Modularity and Specialized Learning: Reexamining Behavior-Based Artificial Intelligence

Joanna Bryson
University of Bath Department of Computer Science
Artificial models of natural Intelligence (AmonI)
Bath, BA2 7AY, United Kingdom
J.J.Bryson@bath.ac.uk

Abstract

Learning, like any search, is only tractable for situated, resource-constrained agents if it is tightly focused. Adaptation is only worth the risks inherent in changing a complicated intelligence if it is very likely to improve the agent's performance on its goal tasks. Modularity is one tool for providing the information a learning system needs: it facilitates the use of a specialized representation suitable to a particular learning task, and provides for specialized perception to inform that representation. This paper begins by examining why behavior-based artificial intelligence, a well-known modular theory of intelligent design, has not so-far been used systematically to support such an approach. It then describes a new design methodology, behavior-oriented design (BOD), which does. Examples, drawn from both mobile robotics and models of learning in non-human primates, show the sorts of information such an approach can support, including both explicit and implicit representations.

1. Introduction

Behavior-based artificial intelligence (BBAI) is one of the best-known modular theories of intelligent design. Historically, however, although researchers have sometimes incorporated learning modules [e.g. 26, 29], there has been no systematic incorporation of learning into pure behavior-based design [though see 17]. Some hybrid systems have been developed which incorporate both BBAI and traditional planning and learning, but these lose the full advantages of modularity. Contemporary multi-agent systems (MAS) are fully modular, but overlook the progress made by BBAI in organizing distributed systems of complicated modules.

In this paper I describe how BBAI can be adapted to fully support modular learning. I begin by reviewing the history of BBAI. Then I discuss my own methodology, Behavior-Oriented Design (BOD), and explain how it exploits specialized learning, with examples drawn from both robotics and from ALife models of non-human primates. BOD allows a system with standard reactive control to behave in an adaptive manner, because its control is reliant on modules containing state, which may reflect current sensor states, predictions based on learning, or more likely a combination of both.

2 Behavior-Based Artificial Intelligence

Behavior-Based Artificial Intelligence (BBAI) is a methodology for constructing intelligent agents which specifies that the attributes of their intelligence should be decomposed into semi-autonomous modules. The expressed behavior of these modules is made coherent through some system of arbitration between these modules. Both the arbitration system and the individual modules are intended to require relatively little processing power or time, so that the agent can respond quickly and appropriately to challenges and opportunities in a complex dynamic environment.

When BBAI was introduced by Brooks [3, 5], its primary purpose as a methodology was to provide a means to create these responsive (or *reactive*) agents. Creating such agents is difficult because a rich environment provides so many things to react to. Any living agent in a complex environment must choose between a large number of possible actions, where each action is itself dependent on a large number of environmental contingencies, and is motivated by competing, mutually exclusive goals[1]. Choosing an optimal next action is impossible [14]. Even choosing a pretty good one requires searching an enormous space of possibilities.

[1]The attribute *living* mandates at least the conflicting goals of acquiring sustenance, avoiding injury, and participating in selective reproduction.

Because an individual agent does not have time for such a search in real time, most of its decision must be made in advance of the agent's active life. However, this does not remove the complexity of the real decision nor the amount of search necessary for a pretty-good decision. For animals, most of this search has been performed by evolution over a period of billions of years. For animats, the analogous role to evolution's is further split between the search conducted by the individual animat designer and that performed by the designer's culture. Designers must anticipate the behaviorally-salient contingencies that their agent may encounter, and provide rapid ways to recognize and select the appropriate response. We do this both through our own analysis and experimentation, but also by exploiting the scaffolding of design knowledge we have previously learned, which itself relies on intelligence scaffolding evolved with our species.

BBAI is a piece of design knowledge that significantly advanced the state of agent design, particularly in the areas of mobile robotics and virtual reality. I believe that the primary reasons for this success were:

- the increased emphasis on providing engineered knowledge thus simplifying the bottom-up processing necessary for the agent's sensing and action-selection, and
- the modular decomposition around individual expressed behaviors, which exploited the designers' existing skills and talents for writing simple programs.

However, after these significant advances, the complexity of the agents being built plateaued before the development of animal-level intelligence. Again, I believe there were two primary causes of this plateau:

1. the fact that at least *some* expertise is best developed by the agent through experience, particularly of the local variations of its own physical plant ('body'), and its own local environment, and
2. the complexity of programming the behavior-arbitration systems increases exponentially as the complexity and number of behavior modules increased.

The first point is key to the thesis of this paper: that modularity presents BBAI with the opportunity to maximally facilitate individual adaptation through providing specialized representations and processes. The second point, although important in the history of BBAI, is really a special case of the first. Modularizing the process of behavior arbitration and providing it with appropriate representations can greatly simplify the design process for a behavior-based agent.

3 A Brief History of Modular AI

This is a brief history of the critical attributes of BBAI systems I outlined above. More extensive reviews of the BBAI literature are also available [6, 8].

3.1 Modules for Perception

I will begin with Fodor's "The Modularity of Mind" [19], both because it introduces many of the concepts familiar to BBAI, and because it presents a theory of intelligence decomposition which is still actively researched in the natural sciences today [e.g. 16, 31].

Fodor introduces the terms "horizontal" vs. "vertical" to describe two different sorts of decomposition of intelligence. *Horizontal* decompositions for Fodor are those which identify processes (e.g. memory, attention, perception, judgement) which underlie all of cognition. *Vertical* decompositions identify particular skills or faculties (e.g. mathematics, language, metaphysics) which each have their own characteristic processes of memory, attention and so forth. Roughly speaking, evidence for horizontal decomposition is the extent to which performance across domains is correlated for a particular individual; evidence for vertical decomposition is the extent to which it is not.

Fodor himself believes that *part* of human intelligence is decomposed in this vertical sense; that part being perception. In Fodor's system, a number of semi-autonomous perceptual modules run simultaneously giving quick, automatic analysis of the perceptual scene. Each module recognizes its own best input, and effectively trumps the other modules when it is best utilized. The output of modules is in the language of thought, which is operated on by a horizontal reasoning system that then produces action.

3.2 Modules as Agents

Another modular theory immediately precursing BBAI was the "Society of Mind" [18, 27]. Minsky's proposal is more substantially vertical than Fodor's, although it still has some horizontal elements. An individual's actions are determined by simpler individual agencies, which are effectively specialists in particular domains. Minsky's agencies exploit hierarchy for organization, so for example the agency of play is composed of agencies of block-play, doll-play and so forth. Arbitration between agencies is also hierarchical, so the play agency competes with the food agency for the individual's attention. Once play establishes control, the block and doll agencies compete.

Minsky's agents have both perception and action, but not memory, which is managed by another network of agencies of a different sort. Memory (K) agencies are interconnected both with each other and with the other, actor (S) agents;

each can activate the other. Keeping the whole system working requires another horizontal faculty: the “B brain” which monitors the main (A) brain for internally obvious problems such as redundancy or feedback cycles. Minsky’s model relates to BBAI mostly as a series of contrasts: it attempts to account for all of human intelligence, but has never been fully implemented.

3.3 Modules as Finite State Machines

In contrast, the term “behavior-based artificial intelligence” was invented to describe a simplified but fully-implemented system used to control multiple, robotic agents. This was the subsumption architecture [3, 5]. The subsumption architecture is purely vertical. The modules were originally finite state machines, and arbitration between them was conducted exclusively by wires connecting the modules — originally literally, eventually as encoded in software. Each wire could connect one module to another’s input or output wires, the signal of which the first module could then either monitor, suppress or overwrite.

Brooks initially asserted that most apparent horizontal faculties (e.g. memory, judgement, attention, reasoning) were actually abstractions emergent from an agent’s expressed behavior, but had no place in the agent’s actual control [5, p. 146–147]. However, his system was rapidly extended to have learning systems either inside modules or local to layers of modules [e.g. 4, 26]. Unfortunately, this promising approach was apparently smothered by the attractive simplicity and radicalism of his deemphasis on representation and centralized control.

3.4 Modules as Slaves and Bitmaps

Of the researchers who did *not* immediately adopt “no representation” as a mantra, most attributed the impressive success of Brooks approach to the fact that he had created abstracted primitives — the action/perception units. Because these primitive units could sort out many of the details of a problem themselves, they made the composition of intelligence under *any* approach relatively easy [25]. Thus behavior systems were incorporated as a component into a large variety of AI architectures which still maintained centralized, logic-based planning and learning systems [e.g. 2, 20]. In fact, due to the difficulty of reasoning about relatively autonomous components, some systems reduced behaviors to “fuzzy rules” [23] or vector spaces [1] which could be easily composed.

Despite the lack of commonality of such approaches to Brooks’ original ideal, they are still often called either behavior-based or hybrid behavior-based systems. Further, by the late nineties, the work of these researchers had so far outstripped that of the “pure” BBAI researchers that two significant publications declared this hybrid approach to have been demonstrated superior to non-hybrid ones [21, 24].

3.5 Agents as Modules

Given the attributes of BBAI outlined in Section 2, in some senses multi-agent systems (MAS) are closer to BBAI than these hybrid behavior-based systems. Each agent performs a particular task, and may have its own private knowledge store and representations which are presumably well suited to its function. However, to date there are fundamental differences between a MAS and a single, modular agent. These differences are due to issues of communication and arbitration between modules / agents. The MAS community is concerned with interoperability between unspecified numbers and types of agents, and with distribution across multiple platforms. This creates an administrative overhead not necessary for a single, modular agent[2].

3.6 Summary

In summary, BBAI was originally conceived and implemented as a clean, simple version of modular hypotheses that were already influential in psychology, linguistics and AI. It has lead to substantial improvements in real-time AI, and still has a great deal of influence not only in robotics [1, 24] but in virtual reality [30, 32]. However, it is famously difficult to program [30, 34]. This difficulty has supported the wide-spread acceptance of hybridization between behavior-based and traditional AI into layered architectures. These hybrids unfortunately lose many of the advantages that BBAI initially had to offer. In the next section, I suggest ways to reclaim the advantages of modularity.

4 Modular Decomposition and Specialized Learning

In the previous section I explained Fodor’s use of the terms “horizontal” and “vertical” to describe modular decompositions along generic function vs. task specific lines (respectively.) I also showed that the original behavior-based AI, the subsumption architecture, used by far the most strictly vertical modular decomposition. In this section I describe my own approach to BBAI and modular decomposition, which is largely determined by variable state for learning.

[2] Where MAS are in fact limited to a single platform and a relatively fixed architecture, I suspect their engineers may in fact be taking the wrong approach, and should consider them to be modular single agents. But this is a topic for another paper [9].

4.1 Modules as Objects

My approach to modular decomposition is not entirely original; it is inspired by object-oriented design [e.g. 15, 28]. Consequently, it is called Behavior-Oriented Design (BOD). Under BOD, modular decomposition is done along the lines of specialized representations underlying adaptive requirements for the agent to be implemented. As it happens, most of these representations tend to support vertical abilities, for example representations underlying navigation or language, but some of them reliably support horizontal abilities, such as behavior arbitration or smoothing motor control.

Although this suggestion is simple, I think it brings a great deal both to BBAI and to the understanding of learning in intelligent systems. Compared to the original BBAI, BOD provides for learning while simplifying behavior arbitration. Compared to hybrid BBAI, BOD provides both a return to full modularity and a reemphasis on facilitating hand design.

In terms of understanding learning and anticipation in intelligent systems, BOD makes explicit the continuum of adaptivity underlying intelligent behavior. The BOD development process [see 8, 12] emphasizes two things:

- increasing the probability of success in learning (or any other type of search) by providing the agent with as much information (bias) as possible, and
- maintaining the simplicity of the agent by trading off complexity between various representations.

4.2 A Module for Behavior Arbitration

BOD particularly emphasizes the tradeoffs to be made between adaptive state for specialized perception and that for action selection through behavior arbitration [8]. This goes back to the notion of whether a module can, on its own, recognize a situation in which it should operate. I believe it is more reasonable for a module to recognize when it *can* operate. To recognize when it *should* operate requires more information than a largely encapsulated, semi-autonomous module ought to have access to.

In any particular context, there may well be more than one module that could or even should operate. This is the familiar problem of *perceptual aliasing*, which was originally seen as a problem of perception, but is in fact just a characteristic of the world. For example, consider a watch-robot intended to patrol an office space composed of corridors and junctions (Figure 1). For some junctions, the direction to go is entirely determined by either the robot's history (where it has most recently been) or its intentions (where it needs to go next.) Of course, we could try to read the robot's history or intentions off of its physical states (such as the direction it is pointing) but these can be perturbed by other subtasks such as avoiding people in the hallway.

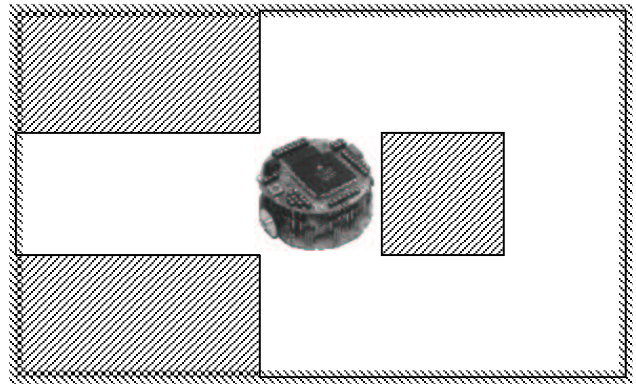

Figure 1. A patrolling robot cannot base its steering decisions entirely on external context and cover the entire maze.

The strategy of making behavior arbitration into a special, horizontal module allows for a tradeoff between the complexity of action selection and the complexity of perception. I have argued at length elsewhere that ideally there should be a structured hierarchical representation underlying behavior arbitration, which represents behavior ordering and prioritization given a particular context [6, 7]. The advantage of such a decomposition is that it simplifies knowledge acquisition by separating acquisition tasks that have minimal correlation between them. The behavior-arbitration module doesn't need to know how task modules recognize context or perform their tasks; task modules don't need to know what other tasks might be performed in the same location at the same time, or what their relative priorities are.

4.3 From Perception to Action

I will briefly return to the example domain of mobile-robot navigation in order to demonstrate the variety of adaptation usefully modeled in behaviors in the BOD system. Although the robot work described here is old [10], I find that the problems of robot perception and action are the most clear and intuitive for explaining the different requirements for variable state.

Figure 2 shows some behaviors I implemented on a radially symmetric, 16 sided Nomad 200 robot. These behaviors allow the robot choose its speed and precise direction given that it has already determined an approximate goal heading. The vertical modules have solid boxes, the horizontal ones (including the robot's body) are dashed. Beginning at the bottom of the figure, the robot provides four

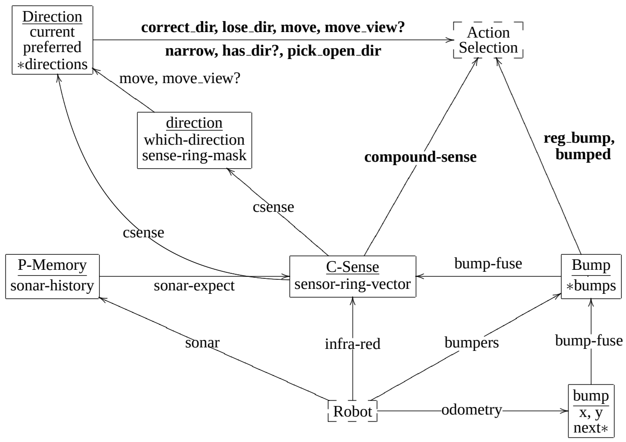

Figure 2. The behaviors involved in moving a robot forward.

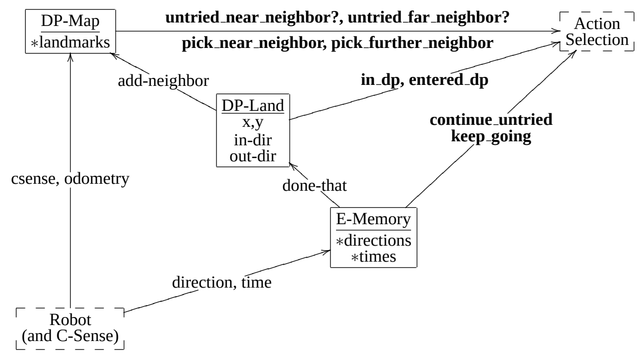

Figure 3. Additional behaviors for map learning.

types of sensory information relevant to picking a safe path. A **direction** behavior will determine the speed and direction for the robot, based on a 16 value array representing the approximate distance from each of the robot's faces to the next obstacle. This array is maintained by **C-sense**.

Sonar, infra-red and bumpers all give information about the location of obstacles. Sonar operates by emitting sound then listening for it to bounce off obstacles. It can be accurate from about 20cm to 6m, but is subject to a variety of deflections and interference which can make objects appear suddenly closer or further away. The behavior **P-Memory** processes this information with a simple 6 item memory buffer. Each time a new sonar reading is received (about 7 times a second) the reading for each sensor is compared with those of the previous half minute. If a major discontinuity is perceived in one reading, it is ignored, and a new one computed based on the previous average value. However, if the new reading persists for 2 more readings, it is then "believed" and becomes the new value for that sonar.

Infra-red sensors do not have the non-linearities of sonar, but have a far more limited range (approximately 0-24cm), and are also influenced by the color of the reflected surface. Infra-red sensors must be used for delicate maneuvers such as passing through doorways which require obstacle detection within the blind zone of the sonars. However, some things will not be detected by either long-range system, and are instead detected by the robots bumpers. The **bump** behaviors each represent one such past event. Since a bump *is* only detectable at the time and location of the event, the robot must compute the bumps approximate location after having disengaged from the obstacle in order to avoid it. This computation is based on odometric data. However, odometry accumulates errors rapidly, so bump events are forgotten after the robot has moved a few yards.

4.4 State as Knowledge

The robot thus brings a diverse array of "knowledge" to the continuous task of choosing a new speed and direction at any given instant. **Direction** and **Action Selection** work in concert for determining which **direction** controls these variables. **Direction** stores the current intended direction, while **Action Selection** determines the behavioral context (e.g. going forward normally toward a goal direction, or backing up after a collision). Each **direction** contains a template for determining discounts on the importance of the values of the array in **C-sound** pertaining to whether the particular array value is directly in front, mostly to the side, or behind the direction of motion before that **direction**'s face. The value of the discounts in the **direction** behaviors was learned off-line by the developer. The values in the **C-sound** array are determined anytime, based on the most recent infra-read reading, the last half second of sonar readings, and perhaps a few minutes of bumper readings.

None of this adaptation would be considered "learning" in the common usage of the term, because it does not change state permanently for the lifetime of the agent. Nevertheless, all this knowledge may be considered predictions which lead to adaptive behavior. For example, the state recording the last direction of motion is used to predict the next one, which in turn determines what values are used in computing the robot's velocities. Similarly, the historic sonar readings are treated as more predictive of the true distance to obstacles than any one current sensor reading. In essence, the only reason to have adaptive state in the robot is because the past can be used to predict the present, and can do so more reliably than sensors on their own.

The same general arrangement was used for map learning (see Figure 3, described further in [8, Section 7.6.3]). Here *decision points* — locations where the robot suddenly has a choice of direction (e.g. when it enters a room or encounters a doorway in a hall) are stored along with the decisions that were made at them, possibly after soliciting advice. Thus the robot can create a crude map, or in this case the British-English term of *plan* for map might be more appropriate. This particular robot does not learn a complete, connected 2-D representation of the world, but rather a set of cues that can be read from the environment in order to make future decisions autonomously.

5 Generic Types of Specialized State

The key observation about the robot example above is that BOD has been used to produce a reactive system which can operate well in a dynamic environment. It does this by exploiting a variety of types of information:

- *Engineering*, provided by the developer (or evolution), which does not change over the lifetime of the agent. This includes both fixed program code and parameters set by off-line tweaking and experimentation.

- *Reactive plans*, which keep track of the robots current decision context and focus its attention on particular behaviors. These are the basic representation underlying the Action Selection module.

- *Learned values of variable state*. Variable state is at the heart of the vertical / task modules. The 'learning' may persist only as very-short-term perceptual memory, as medium-term working memory, or for the lifetime of the agent.

This decomposition can also be found in real animals [13]. The engineered information is roughly equivalent to genetic predispositions, though notice that in real animals, it is more difficult to separate development from learning, since development has evolved to rely on ubiquitous features of the environment as an information source. Reactive plans play a similar role to the behavior of the forebrain in mammals at least, which, when working correctly, selects, sequences and inhibits behavior expression, though again note that in animals this can be more plastic than it is in BOD. Finally, the vertical behaviors I would equate with various sorts of cortical activation and plasticity. Notice that BOD doesn't currently discriminate between plasticity from activation levels and plasticity through long-term changes in connectivity.

These three types of information are not entirely disjoint: the reactive plans are hand coded, and are run in a special action-selection module. Reactive plans are themselves a very elaborate form of specialized variable state. They encode both engineered information in the form of contingencies the designer anticipates the agent will encounter, and variable state indicating recent decision-making context, which constrains choices in the immediate future in order to provide persistence and reduce search.

In fact, all modules mix engineering with variable state. What makes the reactive plans special is that both their representation and the code that exploits it are used in all BOD agents. Extensive research has lead me to believe the BOD reactive plans are simply the best way to do behavior arbitration in a modular single agent [6, 7, 11]. Obviously it would be useful to find other such generically useful representations, since reusing solutions reduces the development time on an agent. In the rest of this section, I will discuss three other biologically-inspired types of learning or plasticity, two of which I am currently developing under BOD.

5.1 Drives and Emotions

Because the reactive plans underlying BOD action selection are relatively fixed, they do not represent well the sorts of variation that the brain represents chemically such as drives for food or sleep, or emotional states such as anger or fear. The correct way to encode these sorts of variation in BOD is as behaviors. However, these "behaviors" are so stereotyped, and have such simple state (essentially a single drive level) that they are effectively their own type.

I have developed a class, *variable-drive-memory*, which has five variables:

- *total*: the level of the drive, which is normally raised by one per time-increment (time-increment is an adjustable value, which allows for varying simulation speed during debugging and experiments.)

- *last-update-time*: total isn't actually updated at every time increment, but rather when it is observed. The new value is computed using total, this value and the current real time.

- *latch*: a binary value for whether this drive will currently affect behavior; aids persistence. directly as well.

- *trigger-time*: when total gets to this value, *latch* is turned to true. If total gets to 0, *latch* is turned to false.

- *gratifactor*: only one value (*total*) is used to represent the level of the drive, but satiating the drive may take longer than activating it (as in hunger) or shorter (as may be true in anger, though see below.) This is the standard rate at which a gratifying action reduces the drive level.

The class also has methods for increasing or decreasing *total*, which might happen acutely or at accelerated rates. For example, if an agent is attacked it might get angry suddenly, or if it is running it might get hungry faster than the default rate. Another method allows switching the value of *latch* if *total* is currently between *trigger-time* and 0. This allows an agent to engage in a consummatory action for a drive opportunistically. For example, an agent that isn't hungry enough to actively seek food may eat some if it is offered in a social setting.

To date I have used this type of behaviors to create a simulation of a primate colony who's members have two drives: one for grooming and one for wandering alone (a stand-in for foraging.) I have been using this model to explore the impact of adding simple social behaviors (such as tolerance of grooming) on the time spent by the group as a whole pursuing their goals [9]. We are currently extending the model to include emotions such as anger and happiness.

5.2 Task Learning

The fact that BOD reactive plans are engineered bars BOD agents from doing something else real animals do: learn new tasks or new vertical modules. Again though, the extent to which animals have this capacity tends to be exaggerated in folk psychology. For example, pigeons can't learn to flap their wings for food or to peck to avoid shock, although they *can* learn to flap their wings to avoid shock or to peck to get food. Dogs can learn a large number of tricks, but not how to unwrap a leash they are tethered to if they have wound it around a pole.

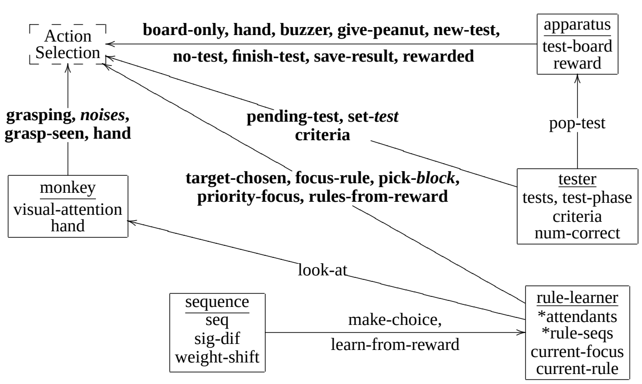

Figure 4. Behaviors used for an artificial life model of transitive inference learning. Notice that this single BOD agent represents two real-world agents, a monkey and a testing apparatus.

I have built a model that learns what is effectively one component of a reactive plan within a particular context. The context is a model of transitive inference learning as performed by animals and children [8, 33]. The model shows simultaneous learning of both context / action pairs, and a set of prioritizations between the different contexts. These prioritizations determine when more than one context applies, which action should be taken. This amounts to a reactive plan — a prioritized set of context / action pairs.

To date we have demonstrated that this models both human and non-human primate learning of transitive inference. I am currently working to extend the model further to model non-human primate learning of other tasks. In this model I anticipate learning not only the context / action pairs and their priorities, but also when new contexts or actions need to be discriminated, and how this impacts the task representation as a whole. The performance context the agent believes itself to be in will determine the set of things it might learn as well as the things it might do.

This sort of learning mechanism also has a biological correlate: the hippocampal learning system. I do not, however, expect that such a general-purpose horizontal hippocampal-learning module would become a typical component of real-time BOD agents. This sort of learning takes a great deal of time even when heavily biased, so defies the BOD principle of guaranteeing successful and timely learning. However, it is necessary for true mammalian learning.

5.3 Phenotype Swaps

Finally, I'd like to describe a very different form of natural plasticity. Hofmann and Fernald [22] have shown that both physical characteristics and expressed behavior can change extremely rapidly (within minutes) following a single traumatic (whether positive or negative) social event. The representations underlying these changes seem to be *phenotypic* in nature, with concurrent changes of gene expression in large numbers of neural synapses. The phenotypes in question determine whether a male Cichlid fish follows a behavior pattern of simple schooling, feeding and growth, or one of aggressive mating and territory defense which does not allow much time for growth. Male cichlid apparently alternate between these phenotypes. Not only behavior, but coloration change immediately after a decisive social event (a fight outcome), while gonad and overall size and shape gradually shift during the following weeks.

I have no immediate plans to model this sort of behavior, but it could be fairly easily done by implementing more than one action-selection plan hierarchy per agent, plus a special arbitration mechanism dedicated to swapping between these two plans. Since top-down expectations influence which behaviors are actively utilized by a BOD agent, this would effectively (though not actually) excite or inhibit other relevant behavior modules.

Whether this sort of adaptation can even be called learning is a somewhat murky question. The representation is not a mental structure, and could not be used or manipulated in any other way. Yet an event (the result of a fight) has been used to select a set of behavior which is only cost effective if that outcome serves to predict a reasonable period of success in near-future events of the same kind. The fish will only receive payoff for the hard work of defending a territory if it does so long enough to reproduce and protect its progeny. Thus this might be a sort of learning, the learning of a new social rank.

6 Conclusions

In this paper, I have described how modularity can be used to facilitate specialized learning for a situated agent. I have described the advantages of the modular approach taken by Behavior-Based Artificial Intelligence (BBAI), and suggested that its greatest strengths lie in its emphasis on the design process for creating intelligent agents, and its decomposition along task lines, which again makes for easier programming. I have argued that progress in this field has been hampered by the difficulty of designing behavior arbitration under early architectures, and by not exploiting the opportunity within a modular system to create specialized representations and learning systems. I have also proposed my own system, Behavior-Oriented Design (BOD) as an alternative model for moving BBAI forward. BOD does exploit specialized representations and provides for a simplified form of behavior arbitration in a single behavior module. As such, it is a useful methodology both for creating interesting artifacts and for modelling natural intelligence.

Acknowledgements

The robot in Figure 1 is a Khepera, ©K-Team SA Switzerland 2002. The Nomad robot work described in Section 4.3 was conducted in the University of Edinburgh Cognitive Neuroscience Laboratory, directed by Brendan McGonigle. Thanks to Will Lowe, Martin Butz, Olivier Sigaud and Samarth Swarup for their comments on earlier drafts. I would like to acknowledge both NSF Grant EIA-0132707 and Marc Hauser for support of this work.

References

[1] Arkin, R. C. (1998). *Behavior-Based Robotics*. MIT Press, Cambridge, MA.

[2] 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.

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

[4] Brooks, R. A. (1991a). Intelligence without reason. In *Proceedings of the 1991 International Joint Conference on Artificial Intelligence*, pages 569–595, Sydney.

[5] Brooks, R. A. (1991b). Intelligence without representation. *Artificial Intelligence*, 47:139–159.

[6] Bryson, J. J. (2000a). Cross-paradigm analysis of autonomous agent architecture. *Journal of Experimental and Theoretical Artificial Intelligence*, 12(2):165–190.

[7] Bryson, J. J. (2000b). Hierarchy and sequence vs. full parallelism in reactive action selection architectures. In *From Animals to Animats 6 (SAB00)*, pages 147–156, Cambridge, MA. MIT Press.

[8] Bryson, J. J. (2001). *Intelligence by Design: Principles of Modularity and Coordination for Engineering Complex Adaptive Agents*. PhD thesis, MIT, Department of EECS, Cambridge, MA. AI Technical Report 2001-003.

[9] Bryson, J. J. (2003). Where should complexity go? Cooperation in complex agents with minimal communication. In Truszkowski, W., Rouff, C., and Hinchey, M., editors, *Innovative Concepts for Agent-Based Systems*, pages 298–313. Springer.

[10] Bryson, J. J. and McGonigle, B. (1998). Agent architecture as object oriented design. In Singh, M. P., Rao, A. S., and Wooldridge, M. J., editors, *The Fourth International Workshop on Agent Theories, Architectures, and Languages (ATAL97)*, pages 15–30. Springer-Verlag.

[11] Bryson, J. J. and Stein, L. A. (2001a). Architectures and idioms: Making progress in agent design. In Castelfranchi, C. and Lespérance, Y., editors, *The Seventh International Workshop on Agent Theories, Architectures, and Languages (ATAL2000)*. Springer.

[12] Bryson, J. J. and Stein, L. A. (2001b). Modularity and design in reactive intelligence. In *Proceedings of the 17th International Joint Conference on Artificial Intelligence*, pages 1115–1120, Seattle. Morgan Kaufmann.

[13] Bryson, J. J. and Stein, L. A. (2001c). Modularity and specialized learning: Mapping between agent architectures and brain organization. In Wermter, S., Austin, J., and Willshaw, D., editors, *Emergent Neural Computational Architectures Based on Neuroscience.*, pages 98–113. Springer.

[14] Chapman, D. (1987). Planning for conjunctive goals. *Artificial Intelligence*, 32:333–378.

[15] Coad, P., North, D., and Mayfield, M. (1997). *Object Models: Strategies, Patterns and Applications*. Prentice Hall, 2nd edition.

[16] Coltheart, M. (1999). Modularity and cognition. *Trends in Cognitive Sciences*, 3(3):115–120.

[17] Dahl, T. S. and Giraud-Carrier, C. (2001). PLANCS: Classes for programming adaptive behaviour based robots. In Cleeremans, A. and Lewicki, P., editors, *AISB'01 Symposium on Nonconscious Intelligence: From Natural to Artificial*.

[18] Doyle, J. (1983). A society of mind. Technical Report 127, CMU Department of Computer Science.

[19] Fodor, J. A. (1983). *The Modularity of Mind*. Bradford Books. MIT Press, Cambridge, MA.

[20] Gat, E. (1991). *Reliable Goal-Directed Reactive Control of Autonomous Mobile Robots*. PhD thesis, Virginia Polytechnic Institute and State University.

[21] 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).

[22] Hofmann, H. A. and Fernald, R. D. (2000). Social status controls somatostatin-neuron size and growth. *Journal of Neuroscience*, 20:1248–1252.

[23] Konolige, K. and Myers, K. (1998). The Saphira architecture for autonomous mobile robots. In Kortenkamp, D., Bonasso, R. P., and Murphy, R., editors, *Artificial Intelligence and Mobile Robots: Case Studies of Successful Robot Systems*, chapter 9, pages 211–242. MIT Press, Cambridge, MA.

[24] Kortenkamp, D., Bonasso, R. P., and Murphy, R., editors (1998). *Artificial Intelligence and Mobile Robots: Case Studies of Successful Robot Systems*. MIT Press, Cambridge, MA.

[25] Malcolm, C., Smithers, T., and Hallam, J. (1989). An emerging paradigm in robot architecture. In *Proceedings of the International Conference on Intelligent Autonomous Systems (IAS)*, volume 2, pages 545–564, Amsterdam. Elsevier.

[26] Matarić, M. J. (1990). A distributed model for mobile robot environment-learning and navigation. Technical Report 1228, Massachusetts Institute of TechnologyArtificial Intelligence Lab, Cambridge, Massachusetts.

[27] Minsky, M. (1985). *The Society of Mind*. Simon and Schuster Inc., New York, NY.

[28] Parnas, D. L., Clements, P. C., and Weiss, D. M. (1985). The modular structure of complex systems. *IEEE Transactions on Software Engineering*, SE-11(3):259–266.

[29] Pebody, M. (1995). Learning and adaptivity: Enhancing reactive behaviour architectures in real-world interaction systems. In Moran, F., Moreno, A., Merelo, J., and Chacon, P., editors, *Advances in Artificial Life (Third European Conference on Artificial Life)*, pages 679–690, Berlin. Springer.

[30] Sengers, P. (1998). Do the thing right: An architecture for action expression. In Sycara, K. P. and Wooldridge, M., editors, *Proceedings of the Second International Conference on Autonomous Agents*, pages 24–31. ACM Press.

[31] Spelke, E. S. (2003). What makes us smart? Core knowledge and natural language. In Gentner, D. and Goldin-Meadow, S., editors, *Advances in the Investigation of Language and Thought*. MIT Press, Cambridge, MA.

[32] Thórisson, K. R. (1998). Real-time decision making in face to face communication. In Sycara, K. P. and Wooldridge, M., editors, *Proceedings of the Second International Conference on Autonomous Agents (Agents '98)*, pages 16–23, Minneapolis. ACM Press.

[33] Wood, M. A., Leong, J. C. S., and Bryson, J. J. (2004). ACT-R is *almost* a model of primate task learning: Experiments in modelling transitive inference. In *The* 26^{th} *Annual Meeting of the Cognitive Science Society (CogSci 2004)*. to appear.

[34] Wooldridge, M. and Jennings, N. R. (1995). Intelligent agents: Theory and practice. *Knowledge Engineering Review*, 10(2):115–152.