



ELSEVIER

Journal of Cognitive Systems Research 2 (2001) 81–93

**Cognitive Systems**  
RESEARCH

www.elsevier.com/locate/cogsys

# Learning in behavior-based multi-robot systems: policies, models, and other agents

Action editor: Ron Sun

Maja J. Matarić

*Computer Science Department, University of Southern California, 941 West 37th Place, Mailcode 0781,  
Los Angeles, CA 90089-0781, USA*

Received 14 January 2001; accepted 18 January 2001

---

## Abstract

This paper describes how the use of behaviors as the underlying control representation provides a useful encoding that both lends robustness to control and allows abstraction for handling scaling in learning, focusing on multi-agent/robot systems. We first define situatedness and embodiment, two key concepts in behavior-based systems (BBS), and then define BBS in detail and contrast it with alternatives, namely reactive, deliberative, and hybrid control. The paper then focuses on the role and power of behaviors as a representational substrate in learning policies and models, as well as learning from other agents (by demonstration and imitation). We overview a variety of methods we have demonstrated for learning in the multi-robot problem domain. © 2001 Elsevier Science B.V. All rights reserved.

*Keywords:* Robot learning; Behavior-based control; Social learning; Imitation learning

---

## 1. Introduction

Learning in physically embedded robots is known to be a difficult problem, due to sensory and effector uncertainty, partial observability of the robot's environment (which, in the multi-robot case, includes other robots), and non-stationarity of the world, especially when multiple learners are involved. Behavior-based systems (BBS) have served as an effective methodology for multiple robot control in a large number of multi-robot problem domains. As a

result, they have been used as the underlying control methodology for multi-robot learning.

Behavior-based systems grew out of the reactive approach to control, in order to compensate for its limitations (lack of state, inability to look into the past or the future) while conserving its strengths (real-time responsiveness, scalability, robustness). In the last decade, behavior-based systems have proven themselves as one of the two favored general methodologies (the other being hybrid systems) for autonomous system control, and as the most popular methodology for physical multi-robot system coordination.

In this paper we discuss how the use of behaviors

---

*E-mail address:* mataric@cs.usc.edu (M.J. Matarić).

as the underlying representation for control provides an effective substrate that facilitates learning, in particular in the multi-agent/robot context. The rest of the paper is organized as follows. Section 2 defines situatedness and embodiment, and discusses these notions relative to multi-agent and multi-robot systems. Section 3 defines and summarizes the key properties of behavior-based systems and compares those to the alternative approaches to control. Section 4 discusses what can be learned with behaviors as a representational substrate, and the remainder of the paper gives specific examples of BBS learning systems and the mechanisms involved. Section 5 discusses policy learning within BBS. The subsequent three sections address model learning in BBS. Section 6 describes how behaviors were originally abstracted to represent landmark information for learning spatial models. Sections 7 and 8 discuss methods for using behavior execution histories as models for agent/robot–environment interaction dynamics. Section 9 describes approaches to learning from other agents and humans. Section 10 concludes the paper.

## 2. Situatedness and embodiment

Behavior-based control arose from the need for intelligent, situated (also called embedded) behavior. *Situatedness* refers to having one's behavior strongly affected by the environment. Examples of situated robots include autonomous highway and city driving (Pomerleau, 1992), coordination of robot teams (Matarić, 1995), and robots in human environments, such as museums (Burgard et al., 2000). In contrast, robots, and agents in general, that exist in fixed, unchanging environments (such as assembly robots and maze-learning agents) are typically not considered situated. The predictability and stability of the environment have a direct impact on the complexity of the agent that must exist in it. Multi-robot systems are an excellent example of the impact of situatedness; individual robots in such systems are situated in a dynamic environment populated with other robots. If multiple robots are learning, i.e., changing their behavior and their environment over time, the complexity of the situatedness is increased.

*Embodiment* is a type of situatedness; it refers to

having a physical body and thus interacting with the environment through the constraints of that body. Physical robots are embodied, as are simulations whose behavior is constrained and affected by (models of) physical laws. Behavior-based control was originally developed for embodied, situated agents, namely robots, but has grown to apply to disembodied situated systems, such as information agents (Maes, 1994). In the case of multi-robot systems, embodiment has a direct reflection in interaction dynamics and thus performance: issues of physical interaction and critical mass play a large role in any problem domain involving multiple physical robots (Goldberg & Matarić, 1997). Embodiment thus plays a critical role in multi-robot learning; methods that do not explicitly take it into account suffer from interference effects. When addressed properly, embodiment can be used to facilitate learning, as we describe in Section 5.

## 3. Behavior-based control and multi-robot control

Behavior-based control is one of four basic classes of control. The others are reactive, deliberative, and hybrid control. For simplicity and clarity, the four can be briefly summarized with the following:

- Reactive control: don't think, react.
- Deliberative control: think hard, then act.
- Hybrid control: think and act independently, in parallel.
- Behavior-based control: think the way you act.

'Don't think, react!' *Reactive control* tightly couples sensory inputs and effector outputs, to allow the robot to quickly respond to changing and unstructured environments (Brooks, 1986). The biological inspiration and correlate to reactive control is 'stimulus-response'; this is a powerful control method: many animals are largely reactive. Thus, this is a particularly popular approach to situated robot control. Its limitations, however, include the robot's inability to have memory, internal representations of the world (Brooks, 1991), or the ability to learn over time. Reactive systems make the tradeoff in favor of fast reaction time and against complexity of reason-

ing. Formal analysis has shown that for environments and tasks that can be characterized a priori, reactive controllers can demonstrate highly effective, and if properly structured, even optimal performance in particular classes of problems (Schoppers, 1987; Agre & Chapman, 1990). In more complex types of environments and tasks, where internal models, memory, and learning are required, reactive control is not sufficient. Learning implies the use of memory; however, most robot learning has in fact been at the level of acquiring reactive controllers, or policies, which map specific sensory inputs to effector outputs. In Section 4, we discuss how the use of behaviors as the representational substrate can further facilitate learning in embodied systems.

‘Think hard, then act.’ In *deliberative control*, the robot uses all of the available sensory information, and all of the internally stored knowledge, to reason about what actions to take. Reasoning is typically in the form of planning, requiring a search of possible state–action sequences and their outcomes, a computationally complex problem if the state space is large or the state is partially observable, both of which are typical in physically situated, embodied systems. In multi-robot systems in particular, the state space rapidly becomes prohibitively large if inputs of other robots are included, or worse yet, if a global space is constructed. In addition to involving search, planning requires the existence of an internal representation of the world, which allows the robot to predict the outcomes of possible actions in various states. In multi-agent and multi-robot systems, planning involves the ability to predict the actions of others, which, in turn, requires models of others. When there is sufficient information for a world model (including adequate models of others), and sufficient time to generate a plan, deliberation is a highly effective method for generating strategic action. Multi-robot systems, however, are situated in noisy, uncertain, and changing environments, where model maintenance becomes extremely difficult and, for large systems, computationally prohibitive. As a result, situated single and multi-robot systems do not typically employ the purely deliberative approach to control.

‘Think and act independently, in parallel.’ *Hybrid control* adopts the best aspects of reactive and deliberative control: it attempts to combine the real-

time response of reactivity with the rationality and optimality of deliberation. As a result, the control system contains two different components, the reactive and the deliberative ones, which must interact in order to produce a coherent output. This is challenging, because the reactive component deals with the robot’s immediate needs, such as avoiding obstacles, and thus operates on a short time-scale and largely at the level of sensory signals. In contrast, the deliberative component uses highly abstracted, symbolic, internal representations of the world, and operates on them on a longer time-scale. If the outputs of the two components are not in conflict, the system requires no further coordination. However, the two parts of the system must interact if they are to benefit from each other. The reactive system must override the deliberative one when the world presents some unexpected and immediate challenge. Analogously, the deliberative component must inform the reactive one in order to guide the robot toward more efficient and optimal strategies. The interaction of the reactive and deliberative components require an intermediary, whose design is typically the greatest challenge of hybrid systems. As a result, these are called ‘three layer systems’, consisting of the reactive, intermediate, and deliberative layers. A great deal of research has been aimed at proper design of such hybrid systems (Giralt, Chatila, & Vaisset, 1983; Firby, 1987; Arkin, 1989; Malcolm & Smithers, 1990; Connell, 1991; Gat, 1998), and they are particularly popular for single robot control.

‘Think the way you act.’ *Behavior-based control* draws inspiration from biology for its design of situated, embodied systems. Behavior-based systems (BBS) get their name from their representational substrate, *behaviors*, which are observable patterns of activity emerging from interactions between the robot and its environment (which may contain other robots). Such systems are constructed in a bottom-up fashion, starting with a set of survival behaviors, such as obstacle-avoidance, which couple sensory inputs to robot actions. Next, behaviors are added that provide more complex capabilities, such as wall-following, target-chasing, exploration, homing, etc. Incrementally, behaviors are introduced to the system until their interaction results in the desired overall capabilities of the robot. Like hybrid systems, behavior-based systems may have different layers,

but the layers do not differ drastically in terms of time-scale and representation used. Importantly, behavior-based systems can store representations, but do so in a distributed fashion. Thus if a robot needs to plan, it does so in a network of communicating behaviors, rather than a centralized planner, and this representational difference carries with it significant computational and performance consequences. BBS, as employed in situated robotics, are not an instance of ‘behaviorism’; behaviorist models of animal cognition involved no internal representations, while behavior-based robot controllers can, thus enabling deliberation and learning.

The level of system situatedness, the nature of the task, and the capabilities of the agent determine which of the above methods is best suited for a given control and learning problem. Behavior-based systems and hybrid systems have the same expressive and computational capabilities: both can store representations and look ahead, but each does it in a very different way. As a result, the two have found rather different niches in mobile robotics. Hybrid systems dominate single robot control, except in time-critical domains that demand reactive systems. Behavior-based systems, on the other hand, dominate multi-robot control because collections of behaviors within the system scale well to collections of robots, resulting in robust, adaptive group behavior. BBS are in general best suited for systems situated in environments with significant dynamic changes, where fast response and adaptivity is crucial, but the ability to look ahead and avoid past mistakes is also required. Those capabilities are distributed over the system’s behaviors, and thus BBS ‘think the way they act’.

Behavior-based control has been applied to various single and multi-robot control problems, including robot soccer (Asada, Uchibe, Noda, Tawaratsumida, & Hosoda, 1994; Werger, 1999), coordinated movement (Matarić, 1995; Parker, 1998; Balch & Hybinette, 2000), cooperative box-pushing (Kube, 1992; Matarić & Gerkey, 2000), and even humanoid control (Brooks & Stein, 1994; Scassellati, 2000; Jenkins, Matarić, & Weber, 2000). In this paper we discuss in detail how using the behavior substrate can be conducive to learning, focusing on multi-robot learning of control policies, models, and form other agents.

#### **4. What can be learned with behaviors?**

The classical goal of machine learning systems is to optimize system performance over its lifetime. In the case of situated learning, in particular in the context of multi-robot systems that face uncertain and changing environments, instead of attaining asymptotic optimality, the aim is toward improved efficiency on a shorter time-scale. Models from biology are often considered, and reinforcement learning is particularly popular, as it focuses on learning directly from environmental feedback (Matarić, 1997b).

A key benefit of the behavior representation is in the way it encodes information used for control. Behaviors are a higher-level representation that elevates control away from low-level parameters, resulting in generality. At the same time, by encompassing and combining sensing and action, the behavior structure helps to reduce the state space of a problem while maintaining pertinent task-specific information. By utilizing the information encoded within behaviors, and the organization of behaviors within a network, various data structures and learning algorithms can be explored. Perhaps the most natural and popular use of behaviors in learning has been as abstractions of actions in the context of acquiring reactive policies that map world states to appropriate behaviors, forming a higher-level representation of standard state–action pairings. Section 5 briefly overviews some work in this area, and gives an example of its application to the multi-robot domain.

Another natural means of using behaviors is for encoding information either about the world or the system itself, for the construction of models. Section 6 describes the first example of using behaviors as a representational substrate, applied to learning spatial models of the environment. Section 7 describes a tree structure representation of histories of behavior activation, used to model the robot’s interaction with its environment and other robots. Section 8 describes an adaptation of semi-Markov chains into so-called augmented Markov models, to statistically represent past behavior activation patterns. This enables a BBS to model its own dynamics at run-time; the resulting models are used to adapt the underlying controller

either by changing the behavior selection strategy or tuning internal behavior parameters, resulting in improvement of performance in individual and multi-robot tasks.

Besides learning behavior selection and models, the BBS framework lends itself to agents learning from each other. Section 9 focuses on a methodology for representing behaviors in a way that allows abstraction, and thus enables learning by observation and imitation, and automated controller construction and exchange between robots in a multi-robot system.

## 5. Learning behavior policies

Effective behavior selection is the key challenge in behavior-based control, as it determines which behavior, or subset of behaviors, controls the agent/robot at a given time. This problem is easily formulated in the reinforcement learning framework as seeking the policy that maps states to behaviors so as to maximize received reward over the lifetime of the agent.

The earliest examples of reinforcement learning (RL) in the context of BBS demonstrated hexapod walking (Maes & Brooks, 1990) and box-pushing (Mahadevan & Connell, 1991). Both decomposed the control system into a small set of behaviors, and used generalized input states, thus effectively reducing the state space. The latter also used modularization to partition the monolithic global policy being learned into three mutually-exclusive policies: one for getting out when stuck, another for finding the box when lost and not stuck, and the third for pushing the box when in contact with one and not stuck.

Our own work explored scaling up reinforcement learning to multi-robot behavior-based systems, where the environment presents further challenges of non-stationarity and credit assignment, due to the presence of other concurrent learners. We studied the problem in the context of a foraging task with four robots, each initially equipped with a small set of basis behaviors (searching, homing, picking up, dropping, following, avoiding) and learning indi-

vidual behavior selection policies, i.e., which behavior to execute under which conditions. Due to interference among concurrent learners, this problem could not be solved directly by standard RL. We introduced *shaping*, a concept popular in psychology (Gleitman, 1981) and subsequently adopted in robot RL (Dorigo & Colombetti, 1997). Shaping pushes the reward closer to the subgoals of the behavior, and thus encourages the learner to incrementally improve its behaviors by searching the behavior space more effectively.

Since behaviors are time-extended and event-driven, receiving reinforcement upon their completion results in a credit assignment problem. We introduced *progress estimators*, measures of progress toward the goal of a given behavior during its execution. This is a form of reward shaping, and it addresses two issues associated with delayed reward: behavior termination and fortuitous reward. Behavior termination in BBS is event-driven; the duration of any given behavior is determined by the interaction dynamics with the environment, and can vary greatly. Progress estimators provide a principled means for deciding when a behavior may be terminated even if its goal is not reached and externally-generated event has not occurred.

Fortuitous reward refers to reward ascribed to a particular situation–behavior (or state–action) pair which is actually a result of previous behaviors/actions. It manifests as follows: previous behaviors lead the system near the goal, but some event induced a behavior switch, and subsequent achievement of the goal is ascribed most strongly to the final behavior, rather than the previous ones. Shaped reward in the form of progress estimators effectively eliminates this effect: because it provides feedback during behavior execution, it rewards the previous, beneficial behaviors more strongly than the final one, thus more realistically dividing up the credit.

We found that in the absence of shaped reward, the four-robot learning system could not converge to a correct policy, because interference from other learners was too frequent and disruptive during time-extended behaviors. The introduction of a progress estimator for only one behavior (homing, where progress toward a goal location was directly measurable) was sufficient to enable efficient learning of

collective foraging. The details of the approach and the results are given in Matarić (1994, 1997b).

In subsequent extended, scaled-up experiments, we found that while reward shaping enabled learning, increasing numbers of concurrent learners decreased the overall speed of learning, due to interference. To reverse this effect, i.e., to enable multi-robot learning to accelerate as a result of multiple learners, we explored another addition to reinforcement learning, namely *spatio-temporally local reward sharing* between agents. Again in the framework of behavior policy learning, we induced robots to share the received credit (reward and punishment) with those local to them in space and time, with the assumption that they were sharing a social context. This simple measure effectively diminished greedy reward maximization in favor of acquiring social behaviors, such as yielding and information-sharing.

The problem scenario we used involved four robots learning two social rules: yielding in congested areas (such as doorways) and sharing information (such as when finding a source of objects). Without shared reward, individual greedy learning could not result in a social policy, because the immediate result of being social, such as yielding to another agent, results in loss of reward. The only means by which a distributed group of agents can learn a social policy is through some sense of global payoff, but that information is not typically available to individuals in many distributed multi-robot problem domains. Thus, by sharing reward locally in space and time, we effectively decreased the locality of the system, without having to introduce global reward. In effect, if a robot yields to another, the reward of the second getting through the door is shared by both and thus reinforces social behavior. This bias effectively guides the learning systems toward the social policy. The details of this work are given in Matarić (1997a,c).

In another example of individual policy learning in a multi-robot scenario, we addressed the problem of tightly-coupled coordination, in the context of cooperative box-pushing by two communicating robots. Each robot was equipped with local contact and light sensors, giving it information about whether it was touching the box and approximately where the goal, marked with a bright light, was located. The information about the goal was limited; depending on

the robot's orientation, it may not have seen the goal at all, or not well. This partial observability of the goal state made learning difficult for each robot individually, and the concurrent learning was made worse as a result. We used communication to ameliorate partial observability: each robot, when in contact with the box, communicated its limited view of the world to the other robot, thus enlarging each other's perspective. Although the robots shared their perceptual state, they kept individual action spaces, and learned individual pushing policies, which corresponded to the side of the box they were on. In a sense, the two robots formed a meta-agent with a shared perceptual mechanism (though communication) and distributed effectors. Interestingly, since stopping and waiting was one of the behaviors in the repertoire, the system repeatedly converged on a turn-taking pushing strategy. This solution minimized the credit assignment problem between the two pushers because the mapping between the actions of each robot and the subsequent reward was made unambiguous (Simsarian & Matarić, 1995).

To summarize, the use of behaviors has an important effect on facilitating learning: it elevates the action representation of the system, thereby reducing the state space, and can be used to shape reward. We have demonstrated the use of reward shaping, reward sharing, and perception sharing, all effective means of addressing challenges in multi-robot learning. Reward shaping manages interference among concurrent learners, reward sharing minimizes greedy 'antisocial' behavior, and perception sharing ameliorates partial observability. The methods are general, but are facilitated by the BBS structure, enabling learning in the challenging multi-robot domain.

Our discussion so far has been confined to policy learning, which is currently the most common learning approach in single- and multi-robot systems. Model learning, however, can exploit the BBS structure to an even greater extent, as presented next.

## 6. Learning models of the environment

Models are more general than policies, since they are not goal/task-specific, and thus can be applied to adapt various controllers. Model learning in behavior

space can take a variety of forms. The first approach we discuss involves using behaviors to represent spatial information. Learning maps of the environment is one of the most basic and popular problems in mobile robotics. Our early work introduced a means of using behaviors as a representational substrate for map learning and path planning, capabilities previously considered outside of the realm of BBS.

The behavior-based system we used, embodied on a robot named Toto, consisted of a navigation layer, a landmark detection layer, and the map and path finding layer. To represent the structure of the environment within a behavior-based system, we used a network of behaviors, assigning an ‘empty’ behavior shell to each newly discovered landmark, and parameterized it with its associated attributes: landmark type (e.g., left-wall, right-wall, corridor, etc.), direction (compass reading) and length. Each new behavior was added to the network (map) by linking it to its topological neighbors with communication links. Each such map behavior was activated whenever its attributes matched the outputs of the landmark detector. As Toto moved about its environment, a topological map of the detected landmarks was constructed, maintained, and updated.

Localization in the network was performed through the combination of three processes, all of which help address partial observability of location information. The first matched all map behaviors to the currently detected landmark. The second used ‘expectation’, message passing between nearest neighbors to indicate which landmark is most likely to become active next. The third, only needed in ambiguous, maze-like environments, used an approximate odometric threshold to eliminate any unlikely matches. The approximate odometry information was needed for detecting cycles in the network, and thus distinguishing new landmarks from those encountered previously.

Path finding was performed in much the same way as the rest of the network processes: by local message passing (or activation spreading) from the goal landmark in the map throughout the rest of the network. The messages contained accumulated path length, so that the shortest path could be chosen at each decision point. Activation was spread continuously, so the robot could move during path planning

and adapt to the new location, as well as account for blocked paths; in those cases the blocked topological link was considered inactive and the continuous path planning found an alternate route, if one existed. This approach was introduced in Matarić (1990a,b), and described in detail in Matarić (1992). Subsequent work explored scaling such distributed map learning to a group of robots, and used graph matching to correlate partial maps across multiple learners (Dedeoglu, Sukhatme, & Matarić, 1999; Dedeoglu & Sukhatme, 2000).

Utilizing the isomorphism between the physical and the behavior network topology is an effective means of embedding a spatial representation into a behavioral one. In the next few sections, we will describe how this process can be made general and applied to non-spatial model learning as well.

## 7. Learning models from behavior history

Behaviors are activated and terminated by events in the environment, and their resulting sequences and combinations encode the dynamics of the robot’s interaction with its world. Behaviors do not explicitly encode state, but include it implicitly in their execution conditions. Furthermore, since behaviors are time-extended, world state changes during their execution. Combined, these properties provide an interesting substrate for model development.

In our first approach to exploiting behavior execution dynamics, we used a tree representation to encode past behavior use, thus capturing frequent paths the robot took in behavior space. The nodes in the tree were executed behaviors, the tree topology represented the paths the robot took in behavior space, and branches were augmented with statistics about path frequency. A robot would construct a tree incrementally, as it repeated its task over several trials. The resulting tree represented a model of behavior execution dynamics which was used to adapt policies at run time, by adapting the behavior selection/arbitration mechanism. We applied this approach to encoding the histories of behavior use in mobile robots learning to find and retrieve a brightly-colored object in a dynamic environment containing large amounts of interference, including other learn-

ing robots as well as other moving robots with stationary policies, i.e., non-learners.

We demonstrated that the use of behavior execution history was effective in recognizing common patterns of interference even without explicit representation of world state (since no such state was represented in the behavior trees). Specifically, although the robot was not correlating specific world locations (such as  $x$ ,  $y$  positions) with interference, it was able to recognize sequences of inefficient behaviors within the tree, and selectively avoid them by making different behavior choices, i.e., altering its behavior selection/arbitration mechanism. In practice this resulted in the robot selectively eliminating certain behaviors from its repertoire (such as wall-following), or selectively favoring them. As a result, different robots learned different, specialized policies, so as to maximize reward and be mutually compatible. One robot used a direct path to the object and back (dubbed ‘aggressive’), while the other circumnavigated the room by following the walls (dubbed ‘passive’). Together, the two robots minimized interference with each other and the non-learners, and maximized individual reward (based on the number of found and delivered objects). Individual robot factors did not play a role in specialization; different robots adapted their controllers to specialized policies in different trials, but specialization to those two particular policies was repeatable. We knew in advance that one robot would be ‘aggressive’ and the other ‘passive’, but could not predict which would be which.

In summary, this approach to capturing behavior execution history was proven effective for constructing models of behavior dynamics on individual robots in a multi-robot domain. The robots were able to adapt their policies (controllers) so as to collectively achieve the task (collecting objects) more efficiently. Details of this work are presented and discussed in Michaud & Matarić (1998a,b).

## 8. Learning models of interaction

More recently, we addressed the problem of modeling the interaction dynamics of BBS in a more general and principled fashion. This is of particular relevance to multi-agent and multi-robot domains, where modeling interaction dynamics can allow

individual robots to adapt to local experience so as to improve the performance of the system as a whole.

We developed *Augmented Markov Models (AMMs)*, a representation based on semi-Markov decision process, but with additional statistics associated with links and nodes (Fig. 1). We also developed an AMM construction algorithm that has minimal computational overhead, making it feasible for on-line real-time applications. One of the main benefits of AMMs is their direct mapping to behaviors within BBS; an AMM of a behaving system is constructed incrementally, with the states of the AMM representing behavior execution, and with state-splitting used to capture the order of the system (see Figs. 2 and 3).

We have demonstrated AMMs as a modeling tool in a number of applications, most featuring multiple robots: fault detection, affiliation determination, hierarchy restructuring, regime detection, and reward maximization. The AMM-based evaluations used in these applications include statistical hypothesis tests and expectation calculations from Markov chain theory. Each of the applications is experimentally verified using up to four physical mobile robots performing elements of a foraging (collection) task. Each robot maintains one or more concurrent AMMs, which it uses to adapt its controller policy. In some multi-robot applications (such as hierarchy restructuring), the robots compare their respective AMMs.

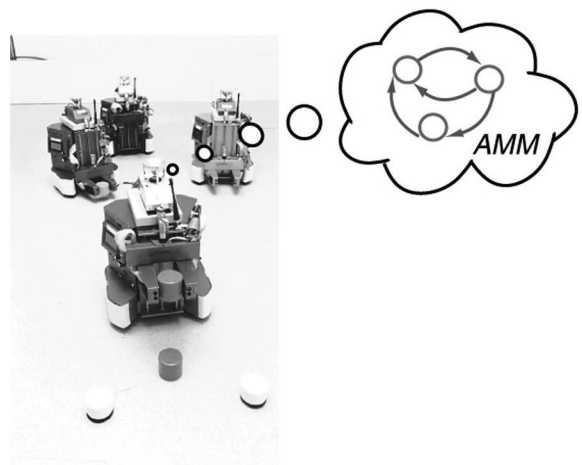


Fig. 1. Each robot maintains one or more AMMs.

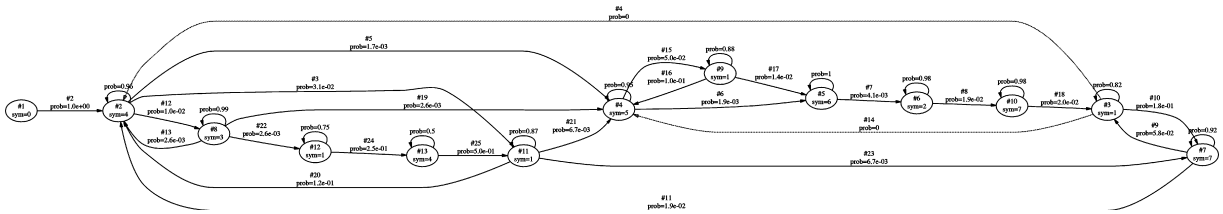


Fig. 2. A 2nd order AMM generated by a foraging robot.

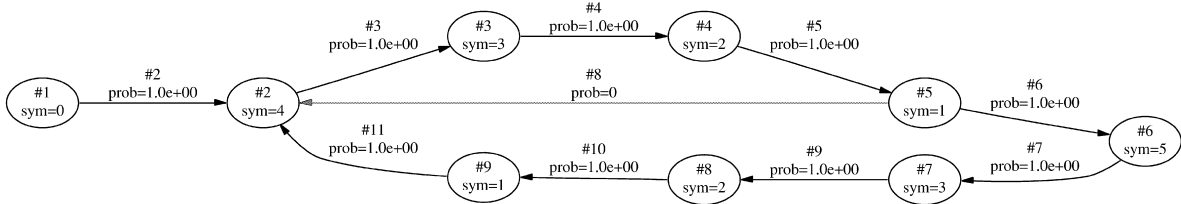


Fig. 3. A 4th order AMM recognized by a robot and translated into 1st order.

In the context of reward maximization, we developed an algorithm that provides a moving average estimate of the state of a non-stationary system, and have applied it to the problem of reward maximization in a non-stationary environment (Goldberg & Matarić, 2000a). The algorithm dynamically adjusts the window size used in the moving average to accommodate the variances and type of non-stationarity exhibited by the system, while discarding outdated and redundant data. Multiple AMMs are learned at different time scales, and statistics about the environment at each time scale are derived from those. The state of the environment is thus estimated indirectly through the robot’s interaction with it. As task execution continues, AMMs are dynamically generated to accommodate the increasing time intervals. Sets of statistics from the models are used to determine whether old data and AMMs are redundant/outdated and can be discarded. In addition, the approach is able to compensate for both abrupt and gradual non-stationarity manifesting at different time scales. Furthermore, it requires no a priori knowledge, uses only local sensing, and captures the notion of time scale. Finally, it works naturally with stochastic task domains where variations between trials may change the most appropriate amount of data for state estimation.

We used a similar approach to address the problem of capturing changes in the environmental dynamics (resulting, at least in part, from the behavior of other agents/learners) based on a robot’s local, individual view of the world. Detecting these shifts allows the robot to appropriately adapt to the different regimes within a task. As above, each robot maintained multiple AMMs at different time scales, so as to capture both abrupt and gradual shifts in the dynamics.

The goal of developing AMMs was to provide a pragmatic, theoretically sound, and general-purpose tool for on-line modeling in complex, noisy, non-stationary systems. As such, AMMs lend themselves in particular to multi-agent and multi-robot learning problems. The structure of AMMs was designed to fit BBS, but is also generally applicable. This work is described in detail in Goldberg & Matarić (1999, 2000a,b).

### 9. Learning from humans and other agents/robots

One of the great benefits but also open challenges of multi-agent learning is the agents’ ability to learn not only from the environment, but from each other

as well as from people. This ability can be as simple as a passive observation of the effects of the actions of others in the environment, or as complex as teacher–student imitation learning.

To exploit the potential of learning from other agents, we have been exploring ways in which the use of behaviors as a common representation substrate can facilitate such learning. We are situating this work in different problem domains (human–robot and robot–robot interaction), in order to test the generality of our methodology, which involves the use of more powerful behavior representations than those discussed so far. We developed the notion of *abstract behaviors*, which separate the activation conditions of a behavior from its output actions (so-called *primitive behaviors*); this allows for a more general set of activation conditions to be associated with the primitive behaviors. While this is not necessary for any single task, and thus not typically employed in BBS, it is what provides generality to the representation. An abstract behavior is a pairing of a given behavior’s activation conditions (i.e., preconditions), and its effects (i.e., postconditions); the result is an abstract and general operator much like those used in classical deliberative systems (see Fig. 4). Primitive behaviors, which typically consist of a small basis set, as is common on well-designed BBS, may involve one or an entire

collection of sequential or concurrently executing behaviors.

Networks of such behaviors are then used to specify strategies or general ‘plans’ in a way that merges the advantages of both abstract representations and BBS. The nodes in the networks are abstract behaviors, and the links between them represent precondition and postcondition dependencies. The task plan or strategy is represented as a network of such behaviors. As in any BBS, when the conditions of a behavior are met, the behavior is activated. Similarly here, when the conditions of an abstract behavior are met, the behavior activates one or more primitive behaviors which achieve the effects specified in its postconditions. The network topology at the abstract behavior level encodes any task-specific behavior sequences, freeing up the primitive behaviors to be reused for a variety of tasks. Thus, since abstract behavior networks are computationally light-weight, solutions for multiple tasks can be encoded within a single system, and dynamically switched, as we have demonstrated in our implementations.

We have developed the methodology for semi-automatically generating such networks off-line as well as at run-time. The latter enables a learning robot to acquire task descriptions dynamically, while observing its environment, and, more importantly,

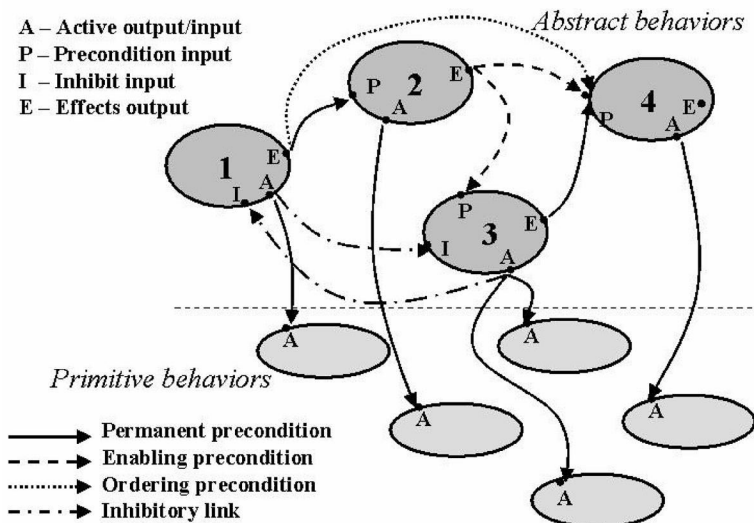


Fig. 4. The structure of abstract behaviors and networks thereof.

while observing other robots and/or a teacher. We have validated this methodology in several tasks involving a mobile robot following a human and acquiring a representation of the human-demonstrated task by observing the activation of its own abstract behavior pre- and post-conditions, thus resulting in a new abstract behavior network representing the demonstrated task. The robot was able to acquire novel behavior sequences and combinations (i.e., concurrently executing behaviors), resulting in successful learning of tasks involving visiting various targets in particular order, picking up, transporting, and delivering objects, dealing with barriers, and maneuvering obstacle courses in specific ways. This work is described in detail in Nicolescu & Matarić (2000a,b; 2001).

While the above-described effort builds on and generalizes earlier work on using behaviors as more abstract representations (described in Section 6), we have also applied the idea to *imitation learning* from a human demonstrator. As above, an agent (in our case a complex humanoid simulation with dynamics) observes a human, through the use of vision sensors or other motion-capture equipment, and maps the observed behavior onto its own known behavior repertoire. While in the above approach what is observed is mapped onto the space of abstract behaviors, resulting in a more abstract ‘model’ of a task, in the case of a humanoid agent, the mapping is done directly onto executable perceptual-motor behaviors. This approach is based on neuroscience evidence (Giszter, Mussa-Ivaldi, & Bizzi, 1993; Rizzolatti et al., 1996; Iaconi et al., 1999) which directly links visually perceived movements with motor cortex activity. When translated into BBS, this results in a finite set of *basis behaviors*, or *perceptual-motor primitives*, being used as a vocabulary for classifying observed movement. The primitives are manipulated through combination operators and can thus be sequenced and superimposed to generate a large movement repertoire. Thus, any observed movement is readily imitated with the best known approximation within the existing primitive-based vocabulary. The error in the resulting imitation, then, is used to enlarge and refine the motor repertoire and facilitate more accurate imitation in the future. This biologically-inspired structuring of the motor and imitation systems allows us to reduce the dimen-

sionality of the otherwise very difficult movement observation, interpretation, and reconstruction problem.

We have demonstrated this form of learning in a humanoid agent that is endowed with a small set of such behaviors, and, as a result, is capable of imitating novel movements including dance and sports. Details about this approach are found in Matarić (2001); an implemented validation of the model is found in Jenkins et al. (2000), and Fod, Matarić & Jenkins (2000) describe a methodology for automatically deriving the primitive vocabulary.

## 10. Summary and conclusions

The aim of this paper has been to discuss how the use of behaviors as the underlying control representation provides a useful encoding that both lends robustness to control and allows abstraction for handling scaling in learning, of key importance to multi-agent/robot systems. We briefly surveyed a variety of methods for learning we have demonstrated within behavior-based systems, in particular focusing on the multi-robot problem domain.

Knowledge representation is typically studied as an independent branch of AI, separate from fields such as robotics and learning. This is likely partly why the notion of behaviors as a representation has been difficult to properly situate within robotics, where methodologies are largely algorithmic in nature. The same holds even more so for machine learning, where representation is typically considered no more than data structure manipulation in service of algorithms. However, empirical results from the last two decades of using BBS have demonstrated that behaviors are an effective representational substrate for both control and learning in robotics, and may have some fundamental features that combine rather than separate issues of representation and computation. This interaction is of particular importance in difficult problems such as multi-robot learning, where challenges of sensor noise, partial observability, delayed reward, and non-stationarity conspire to defy traditional machine learning methods.

Because behaviors are a high-level but non-symbolic representation, and because they are not con-

strained to a tightly defined specification, they provide a rich framework for exploring representational and algorithmic means of addressing difficult problems of situated and embodied control and learning. We have surveyed a collection of BBS-based approaches to single and especially multi-robot learning that have tackled real-world challenges by taking advantage of the behavioral substrate both at the representational and algorithmic levels.

Much work remains to be done both in the theoretical analysis and empirical use of behaviors and BBS. We hope that the examples and discussion provided in this paper encourage such work by pointing to the breadth of utility of the approach.

## Acknowledgements

The author's work on learning in behavior-based systems has been supported by the Office of Naval Research (Grants N00014-95-1-0759 and N0014-99-1-0162), the National Science Foundation (Career Grant IRI-9624237 and Infrastructure Grant CDA-9512448), and by DARPA (Grant DABT63-99-1-0015 and contract DAAE07-98-C-L028). Many thanks to Dani Goldberg for valuable comments and to Monica Nicolescu for much-needed corrections. The author thanks Dani Goldberg, Francois Michaud, Monica Nicolescu, and Kristian Simsarian for numerous valuable insights gained through collaboration.

## References

- Agre, P. E., & Chapman, D. (1990). What are plans for? In: Maes, P. (Ed.), *Designing autonomous agents*, MIT Press, pp. 17–34.
- Arkin, R. C. (1989). Towards the unification of navigational planning and reactive control. In: *AAAI spring symposium on robot navigation*, pp. 1–5.
- Asada, M., Uchibe, E., Noda, S., Tawaratsumida, S., & Hosoda, K. (1994). Coordination of multiple behaviors acquired by a vision-based reinforcement learning. In: *Proceedings, IEEE/RSJ/GI international conference on intelligent robots and systems*, Munich, Germany.
- Balch, T., & Hybinette, M. (2000). Behavior-based coordination of large scale formations. In: *International conference on multi-agent systems (ICMAS-2001)*, Boston, MA.
- Brooks, R. A. (1986). A robust layered control system for a mobile robot. *Journal of Robotics and Automation RA-2*, 14–23.
- Brooks, R. A. (1991). Intelligence without reason. In: *Proceedings, IJCAI-91*, Morgan Kaufman, Sydney, Australia, pp. 569–595.
- Brooks, R. A., & Stein, L. A. (1994). Building brains for bodies. *Autonomous Robots 1*, 7–25.
- Burgard, W., Cremers, A., Fox, D., Hohnel, D., Lakemeyer, G., Schulz, D., Steiner, W., & Thrun, S. (2000). Experiences with an interactive museum tour-guide robot. *Artificial Intelligence 110*(1–2).
- Connell, J. H. (1991). SSS: a hybrid architecture applied to robot navigation. In: *IEEE international conference on robotics and automation*, Nice, France, pp. 2719–2724.
- Dedeoglu, G., & Sukhatme, G. (2000). Landmark-based matching algorithm for cooperative mapping by autonomous robots. In: *Proceedings 5th international symposium on distributed autonomous robot systems*, Knoxville, TN, pp. 251–260.
- Dedeoglu, G., Sukhatme, G., & Matarić, M. (1999). Incremental on-line topological map building for a mobile robot. In: *Proceedings, mobile robots XIV-SPIE*, Boston, MA, pp. 129–139.
- Dorigo, M., & Colombetti, M. (1997). *Robot shaping: an experiment in behavior engineering*, MIT Press, Cambridge, MA.
- Firby, R. J. (1987). An investigation into reactive planning in complex domains. In: *Proceedings, sixth national conference on artificial intelligence*, Seattle, pp. 202–206.
- Fod, A., Matarić, M. J., & Jenkins, O. C. (2000). Automated derivation of primitives for movement classification. In: *Proceedings, first IEEE-RAS international conference on humanoid robotics*, Cambridge, MA, MIT.
- Gat, E. (1998). On three-layer architectures. In: Kortenkamp, D., Bonnasso, R., & Murphy, P. (Eds.), *Artificial intelligence and mobile robotics*, AAAI Press.
- Giralt, G., Chatila, R., & Vaisset, M. (1983). An integrated navigation and motion control system for autonomous multisensory mobile robots. In: Brady, M., & Paul, R. (Eds.), *First international symposium in robotics research*, MIT Press, Cambridge, MA, pp. 191–214.
- Giszter, S. F., Mussa-Ivaldi, F. A., & Bizzi, E. (1993). Convergent force fields organized in the frog's spinal cord. *Journal of Neuroscience 13*(2), 467–491.
- Gleitman, H. (1981). *Psychology*, W.W. Norton, New York.
- Goldberg, D., & Matarić, M. J. (1997). Interference as a tool for designing and evaluating multi-robot controllers. In: *Proceedings, AAAI-97*, AAAI Press, Providence, RI, pp. 637–642.
- Goldberg, D., & Matarić, M. J. (1999). Coordinating mobile robot group behavior using a model of interaction dynamics. In: Etzioni, O., Muller, J. P., & Bradshaw, J. M. (Eds.), *Proceedings, the third international conference on autonomous agents (agents '99)*, ACM Press, Seattle, WA, pp. 100–107.
- Goldberg, D., & Matarić, M. J. (2000a). Learning models for reward maximization. In: *Proceedings, the seventeenth international conference on machine learning (ICML-2000)*, Stanford University, pp. 319–326.
- Goldberg, D., & Matarić, M. J. (2000b). Reward maximization in a non-stationary mobile robot environment. In: *Proceedings, the*

- fourth international conference on autonomous agents (agents 2000), Barcelona, Spain, pp. 92–99.
- Iacononi, M., Woods, R. P., Brass, M., Bekkering, H., Mazziotta, J. C., & Rizzolatti, G. (1999). Cortical mechanisms of human imitation. *Science* 286, 2526–2528.
- Jenkins, O. C., Matarić, M. J., & Weber, S. (2000). Primitive-based movement classification for humanoid imitation. In: Proceedings, first IEEE-RAS international conference on humanoid robotics, Cambridge, MA, MIT.
- Kube, C. R. (1992). *Collective robotic intelligence: a control theory for robot populations*, University of Alberta, Master's thesis.
- Maes, P. (1994). Modeling adaptive autonomous agents. *Artificial Life* 1(2), 135–162.
- Maes, P., & Brooks, R. A. (1990). Learning to coordinate behaviors. In: Proceedings, AAAI-90, Boston, MA, pp. 796–802.
- Mahadevan, S., & Connell, J. (1991). Scaling reinforcement learning to robotics by exploiting the subsumption architecture. In: Eighth international workshop on machine learning, Morgan Kaufmann, pp. 328–337.
- Malcolm, C., & Smithers, T. (1990). Symbol grounding via a hybrid architecture in an autonomous assembly system. In: Maes, P. (Ed.), *Designing autonomous agents*, MIT Press, pp. 145–168.
- Matarić, M. J. (1990a). Environment learning using a distributed representation. In: IEEE international conference on robotics and automation, Cincinnati, pp. 402–406.
- Matarić, M. J. (1990b). Navigating with a rat brain: a neurobiologically-inspired model for robot spatial representation. In: Meyer, J. -A., & Wilson, S. (Eds.), *From animals to animats: international conference on simulation of adaptive behavior*, MIT Press, pp. 169–175.
- Matarić, M. J. (1992). Integration of representation into goal-driven behavior-based robots. *IEEE Transactions on Robotics and Automation* 8(3), 304–312.
- Matarić, M. J. (1994). Reward functions for accelerated learning. In: Cohen, W. W., & Hirsh, H. (Eds.), *Proceedings of the eleventh international conference on machine learning (ML-94)*, Morgan Kaufman, New Brunswick, NJ, pp. 181–189.
- Matarić, M. J. (1995). Designing and understanding adaptive group behavior. *Adaptive Behavior* 4(1), 50–81.
- Matarić, M. J. (1997a). Learning social behavior. *Robotics and Autonomous Systems* 20, 191–204.
- Matarić, M. J. (1997b). Reinforcement learning in the multi-robot domain. *Autonomous Robots* 4(1), 73–83.
- Matarić, M. J. (1997c). Using communication to reduce locality in distributed multi-agent learning. In: Proceedings, AAAI-97, AAAI Press, Providence, RI, pp. 643–648.
- Matarić, M. J. (2001). Sensory-motor primitives as a basis for imitation: linking perception to action and biology to robotics. In: Nehaniv, C., & Dautenhahn, K. (Eds.), *Imitation in animals and artifacts*, MIT Press.
- Matarić, M. J., & Gerkey, B. P. (2000). Principled communication for dynamic multi-robot task allocation. In: Proceedings, international symposium on experimental robotics, Waikiki, Hawaii, pp. 341–352.
- Michaud, F., & Matarić, M. J. (1998). Learning from history for behavior-based mobile robots in non-stationary conditions. *Autonomous Robots*, 5: 335–354; also *Machine Learning*, 31: 141–167.
- Michaud, F., & Matarić, M. J. (1998b). Representation of behavioral history for learning in nonstationary conditions. *Robotics and Autonomous Systems* 29, 187–200.
- Nicolescu, M., & Matarić, M. J. (2000a). Extending behavior-based systems capability using an abstract behavior representation. In: Proceedings, AAAI spring symposium on parallel cognition, North Falmouth, MA, Also USC institute for robotics and intelligent systems technical report IRIS-00-389.
- Nicolescu, M., & Matarić, M. J. (2000b). Learning cooperation from human–robot interaction. In: Proceedings, 5th international symposium on distributed autonomous robotics systems (DARS), Knoxville, TN, pp. 477–478.
- Nicolescu, M., & Matarić, M. J. (2001). Learning and interacting in human–robot domains. In: Dautenhahn, K. (Ed.), *IEEE Transactions on Systems, Man, Cybernetics*, special issue on Socially Intelligent Agents – The Human in The Loop.
- Parker, L. E. (1998). ALLIANCE: an architecture for fault-tolerant multi-robot cooperation. *IEEE Transactions on Robotics and Automation* 14.
- Pomerleau, D. A. (1992). *Neural network perception for mobile robotic guidance*, Carnegie Mellon University, School of Computer Science, Ph.D. thesis.
- Rizzolatti, G., Fadiga, L., Matelli, M., Bettinardi, V., Perani, D., & Fazio, F. (1996). Localization of grasp representation in humans by positron emission tomography: 1. Observation versus execution. *Experimental Brain Research* 111, 246–252.
- Scassellati, B. (2000). Investigating models of social development using a humanoid robot. In: Webb, B., & Consi, T. (Eds.), *Biorobotics*, MIT Press.
- Schoppers, M. J. (1987). Universal plans for reactive robots in unpredictable domains. In: IJCAI-87, Menlo Park, pp. 1039–1046.
- Simsarian, K. T., & Matarić, M. J. (1995). Learning to cooperate using two six-legged mobile robots. In: Proceedings, third European workshop of learning robots, Heraklion, Crete, Greece.
- Werger, B. B. (1999). Cooperation without deliberation: a minimal behavior-based approach to multi-robot teams. *Artificial Intelligence* 110, 293–320.