Integrated Systems Based on Behaviors

Rodney A. Brooks
MIT Artificial Intelligence Laboratory
545 Technology Square #822
Cambridge, MA 02139
brooks@ai.mit.edu

Abstract

Behavior based systems require an orthogonal view of integration issues. In this paper we highlight those issues, discuss what is easy, what is hard, and where the research frontiers lie.

Introduction

At MIT we have been investigating structuring of intelligence based on a decomposition into behaviors, each of which connects sensing to action. Functional modules such as planners and learners do not appear as such, but instead planning, learning, etc., can be observed to be happening in the complete system. The behaviors are the building blocks, and the functionality is emergent. This differs from the traditional approach in which the functional modules are the building blocks and the behaviors are emergent.

Given this fundamental design choice, and once we have imbued our robots with multiple behaviors to handle a variety of circumstances and achieve a variety of tasks, we are faced with the problem of deciding which behavior or behaviors should be active at any particular time. There are two components to this decision—how to select potentially correct behaviors in the circumstances, and how to resolve conflicts between behaviors. More specifically, we need to consider the following issues:

Coherence: Even though many behaviors may be active at once, or may be actively switched on or off, the creature should still appear to an observer to have coherence of action and goals. It should not be rapidly switching between inconsistent behaviors, nor should two behaviors be active simultaneously, if they interfere with each other to the point that neither operates successfully.

Salience: The behaviors that are active should be salient to the situation the creature finds itself in—it should recharge itself when the batteries are low, not when they are full.

Adequacy: The behavior selection mechanism must ensure that the long term goals that the creature designer has for the creature are met. E.g., a floor cleaning robot should successfully clean the floor in normal circumstances, besides doing all the ancillary tasks that are necessary for it to be successful at that.

At the implementation or algorithmic level, there are at least two fundamental questions of organization for a behavior selection mechanism:

- Should the behavior selection mechanism be centralized or decentralized?
- Should the conflict resolution scheme be fixed priority or dynamically reconfigurable?

Many of the problems above must also be addressed by similar approaches, such as [**Rosenschein and Kaelbling 86**] and [**Agre and Chapman 87**]. [**Kaelbling 89**] and [**Chapman 90**] both address the problem through compile-time specification of behavior precedence.

We will relate these questions to recent ethological and neuroscience work, and show some novel ways of organizing behaviors on a 23 actuator, 150 sensor, six legged robot. In particular we will introduce a model of animal hormone systems and discuss how it can be used to integrate many behaviors into a single system.

The approach we use is *behavior-based programming* [**Brooks 90**], which is a refinement of the earlier *subsumption architecture* [**Brooks 86**]. Neither this, nor the other similar architectures, use symbols in the conventional way. They all have virtual *wires* connecting computational units of some sort. The meaning of what passes along a wire depends on where the wire is connected, and is not interpretable symbolically by any other part of the system.

An Example

Attila is a six legged robot. It has 150 sensors, 23 actuators, and 11 onboard processors. Mechanically it is capable of quite complex and impressive behavior. The challenge is to connect the sensing and actuation in such a way that it exhibits such impressive behavior autonomously.

The Physical Robot

The Attila robot [**Angle and Brooks 90**] is 35cm long, and about 30cm wide in its normal stance. Each leg has three degrees of freedom, including one degree suited to climbing which is strong enough for a single leg to lift the entire robot. There is a 19th global degree of freedom which lets the robot keep its legs vertical while its body is pitched—this is very useful for stability on steep slopes. This feature also gives the robot the ability to flip its legs around and walk upside down, should it be flipped onto its back. There is an active 25cm long whisker controlled by two more motors, and a pan–tilt head that can operate at up to 800 degrees per second.

On each leg there are strain gauges giving a three axis read-out, peak strain detectors, position encoders, velocity sensors, a downward looking infrared range (almost contact) sensor in the foot, sensitive to 3mm, and forward and backward looking infrared range sensors that detect obstacles out to about 5cm. On the body there are pitch, roll and flip sensors, along with gyroscopes that are used as part of an inertial navigation system. The whisker has strain gauges mounted on it so that collisions, along with the approximate location along the whisker, can be detected. The head includes a 3m infrared range sensor, a 192×165 pixel CCD camera, a downward angled range sensor, and a downward looking visible light color sensor.

The main processor is a 68000 (actually a Signetics 68070) as is the vision processor. The main processor is programmed in the Behavior Language [**Brooks 90**], based on subsumption [**Brooks 86**]. The vision processor is programmed in C. The remaining 9 processors are small 8 bit microprocessors (Signetics 83C751) with only 64 bytes of RAM. They run servo code for the motors, and control A/D chips to read the sensors. The processors are connected on a LAN known as the I^2C bus, which runs at a peak of 100kilobits per second. We effectively run it at about 16kilobits per second.

The Environment and Task

In an earlier six legged robot, Genghis, we used subsumption to get it to carry out rough terrain predation [**Brooks 89**], and hte Behavior Language to learn to coordinate its leg motions so as not to fall down [**Maes and Brooks 90**].

For Attila we are interested in a much more comprehensive set of behaviors. Furthermore we use an organization for walking inspired by observations of stick insects [**Cruse 90**].

The particular domain that we are using to test Attila is fully autonomous planetary exploration. In the complete scenario the robot is intended to operate for many days without any external commands. It is to take care of recharging its batteries from solar power, shutting itself down in a low energy warm-mode overnight, wander around in local exploratory mode, climb over rough terrain, do long traverses to get to new areas, recover from falls, carry out measurements with scientific instruments, and choose interesting views to digitize and radio back to a relay station.

The Integration Organization

At this writing the behaviors for Attila have only been written for low-level survival and walking. Even so there are many issues of behavior arbitration to be handled.

The lowest level behavior is *sleep*. It causes the robot to shut down all motor systems and active sensors and conserve power. It is induced by falling battery voltage. As the sleep behavior is deactivated the various subsystems of the robot must be reactivated in a reasonable sequence. For this reason the sleep behavior is also active when the robot is first powered up, and is deactivated over the first few seconds.

The next level of behaviors (slight misnomers, *flipped?* and *flipper*) deal with making sure its body and legs are both aligned so that the robot can stand up. If the robot falls on its back this becomes or important, or if it is powered up with its legs and body grossly misaligned it is also important.

The *standup* behavior directly controls the legs to make the robot stand up.

The *tilter* behavior uses the same sensors and actuators as the flipping behaviors, but tries to keep the body oriented parallel to the surface underneath so that the legs of the robot are always exactly vertical. This behavior should operate whenever the robot is standing up or walking about.

A collection of *leg* behaviors, one for each physical leg, control backward and forward strokes, and backward and forward recovery motions of each leg. Besides self calibration we plan to use a learning network in each of these behaviors to minimize total lateral strain on the legs, and thus straighten leg motions without resorting to heavy runtime kinematic computations (the ideal paths would never be followed anyway due to lags in the communications with the servo processors, so an approximation can in practice achieve comparable performance).

A set of *vleg* virtual leg behaviors are used to coordinate walking. Depending on whether the robot is to go forward or backward, and depending on whether it is upright or upside down, the virtual legs are mapped to different physical legs (i.e., *leg* behaviors) via the *mapper* behavior. The virtual legs are modeled after stick insects [**Cruse 90**] in their mutual inhibition and excitation of strokes and recoveries, generating walking rhythms modulated by local obstacle detection, and ground conditions.

More behaviors will be added to Attila over the next few months rounding out its behavior repertoire to the full mission scenario.

The Issue

The key issue is how to coordinate the behaviors competing for the same actuator resources. We need a way of activating and de-activating behaviors. We have adopted the idea of [**Maes 89**] to use an activation level for behaviors, which can switch them on or off.

Maes' work is primarily concerned with the spreading of activation in order to activate appropriate behaviors to achieve subgoals. This mechanism is included in the Behavior Language, but the primary mechanism we use for behavior selection on Attila, at least on the lower level behaviors so far implemented, uses the activation idea in a slightly different way.

On Attila we use an activation system modeled after animal hormone systems. Hormones in biological systems can be viewed as a low bandwidth global communication scheme. However they are not a centralized control system. There is no need for a central agent in the system, beyond the medium for hormone transmission. Release of hormones can happen locally, and activation of behaviors can be purely local without any explicit point to point communication.

The hormone system provides a global repository of state which predisposes appropriate behaviors to be active. It also provides a switching mechanism to smoothly switch on and off appropriate behaviors as the global state changes. In particular this leads to *coherence* and *salience* in the behavior of the robot.

[**Kravitz 88**] describes a model of hormonal control of behavior. Following his own summary the major points of this model are:

1. Sensor inputs enhance the release of hormones.
2. A hormone may be released from tissues or neurons and may act in isolation or in concert with other substances.
3. The hormone finds receptors where it either stimulates a new behavior pattern or enhances or diminishes an existing pattern.
4. The things affected by hormones may include sensory elements, groups of neurons in higher processing centers, and motor or hormonal output systems.
5. The method of stimulation is a gain setting mechanism, biasing the output of the organism in particular directions.
6. The change is apparent in that the organism then responds to particular sensory inputs with altered outputs.
7. Individual organisms may respond to particular hormones with different levels of effectiveness—determined by genetic or environmental influences.
8. The affected circuit may itself further enhance or diminish the release of the hormonal substance.

Kravitz uses the lobster as his primary instantiation of this model. On Attila we use an instantiation of this model which successfully demonstrates seven of the listed eight properties (number seven being the only one not accounted for).

In our model, sensory conditions do not directly release hormones. Rather we use a two stage mechanism so that summation of hormone quantities can be centralized—this is clearly different from the way it might be implemented in a biochemical system.

Any computational process can *excite* a *condition*. Labels we have used for conditions in our programs include such character strings as *drowsiness* and *panic*. Multiple processes can (pseudo) simultaneously excite a single condition—their effects are additive. In our model, the excitation level of a condition is an integer in the range 0 to 15. A process can excite a condition with any increment to this level that it chooses.

The excitation level of a condition decays naturally. As it happens, the default decay rate is linear and one twelfth of a unit per second. The programmer can specify the decay rate as a bilinear function. The semantics are such that above a specified threshold it is possible to have a negative decay rate so that a condition can become self sustaining. The decay rates are set at program compile time and cannot be altered dynamically.

Releasers correspond more directly to hormones. These are the things that actually influence behaviors. A releaser's value at any point in time is an instantaneous function of the current levels of excitement of some set of conditions. There are no a priori restrictions on the value that can be taken by a releaser. For the experiments with Attila the labels used for releasers in our programs include such character strings as *adrenaline* and *sleepine*.

Behaviors are implemented as a collection of augmented finite state machines. Each AFSM is a computational process. A behavior can be *active* or *inactive*. The difference between these states is that different computational processes are allowed to run. A behavior can have three classes of processes:

- **Regular processes:** These processes always run, and any messages they send out reach their destinations.
- **Haltable processes:** These processes only run when the behavior is active. Otherwise they do not run, do not process their inputs and do not send out any messages.
- **Inhibitible processes:** These processes always run but when the behavior is inactive their outputs are inhibited or blocked. They still receive all their inputs however and can retain and change internal state.

Following [**Maes 89**], each behavior has an activation level, and the behavior becomes active when it passes some threshold[1]. It is a continuously recomputed value and does not decay in any sense. The

[1] There is some user controllable hysteresis in the system as the activation level falls below the threshold once again.

original idea of activation was that it should be spread from behavior to behavior as a behavior tried to get certain preconditions met by activating other behaviors. For the hormonal model the activation level can also be a function of releasers.

The computational model of hormones is rather similar to Kravitz' original biological model.

1. *Sensor inputs enhance the release of hormones.* In our experiment, behaviors which process sensor readings can excite a condition, which in turn can contribute to the level of a releaser (or hormone).

2. *A hormone may be released from tissues or neurons and may act in isolation or in concert with other substances.* Any behavior is free to excite a condition and hence change a releaser level. Furthermore, activation levels of behaviors can be functions of many releasers, and thus a releaser may be acting by itself or it may be acting in concert with other releasers.

3. *The hormone finds receptors where it either stimulates a new behavior pattern or enhances or diminishes an existing pattern.* If a releaser pushes a behavior's activation level beyond its threshold then a new behavior pattern can indeed be initiated. Furthermore, since activation levels are accessible to behaviors it is possible to have this control certain parameters of a behavior, for instance the walking speed of the robot.

4. *The things effected by hormones may include sensory elements, groups of neurons in higher processing centers, and motor or hormonal output systems.* In our system all such things are controlled by behaviors, and these can be affected by the hormones.

5. *The method of stimulation is a gain setting mechanism biasing the output of the organism in particular directions.* This was the original idea borrowed from [**Maes 89**]. By having releasers change the activation level of behaviors it can mean that either less or more spreading of activation from another behavior will be needed in order for this behavior to activate. Thus indeed the releasers can bias whether a particular behavior is activated, and thus bias the overall behavior of the robot.

6. *The change is apparent in that the organism now responds to particular sensory inputs with altered outputs.* This is the point of behavior activation. For instance, in the example programmed for this behavior when the robot has been walking over rough terrain, it is much more likely to become "fearful" from having its whiskers stimulated.

7. *Individual organisms may respond to particular hormones with different levels of effectiveness—determined by genetic or environmental influences.* This aspect of the model is not included in our experiments.

8. *The affected circuit may itself further enhance or diminish the release of the hormonal substance.* There are two ways in which our implementation can do this. First, it is possible to make a condition decay negatively above some threshold, effectively keeping the condition's value at its maximum level. Second, the affected circuit might itself excite the condition further.

Evaluation of the Architecture

In this volume [**Mataric 91**] addresses many issues of evaluation of the subsumption architecture and the Behavior Language, including the use of active representations [**Mataric 90**]. In this paper we will confine ourselves to evaluating the biologically inspired aspects of the architecture.

The hormone system is a key mechanism of integration. The components it integrates are behaviors. In the following paragraphs we evaluate its strengths and weaknesses as an integration mechanism within a behavior-based approach.

Generality: We believe the hormone system provides a reasonable substrate for the basic mobility and survival behaviors of an autonomous robot. Other behavior arbitration mechanisms may be needed for higher level behaviors.

Versatility: Any behavior can be linked into the hormone system. The idea is that the hormones describe a few overall states of the system, although they are shaded states which smoothly slide from one to the other. New behaviors can cut out their own new 'state' that they are interested in, by defining appropriate thresholds. In practice many behaviors will be orthogonal, and so only those that interfere with each other need to use the hormone state to resolve activation priorities.

Rationality: The hormone system is meant to coordinate low-level behaviors concerned with survival and navigation. Higher level behaviors that have access to more sophisticated information may eventually want to suppress certain aspects of the hormone system in particular very complex situations.

Ability to add new knowledge: Knowledge is not explicit in behavior-based approaches.

Ability to learn: Currently we have no learning within the hormone system. That could potentially be added in at least two ways. The thresholds for activation of behaviors could be adjusted or the expressions for activation level could be learned in the style of [**Maes and Brooks 90**].

Taskability: This is an orthogonal issue.

Scalability: There are two axes to consider in scalability. One is the total size of the system, and the other is the number of hormones, and hormone induced state. On the first axis, if one considers compiling the complete system to silicon then the amount of area, or wires, devoted to hormone communication is linear in the total size of the system—thus it scales well. On the other hand the number of conditions and releasers can not grow beyond some fairly small number. Their interactions would become unwieldy. (See versatility above.)

Reactivity: The system is very reactive. In fact the hormone system has some damping added into it to make it slightly less reactive to high frequency changes in the environment.

Efficiency: Questions of efficiency are different within subsumption like systems, from in classical systems. The hormone system adds very little overhead to existing behaviors—in practice only a few machine instructions per behavior need be executed per cycle through the underlying scheduler which simulates parallelism.

Psychological or neuroscientific validity: The model is based on recent neuroscientific studies [**Kravitz 88**].

Conclusion

As we get more complex behavior-based systems we will need many different levels of integration. In this paper we have discussed an approach to integration of the low level behaviors necessary for basic survival and navigation. As we move to higher level behaviors we will need to have other behavior integration mechanisms operating in parallel.

Acknowledgements

The research reported here was done at the MIT Artificial Intelligence Laboratory. Support for this research was provided in part by the University Research Initiative under Office of Naval Research contract N00014–86–K–0685, and in part by the Advanced Research Projects Agency under Office of Naval Research contract N00014–85–K–0124.

References

[**Agre and Chapman 87**] "Pengi: An Implementation of a Theory of Activity", Philip E. Agre and David Chapman, *Proceedings, AAAI-87*, Morgan Kaufmann, Los Altos, CA, 1987.

[**Angle and Brooks 90**] "Small Planetary Rovers", Colin M. Angle and Rodney A. Brooks, *IEEE International Workshop on Intelligent Robots and Systems*, Tsuchiura, Japan, July 1990, 383–388.

[**Brooks 86**] "A Robust Layered Control System for a Mobile Robot", Rodney A. Brooks, *IEEE Journal of Robotics and Automation*, RA-2, April 1986, 14-23.

[**Brooks 89**] "A Robot that Walks; Emergent Behavior from a Carefully Evolved Network", Rodney A. Brooks, *Neural Computation*, 1:2, Summer 1989.

[**Brooks 90**] "The Behavior Language; User's Guide", Rodney A. Brooks, *MIT AI Lab Memo 1127*, 1990.

[**Chapman 90**] "Vision, Instruction and Action", David Chapman, *MIT AI Lab Tech Report #1204*, April 1990.

[**Cruse 90**] "What Mechanisms Coordinate Leg Movement in Walking Arthropods", Holk Cruse, *Trends in Neuroscience (TINS)*, 13:1, 1990, 15–21.

[**Kaelbling 88**] "Goals as Parallel Program Specifications", Leslie P. Kaelbling, *Proceedings, AAAI-88*, Saint Paul, MN, August, 60–65.

[**Kravitz 88**] "Hormonal Control of Behavior: Amines and the Biasing of Behavioral Output in Lobsters", Edward A. Kravitz, *Science 241*, September 30, 1775–1781.

[**Maes 89**] "The Dynamics of Action Selection", Pattie Maes, *IJCAI-89*, Detroit, MI, 991–997.

[**Maes and Brooks 90**] "Learning to Coordinate Behaviors", Pattie Maes and Rodney A. Brooks, *AAAI-90*, Boston, 796–802.

[**Mataric 90**] "A Distributed Model for Mobile Robot Environment Learning and Navigation", Maja J Mataric, *MIT AI Lab Tech Report #1228*, May 1990.

[**Mataric 91**] "Behavioral Synergy Without Explicit Integration", Maja J Mataric, *these proceedings*, March 1990.

[**Rosenschein and Kaelbling 86**] "The synthesis of digital machines with provable epistemic properties", Stan Rosenschein and Leslie Kaelbling, *Proceedings of the Conference on Theoretical Aspects of Reasoning About Knowledge*, 1986.