In *Special Issue of IEEE Transactions on Systems, Man, and Cybernetics, Part A: Systems and Humans*
Vol. 31, No. 5, Sept. 2001, Pages 419-430, C. C. White and K. Dautenhahn (eds.)

Learning and Interacting in Human-Robot Domains

Monica N. Nicolescu and Maja J Matarić

Abstract—
Human-agent interaction is a growing area of research; there are many approaches that address significantly different aspects of agent social intelligence. In this paper, we focus on a robotic domain in which a human acts both as a teacher and a collaborator to a mobile robot. First, we present an approach that allows a robot to learn task representations from its own experiences of interacting with a human. While most approaches to learning from demonstration have focused on acquiring policies (i.e., collections of reactive rules), we demonstrate a mechanism that constructs high-level task representations based on the robot's underlying capabilities. Second, we describe a generalization of the framework to allow a robot to interact with humans in order to handle unexpected situations that can occur in its task execution. Without using explicit communication, the robot is able to engage a human to aid it during certain parts of task execution. We demonstrate our concepts with a mobile robot learning various tasks from a human, and, when needed, interacting with a human to get help performing them.

Keywords— **Robotics, Learning and Human-Robot Interaction**

I. Introduction

Human-agent interaction is a growing area of research, spawning a remarkable number of directions for designing agents that exhibit social behavior and interact with people. These directions address many different aspects of the problem and require different approaches to human-agent interaction based on whether they are software agents or embedded (robotic) systems.

The different human-agent interaction approaches have two major challenges in common. The first is to build agents that have the ability to to learn through social interaction with humans or with other agents in the environment. Previous approaches have demonstrated social agents that could learn and recognize models of other agents [1], imitate demonstrated tasks (maze learning of [2]) or use natural cues (such as models of joint attention [3]) as means for social interaction.

The second challenge is to design agents that exhibit social behavior, which allows them to engage in various types of interactions. This is a very large domain, with examples including assistants (helpers) [4], competitor agents [5], teachers [6], [7], [8], entertainers [9] and toys [10].

In this paper we focus on the physically embedded robotic domain and present an approach that unifies the two challenges, where a human acts both as a teacher and a collaborator for a mobile robot. The different aspects of this interaction help demonstrate the robot's learning and social abilities.

Teaching robots to perform various tasks by presenting demonstrations is being investigated by many researchers. However, the majority of the approaches to this problem to date has been limited to learning policies, collections of reactive rules that map environmental states with actions. In contrast, we are interested in developing a mechanism that would allow a robot to learn representations of high level tasks, based on the underlying capabilities already available to the robot. Our goal is to enable a robot to automatically build a controller that achieves a particular task from the experience it had while interacting with a human. We present the behavior representation that enables these capabilities, and describe the process of learning task representations from experienced interactions with humans.

In our system, during the demonstration process, the human-robot interaction is limited to the robot following the human and relating the observations of the environment to its internal behaviors. We extend this type of interaction to a general framework that allows a robot to convey its intentions by suggesting them through actions, rather than communicating them through conventional signs, sounds, gestures, or marks with previously agreed-upon meanings. Our goal is to employ these actions as a vocabulary that a mobile robot could use to induce a human to assist it for parts of tasks that it is not able to perform on its own.

This paper is organized as follows. Section II presents the behavior representation that we are using and Section III describes learning task representations from experienced interactions with humans. In Section IV, we present the interaction model and the general strategy for communicating intentions. In Section V we present experimental demonstrations and validation of learning task representations from demonstration, including experiments where the robot engaged a human in interaction through actions indicative of its intentions. Sections VI and VII discuss different related approaches and present the conclusions on the described work.

II. Behavior representation

We are using a behavior-based architecture [11], [12] that allows the construction of a given robot task in the form of behavior networks [13]. This architecture provides a simple and natural way of representing complex sequences of behaviors and the flexibility required to learn high-level task representations.

In our behavior network, the links between nodes/behaviors represent precondition-postcondition dependencies; thus the activation of a behavior is dependent not only on its

The authors are with the Robotics Research Laboratory, University of Southern California, Los Angeles, CA 90089-0781. E-mail: monica|mataric@cs.usc.edu. This work is supported by DARPA Grant DABT63-99-1-0015 under the Mobile Autonomous Robot Software (MARS) program and by the ONR Defense University Research Instrumentation Program Grant.

own preconditions (particular environmental states) but also on the postconditions of its relevant predecessors (*sequential preconditions*).

We introduce a representation of goals into each behavior, in the form of abstracted environmental states. The *met/not met* status of those goals is continuously updated, and communicated to successor behaviors through the network connections, in a general process of activation spreading which allows for arbitrary complex tasks to be encoded. Embedding goal representations in the behavior architecture is a key feature of our behavior networks and, as we will see, critical aspect of learning task representations.

We distinguish between three types of *sequential preconditions* which determine the activation of behaviors during the behavior network execution:

- **Permanent preconditions**: preconditions that must be met during the entire execution of the behavior. A change from met to not met in the state of these preconditions automatically deactivates the behavior. These preconditions enable the representation of sequences of the following type: *the effects of some actions must be permanently true during the execution of this behavior.*
- **Enabling preconditions**: preconditions that must be met immediately before the activation of a behavior. Their state can change during the behavior execution, without influencing the activation of the behavior. These preconditions enable the representation of sequences of the following type: *the achievement of some effects is sufficient to trigger the execution of this behavior.*
- **Ordering constraints**: preconditions that must have been met at some point before the behavior is activated. They enable the representation of sequences of the following type: *some actions must have been executed before this behavior can be executed.*

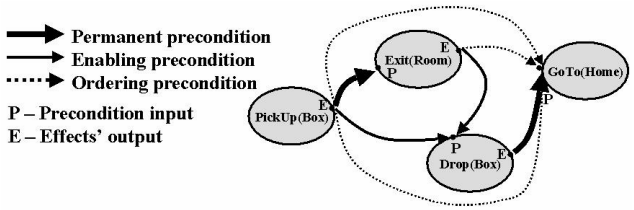


Fig. 1. Example of a behavior network

From the perspective of a behavior whose goals are **Permanent preconditions** or **Enabling preconditions** for other behaviors, these goals are what the planning literature [14] calls goals of *maintenance* and of *achievement*, respectively [14]. In a network, a behavior can have any combination of the above preconditions. The goals of a given behavior can be of maintenance for some successor behaviors and of achievement for others. Thus, since in our architecture there is no unique and consistent way of describing the conditions representing a behavior's goals, we distinguish them by the role they play as preconditions for the successor behaviors. Figure 1 shows a generic behavior network and the three types of precondition-postcondition links.

A default **Init** behavior initiates the network links and detects the completion of the task. **Init** has as predecessors all the behaviors in the network. All behaviors in the network are continuously *running* (i.e., performing the computation described below), but only one behavior is *active* (i.e., sending commands to the actuators) at a given time.

Similar to [15], we employ a continuous mechanism of activation spreading, from the behaviors that achieve the final goal to their predecessors (and so on), as follows: each behavior has an *Activation level* that represents the number of successor behaviors in the network that require the achievement of its postconditions. Any behavior with activation level greater than zero sends activation messages to all predecessor behaviors that do not have (or have not yet had) their postconditions met. The activation level is set to zero after each execution step, so it can be properly re-evaluated at each time, in order to respond to any environmental changes that might have occurred.

The activation spreading mechanism works together with precondition checking to determine whether a behavior should be active, and thus able to execute its actions. A behavior is activated iff:

(The *Activation level* != 0) AND
(All **ordering constraints** = $TRUE$) AND
(All **permanent preconditions** = $TRUE$) AND
((All **enabling preconditions** = $TRUE$) OR
(the behavior was active in the previous step))

In the current implementation, checking precondition status is performed serially, but the process could also be implemented in parallel hardware.

The behavior network representation has the advantage of being adaptive to environmental changes, whether they be favorable (achieving the goals of some of the behaviors, without them being actually executed) or unfavorable (undoing some of the already achieved goals). Since the conditions are continuously monitored, the system executes the behavior that should be active according to the current environmental state.

III. Learning from human demonstrations

A. The demonstration process

In a demonstration, the robot follows a human teacher and gathers observations from which it constructs a task representation. The ability to learn from observation is based on the robot's ability to relate the observed states of the environment to the known effects of its own behaviors.

In the implementation presented here, in this *learning* mode, the robot follows the human teacher using its **Track(color, angle, distance)** behavior. This behavior merges information from the camera and the laser-rangefinder to track any target of a known color at a distance and angle w.r.t the robot specified as behavior parameters (described in more detail in Section IV).

During the demonstration process, all of the robot's behaviors are continuously monitoring the status of their postconditions. Whenever a behavior signals the achievement of its effects, this represents an example of the robot having seen something it is able to do. The fact that the

behavior postconditions are represented as abstracted environmental states allows the robot to interpret high-level effects (such as approaching a target, a wall, or being given an object). Thus, embedding the goals of each behavior into its own representation enables to robot to perform a mapping between what it observes and what it can perform. This provides the information needed for learning by observation. This also stands in contract with traditional behavior-based systems, which do not involve explicit goal representation and thus any computational reflection.

Of course, if the robot is shown actions or effects for which it does not have any behavior representation, it will not be able to observe or learn from those experiences. For the purposes of our research, it is reasonable to accept this constraint; we are not aiming at teaching a robot new behaviors, but at showing the robot how to use its existing capabilities in order to perform more complicated tasks.

Next, we present the algorithm that constructs the task representation from the observations the robot has gathered during the demonstration.

B. Building the task representation from observations

During the demonstration, the robot acquires the status of the postconditions for all of its behaviors, as well as the values of the relevant behavior parameters. For example, for the **Tracking** behavior, which takes as parameters a desired angle and distance to a target, the robot continuously records the observed angle and distance whenever the target is visible (i.e., the **Tracking** behavior's postconditions are true). The last observed values are kept as learned parameters for that behavior.

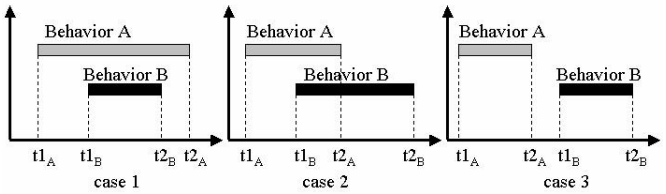


Fig. 2. Precondition types

Before describing the algorithm, we present a few notational considerations. Similar to the interval-based time representation of [16], we consider that for any behaviors A and B, the postconditions of A being met and behavior B being active are time extended events that take place during the intervals $[t1_A, t2_A]$ and $[t1_B, t2_B]$, respectively (Figure 2).

- If $t1_B \geq t1_A$ and $t1_B \leq t2_A$, behavior A is a predecessor of behavior B. Moreover, if $t2_B \leq t2_A$, the postconditions of A are permanent preconditions for B (case 1). Else, the postconditions of A are enabling preconditions for B (case 2).
- If $t1_B > t2_A$, behavior A is a predecessor of behavior B and the postconditions of A are ordering constraints for B (case 3).

Behavior network construction

1. Filter data to eliminate false indications of behavior effects. These cases are detected by having very small durations or unreasonable values of the behavior parameters.

2. Build a list of intervals for which the effects of any behavior have been true, **ordered** *by the time these events happened.* These intervals contain information about the behavior they belong to and the values of the parameters (if any) at the end of the interval. Multiple intervals related to the same behavior generate different **instances** of that behavior.

3. Initialize the behavior network as empty.

4. For each interval in the list, add to the behavior network an instance of the behavior it corresponds to. Each behavior is identified by a unique ID to differentiate between possible multiple instances of the same behavior.

5. For each interval I_j in the list:

For each interval I_k at its right in the list:

Compare the end-points of the interval I_j with those of all other intervals I_k on its right in the list: (we denote the behavior represented by I_j as J and the behaviors represented in turn by I_k with K)

- If $t2_j \geq t2_k$, then the postconditions of J are **permanent preconditions** for K (case 1). Add this permanent link to behavior K in the network.
- If $t2_j < t2_k$ and $t1_k < t2_j$, then the postconditions J are **enabling preconditions** for K (case 2). Add this enabling link to behavior K in the network.
- If $t2_j < t1_k$, then the postconditions of J are **ordering constraints** for K (case 3). Add this ordering link to behavior K in the network.

The general idea of the algorithm is to find the intervals when the postconditions of the behaviors were true (as detected from observations), and to determine the temporal ordering of those: whether they occurred in strict sequence or if they overlapped. The resulting list of intervals is ordered temporally, so one-directional comparisons are sufficient; no reverse precondition-postcondition dependencies could exist.

IV. Communication by acting - a means for robot-human interaction

Our goal is to extend a robot's model of interaction with humans so that it can induce a human to assist it by being able to express its intentions in a way that humans could easily understand. The ability to communicate relies on the existence of a shared *language* between a "speaker" and a "listener". The quotes above express the fact that there are multiple forms of *language*, using different means of communication, some of which are not based on spoken language, and therefore the terms are used in a generic way. In what follows, we discuss the different means which can be employed for communication and their use in current approaches to human-robot interaction. We then describe our own approach.

A. Language and communication in human-robot domains

Webster's Dictionary gives two definitions for *language*, differentiated by the elements that constitute the basis for communication. Interestingly, the definitions correspond well to two distinct approaches to communication in the human-robot interaction domain.

Definition 1. **Language** = *a systematic means of communicating ideas or feelings by the use of conventionalized signs, sounds, gestures, or marks having understood meanings.*

Most of the approaches to human-robot interaction so far fit in this category, since they rely on using predefined, common *vocabularies* of gestures [17], signs or words. These can be said to be using a *symbolic language*, whose elements explicitly communicate specific meanings. The advantage of these methods is that, assuming an appropriate vocabulary and grammar, arbitrarily complex information can be directly transmitted. However, as we are still far from a true dialogue with a robot, most approaches that use *natural language* for communication employ a limited and specific vocabulary which has to be known in advance by both the robot and the human users. Similarly, for *gesture* and *sign languages*, a mutually predefined, agreed-upon vocabulary of symbols is necessary for successful communication. In this work, we address communication without such explicit prior vocabulary sharing.

Definition 2. **Language** = *the suggestion by objects, actions or conditions of associated ideas or feelings.*

Implicit communication, which does not involve a symbolic agreed-upon vocabulary, is another form of using *language*, and plays a key role in human interaction. Using evocative actions, people (and other animals) convey emotions, desires, interests, and intentions. Using this type of communication for human-robot interaction, and human-machine interaction in general, is becomming very popular. For example, it has been applied to humanoid robots (in particular head-eye systems), for communicating emotional states through face expressions [18] or body movements [19], where the interaction is performed through *body language*. This idea has been explored in autonomous assistants and interface agents as well [20]. Action-based communication has the advantage that it need not be restricted to robots or agents with a humanoid body or face: structural body similarities between the interacting agents are not required to achieve successful interaction. Even if there is no exact mapping between a mobile robot's physical characteristics and those of a human user, the robot may still be able to convey a message, since communication through action also draws on human common sense [21]. In the next secton we describe how our approach achieves this type of communication.

B. Approach: communicating through actions

Our goal is to use implicit ways of communication that do not rely on a symbolic language between a human and a robot, but instead to use actions, whose outcomes are common regardless of the specific body performing them. We first present a general example that illustrates the basic idea of our approach.

Consider a prelinguistic child who wants a toy that is out of his reach. To get it, the child will try to bring a grown-up to the toy and will then point and even try to reach it, indicating his intentions. Similarly, a dog will run back and forth to induce its owner to come to a place where it has found something it desires. The ability of the child and the dog to demonstrate their intentions by calling a helper and mock-executing an action is an expressive and natural way to communicate a problem and need for help. The capacity of a human observer to understand these intentions from exhibited behavior is also natural since the actions carry intentional meanings, and thus are easy to understand.

We apply the same strategy in the robot domain. The action-based communication approach we propose for the purpose of suggesting intentions is general and can be applied across different tasks and physical bodies/platforms. In our approach, a robot performs its task independently, but if it fails in a cognizant fashion, it searches for a human and attempts to induce him to follow it to the place where the failure occurred and demonstrates its intentions in hopes of obtaining help. Next, we describe how this communication is achieved.

Immediately after a failure, the robot saves the current state of the task execution (failure context), in order to be able to later restart execution from that point. This information consists of the state of the **ordering constraints** for all the behaviors and an ID of the behavior that was active when the failure occurred.

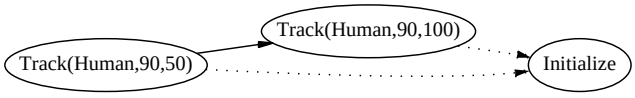
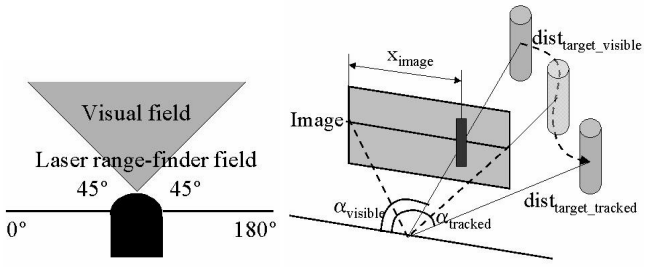


Fig. 3. Behavior network for calling a human

Next, the robot starts the process of finding and luring a human to help. This is implemented as a behavior-based system, and thus capable of handling failures, and uses two instances of the **Track(Human, angle, distance)** behavior, with different values of the *Distance* parameter: one for getting close (50cm) and one for getting farther (1m) (Figure 3). As part of the first tracking behavior, the robot searches for and follows a human until he stops and the robot gets sufficiently close. At that point, the preconditions for the second tracking behavior are active, so the robot backs up in order to get to the farther distance. Once the outcomes of this behavior have been achieved (and detected by the **Init** behavior), the robot re-instantiates the network, resulting in a back and forth *cycling* behavior, much like a dog's behavior for enticing a human to follow. When the detected distance between the robot and the human becomes smaller than the values of the *Distance* parameter for any one of its **Track** behaviors for some period of time, the cycling behavior is terminated.

The **Track** behavior enables the robot to follow colored targets at any distance in the [30, 200] cm range and any angle in the [0, 180] degree range. The information from the camera is merged with data from the laser range-finder in order to allow the robot to track targets that are outside of its visual field (see Figure 4). The robot uses the camera to first detect the target and then to track it after it goes out of the visual field. As long as the target is visible to the camera, the robot uses its position in the visual field (x_{image}) to infer an approximate angle to the target $\alpha_{visible}$

(the "approximation" in the angle comes from the fact that we are not using precise calibrated data from the camera and we compute it without taking into consideration the distance to the target). We get the real distance to the target $dist_{target\_visible}$ from the laser reading in a small neighborhood of the $\alpha_{visible}$ angle. When the target disappears from the visual field, we continue to track it by looking in the neighborhood of the previous position in terms of angle and distance which are now computed as $\alpha_{tracked}$ and $dist_{target\_tracked}$. Thus, the behavior gives the robot the ability to keep track of positions of objects around it, even if they are not currently visible, akin to working memory. This is extremely useful during the learning process, as discussed in the next section.



(a) Space coverage using laser rangefinder and camera

(b) Principle for target tracking by merging vision and laser data

Fig. 4. Merging laser and visual information for tracking

After capturing the human's attention, the robot switches back to the task it was performing (i.e., loads the task behavior network and the failure context that determines which behaviors have been executed and which behavior has failed), while making sure that the human is following. Enforcing this is accomplished by embedding two other behaviors into the task network, as follows:

- Add a **Lead** behavior as a **permanent** predecessor for all network behaviors involved in the failed task. The purpose of this behavior is to insure the human follower does not fall behind, and it is achieved by adjusting the speed of the robot such that the human follower is kept within desirable range behind the robot. Its postconditions are true as long as there is a follower sensed by the robot's rear sonars. If the follower is lost, none of the behaviors in the network is active, as task execution cannot continue. In this case, the robot starts searching again for another helper. After a few experiences with *unhelpful* humans, the robot will again attempt to perform the task on its own. If a human provides useful assistance, and the robot is able to execute the previously failed behavior, **Lead** is removed from the network and the robot continues with task execution as normal.
- Add a **Turn** behavior as an **ordering** predecessor of the **Lead** behavior. Its purpose is to initiate the leading process, which in our case involves the robot turning around (in place, for 5 seconds) and beginning task execution.

Thus, the robot retries to execute its task from the point where it has failed, while making sure that the human helper is near by. Executing the previously failed behavior will likely fail again, effectively expressing to the human the robot's problem.

In the next section we describe the experiments we performed to test the above approach to human-robot interaction, involving cases in which the human is helpful, unhelpful, or uninterested.

V. Experimental results

In order to validate the capabilities of the approach we have described, we performed several sets of evaluation experiments that demonstrate the ability of the robot to learn high-level task representations and also to naturally interact with a human in order to receive appropriate assistance when needed.

We implemented and tested our concepts on a Pioneer 2-DX mobile robot, equipped with two rings of sonars (8 front and 8 rear), a SICK laser range-finder, a pan-tilt-zoom color camera, a gripper, and on-board computation on a PC104 stack (Figure 5).

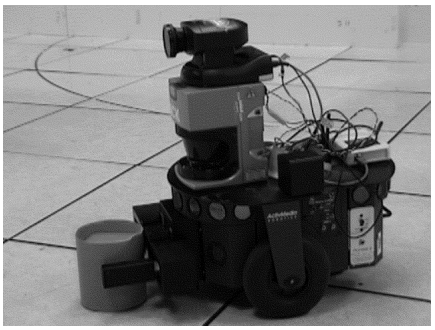


Fig. 5. A Pioneer 2DX robot

A. Evaluation criteria

We start by describing the evaluations criteria we used in order to analyze the results of our experiments, specifically the notions of success and failure.

The first challenge we addressed enables a robot to *learn high-level task representations from human demonstrations*, relying on a behavior set already available to the robot. Within this framework, we define an experiment as **successful** iff all of the following properties hold true:

- the robot learns the correct task representation from the demonstration;
- the robot correctly reproduces the demonstration;
- the task performance finishes within a certain period of time (in the same and also in changed environments);
- the robot's reports on its reproduced demonstration (sequence and characteristics of demonstrated actions) and user observation of the robot's performance match and represent the task demonstrating by the human.

Conversely, we characterize an experiment as having **failed** if any one of the properties below holds true:

- the robot learns an incorrect representation of the demonstration;
- the time limit allocated for the task was exceeded;
- the robot performs an incorrect reproduction of a correct representation.

The second challenge we addressed enables a robot to *naturally interact with humans*, which is harder to evaluate by exact metrics such as the ones that we used above. Consequently, here we rely more on the reports of the users that have interacted with the robot, and take into consideration if the final goal of the task has been achieved (with or without the human's assistance). In these experiments we assign the robot the same tasks that it has learned during the demonstration phase, but we change the environment up to the point where the robot would not be able to execute them without a human's assistance.

Given the above, we define an experiment as **successful** iff all of the following conditions hold true:

- the robot is able to get the human to come along to help if a human is available and willing;
- the robot can signal the failure in an expressive and understandable way such that the human could understand and help the robot with the problem;
- the robot can finish the task (with or without the human's help) under the same constraints of correctness as above.

Conversely, we characterize an experiment as having **failed** if any one of the properties below holds true:

- the robot is unable to find a present human or to entice a willing human to help by performing actions indicative of its intentions;
- the robot is unable to signal the failure in a way the human can understand;
- the robot is unable to finish the task due to one of the reasons above.

B. Experiments in learning from demonstration

In order to validate our learning algorithm we designed three different experiments which rely on navigation and object manipulation capabilities of the robot. Initially, the robot was given a behavior set that allowed it to track colored targets, open doors, pick up, drop, and push objects. The behaviors were implemented using AYLLU [22], an extension of the C language for development of distributed control systems for mobile robot teams.

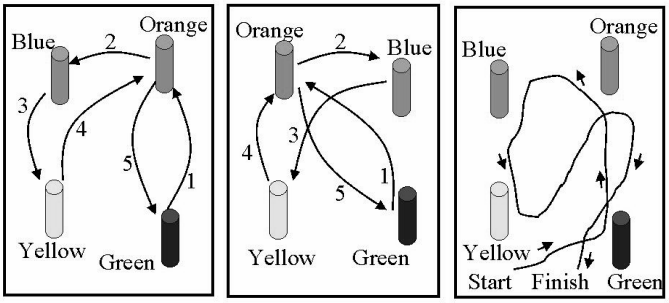
We performed three different experiments in a 4m x 6m arena. During the demonstration phase a human teacher led the robot through the environment while the robot recorded the observations relative to the postconditions of its behaviors. The demonstrations included:

- teaching a robot to visit a number of targets in a particular order;
- teaching a robot to move objects from a particular source to a particular destination location;
- teaching a robot to slalom around objects.

We repeated these *teaching* experiments more than five times for each of the demonstrated tasks, to validate that the **behavior network construction** algorithm reliably constructs the same task representation for the same demonstrated task. Next, using the behavior networks constructed during the robot's observations, we performed experiments in which the robot reliably repeated the task it had been shown. We tested the robot in executing the task five times in the same environment as the one in the learning phase, and also five times in a changed environment. We present the details and the results for each of the tasks in the following sections.

B.1 Learning to visit targets in a particular order

The goal of this experiment was to teach the robot to reach a set of targets in the order indicated by the arrows in Figure 6(a). The robot's behavior set contains a **Tracking** behavior, parameterizable in terms of the colors of targets that are known to the robot. Therefore, during the demonstration phase, different instances of the same behavior produced output according to their settings.



(a) Experimental setup (1) (b) Experimental setup (2) (c) Approximate robot trajectory

Fig. 6. Experimental setup for the target visiting task

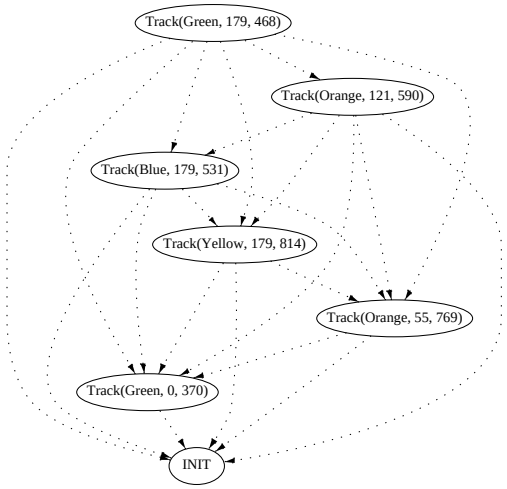
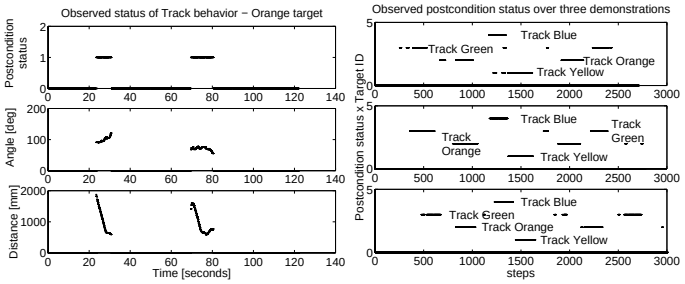


Fig. 7. Task representation learned from the demonstration of the **Visit targets** task

Figure 7 shows the behavior network the robot constructed as a result of the above demonstration.

As expected, all the precondition-postcondition dependencies between behaviors in the network are **ordering** type constraints; this is evident in the robot's observation data presented in Figure 8. The intervals during which different behaviors have their postconditions met did not overlap (case 3 of the learning algorithm) and therefore the ordering is the only constraint that has to be imposed for this task representation. More than five trials of the same demonstration were performed in order to verify the reliability of the network generation mechanism. All of the

produced controllers were identical and validated that the robot learned the correct representation for this task.



(a) Observed values of a Track behavior's parameters and the status of its postconditions

(b) Observed status of all the behaviors' postconditions during three different demonstrations

Fig. 8. Observation data gathered during the demonstration of the **Visit targets** task

Figure 9 shows the time (averaged over five trials) at which the robot reached each of the targets it was supposed to visit (according to the demonstrations) in an environment identical to the one used in the demonstration phase. As can be seen from the behavior network controller, the precondition links enforce the correct order of behavior execution. Therefore, the robot will visit a target only after it knows that it has visited the ones that are predecessors to it. However, during execution the robot might pass by a target that it was not supposed to visit at a given time. This is due to the fact that the physical targets are sufficiently distant from each other such that the robot could not see them directly from each other. Thus, the robot has to wander in search of the next target while incidentally passing by others; this is also the cause behind the large variance in traversal times. As is evident from the data, due to the randomness introduced by the robot's wandering behavior, it may take less time to visit all six targets in one trial than it does to visit only the first two in another trial.

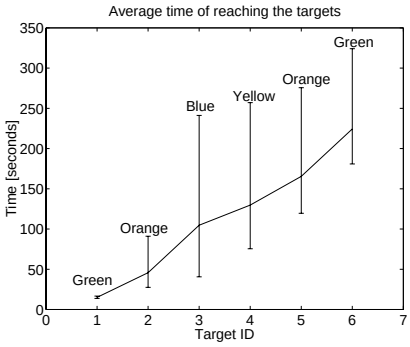


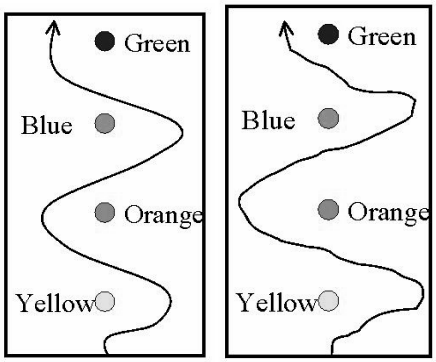
Fig. 9. Averaged time of the robot's progress while performing the **Visit targets** task

The robot does not consider these visits as achievements of parts of its task, since it is not interested in them at that point of task execution. The robot performs the correct task as it is able to discern between an intended and an incidental visit to a target. All the intended visits occur in the same order as demonstrated by a human. Unintended visits, on the other hand, vary from trial to trial as a result of different paths the robot takes as it wanders in search of targets, and are not recorded by the robot in the task achievement process.

In all experiments the robot met the time constraint, finishing the execution within 5 minutes, the allocated amount of time for this task.

B.2 Learning to slalom

In this experiment, the goal was to teach a robot to slalom through four targets placed in a line, as shown in Figure 10(a). We changed the size of the arena to 2m x 6m for this task.



(a) Experimental setup

(b) Approximate robot trajectory

Fig. 10. The **Slalom** task

During 8 different trials the robot learned the correct task representation as shown in the behavior network from Figure 11. For this case, we can observe that the relation between behaviors that track consecutive targets is of **enabling** precondition type. This correctly represents the demonstration, since, due to the nature of the experiment and of the environmental setup, the robot began to track a new target while still near the previous one (case 2 of the learning algorithm).

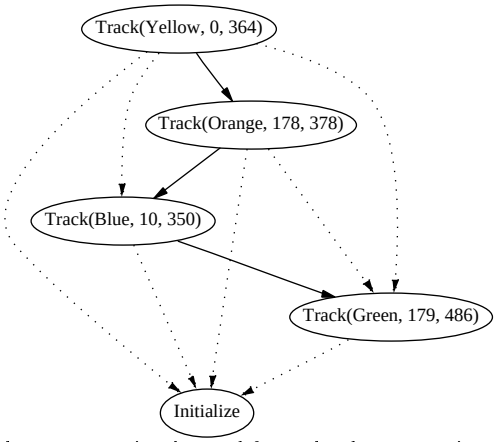


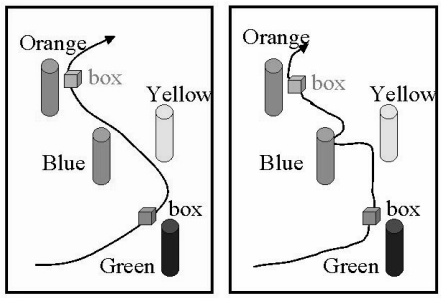
Fig. 11. Task representation learned from the demonstration of the **Slalom** task

We performed 20 experiments, in which the robot correctly executed the slalom task in 85% of the cases. The failures consisted of two types: 1) the robot, after passing one "gate," could not find the next one due to the limitations of its vision system; and 2) the robot, while searching

for a gate, turned back towards the already visited gates. Figure 10(b) shows the approximate trajectory of the robot succesfully executing the slalom task on its own.

B.3 Learning to traverse "gates" and move objects from one place to another

The goal of this experiment was to extend the complexity and thus the challenge of learning the demonstrated tasks in two ways. First, we added object manipulation to the tasks, using the robot's ability to pick up and drop objects. Second, we added the need for learning behaviors that involved co-execution, rather than only sequencing, of the behaviors in the robot's repertoire.



(a) Traversing gates and moving objects (b) Approximate trajectory of the robot

Fig. 12. The **Object manipulation** task

The setup for this experiment is presented in Figure 12(a). Close to the green target there is a small orange box. In order to teach the robot that the task is to pick up the orange box placed near the green target (the source), the human led the robot to the box, and when sufficiently near it, placed the box between the robot's grippers. After leading the robot through the "gate" formed by the blue and yellow targets, when reaching the orange target (the destination), the human took the box from the robot's gripper. The learned behavior network representation is shown in Figure 13. Since the robot started the demonstration with nothing in the gripper, the effects of the **Drop** behavior were met, and thus an instance of that behavior was added to the network. This ensures correct execution for the case when the robot might start the task while holding something: the first step would be to drop the object being carried.

During this experiment, all three types of behavior preconditions were detected: during the demonstration the robot is carrying an object for the entire time while going through the gate and tracking the destination target, the links between **PickUp** and the behavior corresponding to the actions above are **permanent** preconditions (case 1 of the learning algorithm). **Enabling** precondition links appear between behaviors for which the postconditions are met during intervals that only temporarily overlap, and finally the **ordering** constraints enforce a topological order between behaviors, as it results from the demonstration process.

The ability to track targets within a [0, 180] degree range allows the robot to learn to naturally execute the part of the task involving going through a gate. This experience is mapped onto the robot's representation as follows: "track the yellow target until it is at 180 degrees (and 50cm) with respect to you, then track the blue target until it is at 0 degrees (and 40cm)." At execution time, since the robot is able to track both targets even after they disappeared from its visual field, the goals of the above **Track** behaviors were achieved with a smooth, natural trajectory of the robot passing through the gate.

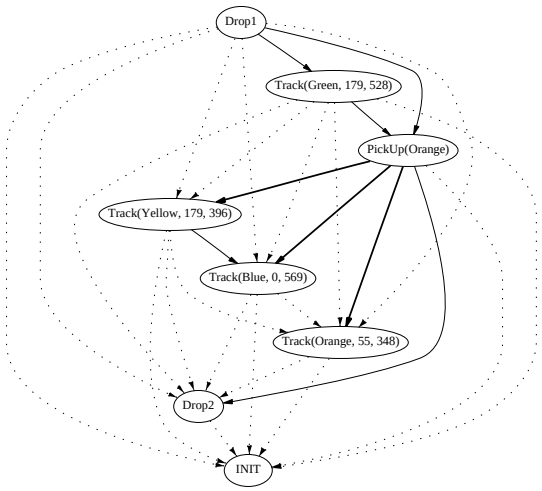


Fig. 13. Task representation learned from the demonstration of the **Object manipulation** task

Due to the increased complexity of the task demonstration, in 10% of the cases (out of more than 10 trials) the behavior network representations built by the robot were not completely accurate. The errors represented specialized versions of the correct representation, such as: **Track** the green target from a certain angle and distance, followed by the same **Track** behavior but with different parameters - when only the last was in fact relevant.

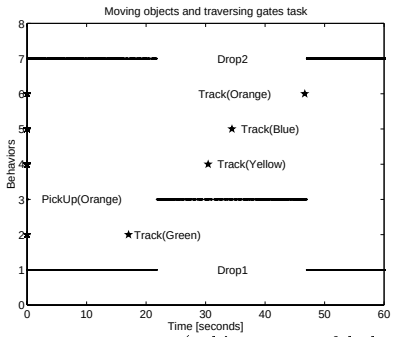


Fig. 14. The robot's progress (achievement of behavior postconditions) while performing the **Object manipulation** task

The robot correctly executed the task in 90% of the cases. The failures were all of the type involving exceeding the allocated amount of time for the task. This happened when the robot failed to pick up the box because it was too close to it and thus ended up pushing it without being able to perceive it. This failure results from the undesirable arrangement and range of the robot's sensors, not to any algorithmic issues. Figure 14 shows the robot's progress during the execution of a successful task, specifically the

intervals of time during which the postconditions of the behaviors in the network were true: the robot started by going to the green target (the source), then picked up the box, traversed the gate, and followed the orange target (the destination) where it finally dropped the box.

B.4 Discussion

The results obtained from the above experiments demonstrate the effectiveness of using human demonstration combined with our behavior architecture as a mechanism for learning task representations. The approach we presented allows a robot to automatically construct such representations from a single demonstration. The summary of the experimental results is presented in Table I. Furthermore, the tasks the robot is able to learn can embed arbitrarily long sequences of behaviors, which become encoded within the behavior network representation.

TABLE I

SUMMARY OF THE EXPERIMENTAL RESULTS.

Experiment name	Trials	Successes	
		Nr.	Percent
Six targets (learning)	5	5	100 %
Six targets (execution)	5	5	100 %
Slalom (learning)	8	8	100 %
Slalom (execution)	20	17	85 %
Object move (learning)	10	9	90 %
Object move (execution)	10	9	90 %

Analyzing the task representations the robot built during the experiments above, we observe the tendency toward over-specialization. The behavior networks the robot learned enforce that the execution go through all demonstrated the steps of the task, even if some of them might not be relevant. Since, during the demonstration, there is no direct information from the human about what is or is not relevant, and since the robot learns the task representation from even a single demonstration, it assumes that everything that it notices about the environment is important and represents it accordingly.

As any one-shot learning system, our system learned a correct, but potentially overly specialized representation of the demonstrated task. Additional demonstrations of the same task would allow it to generalize at the level of the constructed behavior network. Standard methods for generalization can be directly applied to address this issue within our framework. An alternative approach to addressing overspecialization is to allow the human to signal to the robot the saliency of particular events, or even objects. While this does not eliminate irrelevant environment state from being observed, it biases the robot to notice and (if capable) capture the key elements. In our future work we will explore both of the above approaches.

C. Interacting with humans - communication by acting

In the previous section we presented examples of learning task representation from human demonstrations. The experiments that we present next focus on another level of robot-human interaction: performing actions as a means of communicating intentions and needs.

In order to test the interaction model we described in Section IV, we used the same set of tasks as in the previous section, but changed the environment so the robot's execution of the task became impossible without some outside assistance. The failure to perform any one of the steps of the task induced the robot to seek help and to perform evocative actions in order to catch the attention of a human and get him to the place where the problem occurred. In order to communicate the nature of the problem, the robot repeatedly tried to execute the failed behavior in front of its helper. This is a general strategy that can be employed for a wide variety of failures. However, as demonstrated in our third example below, there are situations for which this approach is not sufficient for conveying the message about the robot's intent. In those, explicit communication, such as natural language, is more effective. We discuss how different types of failures require different modes of communication for help.

In our validation experiments, we asked a person that had not worked with the robot before to be close during the tasks execution and expect to be engaged in interaction. During the experiment set, we encountered different situations, corresponding to different reactions of the human in response to the robot. We can group these cases into the following main categories:

- **uninterested**: the human was not interested in, did not react to, or did not understand the robot's calling for help. As a result, the robot started to search for another helper.
- **interested, unhelpful**: the human was interested and followed the robot for a while but then abandoned it. As in the previous case, when the robot detected that the helper was lost, it started to look for another one.
- **helpful**: the human followed the robot to the location of the problem and assisted the robot. In these cases the robot was able to finish the execution of the task, benefiting from the help it had received.

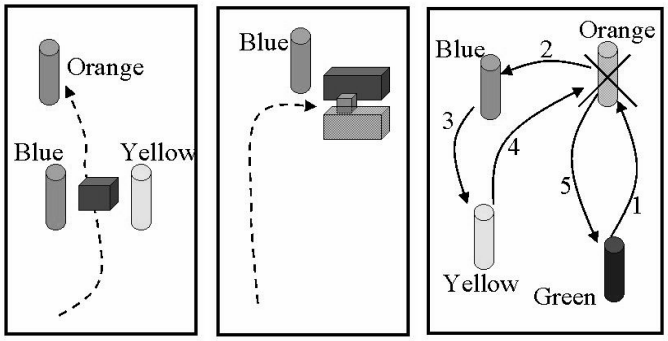
We purposefully constrained the environment in which the task was to be performed, in order to encourage human-robot interaction. The helper's behavior, consquently, had a decisive impact on the robot's task performance: when uninterested or unhelpful, failure ensued either due to exceeding time constraints or to the robot giving up the task after trying for too many times. However, there were also cases when the robot failed to find or entice the human to come along, due to visual sensing limitations or the robot failing to expressively execute its *calling* behavior. The few cases in which a failure occurred despite the assistantce of a helpful human, are presented below, along with a description of each of the three experimental tasks and overall results.

C.1 Traversing blocked gates

In this section we discuss an experiment in which a robot is given a task similar to the one learned by demonstration (presented in Section V-B.3), traversing gates formed by two closely placed colored targets. The environment (see

Figure 15(a)) is changed in that the path between the targets is blocked by a large box that prevents the robot from going through.

Expressing intentionality of performing this task is done by executing the **Track** behavior, which allows the robot to make its way around one of the targets. While trying to reach the desired distance and angle to the target, hindered by the large box, the robot shows the direction it wants to go in, which is blocked by the obstacle.



(a) Going through a gate (b) Picking up an inaccessible box (c) Visiting a missing target

Fig. 15. The human-robot interaction experiments setup

We performed 12 experiments in which the human proved to be helpful. Failures in accomplishing the task occurred in three of the cases, in which the robot could not get through the gate even after the human had cleared the box from its way. For the rest of the cases the robot successfully finished the task with the human's assistance.

C.2 Moving inaccessible located objects

A part of the experiment described in Section V-B.3 involved moving objects around. In order to induce the robot to seek help, we placed the desired object in a narrow space between two large boxes, thus making it inaccessible to the robot (see Figure 15(b)).

The robot expresses the intentions of getting the object by simply attempting to execute the corresponding **PickUp** behavior. This forces the robot to lower and open its gripper and tilt its camera down when approaching the object. The drive to pick up the object is combined with the effect of avoiding large boxes, causing the robot to go back and forth in front of the narrow space and thus convey an expressive message about its intentions and its problem.

From 12 experiments in which the human proved to be helpful, we recorded two failures in achieving the task. These failures were due to the robot losing track of the object during the human's intervention and being unable to find it again before the allocated time expired. For the rest of the cases the help received allowed the robot to successfully finish the task execution.

C.3 Visiting non-existing targets

In this section we present an experiment that does not fall into the category of the tasks mentioned above and is an example for which the framework of communicating through actions should be extended to include more explicit means of communication. Consider the task of visiting a number of targets (see Section V-B.1), in which one of the targets has been removed from the environment (Figure 15(c)). The robot gives up after some time of searching for the missing target and goes to the human for help. By applying the same strategy of executing in front of the helper the behavior that failed, the result will be a continuous wandering in search of the target from which it is hard to infer what the robot's goal and problem are. It is evident that the robot is looking for something - but without the ability to name the missing object, the human cannot intervene in a helpful way.

D. Discussion

The experiments presented above demonstrate that implicit yet expressive action-based communication can be successfully used even in the domain of mobile robotics, where the robots cannot utilize physical structure similarities between themselves and the people they are interacting with.

From the results, our observations, and the report of the human subject interacting with the robot throughout the experiments, we derive the following conclusions about the various aspects of the robot's social behavior:

- **Capturing a human's attention** by approaching and then going back and forth in front of him is a behavior typically easily recognized and interpreted as soliciting help.
- **Getting a human to follow** by turning around and starting to go to the place where the problem occurred (after capturing the human's attention) requires multiple trials in order for the human to completely follow the robot the entire way. This is due to several reasons: first, even if interested and realizing that the robot wants something from him, the human may not actually believe that he is being called by a robot in a way in which a dog would do it and does not expect that *following* is what he should do. Second, after choosing to go with the robot, if wandering in search of the place with the problem takes too much time, the human gives up not knowing whether the robot still needs him.
- **Conveying intentions** by repeating the actions of a failing behavior in front of a helper is easily achieved for tasks in which all the elements of the behavior execution are observable to the human. Upon reaching the place of the robot's problem, the helper is already engaged in interaction and is expecting to be shown something. Therefore, seeing the robot trying and failing to perform certain actions is a clear indication of the robot's intentions and need for assistance.

VI. Related work

The work presented here is most related to two areas of robotics research: robot learning and human-robot interaction. Here we discuss its relation to both areas and state the advantages gained by combining the two in the context of adding social capabilities to agents in human-robot domains.

Teaching robots new tasks is a topic of great interest in robotics. Specifically in the context of behavior-based robot learning, the majority of approaches have been at the level of learning policies, situation-behavior mappings. The method, in various forms, has been successfully applied to single-robot learning of various tasks, most commonly navigation [23], also hexapod walking [24], box-pushing [25], and has also been successfully applied to multi-robot learning [26].

Another relevant approach has been in teaching robots by demonstration, also referred to as imitation. [2] demonstrated simplified maze learning, i.e., learning turning behaviors, by following another robot teacher. The robot uses its own observations to relate the changes in the environment with its own forward, left, and right turn actions. [1] describes how robots can build models of other robots that they are trying to imitate by following them, and by monitoring the effects of those actions on their internal state of *well being*. [27] used model-based reinforcement learning to speed-up learning for a system in which a 7 DOF robot arm learned the task of balancing a pole from a brief human demonstration. Other work in our lab is also exploring imitation based on mapping observed human demonstration onto a set of behavior primitives, implemented on a 20 DOF dynamic humanoid simulation [28], [29]. The key difference between the work presented here and those above is at the level of learning. The work above focuses on learning at the level of action imitation (and thus usually results in acquiring reactive policies), while we are concerned with learning high-level, sequential tasks.

A connectionist approach to the problem of learning from human or robot demonstrations using a *teacher following* paradigm is presented in [30], [31]. The architecture allows the robots to learn a vocabulary of "words" representing properties of objects in the environment or actions shared between the teacher and the learner and also to learn sequences of "words" representing the teacher's actions.

One of the most important forms of body language, which has received a great deal of attention among researchers, is the communication of emotional states through face expressions. In some cases, the robot's emotional state is determined by physical interaction such as touch: [19] present a LEGO robot that is capable of displaying several emotional expressions in response to physical contact. In others, visual perception is used as a social cue that influences the robot's physical state: Kismet [18] is capable of conveying intentionality through its facial expressions and behavior. There, the eye movements, controlled by a repertoire of active vision behaviors, are modeled after humans and therefore have communicative value. Other researchers (e.g., [32], [33]) have also addressed the problem of human-robot interaction from the perspective of using humanoid robots, and this is quickly becoming a fast-growing area of reserach.

While facial expressions are a natural means of interaction for a humanoid, or in general a "headed," robot, they cannot be entirely applied to the domain of mobile robots, where the platforms typically have a very different, and non-anthropomorphic physical structure. [34] discusses the role of artificial emotions in social robotics for teams of mobile robots, as they could serve as a basis for mechanisms of social interaction. Aspects such as managing group heterogeneity, history of affects over time, and deriving shared meanings are all relevant for the domain of robot teams - and if addressed from the perspective of *artificial emotions* could help develop social interactions at the level of the robot group.

Human-robot or robot-robot interaction in the mobile robots domain have been mostly addressed from the perspective of using explicit methods of communication. [35] presents a system that includes, besides robots, people, automated instruments, and computers in order to implement multimodal interaction. The approach integrates speech generation with gesture recognition and gesture generation as means of communication within this heterogeneous team.

The use of implicit methods of interaction between robots is also addressed in [1], who presented an approach very related to ours. There, the robots interact by maintaining body contact, either to learn about each other's internal models or to detect if continuing the interaction is beneficial for the robot's current internal state. The interaction allows robots with different sensory capabilities to learn how to combine their abilities in order to climb hills, an action that they could not perform alone. In our approach, we demonstrate that the use of implicit, action-based methods for communicating and expressing intentions can be extended to the human-robot domain, despite the structural differences between mobile robots and humans.

VII. Conclusions

We have addressed two different but related research problems, both dealing with aspects of designing socially intelligent agents: learning from experienced demonstration and interacting with humans using implicit, action-based communication.

First, we presented a methodology that extends the framework of learning from demonstration by allowing a robot to construct high-level representations of tasks presented by a human teacher. The robot learns by relating the observations to the known effects of its behavior repertoire. This is made possible by using a behavior architecture that embeds representations of the robot's behavior goals. We have demonstrated that the method is robust and can be applied to a variety of tasks involving the execution of long, and sometimes repeated, sequences of behaviors as well as concurrently executed behaviors.

Second, we argued that the means of communication and interaction of mobile robots which do not have anthropomorphic, animal, or pet-like appearance and expressiveness should not necessarily be limited to explicit types of interaction, such as speech or gestures. We demonstrated that simple actions could be used in order to allow a robot to successfully interact with users and express its intentions. For a large class of intentions such as: *I want to do* **"this"**

- *but I can't*, the process of capturing a human's attention and then trying to execute the action and failing is expressive enough to effectively convey the message, and thus obtain assistance.

References

[1] Kerstin Dautenhahn, "Getting to know each other - artificial social intelligence for autonomous robots," *Robotics and Autonomous systems*, vol. 16, pp. 333–356, 1995.

[2] Gillian Hayes and John Demiris, "A robot controller using learning by imitation," in *Proc. of the Intl. Symp. on Intelligent Robotic Systems, Grenoble*, 1994, pp. 198–204.

[3] Brian Scasellatti, "Investigating models of social development using a humanoid robot," in *Biorobotics*, Barbara Webb and Thomas Consi, Eds. MIT Press, to appear, 2000.

[4] T. Matsui et al., "An office conversation mobile robot that learns by navigation and conversation," in *Proc., Real World Computing Symp.*, 1997, pp. 59–62.

[5] Peter Stone, Patrick Riley, and Manuela Veloso, "Defining and using ideal teammate and opponent agent models," in *Proc., IAAI, the Twelfth Annual Conf.*, 2000, pp. 1040–1045.

[6] A. David and M. P. Ball, "The video game: a model for teacher-student collaboration," *Momentum*, vol. 17, no. 1, pp. 24–26, 1986.

[7] C. Murray and K. VanLehn, "DT Tutor: A decision-theoretic, dynamic approach for optimal selection of tutorial actions," in *Proc., ITS, 6th Intl. Conf.*, 2000, pp. 153–162.

[8] V. J. Shute and J. Psotka, "Intelligent tutoring systems: past, present and future," *Handbook of Research on Educational Communications and Technology*, 1996.

[9] Sebastian Thrun et al., "A second generation mobile tour-guide robot," in *Proc. of IEEE, ICRA*, 1999.

[10] Francois Michaud and Caron S., "Roball - an autonomous toy-rolling robot," in *Proc. of the Workshop on Interactive Robotics and Entertainment*, 2000.

[11] Maja J. Matarić, "Behavior-based control: Examples from navigaton, learning, and group behavior," *Journal of Experimental and Theoretical Artificial Intelligence*, vol. 9, no. 2–3, pp. 323–336, 1997.

[12] Ronald C. Arkin, *Behavior-Based Robotics*, MIT Press, CA, 1998.

[13] Monica N. Nicolescu and Maja J Mataric, "Extending behavior-based systems capabilities using an abstract behavior representation," Tech Report IRIS-00-389, Institute for Robotics and Intelligent Systems, University of Southern California, Los Angeles, California, 2000.

[14] Stuart Russell and Peter Norvig, *Artificial Intelligence: A Modern Approach*, Prentice Hall, NJ, 1995.

[15] Pattie Maes, "Situated agents can have goals," *Journal for Robotics and Autonomous Systems*, vol. 6, no. 3, pp. 49–70, June 1990.

[16] James F. Allen, "Maintaining knowledge about temporal intervals," *Communications of the ACM*, vol. 26, no. 11, pp. 832–843, 1983.

[17] David Kortenkamp, Eric Huber, and R. Peter Bonasso, "Recognizing and interpreting gestures on a mobile robot," *Proc., AAAI*, pp. 915–921, 1996.

[18] Cynthia Breazeal and Brian Scassellati, "How to build robots that make friends and influence people," in *Proc., IROS, Kyonju, Korea*, 1999, pp. 858–863.

[19] Lola D. Canamero and Jakob Fredslund, "How does it feel? emotional interaction with a humanoid lego robot," Tech Report FS-00-04, AAAI Fall Symposium, 2000.

[20] Tomoko Koda and Pattie Maes, "Agents with faces: The effects of personification of agents," in *Proceedings, HCI*, 1996, pp. 98–103.

[21] Daniel C. Dennett, *The Intentional Stance*, MIT, CAM, 1987.

[22] Barry Brian Werger, "Ayllu: Distributed port-arbitrated behavior-based control," in *Proc., DARS, the 5th Intl. Symp.*, 2000, pp. 25–34.

[23] Marco Dorigo and Marco Colombetti, *Robot Shaping: An Experiment in Behavior Engineering*, MIT, CAM, 1997.

[24] Pattie Maes and Rodney A. Brooks, "Learning to coordinate behaviors," in *Proc., AAAI*, Boston, MA, 1990, pp. 796–802.

[25] Sridhar Mahadevan and Jonathan Connell, "Scaling reinforcement learning to robotics by exploiting the subsumption architecture," in *Eighth Intl. Workshop on Machine Learning*, 1991, pp. 328–337.

[26] Maja J Matarić, "Reinforcement learning in the multi-robot domain," *Autonomous Robots*, vol. 4, no. 1, pp. 73–83, 1997.

[27] Stefan Schaal, "Learning from demonstration," in *Advances in Neural Information Processing Systems 9*, M.C. Mozer, M. Jordan, and T. Petsche, Eds. 1997, pp. 1040–1046, MIT.

[28] Maja J Matarić, "Sensory-motor primitives as a basis for imitation: Linking perception to action and biology to robotics," in *Imitation in Animals and Artifacts*, Chrystopher Nehaniv and Kerstin Dautenhahn, Eds. MIT, to appear, 2001.

[29] Odest Chadwicke Jenkins, Maja J Matarić, and Stefan Weber, "Primitive-based movement classification for humanoid imitation," in *Proc., First IEEE-RAS Intl. Conf. on Humanoid Robotics*, Cambridge, MA, MIT, 2000.

[30] A. Billard and K. Dautenhahn, "Grounding communication in autonomous robots: an experimental study," *Robotics and Autonomous Systems, Special Issue on Scientific methods in mobile robotics*, vol. 24:1-2, pp. 71–79, 1998.

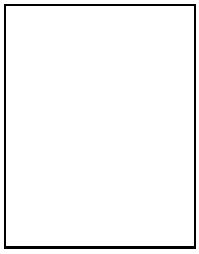
[31] A. Billard and G. Hayes, "Drama, a connectionist architecture for control and learning in autonomous robots," *Adaptive Behaviour Journal*, vol. 7:2, pp. 35–64, 1998.

[32] S. Hirano A. Takanishi and K. Sato, "Development of an anthropomorphic head-eye system for a humanoid robot," in *Proc., IEEE ICRA*, IEEE Press, Ed., 1998, pp. 59–62.

[33] K. Shibuya T. Morita and S. Sugano, "Design and control of mobile manipulation system for human symbiotic humanoid," in *Proc., IEEE ICRA*, IEEE Press, Ed., 1998, pp. 59–62.

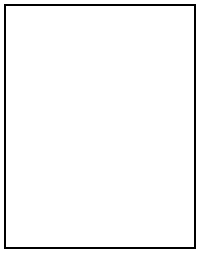
[34] Francois Michaud, Paolo Pirjanian, Audet J., and Letourneau D., "Artificial emotion and social robotics," in *Proc., DARS, the 5th Intl. Symp.*, 2000, pp. 198–204.

[35] H. Takeda, N. Kobayashi, Y. Matsubara, and T. Nishida, "Towards ubiquitous human-robot interaction," in *Working Notes, IJCAI-97 Workshop on Intelligent Multimodal Systems*, 1997, pp. 1–8.



Monica N. Nicolescu was born in Bucharest, Romania, in 1971. She received her B.S. in Computer Science (CS) from the Polytechnic University Bucharest, Romania in 1995 and her M.S. in CS from the University of Southern California (USC), Los Angeles, in 1999. She is currently working toward her Ph.D. at the USC Robotics Laboratory.

She has been a Research Assistant at the USC Robotics Laboratory since 1998 and a student member of the American Association for Artificial Intelligence since 1999. From 1990 to 1995 she was awarded a Romanian Governmental Merit-Based Fellowship and in 2000 she received a USC International Student Award. Her current research includes work on human-robot interaction and learning high-level representations, in the field of autonomous mobile robots.



Maja J Matarić was born in Beograd, Jugoslavia, in 1965. She received her B.S. in Computer Science (CS) from the University of Kansas in 1987, her M.S. in CS from MIT in 1990, and her Ph.D. in CS and Artificial Intelligence from MIT in 1994. She is an Associate Professor of Computer Science and Neuroscience at the University of Southern California (USC), the Director of the USC Robotics Labs, and the Associate Director of IRIS (Institute of Robotics and Intelligent Systems). She is a recipient of the NSF Career Award, the IEEE Early Career Award, and the MIT TR100 Innovation Award. She has worked at NASA's Jet Propulsion Lab, the Free University of Brussels AI Lab, LEGO Cambridge Research Labs, GTE Research Labs, the Swedish Institute of Computer Science, and ATR Human Information Processing Labs. Her research is in the areas of control and learning in behavior-based multi-robot systems and skill learning by imitation based on sensory-motor primitives.