Reinforcement Learning with Human Teachers: Evidence of Feedback and Guidance with Implications for Learning Performance

Andrea L. Thomaz and **Cynthia Breazeal**
MIT Media Lab
20 Ames St. E15-485, Cambridge, MA 02139
alockerd@media.mit.edu, cynthiab@media.mit.edu

Abstract

As robots become a mass consumer product, they will need to learn new skills by interacting with typical human users. Past approaches have adapted reinforcement learning (RL) to accept a human reward signal; however, we question the implicit assumption that people shall only want to give the learner feedback on its past actions. We present findings from a human user study showing that people use the reward signal not only to provide feedback about past actions, but also to provide *future directed* rewards to *guide* subsequent actions. Given this, we made specific modifications to the simulated RL robot to incorporate guidance. We then analyze and evaluate its learning performance in a second user study, and we report significant improvements on several measures. This work demonstrates the importance of understanding the human-teacher/robot-learner system as a whole in order to design algorithms that support how people want to teach while simultaneously improving the robot's learning performance.

Introduction

As robots enter the human environment as consumer products, the ability for ordinary people to easily teach them new tasks will be key to their success. Past works have addressed some of the hard problems of learning in the real world, e.g., real-time learning in environments that are partially observable, dynamic, continuous (Mataric 1997; Thrun & Mitchell 1993; Thrun 2002). However, learning quickly from interactions with a human teacher poses additional challenges (e.g., limited human patience, ambiguous human input, etc.).

The design of machines that learn by interacting with ordinary people is a relatively neglected topic in machine learning. To address this, we advocate a systems approach that integrates machine learning into a Human-Robot Interaction (HRI) framework. Our first goal is to understand the nature of the teacher's input to adequately support *how people want to teach*. Our second goal is to then incorporate these insights into standard machine learning frameworks to improve a robot's learning performance.

To contribute to each of these goals, we use a computer game framework to log and analyze interactive training sessions that human teachers have with a Reinforcement Learning (RL) agent. We study RL because of its popularity as a technique for teaching robots and game characters new skills by giving the human access to the agent's reward signal (Blumberg *et al.* 2002; Kaplan *et al.* 2002; Isbell *et al.* 2001; Evans 2002; Stern, Frank, & Resner 1998).

In experiments with this agent, we discovered that human teaching behavior introduces a new wrinkle not addressed by the traditional RL framework—namely that people use the reward signal to give *anticipatory rewards*, or future directed *guidance* for the agent (in addition to providing feedback on past actions). We then modified the RL agent and the game interface to specifically take advantage of this observed guidance behavior. In a follow up experiment, we show that guidance significantly improves the learning performance of the agent across several metrics including the speed of task learning, the efficiency of state exploration, and a significant drop in the number of failed trials encountered during learning. Therefore, by understanding the coupled human-teacher/robot-learner system, we demonstrate that it is possible to design algorithms that support how people want to teach while simultaneously improving the machine's ability to learn.

HRI Meets Machine Learning

A review of related works in machine learning yields several interesting dimensions upon which human-trainable systems can be characterized. One interesting dimension is implicit verses explicit training. For instance, personalization agents and adaptive user interfaces rely on the human as an implicit teacher to model human preferences or activities through passive observation of the user's behavior (Lashkari, Metral, & Maes 1994; Horvitz *et al.* 1998). In contrast, this work addresses explicit training where the human teaches the learner through interaction.

For systems that learn via interaction, another salient dimension is whether the human or the machine leads the interaction. For instance, active learning or learning with queries is an approach that explicitly acknowledges a human in the loop (Cohn, Ghahramani, & Jordan. 1995; Schohn & Cohn 2000). Through queries, the algorithm is in control of the interaction without regard of what the hu-

Figure 1: Sophie's Kitchen. The vertical bar is the interactive reward and is controlled by the human.

man will be able to provide in a real scenario. In contrast, this work addresses the human-side of the interaction and specifically asks *how do humans want to teach machines?*

A third interesting dimension is the balance between having the machine rely on human guidance verses its own exploration to learn new tasks. A number of systems rely on a human guidance paradigm where the learning problem is essentially reduced to programming through natural interfaces — with little if any exploration on the part of the machine, yielding a dependence on having a human present to learn (e.g., learning by demonstration (Nicolescu & Matarić 2003; Schaal 1999; Voyles & Khosla 1998; Lieberman 2001), by tutelage (Lockerd & Breazeal 2004), or straight communication (Lauria *et al.* 2002)). In contrast, modified reinforcement-based approaches (e.g., the human contributes to the reward function, or supervises the action selection) are positioned strongly along the exploration dimension (Blumberg *et al.* 2002; Kaplan *et al.* 2002; Isbell *et al.* 2001; Kuhlmann *et al.* 2004; Evans 2002; Clouse & Utgoff 1992; Smart & Kaelbling 2002).

In contrast, an important goal of this work is to create learning systems that can dynamically slide along the exploration-guidance spectrum, to leverage a human teacher when present as well as learn effectively on its own. While there are known practical issues with RL (training time requirements, representations of state and hidden state, practical and safe exploration strategies), we believe that an appropriate reformulation of RL-based approaches to include input from a human teacher could alleviate these current shortcomings. To do this properly, we must deeply understand the human teacher as a unique contribution that is distinct from other forms of feedback coming from the environment.

Experimental Platform: Sophie's Kitchen

To investigate how human interaction can and should change the machine learning process, we have implemented a Java-based computer game platform, *Sophie's Kitchen*, to experiment with learning algorithms and enhancements.

Sophie's MDP

In our system, a World $W = (L, O, \Sigma, T)$ is a finite set of k locations $L = \{l_1, \ldots, l_k\}$ and n objects $O = \{o_1, \ldots, o_n\}$. Each object can be in one of an object-specific number of mutually exclusive object states. Thus, Ω_i is the set of states for object o_i, and $O^* = (\Omega_1 \times \ldots \times \Omega_n)$ is the entire object configuration space. The task scenario used is a kitchen world (see Fig. 1), where the agent (Sophie) learns to bake a cake. This world has five objects: `Flour`, `Eggs`, a `Spoon`, a `Bowl` (with five states: `empty`, `flour`, `eggs`, `both`, `mixed`), and a `Tray` (with three states: `empty`, `batter`, `baked`). The world has four locations: `Shelf`, `Table`, `Oven`, `Agent` (i.e., the agent in the center surrounded by a shelf, table and oven). W is also defined by a set of legal states $\Sigma \subset (L \times L^O \times O^*)$. Thus, a world state $s(l_a, l_{o_1} \ldots l_{o_n}, \omega)$ consists of the agent's location, and the location and configuration, $\omega \in O^*$, of each object.

W has a transition function $T : \Sigma \times A \mapsto \Sigma$. The action space A is fixed and is defined by four atomic actions: Assuming the locations L are arranged in a ring, the agent can always `GO left` or `right` to change location; she can `PICK-UP` any object in her current location; she can `PUT-DOWN` any object in her possession; and she can `USE` any object in her possession on any object in her current location. The agent can hold only one object at a time. Each action advances the world state according to T. For example, executing `PICK-UP <Flour>` advances the state of the world such that the `Flour` has location `Agent`. `USE`ing an ingredient on the `Bowl` puts that ingredient in it; using the `Spoon` on the `both Bowl` transitions its state to `mixed`, and so on.

In the initial state, S_0, all objects and the agent are at location `Shelf`. A successful completion of the task will include putting flour and eggs in the bowl, stirring the ingredients using the spoon, then transferring the batter into the tray, and finally putting the tray in the oven. Some end states are so-called *disaster* states (for example—putting the eggs in the oven), which result in a negative reward ($r = -1$), the termination of the current trial, and a transition to state S_0.

This state space of is on the order of 10,000, with between 2 and 7 actions available in each state. The task is hierarchical, having subgoals that are necessary to achieve but not in a particular order, and has multiple solutions.

The algorithm we implemented for the experiments presented in this paper is a standard Q-Learning algorithm (learning rate $\alpha = .3$ and discount factor $\gamma = .75$) (Watkins & Dayan 1992). We chose Q-Learning as the instrument for this work because it is a simple and widely understood formulation of RL, thus affording the transfer of these lessons to any reinforcement-based approach.

Interactive Rewards Interface

A central feature of *Sophie's Kitchen* is the interactive reward interface. Using the mouse, a human trainer can—at any point in the operation of the agent—award a scalar reward signal $r = [-1, 1]$. The user receives visual feedback enabling them to tune the reward signal before sending it to the agent. Choosing and sending the reward does not halt the progress of the agent, which runs asynchronously to the interactive human reward.

The interface also lets the user make a distinction between rewarding the whole state of the world or the state of a particular object (object specific rewards). An object specific reward is administered by doing a feedback message on a

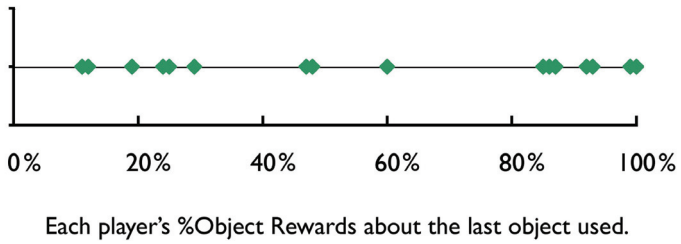


Figure 2: Many people had object rewards rarely about the last object, thus rarely used in a feedback orientation.

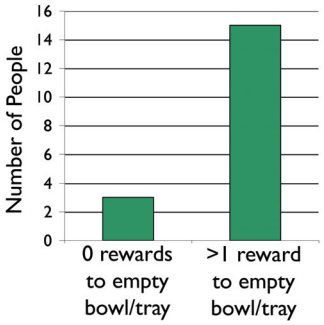


Figure 3: 15 of the 18 players gave rewards to the bowl/tray empty on the shelf, assumed to be guidance.

particular object (objects are highlighted when the mouse is over them to indicate that any subsequent reward will be object specific). We have this distinction due to a hypothesis that people would prefer to communicate feedback about particular aspects of a state rather than the entire state. However, object specific rewards are used only to learn about the human trainer's behavior and communicative intent; the learning algorithm treats all rewards in the traditional sense of pertaining to a whole state.

Evidence of Anticipatory Guidance Rewards

The purpose of our initial experiment with *Sophie's Kitchen* was to understand, when given a single reward channel (as in prior works), how do people use it to teach the agent? We had 18 paid participants play a computer game, in which their goal was to get the virtual robot, Sophie, to learn how to bake a cake on her own. Participants were told they could not tell Sophie what to do, nor could they do actions directly, but they could send Sophie the following messages via a mouse to help her learn the task:

- Click and drag the mouse up to make a green box, a positive message; and down for red/negative (Figure 4(a)).
- By lifting the mouse button, the message is sent to Sohpie, she sees the color and size of the message.
- Clicking on an object, this tells Sophie your message is about that object. As in, "Hey Sophie, this is what I'm talking about...". If you click anywhere else, Sophie assumes your feedback pertains to everything in general.

The system maintains an activity log and records time step and real time of each of the following: state transitions, actions, human rewards, reward aboutness (if object specific), disasters, and goals. Additionally, we conducted an informal interview after subjects completed the task.

Experimental Results:

From this experiment we had several findings about human training behavior and strategies. For this paper we present one important finding about the multiple communicative intents people have beyond the simple positive and negative feedback that the algorithm expects. In particular, that people want to guide the agent.

Even though our instructions clearly stated that communication of both general and object specific rewards were *feedback* messages, we found many people assumed that object specific rewards were future directed messages or guidance for the agent. Several people mentioned this in the interview, and we also find behavioral evidence in the game logs.

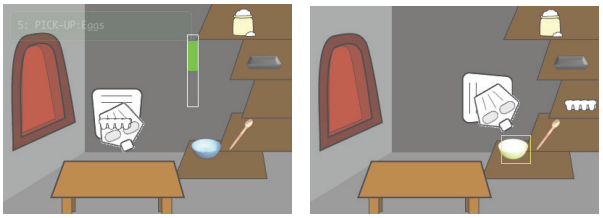
An object reward used in a standard RL sense, should pertain to the last object the agent used. Figure 2 has a mark for each player, indicating the percentage of object specific rewards that were about the last object the agent used: 100% would indicate that the player always used object rewards in a feedback connotation, and 0% would mean they never used object rewards as feedback. We can see that several players had object rewards that were rarely correlated to the last object (i.e., for 8 people less than 50% of their object rewards were about the last object).

Interview responses suggested these people's rewards actually pertain to the future, indicating what they want (or do not want) the agent to use next. A single test case is used to show how many people used object rewards as a guidance mechanism: When the agent is facing the shelf, a guidance reward could be administered (i.e., what to pick up). Further, a positive reward given to either the empty bowl or empty tray on the shelf could *only* be interpreted as guidance since this state would not be part of any desired sequence of the task (only the initial state). Thus, rewards to empty bowls and trays in this configuration serve to measure the prevalence of guidance behavior.

Figure 3 indicates how many people tried giving rewards to the bowl or tray when they were empty on the shelf. Nearly all of the participants, 15 of 18, gave rewards to the bowl or tray objects sitting empty on the shelf. This leads to the conclusion that many participants tried using the reward channel to guide the agent's behavior to particular objects, giving rewards for actions the agent was *about to do* in addition to the traditional rewards for what the agent had just done. These *anticipatory* rewards observed from everyday human trainers will require new attention in learning algorithms in order for the agent to correctly interpret their human partner.

Modifications to Leverage Human Guidance

This guidance behavior suggests that people want to speak directly to the action selection part of the algorithm to in-



(a) Feedback message. (b) Guidance message.

Figure 4: The embellished communication channel includes the feedback messages as well as guidance messages. In 4(a), feedback is given by left-clicking and dragging the mouse up to make a green box (positive) and down for red (negative). In 4(b), guidance is given by right-clicking on an object of attention, selecting it with the yellow square.

fluence the exploration strategy. To accomplish this, we added a guidance channel of communication to distinguish this intention from feedback. Clicking the right mouse button draws an outline of a yellow square. When the yellow square is administered on top of an object, this communicates a guidance message to the learning agent and the content of the message is the object. Figure 4(b) shows the player guiding Sophie to pay attention to the bowl. Note, the left mouse button still allows the player to give feedback as described previously.

Algorithm 1 describes the standard Q-Learning algorithm used for the initial interactive training sessions with *Sophie's Kitchen*, a slight delay happens in step 3 as the agent's action is animated and also to allow the human time to issue interactive rewards.

Conceptually, our modified version gives the algorithm a pre-action and post-action phase in order to incorporate the new guidance input. In the pre-action phase the agent registers guidance communication to bias action selection, and in the post-action phase the agent uses the reward channel in the standard way to evaluate that action and update a policy. The modified learning process is shown in Algorithm 2.

The agent begins each iteration of the learning loop by pausing to allow the teacher time to administer guidance (1.5 seconds). The agent saves the object of the human's guidance messages as g. During the action selection step, the default behavior chooses randomly between the set of actions with the highest Q-values, within a bound β. However, if any guidance messages were received, the agent will *instead* choose randomly between the set of actions that have to do with the object g. In this way the human's guidance messages bias the action selection mechanism, narrowing the set of actions the agent considers.

Algorithm 1 Q-Learning with Interactive Rewards:
s = last state, s' = current state, a = last action, r = reward

1: **while** learning **do**
2: a = random select weighted by $Q[s,a]$ values
3: execute a, and transition to s'
(small delay to allow for human reward)
4: sense reward, r
5: update values:

$$Q[s,a] \leftarrow Q[s,a] + \alpha(r + \gamma(max_{a'} Q[s',a']) - Q[s,a])$$

6: **end while**

Algorithm 2 Interactive Q-Learning modified to incorporate interactive human guidance in addition to feedback.

1: **while** learning **do**
2: **while** waiting for guidance **do**
3: **if** receive human guidance message **then**
4: $g = guide\text{-}object$
5: **end if**
6: **end while**
7: **if** received guidance **then**
8: a = random selection of actions containing g
9: **else**
10: a = random selection weighted by $Q[s,a]$ values
11: **end if**
12: execute a, and transition to s'
(small delay to allow for human reward)
13: sense reward, r
14: update values:

$$Q[s,a] \leftarrow Q[s,a] + \alpha(r + \gamma(max_{a'} Q[s',a']) - Q[s,a])$$

15: **end while**

Evaluation

Expert Data

To evaluate the potential effects of guidance we collected data from expert training sessions, in two conditions:

1. `No guidance`: has feedback only and the trainer gives one positive or negative reward after every action.
2. `Guidance`: has both guidance and feedback available; the trainer uses the same feedback behavior and additionally guides to the desired object at every opportunity.

For the user's benefit, we limited the task for this testing (e.g., taking out the spoon/stirring step, among other things). We had one user follow the above expert protocol for 10 training sessions in each condition (results in Table 1). The `guidance` condition is faster: The number of training trials needed to learn the task was significantly less, 30%; as was the number actions needed to learn the task, 39% less. In the `guidance` condition the number of unique states visited was significantly less, 40%; thus the task was learned more efficiently. And finally the `guidance` condition provided a more successful training experience. The number of trials ending in failure was 48% less, and the number of failed trials before the first successful trial was 45% less.

Non-Expert Data

Having found guidance has the potential to drastically improve several metrics of the agent's learning performance,

Table 1: An **expert** user trained 20 agents, following a strict best-case protocol; yielding theoretical best-case learning effects of guidance. (F = failures, G = first success).

Measure	Mean no guide	Mean guide	chg	t(18)	p
# trials	6.4	4.5	30%	2.48	.01
# actions	151.5	92.6	39%	4.9	<.01
# F	4.4	2.3	48%	2.65	<.01
# F before G	4.2	2.3	45%	2.37	.01
# states	43.5	25.9	40%	6.27	<.01

our final evaluation looks at how the agent performs with ordinary human trainers. We solicited 11 more people to play the *Sophie's Kitchen* game using both feedback and guidance messages, adding the following guidance instructions:

> *You can direct Sophie's attention to particular objects with guidance messages. Click the right mouse button to make a yellow square, and use it to help guide Sophie to objects, as in 'Pay attention to this!'*

We compare the game logs of these players (`guidance`), to 17 who played without the guidance signal (`no guidance`); summarized in Table 2.

`Guidance` players were faster than `no guidance` players. The number of training trials needed to learn the task was 48.8% less, and the number actions needed was 54.9% less. Thus, the ability for the human teacher to guide the agent's attention to appropriate objects at appropriate times creates a significantly faster learning interaction.

The `guidance` condition provided a significantly more successful training experience. The number of trials ending in failure was 37.5% less, and the number of failed trials before the first successful trial was 41.2% less. A more successful training experience is particularly desirable when the learning agent is a robot that may not be able to withstand very many failure conditions. Additionally, a successful interaction, especially reaching the first successful attempt sooner, may help the human teacher feel that progress is being made and prolong their engagement in the process.

Finally, agents in the `guidance` condition learned the task by visiting a significantly smaller number of unique states, 49.6% less than the `no guidance` condition. Moreover, we analyze the percentage of time spent in a good portion of the state space, defined as $G = \{$every unique state in $X\}$, where $X = \{$all non-cyclic sequences, $S_0, ..., S_n$, such that $n \leq 1.25(min\_sequence\_length)$, and $S_n =$ a goal state$\}$. The average percentage of time that `guidance` players spent in G was 72.4%, and is significantly higher than the 60.3% average of `no guidance` players. Thus, attention direction helps the human teacher keep the exploration of the agent within a smaller and more positive (useful) portion of the state space. This is a particularly important result since that the ability to deal with large state spaces has long been a criticism of RL. A human partner may help the algorithm overcome this challenge.

Table 2: **Non-expert** players trained Sophie with and without guidance communication and also show the positive learning effects of guidance. (F = failures, G = first success).

Measure	Mean no guide	Mean guide	chg	t(26)	p
# trials	28.52	14.6	49%	2.68	<.01
# actions	816.44	368	55%	2.91	<.01
# F	18.89	11.8	38%	2.61	<.01
# F before G	18.7	11	41%	2.82	<.01
# states	124.44	62.7	50%	5.64	<.001
% good states	60.3	72.4		-5.02	<.001

Discussion

Robotic and software agents that operate in human environments will need the ability to learn new skills and tasks 'on the job' from everyday people. It is important for designers of learning systems to recognize that while the average consumer is not familiar with machine learning techniques, they are intimately familiar with various forms of social learning (e.g., tutelage, imitation, etc.). This raises two important and related research questions for the machine learning community. 1) How do people want to teach machines? 2) How do we design machines that learn effectively from natural human interaction? In this paper we have demonstrated the utility of a *socially guided machine learning* approach, exploring the ways machines can be designed to more fully take advantage of a natural human teaching interaction.

Through empirical studies of people teaching an RL agent, we show that *people assume they can guide the agent* in addition to provide feedback. In *Sophie's Kitchen* we observed people's desire to guide the character to an object of attention, even when we explicitly told people that only feedback messages were supported. This behavior goes against RL's fundamental assumptions about the reward signal, and is a topic that has not been addressed in prior works.

In their guidance communication, people mean to bias the action selection mechanism of the RL algorithm. When we allow this, introducing a separate interaction channel for attention direction and modifying the action selection mechanism of the algorithm, we see a significant improvement in the agent's learning performance.

We use a widely understood formulation of RL to afford the transfer of the lessons and modifications presented here to any reinforcement-based approach. We have shown significant improvements in an RL domain suggesting that a situated learning interaction with a human partner can help overcome some of the well recognized problems of RL (e.g., speed and efficiency of exploration).

Our empirically informed modifications indicate ways that an interactive agent can be designed to learn better and faster from a human teacher across a number of dimensions. Guidance allows the agent to learn tasks using fewer executed actions over fewer trials. Our modifications also lead to a more efficient exploration strategy that spent more time in relevant states. A learning process, as such, that is seen

as less random and more sensible will lead to more understandable and believable agents. Guidance also led to fewer failed trials and less time to the first successful trial. This is a particularly important improvement for interactive agents in that it implies a less frustrating experience, creating a more engaging interaction for the human partner.

Importantly, in this work we acknowledge that the ground truth evaluation for systems meant to learn from people is performance with non-expert humans. This topic deserves more attention from the machine learning community, as it will be important for progress towards a social learning scenario for machines. This series of experiments with an interactive learning agent illustrates the effectiveness of this approach for building machines that can learn from ordinary people. The ability to utilize and leverage social skills is far more than a nice interface technique. It can positively impact the dynamics of underlying learning mechanisms to show significant improvements in a real-time interactive learning session with non-expert human teachers.

Conclusion

Machines meant to learn from people can leverage and support the ways in which people naturally approach teaching. In this paper we have addressed this with an approach that integrates machine learning into a Human-Robot Interaction framework, with the goal of first understanding and then designing for non-expert human teachers. Having found that people try to guide a Reinforcement Learning agent with their feedback channel, we modified the RL algorithm and interface to include an embellished channel of communication that distinguishes between guidance and feedback. An evaluation with human users shows that this empirically informed modification improves several dimensions of learning including the speed of task learning, the efficiency of state exploration, and a significant drop in the number of failed trials encountered during learning.

References

Blumberg, B.; Downie, M.; Ivanov, Y.; Berlin, M.; Johnson, M.; and Tomlinson, B. 2002. Integrated learning for interactive synthetic characters. In *Proceedings of the ACM SIGGRAPH*.

Clouse, J., and Utgoff, P. 1992. A teaching method for reinforcement learning. In *Proc. of the Nineth International Conf. on Machine Learning (ICML)*, 92–101.

Cohn, D.; Ghahramani, Z.; and Jordan., M. 1995. Active learning with statistical models. In Tesauro, G.; Touretzky, D.; and Alspector, J., eds., *Advances in Neural Information Processing*, volume 7. Morgan Kaufmann.

Evans, R. 2002. Varieties of learning. In Rabin, S., ed., *AI Game Programming Wisdom*. Hingham, MA: Charles River Media. 567–578.

Horvitz, E.; Breese, J.; Heckerman, D.; Hovel, D.; and Rommelse, K. 1998. The lumiere project: Bayesian user modeling for inferring the goals and needs of software users. In *In Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence*, 256–265.

Isbell, C.; Shelton, C.; Kearns, M.; Singh, S.; and Stone, P. 2001. Cobot: A social reinforcement learning agent. *5th Intern. Conf. on Autonomous Agents*.

Kaplan, F.; Oudeyer, P.-Y.; Kubinyi, E.; and Miklosi, A. 2002. Robotic clicker training. *Robotics and Autonomous Systems* 38(3-4):197–206.

Kuhlmann, G.; Stone, P.; Mooney, R. J.; and Shavlik, J. W. 2004. Guiding a reinforcement learner with natural language advice: Initial results in robocup soccer. In *Proceedings of the AAAI-2004 Workshop on Supervisory Control of Learning and Adaptive Systems*.

Lashkari, Y.; Metral, M.; and Maes, P. 1994. Collaborative Interface Agents. In *Proceedings of the Twelfth National Conference on Artificial Intelligence*, volume 1. Seattle, WA: AAAI Press.

Lauria, S.; Bugmann, G.; Kyriacou, T.; and Klein, E. 2002. Mobile robot programming using natural language. *Robotics and Autonomous Systems* 38(3-4):171–181.

Lieberman, H., ed. 2001. *Your Wish is My Command: Programming by Example*. San Francisco: Morgan Kaufmann.

Lockerd, A., and Breazeal, C. 2004. Tutelage and socially guided robot learning. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*.

Mataric, M. 1997. Reinforcement learning in the multi-robot domain. *Autonomous Robots* 4(1):73–83.

Nicolescu, M. N., and Matarić, M. J. 2003. Natural methods for robot task learning: Instructive demonstrations, generalization and practice. In *Proceedings of the 2nd Intl. Conf. AAMAS*.

Schaal, S. 1999. Is imitation learning the route to humanoid robots? *Trends in Cognitive Sciences* 3:233242.

Schohn, G., and Cohn, D. 2000. Less is more: Active learning with support vector machines. In *Proc. 17th ICML*, 839–846. Morgan Kaufmann, San Francisco, CA.

Smart, W., and Kaelbling, L. 2002. Effective reinforcement learning for mobile robots.

Stern, A.; Frank, A.; and Resner, B. 1998. Virtual petz (video session): a hybrid approach to creating autonomous, lifelike dogz and catz. In *AGENTS '98: Proceedings of the second international conference on Autonomous agents*, 334–335. New York, NY, USA: ACM Press.

Thrun, S. B., and Mitchell, T. M. 1993. Lifelong robot learning. Technical Report IAI-TR-93-7.

Thrun, S. 2002. Robotics. In Russell, S., and Norvig, P., eds., *Artificial Intelligence: A Modern Approach (2nd edition)*. Prentice Hall.

Voyles, R., and Khosla, P. 1998. A multi-agent system for programming robotic agents by human demonstration. In *Proceedings of AI and Manufacturing Research Planning Workshop*.

Watkins, C., and Dayan, P. 1992. Q-learning. *Machine Learning* 8(3):279–292.