Modularity and Specialized Learning:
Reexamining Behavior-Based Artificial Intelligence

Joanna J. Bryson
Artificial models of natural Intelligence
University of Bath, BA2 7AY, United Kingdom
J.J.Bryson@bath.ac.uk

Abstract

Learning, like any search, is only tractable if it is tightly focused. Modularity can provide the information a learning system needs by supporting specialized representation. Behavior-based artificial intelligence is a well-known modular theory of intelligent design, but has not been used systematically in this way. This paper describes a new design methodology, behavior-oriented design (BOD), which does. Examples drawn from mobile robotics and models of learning in non-human primates show the diversity of information this approach supports, from implicit and perceptual learning to tasks, maps and relationships.

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. 24, 28], 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 BBAI advances 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, using examples from both robotics and ALife models of non-human primates. BOD allows a system with preprogrammed reactive control to behave in an adaptive manner, because its control relies on modules containing variable state. These modules combine current sensor readings with predictions based on learning.

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 complex dynamic environments.

When BBAI was introduced by Brooks [3, 5], its primary purpose 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.Choosing an optimal next action is impossible [15]. Even choosing a pretty good one requires searching an enormous space of possibilities.

Because an individual agent does not have time for such a search in real time, most of its decisions must be made in advance of its active life. However, this does not remove the complexity of the decision nor the amount of search necessary for a pretty-good choice. 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, and through exploiting the scaffolding of design knowledge we have previously learned.

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 are:

- the increased emphasis on providing engineered knowledge, which is a side effect of the emphasis on bottom-up control (sensing not representing), and
- the modular decomposition around individual expressed behaviors. This exploits the designers' existing skills and talents for writing simple programs.

After these significant advances, the complexity of the agents being built seems to have plateaued before the development of animal-level intelligence [22, 26]. Again, I believe there were two primary causes:

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 increases.

The first point is key to the thesis of this paper: 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 which will support the claims I outlined above. More extensive reviews of the BBAI literature are available [6, 8], as are more thorough comparisons to neural and psychological theories [8, 10].

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. 14].

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, judgment) 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 believes that *part* of human intelligence is decomposed in this vertical sense; those parts being perception and (separately) action. 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 these modules is in the language of thought, which is operated on by a horizontal reasoning system. He presumes that the reasoning system's output is interpreted into action in a similar way, but theorizes less about this process.

Another precursor of BBAI is the "Society of Mind" [18, 29]. 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 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 such as block-play and doll-play. 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.

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, judgment, 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, 28]. Unfortunately, this promising approach was apparently smothered by the attractive simplicity and radicalism of his deemphasis on representation and centralized control.

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 modules. 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 [27]. 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, 21]. Unfortunately, the difficulty of reasoning about relatively autonomous components motivates the trivialization of behavior modules, e.g. to "fuzzy rules" [25] or vector spaces [1] which can 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 [22, 26].

Given the attributes of BBAI outlined earlier, in some senses multi-agent systems (MAS) are closer to BBAI than 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 can be seen in the emphasis on communication and negotiation between modules / agents [35]. 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[1].

In summary, BBAI was originally conceived and implemented as a clean, simple version of modular hypotheses that were already influential in psychology and AI. It lead to substantial improvements in real-time AI, and still has a great deal of influence not only in robotics [1, 26] but in virtual reality [33]. However, it is famously difficult to program [33, 35]. This difficulty has supported the widespread acceptance of hybridization between behavior-based and traditional AI. Unfortunately, these hybrids lose many of the advantages of modularity. The next section suggests ways to reclaim these advantages.

4 Modularity and 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 the most strictly vertical modular decomposition. In this section I describe my own approach to BBAI and modular decomposition — that is, the problem of deciding how many modules an agent needs and what should be their capacities.

I believe modular decomposition should be determined by the requirements of variable state needed for learning. This idea is not entirely original; it is inspired by object-oriented design [e.g. 16, 30]. Consequently, I call it 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. Most of these representations will 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, including animals. 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 in intelligent systems, BOD makes explicit the continuum of adaptivity underlying intelligent behavior. The BOD development process [see 8] 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.

4.1 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.

[1] Where MAS are in fact limited to a single platform and a relatively fixed architecture, I suspect their engineers should consider them to be modular single agents [9].

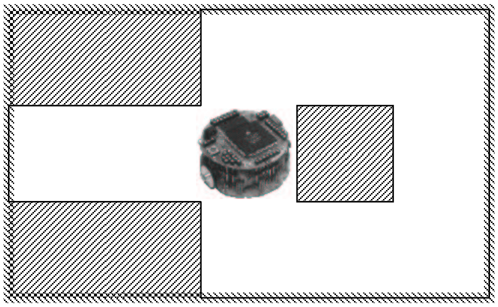

Figure 1. A patrolling robot cannot base its steering decisions entirely on external context and cover the entire maze.

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.) 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. It is far simpler to keep a brief history of decisions or intentions in the specialized state that supports arbitration.

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.

4.2 From Perception to Knowledge

I will use the 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 [12], the problems of robot perception and action provide clear and intuitive explanations for the different requirements for variable state. The robot, a radially symmetric, 16 sided Nomad 200, navigated in a smooth, continuous fashion around an office environment, negotiating doorways barely wider than itself and avoiding obstacles. It also learned simple paths from instruction.

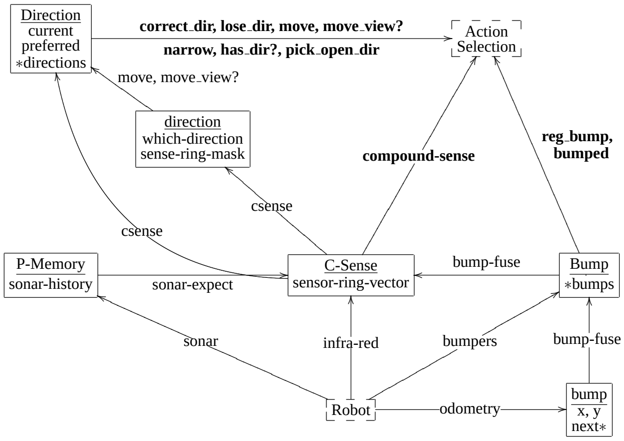

Figure 2. Behaviors for moving a robot.

Figure 2 shows behaviors that allow the robot to 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 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** (compound 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. Perceptual memory, **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 bump's 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.

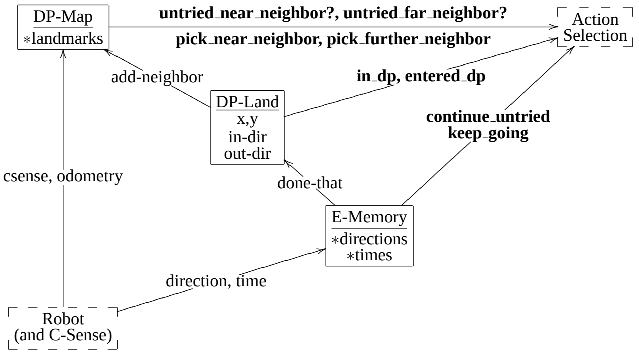

Figure 3. Behaviors added for map learning.

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-Sense** 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 discount templates in the **direction** behaviors was learned off-line by the developer. The values in the **C-Sense** array are determined at any time, 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. 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.

This argument extends to the modules that do 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 map (or learn a plan) from an instructor. 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. Nevertheless, it behaves as if it has learned its way around. Now the common usage of 'learning' does apply, but the knowledge system is fundamentally the same.

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 [10, for more details]. The engineered information is roughly equivalent to genetic predispositions, though 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 vertebrate forebrain [31], which, when working correctly, selects, sequences and inhibits behavior expression [13], 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. BOD does not 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 an 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 [7, 8]. Obviously it would be useful to find other such generically useful representations, since reusing solutions reduces development time. 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. Drives and emotions represent and intermediate time course for intelligent state between the electrical / neural firing rate (which BOD represents in reactive plans) and long-term potentiation (which BOD stores in modules.) Reactive agents without this sort of state can seem erratic [33]. We are currently exploring how including this sort of durative decision state influences action selection, both from the perspective of believability (for VR agents) and for evolving social behavior (in Artificial Life agents.

The way to encode variable state in BOD is as behavior modules. However, these behaviors are so stereotyped, and have such simple state (essentially a single drive level) that they are effectively their own type. We are consequently developing a standardized representation for modeling of emotions and drives. Although the drive level itself is simple variable, each drive or emotion has its own onset and decay characteristics [34]. Further, the interactions between these states — with each other and with standard action selection — varies. For example, there must either be a latching or a blending mechanism to decide which of two conflicting drives or emotions is expressed.

To date we have used this type of behaviors both to create realistic real-time facial animation [34] and to create a simulation of a primate colony. The primate colony is the first exploration of combining all three types of intelligent state together. Its members have two drives: one for grooming (a stand in for general social behavior) 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 social model to include emotions or drives such as anger and affinity in an effort to model differences in different species of primates social structures.

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 [see further 20, 32].

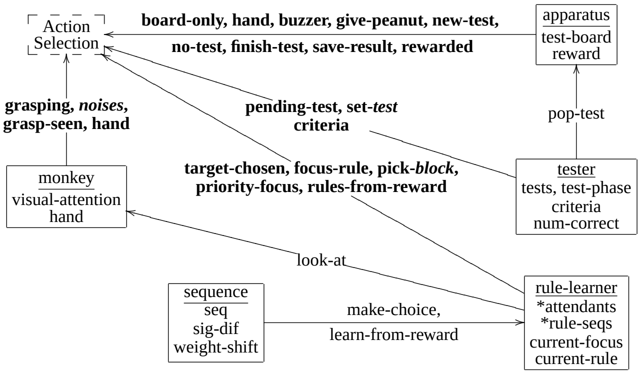

Figure 4. Behaviors used for an artificial life model of transitive inference learning.

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, 11]. 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 this model. True task learning should include not only 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 task-learning mechanism also has a biological correlate: the hippocampal learning system [11]. Nevertheless, such a general-purpose horizontal task-learning module should probably not become an expected component of all BOD agents. Such open-ended learning takes a great deal of time even with heavy bias, so defies the BOD principle of guaranteeing successful and timely learning. However, it is necessary for true mammalian intelligence.

5.3 Phenotype Swaps

Finally, I'd like to describe a very different form of natural plasticity. Hofmann and Fernald [23] 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.

Is this learning? The representation involves no mental structure, and could not be used or manipulated in any other way. Yet an event (the result of a fight) selects 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. Again, adaptation is a continuum, and this perhaps the ultimate vertical module.

6 Conclusions

In this paper, I have described how modularity can be used to facilitate specialized learning, and shown how this is central to intelligent behavior. I have concentrated on how this perspective illuminates the history of modular AI, but I believe it also informs the modularity debate in psychology, and provides some explanation for the modularization we see in the brain. In the future I hope that modular AI will be able to do for psychology and systems neuroscience what neural networks research has done for neuroscience — provide testbeds and intuition pumps to help the natural sciences form and refine their models.

Acknowledgments

Thanks to Will Lowe, Marc Hauser (NSF Grant EIA-0132707), Martin Butz, Olivier Sigaud and Samarth Swarup. The robot in Figure 1 is a Khepera, ©K-Team SA Switzerland 2002. The Nomad robot work was in the University of Edinburgh Cognitive Neuroscience Laboratory.

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. (2004). Modular representations of cognitive phenomena in AI, psychology and neuroscience. In Davis, D., editor, *Visions of Mind*. Idea Group.

[11] Bryson, J. J. and Leong, J. C. S. (2004). Representations underlying transitive choice in humans and other primates. submitted.

[12] 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* 4^{th} *International Workshop on Agent Theories, Architectures, and Languages (ATAL97)*, pages 15–30. Springer-Verlag.

[13] Carlson, N. R. (2000). *Physiology of Behavior*. Allyn and Bacon, Boston.

[14] Carruthers, P. (2003). The cognitive functions of language. *Brain and Behavioral Sciences*, 25(6).

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

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

[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] Gallistel, C., Brown, A. L., Carey, S., Gelman, R., and Keil, F. C. (1991). Lessons from animal learning for the study of cognitive development. In Carey, S. and Gelman, R., editors, *The Epigenesis of Mind*, pages 3–36. Lawrence Erlbaum, Hillsdale, NJ.

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

[22] 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).

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

[24] Horswill, I. D. (1993). *Specialization of Perceptual Processes*. PhD thesis, MIT, Department of EECS, Cambridge, MA.

[25] Konolige, K. and Myers, K. (1998). The Saphira architecture for autonomous mobile robots. In [26], pages 211–242. MIT Press, Cambridge, MA.

[26] 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.

[27] 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.

[28] 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.

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

[30] 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.

[31] Prescott, T. J., Redgrave, P., and Gurney, K. (1999). Layered control architectures in robots and vertebrates. *Adaptive Behavior*, (7):99–127.

[32] Roper, T. J. (1983). Learning as a biological phenomena. In Halliday, T. R. and Slater, P. J. B., editors, *Genes, Development and Learning*, volume 3 of *Animal Behaviour*, chapter 6, pages 178–212. Blackwell Scientific Publications, Oxford.

[33] Sengers, P. (1998). Do the thing right: An architecture for action expression. In Sycara, K. P. and Wooldridge, M., editors, *Proceedings of the* 2^{nd} *International Conference on Autonomous Agents*, pages 24–31. ACM Press.

[34] Tanguy, E. A. R., Willis, P. J., and Bryson, J. J. (2003). A layered Dynamic Emotion Representation for the creation of complex facial animation. In Rist, T., Aylett, R., Ballin, D., and Rickel, J., editors, *Intelligent Virtual Agents*, pages 101–105. Springer.

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