

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

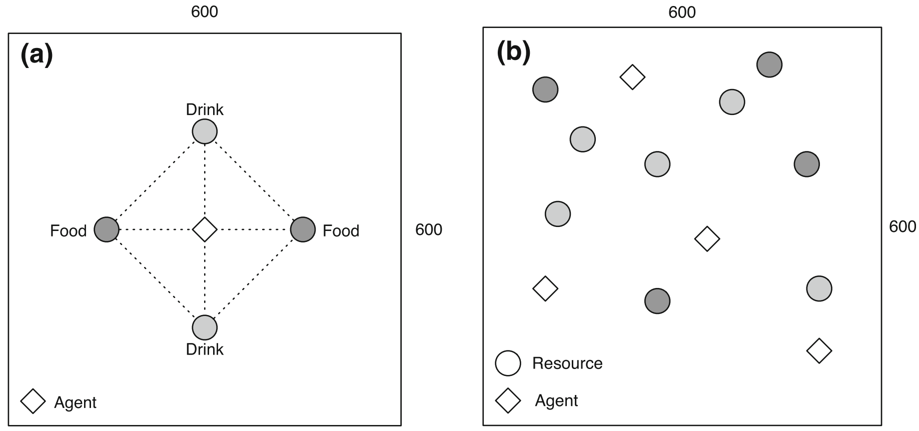
$$\phi_i \geq \delta_i + \sum_{j=1}^{n} \left(E_j^r + E_j^c \right) \tag{4}$$

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

$$E_i^c = \sum_{j=1}^{m} \left(E_{ij}^r + E_{ij}^c \right) \tag{5}$$

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. Consequently, heuristic values must be used. Nature selects for agents with appropriate or at least adequate thresholds; here we test a

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



controlled environment

more realistic environment

Fig. 2 The two simulation environments used to test the overall efficiency of the agents: a completely controlled scenario (**a**) where energy sources are the 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

range of values for ψ to find which is appropriate for our particular simulations.

Experiment and Simulation Details

Our experiments are organised into two sets. The first set uses *sim1*, a very well-defined set-up that allows a great degree of control over all aspects investigated, particularly the frequency of interruption (see Fig. 2a). The second set use *sim2* (Fig. 2b), a more realistic simulator where interruptions are caused by the dynamics of the environment itself. For our experiments, we consider two types of interrupts. The first type occurs when the source of satisfaction is depleted or otherwise removed (e.g., an agent looses its current grooming partner). The second type of interrupt is caused by higher-priority drives that are triggered.

In both simulations, there are five identical agents. Furthermore, *sim1* positions the energy sources such that they are maximum distance from one another[3]. 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 somewhat 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 unit per time step.

The experiments are executed over 15 distinct trials. Each trial executes the simulation for 5,000 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 0.995. 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 E falls to zero after a behaviour has been triggered.

Results

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

Table 1 Comparing latched and unlatched behaviours

Action	No latch	Latched			Significance		
	$\phi = \delta$	10	50	100	0–10	10–50	50–100
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	1,414	2,037		*	*
Total	886	903	2,824	3,498			

The latches are chosen to be $\phi - \delta \in \{0, 10, 50, 100\}$

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

Table 2 The performance of the agents given $\phi - \delta \in \{10, 50, 100\}$ and 1, 3 or 5 interruptions

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	1,324	962	917	2,627	1,493	1,028					

Significance is checked for $\phi = 100$. Cases without interruptions (0) are taken from the results shown in Table 1 (not shown in this table)

size 10 does increase the activity of the primary actions for behaviours B_1 and B_2 but still does not allow for the lower-priority behaviours B_3 and B_4 to be executed. Once the latch increases sufficiently in size, so does the activity of the lower-priority behaviours. This result is not surprising. Note though that too large a latch might also lead to neglect of lower-priority behaviours, since the highest-level goals might never be satisfied.

The next experiment investigates the efficiency 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 primary 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.

Table 4 Significance results for Table 3

	1			3			5			*vs*. strict		
	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}			*		*	*		*	*	*	*	*

Increasing p has the most impact on the lowest-priority behaviour. The right-most column compares the strictly and flexibly latched implementation for the different frequencies of interruptions

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 that $\psi = 105$. The results are shown in Table 3, and a setting of $\psi = \delta$ seems most successful. However, as shown in Table 4, the differences are usually not significant. In the absence of significant difference, the zero setting is still to be preferred as it also allows us to simplify the action-selection mechanism. We can effectively eliminate ψ altogether but *always* reconsider priorities when interrupted. Comparing the flexible latch to the strict latch shows a significant improvement in at least one behaviour's primary action for any number of interruptions tested (compare Table 2 with Table 3; significance is indicated in the right-most column of Table 4).

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. Furthermore, as a reference value, the unlatched and uninterrupted latched cases are also shown. It is evident that the performance of the strict latch degrades very quickly while the flexible latch substantially reduces the impact of interruptions.

Table 3 The performance of the agents with flexible latching

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}	1,723	1,689	1,528	1,312	1,458	1,342	1,059	651	1,222	1,150	880	495
Total	3,195	3,146	2,991	2,797	2,906	2,799	2,516	2,098	2,666	2,592	2,318	1,923

$\psi = \delta + p(\phi - \delta)$ where $p \in \{0, 0.25, 0.5, 0.75\}$, $\delta = 200$, $\phi = 300$ and frequency of interruptions equal to 1, 3 and 5. Significance of results shown in Table 4

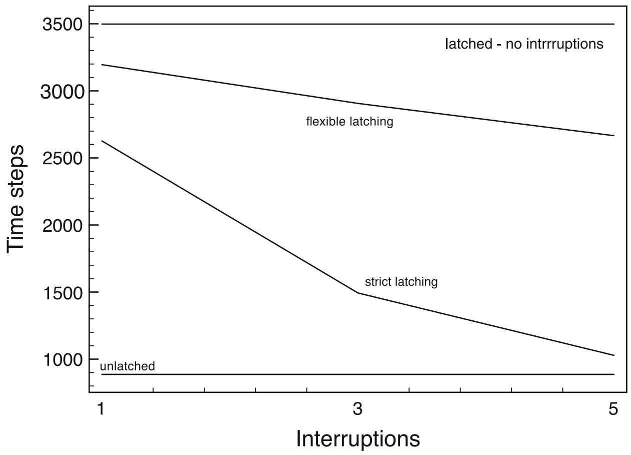


Fig. 3 A graphical comparison of strict and flexible latching $\left(\sum_{i=1}^{4} B_i^{\alpha}\right)$. The y axis is the number of program cycles available for pursuing the lower-priority behaviours 3 and 4. The top and bottom lines are shown for reference, indicating the latched but uninterrupted and unlatched cases. For uninterrupted latches, the strict and flexible cases are indistinguishable

Death Rates

In the previous experiments, efficiency was judged by the capacity to devote time to all behaviours. For these experiments, the value of δ has been set such that agents would always survive. In nature, such a threshold would evolve in species like primates that invest a great deal in individual survival and life histories. Nevertheless, exceptionally extreme environments or other unusual circumstances may cause a threshold setting to become (temporarily) insufficient.

In the present experiment, we set δ such that survival in an uncertain environment is no longer guaranteed ($\delta = 40$). We then compare death rates between strict and flexible latches. The latch is also set at a relatively low level of $\phi = 45$. The results are shown in Table 5. The flexible latch shows a significantly reduced death rate in all three relevant conditions (as determined by the number of interruptions). Furthermore, it is interesting to note that now, even with the smaller latch, the flexible implementation performs significantly better in almost all cases when compared to the strictly latched version.

Table 5 A comparison of death rates for agents with lower values of δ than are entirely sustainable in the environmental context. Tests are run with strict or flexible latching and with from 0–3 interruptions. Without interruptions, whether the latch is flexible is irrelevant

Action	0	Strict latch			Flexible latch			Significance		
		1	2	3	1	2	3	1–1	2–2	3–3
B_1^a	478	423	34	34	475	387	360		*	*
B_2^a	477	415	37	32	475	390	360	*	*	*
B_3^a	460	256	0	0	444	324	255	*	*	*
B_4^a	1,402	90	0	0	750	295	140	*	*	*
Total	2,816	1,185	71	66	2,144	1,397	1,115			
Dead	0	601	4,551	4,551	0	861	1,143	*	*	*

Finally, it is possible to reduce the death rate even further. In another scenario, we utilise the agents' ability to deal with interruptions: equal-priority behaviours are allowed to interrupt one another if they reach a critical threshold ψ. We set $\psi = 20$, as per the calculations described in Sect. "Threshold Section" above. This critical threshold essentially corresponds to the minimum energy required to satisfy a single need. The addition of the threshold changes the death rates from 0, 861, 1,143 to 60, 417, 472. Interestingly, the death rate is actually slightly higher in the first case but noticeable lower in the other two cases. The differences are relatively weakly significant for this N, with a confidence of $p < 0.05$ for both the two- and three-interrupt conditions.

Random Environment: *Sim2*

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

Table 6 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 whether or not it is flexible and indeed the flexibility makes no significant difference at this size latch. The change is due to the random environment providing more opportunities, which either implementation is able to exploit. Once the size of the latch increases, flexibility creates a noticeable (as well as significant) difference for behaviour B_4, but no difference for B_3. This indicates B_3's primary action is already executed sufficiently even without the flexibility in the latch—the flexibility in the environment provides sufficient opportunities for it to satiate at the threshold levels we have specified. Nevertheless, the massive increase of opportunity for expressing the exploratory behaviour shows the power of flexible latching.

Discussion

We have considered three variants of a simple threshold-based action selection mechanisms. The completely unlatched condition may seem unrealistic, but several well-known reactive architectures have added latching only as

Table 6 Comparing the unlatched, strictly and flexibly latched implementations in *sim2* using latch sizes of $\phi - \delta \in \{10, 50, 100\}$ and $\psi = \phi$. All cases have frequent interruptions (see main text)

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	1,289		*	*
Total	903	1,156	1,574	2,131	1,084	2,058	2,697			
Dead	0	0	0	0	0	0	0			

an afterthought, handled with rather inelegant exception mechanisms [16, 35]. Others assume latching can be handled by intelligent planning [6, 39]. This, however, requires a high cognitive load and in general, reasoning about time and distant rewards is difficult even for cognitive, symbolic systems [1].

The basic latched approach is inspired by theories of affect and action selection, as well as basic control theory. LeDoux [29] for example promotes the theory that emotions place the brain in a cognitive context appropriate for a particular course of action. Neuroscience tells us that interrupting such emotional responses is a cognitive capacity requiring frontal-lobe inhibition of the emotional response [14]. Of course, the frontal-lobe inhibition system must itself be a fairly automatic gating mechanism. But this mechanism provides an opportunity for an alternative plan to become most salient [34].

Our system for determining appropriate thresholds for the flexible latches is also inspired by animal mechanisms through ethology. In particular, Dunbar's time budget theory [17, 25] suggests that animal drives have evolved to ensure individuals are likely to spend the appropriate amount of time in behaviours, where *appropriate* is determined by what is adaptive. Our work here can be seen both as support for this theory and possibly as an elaboration, to the extent that our mechanism helps connect the time budget to the underlying neuroscience others have proposed (e.g. [34].)

In AI in contrast, there have been surprisingly few recent attempts to propose general-purpose architectural features for homeostatic control. Those that exist tend to create detailed biomimetic representations of hormone levels [27, 41]. Gadanho [20] has a similar perspective to our work, using emotions to control the temporal expression of behaviour. However, she focuses on modelling specific emotions and their impact on reinforcement learning systems, rather than focusing directly on control mechanisms. In contrast, our flexible latch is simple to implement and incorporate into any standard module-based agent architecture. Also, she uses rising levels of emotions as the *source* of interruptions, rather than dealing with inefficiencies caused by interruptions generated by the external environment.

Interestingly, several established models of consciousness are similar to our new model of flexibly latched drives. Norman and Shallice [33] 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. Our system of flexible latching also operates by recognising interruptions. It would be plausible in a system with modules capable of deliberation to have interruptions trigger these rather than the simple reassessment of existing goals demonstrated above. More recently, Shanahan [36] proposes a model of mutually inhibiting motives in a global workspace. We do not entirely support Shanahan's model. Mechanisms such as our flexible latching or Shanahan's global workspace model cannot account for detailed action selection because they do not scale well. Tyrrell [40] demonstrates this with an extensive critique of a similar spreading-activation architecture, The Agent Neural Architecture [30] (more commonly referred to as Maes' Nets [19]). The problem is simple combinatorics—a problem that architectures like ACT-R and IDA address by focussing on a single-plan subset of the full network [2, 19]. This focussing makes these architectures functionally similar to script-based dynamic-planning systems, such as POSH (used here), although their actual action-selection mechanisms are far more complex. However, as this paper makes clear, we do think that a system like Shanahan's or Maes' could well account for high-level goal arbitration.

IDA is a cognitive architecture specifically designed to implement a theory of consciousness [3]. IDA is not only a model, but also a working AI architecture which has been used to create recommender systems for the US Navy. Its newest version, LIDA provides the functionality of flexible latches through "timekeeper codelets" [4, p. 30] which keep a proposed action salient long enough for a variety of options to be debated. This system could well be effective and is certainly more conducive to human-like meta cognition than the system proposed here. However, our flexible latches are simpler and probably sufficient for most autonomous AI applications.

The problems Tyrrell identified with spreading-activation models are to some extent addressed by Goetz & Walters [22], who recommend generating a system of attractors in the networks. This achieves an effect similar to the latching shown here. Again the mechanism and architecture presented here are much simpler than spreading activation, even without the attractor system [9].

The difficulties in scaling spreading-activation networks draw attention to an important limit of our work. Although we have shown substantial efficiency improvements, temporal costs still increase linearly with the number of interruptions. Further, some forms of interruptions will necessarily increase with the number of potential behaviours—in particular 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.

What we present here is a cognitively minimal mechanism that makes substantial improvements to an otherwise reactive action-selection system. Elsewhere, we explore in more detail the earlier suggestion that due to LeDoux that the psychological entities called *drives* and *emotions* may be seen as a chemically based latching system, evolved to provide persistence and coherence to the otherwise electrically based action selection provided by the central nervous system [13]. We hypothesise that in nature, each drive or emotion—with its associated pattern of hormonal regulators and species-typical actions—might be viewed as serving one such high-level goal or need. We recommend that a system such as our flexible latch should similarly be used for each high-level goal an agent has that requires a time budget in an artificial cognitive system.

Conclusions

In this paper, we have presented a relatively simple way to introduce flexible latching into an autonomous system and presented an analysis of how to determine appropriate thresholds that govern the execution of lower-priority behaviours. The agents we considered have been specified using the behaviour-oriented design methodology: each agent consists of a set of modules that produce specific behaviours as well as a dynamic plan that prioritises amongst these behaviours. We take this as a fairly standard modular architecture using scripted dynamic plans for action selection and then demonstrate how to extend that action selection to improve its efficiency.

We demonstrate our system using four behaviours derived from a tool for modelling primate social behaviour. Two behaviours—eating and drinking—are essential to the immediate survival of the agent and are of highest (and equal) priority. The third, grooming, represents a mission-critical behaviour though it is not essential for immediate survival. This and the fourth, default behaviour (exploring), can only be executed if the higher-priority behaviours are managed efficiently. Each behaviour is composed of a number of individual actions, and we distinguish between *primary* and *secondary* actions. Secondary actions are those required to perform the primary action; the primary action is the core consummatory action of the behaviour and satisfies the agent's need that triggers the behavioural module. Efficient execution of behaviours requires the agents to (a) minimise the execution of secondary actions and (b) acquire sufficient satisfaction (energy in our case) to be able to carry out lower-priority behaviours.

The behaviour- (or action-) selection mechanism we have introduced consists of three thresholds: a lower threshold δ that triggers the behaviour depending on the agent's internal state, an intermediate threshold, ψ, that acts in case the agent is interrupted and an upper threshold, ϕ, that causes the behaviour to terminate. The addition of these thresholds does not alter the priorities of the behaviours (which are still governed by the dynamic plan) but may delay (or not) the execution of lower-priority behaviours and may have a significant impact on the ratio of secondary to primary actions performed by the agent. We demonstrated their efficacy in two experimental settings. Without latching (i.e., only a lower threshold), the agent dithers between food sources, leaving no time to execute lower-priority behaviours. Latching (i.e., lower and upper threshold) allows for *persistence* but may be hugely inefficient in the presence of interruptions. The persistent pursuit of unsatisfied behaviours may lead to an unsustainable frequency of secondary task executions.

The experiments allowed us to determine the most useful setting for the intermediate threshold, above which an interrupted agent may reconsider its behaviour priorities. The results demonstrate the utility of latching, as long as the latch is sufficiently large, in cases where there is a significant cost of switching between goals. *Flexible* latching addresses a reduction in performance of latches when there are interruptions. We found that the intermediate threshold is often not required, or more precisely, can be set to be equal to the lower threshold. In our initial set of experiments, it was optimal for agents to reconsider priorities *whenever* interrupted. This result may not hold if interruptions are more frequent and/or the size of the latch is smaller (as in the case of the final set of experiments), since either case would increase the probability that persistence is needed. Finally, we also explored the case where the agent may die if essential behaviours are carried out inefficiently. We found that latching significantly improves the rate of survival of the agent.

We have discussed how this mechanism, despite its simplicity, or because of it, may be relevant to numerous existing artificial cognitive architectures, and we have drawn parallels to natural decision-making processes. Although the validation presented here is admittedly limited, these results do match expectations derived from our observations in nature concerning the life-history strategies for species that tend to be correlated with more cognitive ability. At the same time, the work presented here also allows for extremely simple implementations such as hand-coding heuristic indicators of interruption.

There are numerous possible avenues to be explored in the near future. In our experiments, we chose the same thresholds for all behaviours, allowing a centralised approach that involves little overhead. However, it would be interesting to highlight potential differences in the efficiency of an agent's action selection when all behaviours have individual threshold settings. Furthermore, the thresholds may be adjusted dynamically over time (e.g., using a simple feedback control loop) or in artificial life contexts might be individually evolved.

Acknowledgments Hagen Lehmann first identified that strict latching seemed inappropriate for modelling primate behaviour; his model provided the initial behaviour code that was extended for this research. The research was conducted by PR based on a suggested fix by JJB. It was first presented at the AAAI 2008 Fall Symposium on Biologically Inspired Cognitive Architectures; we thank the organisers of that meeting, the other participants and the reviewers both for AAAI and Cognitive Computation. Research funded by the British Engineering and Physical Sciences Research Council (EPSRC) grants GR/S79299/01 and EP/E058884/1.

References

1. Ainslie G. 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; 2005.
2. Anderson JR, Matessa MP, Lebiere C. ACT-R: a theory of higher level cognition and its relation to visual attention. Human Comput Interaction. 1997;12(4):439–62.
3. Baars BJ, Franklin S. How conscious experience and working memory interact. Trends Cognit Sci. 2003;7(4):166–72.
4. Baars BJ, Franklin S. Consciousness is computational: The LIDA model of global workspace theory. Int J Machine Consciousness. 2009;11:23–32.
5. Bargh JA, Chartrand TL. The unbearable automaticity of being. Am Psychol. 1999;54(7):462–79.
6. Bonasso RP, Firby RJ, Gat E, Kortenkamp D, Miller DP, Slack MG. Experiences with an architecture for intelligent, reactive agents. J Exper Theor Artificial Intelligence. 1997;9(2/3):237–56.
7. Brooks RA. Intelligence without representation. Artificial Intelligence. 1991;47:139–59.
8. Bryson JJ. Cross-paradigm analysis of autonomous agent architecture. J Exper Theor Artificial Intelligence. 2000;12(2):165–90.
9. Bryson JJ. Hierarchy and sequence versus full parallelism in reactive action selection architectures. In: From animals to animats 6 (SAB00). Cambridge, MA: MIT Press; 2000, p. 147–156.
10. Bryson JJ. Action selection and individuation in agent based modelling. In: Sallach DL, Macal C, editors. Proceedings of agent 2003: challenges in social simulation. Argonne, IL: Argonne National Laboratory; 2003, p. 317–330.
11. Bryson JJ. Mechanisms of action selection: introduction to the theme issue. Adaptive Behav. 2007;15(1):5–8.
12. Bryson JJ, Stein LA. Modularity and design in reactive intelligence. In: Proceedings of the 17th International Joint Conference on artificial intelligence. Seattle: Morgan Kaufmann; 2001, p. 1115–1120.
13. Bryson JJ, Tanguy EAR. Simplifying the design of human-like behaviour: emotions as durative dynamic state for action selection. Int J Synthetic Emotions. 2010;1(1):30–50.
14. Carlson NR. Physiology of behavior, 7th ed. Boston: Allyn and Bacon; 2000.
15. Chapman D. Planning for conjunctive goals. Artificial Intelligence. 1987;32:333–78.
16. Connell JH. Minimalist mobile robotics: a colony-style architecture for a mobile robot. Cambridge, MA: Academic Press; 1990. also MIT TR-1151.
17. Dunbar RIM. Coevolution of neocortical size, group size and language in humans. Behav Brain Sci. 1993;16(4):681–735.
18. Fikes RE, Hart PE, Nilsson NJ. Learning and executing generalized robot plans. Artificial Intelligence. 1972;3:251–88.
19. Franklin S. A "consciousness" based architecture for a functioning mind. Visions of mind: Architectures for Cognition and Affect; 2005, p. 149–175.
20. Gadanho SC. Reinforcement learning in autonomous robots: an empirical investigation of the role of emotions. Ph.D. thesis, University of Edinburgh; 1999.
21. Gigerenzer G, Todd PM, editors. Simple heuristics that make us smart. Oxford: Oxford University Press; 1999.
22. Goetz P, Walters D. The dynamics of recurrent behavior networks. Adaptive Behav. 1997;6(2):247–83.
23. Hawes N, Wyatt J, Sloman A. Exploring design space for an integrated intelligent system. Knowledge-Based Syst. 2009;22(7):509–15.
24. Hexmoor H, Horswill I, Kortenkamp D. Special issue: Software architectures for hardware agents. J Exper Theoretical Artificial Intelligence. 1997; 9(2/3):323–36.
25. Korstjens AH, Verhoeckx IL, Dunbar RIM. Time as a constraint on group size in spider monkeys. Behav Ecol Sociobiol. 2006;60(5):683–94.
26. Kortenkamp D, Bonasso RP, Murphy R, editors. Artificial intelligence and mobile robots: case studies of successful robot systems. Cambridge, MA: MIT Press; 1998.
27. Krichmar JL. The neuromodulatory system: a framework for survival and adaptive behavior in a challenging world. Adaptive Behav. 2008;16(6):385–99.
28. Laird JE, Rosenbloom PS. The evolution of the Soar cognitive architecture. In: Steier DM, Mitchell TM, editors. Mind matters. Hillsdale: Erlbaum; 1996.
29. LeDoux J. The emotional brain: the mysterious underpinnings of emotional life. Simon and Schuster, New York; 1996.
30. Maes P. The agent network architecture (ANA). SIGART Bulletin. 1991;2(4):115–20.
31. Millington I, Funge J. Artificial intelligence for games. Amsterdam: Elsevier; 2009.

32. Nilsson NJ. Teleo-reactive programs for agent control. J Artificial Intelligence Res. 1994;1:139–58.
33. Norman DA, Shallice T. Attention to action: willed and automatic control of behavior. In: Davidson R, Schwartz G, Shapiro D, editors. Consciousness and self regulation: advances in research and theory, vol. 4. New York: Plenum; 1986, p. 1–18.
34. Redgrave P, Prescott TJ, Gurney K. Is the short-latency dopamine response too short to signal reward error? Trends Neurosci. 1999;22(4):146–51.
35. Rosenblatt K, Payton D. 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; 1989.
36. Shanahan MP. Global access, embodiment, and the conscious subject. J Consciousness Stud. 2005;12(12):46–66.
37. Simon HA. Theories of bounded rationality. In: Radner CB, Radner R (eds) Decision and organization. Amsterdam: North Holland; 1972, p. 161–176.
38. Sloman A. What sort of architecture is required for a human-like agent. Foundations of Rational Agency; 1999, p. 35–52.
39. Sun R. Motivational representations within a computational cognitive architecture. Cognit Comput. 2009;1(1):91–103.
40. Tyrrell T. An evaluation of Maes's bottom-up mechanism for behavior selection. Adaptive Behav. 1994;2(4):307–48.
41. Vargas P, Moioli R, de Castro LN, Timmis J, Neal M, Von Zuben FJ. Artificial homeostatic system: a novel approach. Eighth European Conference of advances in artificial life, ECAL; 2005, p. 754–764.
42. Winston PH. Artificial intelligence. Boston, MA: Addison-Wesley; 1984.