# Encoding Intelligent Agents for Uncertain, Unknown, and Dynamic Tasks: From Programming to Interactive Artificial Learning

**Jacob W. Crandall**
Information Technology Program
Masdar Institute of Science and Technology
Abu Dhabi, UAE
jcrandall@mist.ac.ae

**Michael A. Goodrich** and **Lanny Lin**
Computer Science Department
Brigham Young University
Provo, UT 84602
mike@cs.byu.edu
lanny.lin@gmail.com

## Abstract

In this position paper, we analyze ways that a human can best be involved in interactive artificial learning against a backdrop of traditional AI programming and conventional artificial learning. Our primary claim is that interactive artificial learning can produce a higher return on human investment than conventional methods, meaning that performance of the agent exceeds performance of traditional agents at a lower cost to the human. This claim is clarified by identifying metrics that govern the effectiveness of interactive artificial learning. We then present a roadmap for achieving this claim, identifying ways in which interactive artificial learning can be used to improve each stage of training an artificial agent: configuring, planning, acting, observing, and updating. We conclude by presenting a case study that contrasts programming using conventional artificial learning to programming using interactive artificial learning.

## Introduction

Artificial agents, both embodied and otherwise, are increasingly being used to perform difficult tasks in uncertain, unknown, and dynamic environments. As the environment and tasks become more complex, encoding agent behaviors requires system designers to invest more effort in evaluating and encoding how an agent should act. Performing this task in traditional ways can be very difficult since system designers are often not domain experts. Furthermore, even when system designers have domain expertise, traditional methods for encoding agent behavior require considerable trial and error to understand how the agent will and should respond in various environmental conditions.

To date, the most common and effective methodology for encoding intelligent agents is *traditional AI programming*. System designers of such agents must understand both how experts perform domain specific tasks and how the environment triggers and modulates agent behaviors. Thus, agent designers must either consult with domain experts or become domain experts themselves to acquire knowledge of the specific tasks that the agents must perform. Furthermore, implementing these tasks within real environments requires considerable trial and error to reveal important details about the environment. Since these details are often unknown *a priori*, everyday end-users have the choice of implementing creative (sometimes cumbersome) workarounds (Woods et al. 2004), programming the technologies themselves (Yim 2006), or not using the technologies at all.

A second method for encoding intelligent agents is *conventional artificial learning*. Ideally, conventional artificial learning gives agents the ability to self-develop intelligent behavior, thus eliminating dependence on technology and domain experts. Unfortunately, conventional artificial learning has had only limited application to many real-world problems for at least two reasons. First, conventional artificial learning algorithms do not learn fast enough. Second, in practice, current conventional artificial learning algorithms still require technology and domain experts to specify and develop task specific learning representations, reward structures, and parameter settings. Thus, when the agent's task and environment are not known *a priori*, technology experts must tune the learning mechanisms until they achieve the desired behavior.

Recently, research efforts have begun to focus on a third method for encoding agent intelligence, called *interactive artificial learning* (IAL). In IAL, the end-user interacts with the agent via natural human-machine interactions throughout the learning process. Various approaches to IAL have been developed and analyzed, including interactive pattern classification (Fails and Olsen 2003), imitation learning (e.g., (Saunders, Nehaniv, and Dautenhahn 2005; Demiris and Meltzoff 2008)), teaching by demonstration (e.g., (Argall, Browning, and Veloso 2007; Nicolescu and Mataric 2003)), and interactive reinforcement learning (e.g., (Thomaz and Breazeal 2008)). In each case, the human iteratively interacts with the agent to help it learn intelligent behavior more quickly.

The goal of IAL is to develop agent capabilities that require minimal input and skill from the end-user. This notion is captured by *return on human investment*, measured as the ratio of agent capabilities to human involvement[1]. Figure 1 conceptually illustrates potential return on human investment for the three general programming paradigms we have discussed. Traditional AI programming requires much effort

[1] We do not distinguish between return on *designer* investment and return on *user* investment. Future work should study the interrelationship between work performed by designers and users.

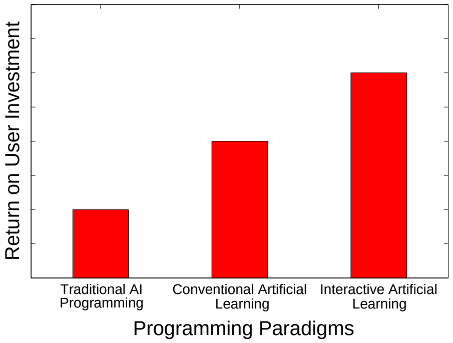

Figure 1: The various approaches to encoding agent intelligence have different returns on human investment.

from technology and domain experts, resulting in a low return on human investment. Conventional artificial learning potentially requires less work, but it still typically requires significant investment from technology experts (in the form of parameter tweaking, etc.) before the agent can learn successfully. IAL has the highest potential return on human investment of the three paradigms, though many challenges must be overcome before this potential will be brought to fruition on a large scale.

To realize the potential return on human investment, the key characteristics of IAL must be identified, and methods for implementing these key characteristics must be developed. Toward this end, in this paper, we leverage lessons learned from conventional artificial learning to identify key characteristics and recommendations for IAL. We analyze existing approaches to IAL with respect to these characteristics and recommendations. We then use a case study of conventional artificial learning to better illustrate our findings and recommendations. However, to better clarify our objectives, we first formalize metrics for IAL.

## Metrics

The metrics for the quality of conventional machine learning algorithms are generally agreed on and include things like time to convergence, proximity to the optimal solution, etc. (Mitchell 1997; Thrun 1992). Unlike conventional machine learning, much less is written about how to measure the quality of a solution to a problem involving IAL, though the amount of user input and the speed with which the learning algorithm converges are key since both affect user workload (Fails and Olsen 2003; Chernova and Veloso 2008).

Figure 2 shows an abstract illustration of how human input can improve the performance of a learning agent over time. A key aspect of this figure is that learning should improve behavior over time. We refer to this growth as the *cumulative acquisition function* (CAF) to indicate the increasing capability acquired over time in a well-designed system.

The shape of this abstract curve suggests the first potential metric for evaluating IAL: *competence*. This metric can be defined as the set of capabilities that an agent can (eventually) acquire over time using IAL. The set of acquired capabilities can be measured, for example, using performance on a specific set of benchmark tasks.

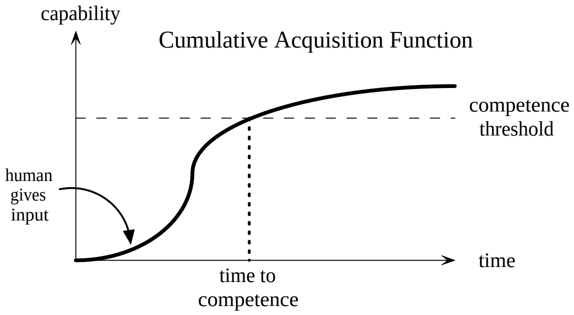

Figure 2: Robot capability grows with human input over time.

Unfortunately, maximum capability ignores the amount of work and patience that a human will invest in helping the agent grow. Although several factors contribute to the perception of work and patience, the amount of time elapsed before the robot becomes competent is one key element of such perception. Setting a threshold that delineates when the agent is competent enough to be useful (by, for example, satisfying its design specifications) introduces the second potential metric for evaluating IAL: *time to competence*. This metric is defined as the amount of time that will elapse before the agent crosses the competence threshold, as illustrated in Figure 2. This time can be measured, for example, as the amount of time before a robot learns to perform acceptably well on a required set of benchmark tasks.

We claim that the expected time to competence for IAL is potentially much lower than for conventional artificial learning or traditional AI programming. The motivation for this unproven claim is that involving the human at the right times and in the right ways causes the learning agent to rapidly acquire expertise by explicitly supporting the necessary trial and error interactions between human and agent. However, decreasing time to competence begs the question of how much work is required from a human. For example, although an agent may achieve higher competence in less time, such competence may potentially require more work from a human depending on how such work is measured (number of interactions, frequency of interactions). In the next sections, we identify areas where human input can potentially be used to minimize time to competence without imposing undue burdens on the human. Prior to doing so, we present a metric that reflects the amount of return on the human's investement.

The abstract CAF curve depicted in Figure 2 includes an inflection where capabilities transition from slow growth to rapid growth. This inflection point represents an interval of time where the human gives input. Clearly, if the CAF for one interactive learning algorithm grows more quickly than another given a fixed amount of human input, then the algorithm that produces the higher growth will be considered superior. This suggests a third potential metric for evaluating IAL: *leverage*. We define leverage as the instantaneous

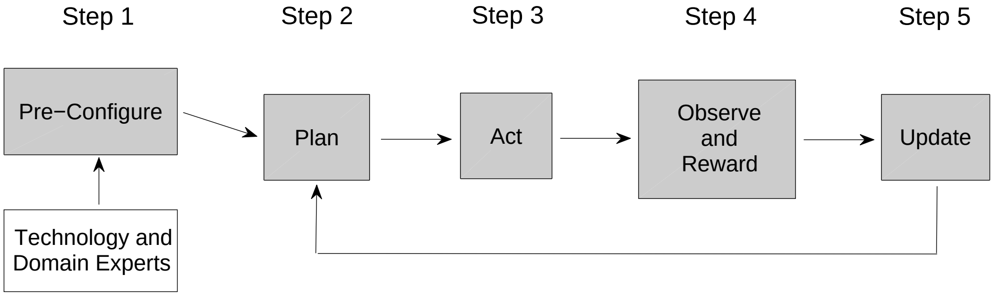

Figure 3: Conventional artificial learning process.

impact of a human-agent interaction on the agent's competence. It can be viewed as the slope of the CAF as a function of human input, and can be be measured by counting the number of interactions (or measuring the amount of time) required to produce a fixed increment in the CAF.

Although the CAF curve depicted in Figure 2 shows monotonic growth, there are several factors that may detract from an agent's competence. If the agent is oversensitive to perturbations in the environment or changes in learning parameters, than its performance may begin to decline in the presence of change. Perhaps equally important, the human working with the agent will also be learning: learning what causes the agent to change, identifying different ways to represent the problem, and discovering new aspects of the problem to be solved. Thus, IAL inherently includes at least two learning agents: the agent and the human. The potential sensitivity to perturbations and the fact that the algorithm will need to learn in the presence of other learning agents suggest a fourth potential metric for evaluating interactive machine learning: *robustness*. This metric can be measured, for example, using sensitivity analysis that shows how the growth of the CAF does not negatively change with perturbations, or by evaluating the algorithm's ability to learn productive equilibria of the multi-agent problem.

These metrics suggest several practical questions about how to design an interactive learning process. Three of these questions seem particularly important.

1. When should humans be involved in the learning process, and who determines human involvement? The decision of when and how to involve humans in the learning process is similar to the question of who has authority in human-robot interaction (Sheridan and Verplank 1978); does the human have sole authority to modify the process of learning or can the algorithm request human involvement as needed? Can the algorithm reject human input? Additionally, including human input in the learning process may mean that the human must be aware of how the agent learns and vice versa.

2. How should a system be designed to support these interactions? This includes (a) designing interfaces that show the agent's progress on the specific task as well as progress toward its maximum potential progress, (b) providing situation awareness of the learning state of the agent, and (c) supporting the user in specifying rewards, representations, etc.

3. What learning algorithms are most appropriate and for which steps of the process? This includes identifying algorithms that are general purpose, that support human input, and that are powerful enough to learn complicated tasks. Additionally, since humans are likely to adapt to changes in the agent's capabilities, the algorithm used by the agent should be capable of learning in the presence of other learning agents.

To begin to answer these questions, we analyze the conventional artificial learning process, and seek to learn necessary characteristics of IAL from this process.

## Programming Using Conventional Artificial Learning

An idealized process of conventional artificial learning is shown in Figure 3. In the first step of conventional artificial learning, called *pre-configuration*, the system designer defines the learning mechanisms, reward structure, parameter values, and state and feature representations used by the agent to learn. After pre-configuration, the algorithm learns based on the decisions made in this beginning step, beginning with using its learning mechanisms to *plan* its next action or behavior. The agent then *acts* out its plan, *observes* the outcomes of its actions, and *updates* its internal state, utility estimates, etc. based on the learning mechanisms defined in the pre-configuration step. The agent then plans a new action, and the process repeats.

The success of the idealized learning process depicted in Figure 3 is heavily dependent on the choices made by the system designer in the pre-configuration step. Since system designers often cannot envision *a priori* the various attributes of the problem the agent must learn, this conventional learning process is often unsuccessful. Upon realizing that the agent has failed to learn effectively, the system designer must analyze the agent's policies, utilities, etc. throughout the learning process to determine why it failed. After determining why the algorithm failed, the system designer then re-configures the learning algorithm by tweaking the algorithm's tunable parameters, learning mechanisms, and reward structure (Figure 4). The conventional learning

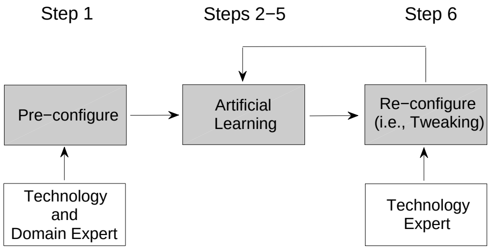

Figure 4: Classical learning process – what really happens.

process of plan, act, observe, and update is then repeated until success (after which, in the academic setting, the system designer publishes a paper describing the "successful" learning algorithm).

The need to tweak the algorithm and repeat until success results in a low return on human investment. Furthermore, it requires system designers to be technology and domain experts. In efforts to reduce these concerns, much effort has been placed on developing algorithms that automatically determine good features, states, and parameters values (e.g., (Kaiser 2007; Kohavi and John 1995)). However, to date, most situations still require technology and domain experts to spend large amounts of time developing new learning mechanisms, tuning parameters, etc.

Rather than exclusively seeking to develop increasingly sophisticated learning algorithms, we advocate that IAL algorithms should be developed. These algorithms acknowledge the need for the human to play the role of a teacher and collaborator in the learning process. Thus, rather than remove the human from the loop, IAL reframes the learning process to one of iterative improvement that seeks to exploit human capabilities and knowledge in order to increase the return on human investment.

## Programming Intelligent Agents Using Interactive Artificial Learning

A generalized process of IAL is depicted in Figure 5. IAL involves the same steps as conventional artificial learning. However, in IAL, the end-user, who is not required to be a technology or domain expert, can potentially be involved in each step of the learning process. In so doing, decisions and assumptions in configuring the algorithm can be modified throughout the learning process, thus decreasing the algorithms dependence on the decisions made during configuration.

Involving the user in iterative interactions with the agents in the various steps of the learning process is important for a number of reasons. First, it allows the agent to benefit from the knowledge, intuition, and goals articulated by the user. Second, these interactions allow the user to build situation awareness of not just what is going on in the world, but also how much the agent knows. This information will help the user to provide more efficient input to the agent. To better understand how human-machine interactions can influence the learning process, we discuss each step separately. Furthermore, we note how various approaches to IAL from the literature have addressed these interactive needs.

### Configuration

Prior to learning, the user configures the algorithm by specifying various representations, learning mechanisms, parameter values, etc. However, unlike the pre-configuration step in conventional artificial learning in which the human imposes these decisions on the agent, we envision an interactive process in which the user and the agent collaborate to define and initialize the learning algorithm. For example, the agent and the user could collaborate to select a desirable artificial learning mechanism from a toolbox of learning algorithms, and the agent could help the user to select reasonable learning parameters. Additionally, collaborative tools could be designed to support an intuitive process for selecting state representations, relevant features, etc. and to create a reward structure that supports scaffolding (Saunders, Nehaniv, and Dautenhahn 2006). While these selections need not be perfect since they can be altered throughout the learning process, this collaborative process could significantly increase the agents ability to quickly become competent. Thus, *configuration* (rather than *pre-configuration*) becomes very much part of the complete IAL process.

### Planning

The planning step consists of determining a policy or strategy based on the knowledge, rules, and representations held by the system. In conventional artificial learning, this means generating a policy given the utility estimates and policy-construction rules specified in the pre-configuration step. However, when the user is involved in planning, these utility estimates and policy-construction rules can be augmented with the user's knowledge and intuition to greatly augment the learning process. To do so, the "black-box" methodologies utilized in conventional artificial learning must be replaced by a more collaborative experience in which the user and the agent can ask each other questions, provide answers and make hypothesis, and then negotiate an effective strategy. During this collaborative process, both the user and the agent could improve their understanding of the problem.

With the exception of (Thomaz and Breazeal 2008), we are unaware of approaches to IAL that address such collaborative processes in detail. As part of this position paper, we propose three characteristics that an IAL should have for planning. First, the artificial agent must be able to communicate its internal state, utilities, and policies. For example, Thomaz and Breazeal used visual cues for a robot to communicate uncertainty in which action to implement (Thomaz and Breazeal 2008). Other methodologies could involve data visualization algorithms.

Second, a collaborative planning process for IAL should allow the user and the agent to explore the estimated effects of different actions. For example, (Goodrich et al. 2007) allowed a human to investigate how altering a planning algorithm's parameters affected the system's plan before committing to the plan. Additionally, (Demiris and Meltzoff 2008) discussed using forward and inverse models in imitation learning. Such forward models could be used to pre-

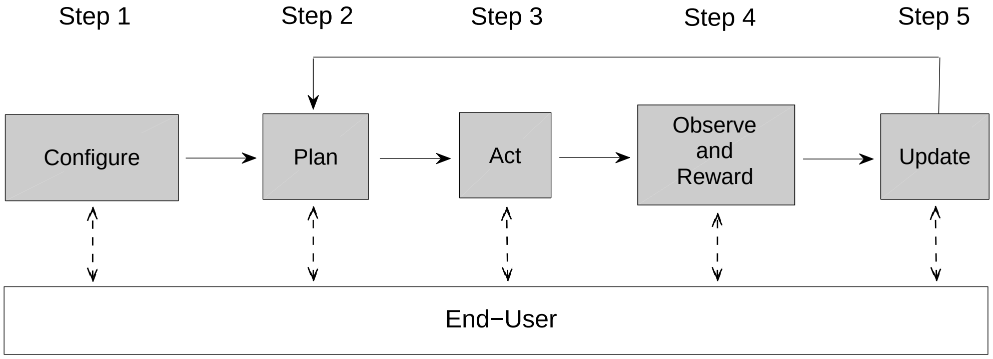

Figure 5: The interactive artificial learning process.

dict actions, thus improving the collaborative process (if presented in a way that the human can understand). Principles learned from research in decision support systems could inform the development of these collaborative tools.

Third, an IAL should use the user's input in the planning stage to improve its utility estimates, learning mechanisms, etc. However, as in other input provided by the human throughout the learning process, we cannot assume that the human will always be correct. As such, the agent must be able to tolerate incorrect feedback. To do this, we anticipate that the agent will need to distinguish what it learns in the planning steps from what it learns from its own experiences (in the observe and update steps).

## Acting

As in conventional artificial learning, the acting step in IAL involves carrying out the intended action. Typically, this involves invoking the policy determined in the planning step, though it can possibly include situations in which the human pre-empts this policy.

The effects of human involvement in the acting step have been one of the most studied aspects of IAL. Both imitation learning (e.g., (Demiris and Meltzoff 2008; Saunders, Nehaniv, and Dautenhahn 2005)) and teaching by demonstration (e.g., (Argall, Browning, and Veloso 2007)) fall in this realm. In both cases, the artificial agent observes the actions taken by the human, and learns thereby. If the human acts appropriately, the agent can learn many aspects of its policy space very quickly. In effect, the human is able to guide the agent's learning so that it can quickly learn desirable actions.

One potential reason for this speed up in learning is that training examples tend to cluster in critical areas of the decision boundary (Fails and Olsen 2003). This leads to the question of how to decide when human input will be most useful; does the human intervene or does the machine request help? (Thomaz and Breazeal 2008) gave this responsibility to the user, while (Grollman and Jenkins 2007) and (Chernova and Veloso 2008) implemented procedures in which the agent requested the user's input when its uncertainty was too high. We note that, in these cases, the particular strategies were chosen based on the context in which the agents were used.

One other potential reason for involving the user in the acting step of IAL is to increase the user's awareness of the agent's task and environment. The act of implementing the action could cause the user to think more deeply about the challenges the agent is encountering, thus allowing the user to provide more useful input to the agent in other steps of the learning process. While we are not aware of this approach being implemented in the IAL literature, there are parallels in the literature on adjustable autonomy in human-robot teams. For example, Kaber and Endsley explore the effects of periodic manual control on the human operator's situation awareness (Kaber and Endsley 2004).

## Observing and Rewarding

After acting, the agent and (when desired) the human observe the consequences of the action that was taken. Based on the system's goals and preferences, a payoff is ascribed to this outcome. Involving the user in this step is desirable for three reasons. First, the user can be used to determine the utility of the outcome (e.g., (Thomaz and Breazeal 2008)). While this could be done in the configuration step when the user defines the reward structure of the environment, it is often difficult to specify the reward structure *a priori*. Additionally, the reward structure could change over time.

We note that rewards need not be scalars, as real-world reward signals can be much richer than a single payoff. A learning algorithm may require a vector of rewards, one for each attribute of the problem (e.g., (Goodrich and Quigley 2004; Crandall and Goodrich 2004)). While it may be unnecessary or impractical for the human to specify all reward signals, the human may be asked to specify part of it. For example, an action may be associated with both monetary and social rewards. While the monetary reward might be easily observable and quantifiable, the agent may have more difficulty detecting social consequences. Thus, the user could communicate this reward to the agent.

A second reason for involving the human in the observing and rewarding step is to employ scaffolding. Since a task may be too difficult for the agent to learn by itself initially, it is often necessary to help the agent first learn a simpler task. This can be accomplished by reinforcing behavior that accomplishes the simpler task, even though the actual observed outcome does not fully meet the agent's goals. Examples of using rewards to scaffold the environment in IAL include the approaches used by (Saunders, Nehaniv, and Dautenhahn 2006) and (Thomaz and Breazeal 2008).

Third, the human can be used to help identify and annotate an outcome that the agent does not see or understand. Such a scenario could develop due to deficiencies in the agent's sensor system or in its ability to extract meaning from the data gathered from its sensors.

### Updating

After observing the outcome of the performed action, the agent updates its utility estimates and internal state. Additionally, in IAL the update step provides an opportunity for the agent, with the help of the user, to update its learning mechanisms, state and feature representations, etc. We anticipate that the kinds of human-machine interactions we suggested for the planning step can be beneficial in the update step as well. For example, (Fails and Olsen 2003) provide the user with a visual representation of how the agent perceives the world (via a classifier) after the agent performs the update. This information can help the user understand what information the agent lacks. Additionally, the user involved in the update step could help the agent solve critical issues such as credit assignment.

As in planning, updating in conventional artificial learning often involves processes that are difficult for the user to understand. For the user to become effectively involved in this step, these "black-box" processes must be replaced by collaborative interactions that allow the agent and user to effectively communicate knowledge, intuition, and ideas. Thus, the characteristics of successful interactions in the planning step will also be applicable in the update step.

Despite the potential benefits of human-machine collaborations in the update step, with the exception of (Fails and Olsen 2003), we are not aware of any papers that address this issue in detail. Future work in this area is needed to determine effective interactive processes for updating.

### Summary

Each step of the conventional artificial learning process can be converted into a collaborative interaction between the user and the agent. These interactions will potentially serve to quickly increase the agent's competence, while providing the user with a better situation awareness of the internal state and learning mechanisms of the agent. Thus, we argue that successful IAL algorithms should implement each of these types of interactions to improve the return on human investment. To better illustrate these claims, we present a case study in the next section.

## Case Study – Multi-agent Learning

In the previous two sections, we discussed the conventional and interactive artificial learning processes. The conventional artificial learning process, while well-studied, does not yield a high return on human investment due to the considerable trial and error that is required to understand and implement how the agent should respond in various environmental conditions. However, conventional artificial learning illustrates the important human-machine interactions that should be considered in IAL. To better illustrate what we mean, we present a case study in which we developed a multi-agent learning algorithm.

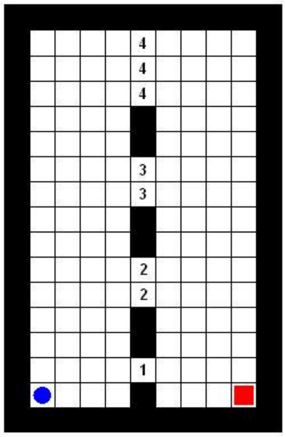

Figure 6: Prisoners' dilemma game in maze form.

| | cooperate | defect |
|---|---|---|
| ccooperate | (24, 24) | (8, 30) |
| defect | (30, 8) | (15, 15) |

Table 1: Generalized payoff matrix for the maze game.

### Task and Environment

We sought to create an agent that could learn successfully in the game, shown in Figure 6, that modeled an iterated prisoner's dilemma (Crandall and Goodrich 2004). In the game, two agents (the circle and the square in the figure) began on opposites corners of the maze. The agents were separated by a wall containing four different gates which were initially open. The goal of each agent was to move across the world to the other agent's starting position in as few moves as possible. The physics of the game were as follows:

1. Each agent could move up, down, left, and right. Moves into walls or closed gates resulted in the agent remaining where it was before the action was taken.
2. If both agents attempted to move through gate 1 at the same time, gates 1 and 2 closed (without allowing either of the agents passage).
3. If one agent moved through gate 1 and the other agent did not, then gates 1, 2, and 3 closed (after the defecting agent moved through the gate).
4. If one agent moved through any of the gates, then gate 1 closed.
5. When an agent reached its goal state, it received a reward of $r = 40 - n$, where $n$ is the number of steps taken to reach the goal.

After both agents reached their respective goals, the maze was reset, and the game repeated.

In this game, when an agent attempts to move through gate 1, it is said to have defected. Otherwise, it is said to have cooperated. Viewed in this way, the game turns into the prisoner's dilemma matrix game shown in Table 1. In repeated prisoner's dilemmas, successful agents learn to cooperate with other successful agents, and they learn to defect against agents that tend to defect (Axelrod 1984).

This simple game abstracts many real-world problems, including self-configuring sensor networks and online markets. It is challenging for learning agents to solve since (1) the game has a large state space (especially when the position and actions of the other agent are considered part of the agent's state), (2) the environment in which the agent must learn in is non-stationary when the other agent is learning, and (3) the agents have conflicting interests, though both can benefit from mutual cooperation.

In addition to being a challenging game for artificial agents, the game is also challenging for humans to learn to play. However, humans do learn to avoid being exploited and to (usually) play cooperatively with each other. We sought to duplicate this successful behavior with an artificial learning agent using conventional artificial learning.

## Programming Using Conventional Artificial Learning

Our experience developing a successful multi-agent learning algorithm for this game followed the following stages:

1. We pre-configured our agent using traditional reinforcement learning algorithms from the literature. However, these algorithms took a long time to learn. Furthermore, they learned to defect when playing against other learning agents, resulting in low payoffs. True to the conventional artificial learning paradigm, we began to tweak the agent's learning mechanisms, parameters, and state representations in hopes of obtaining better results.

2. After several iterations failed to produce a successful learning algorithm, we began to realize that we did not have enough domain knowledge. To compensate, we developed a GUI that allowed us to play the game against each other and against artificial agents employing various learning algorithms. This provided us with intuition for a new learning algorithm.

3. The new algorithm, called SPaM, learned two customized utility functions and a mechanism to determine agent behavior based on these utilities (Crandall and Goodrich 2004). We then spent significant amounts of time using trial and error to refine the algorithm's learning mechanisms and representations. Much of this time was spent viewing the agent's utility estimates as it learned in the game, studying why certain utility estimates were "incorrect," and tweaking the algorithm to compensate.

SPaM performs very effectively in this prisoner's dilemma maze game. It quickly teaches many associates (including humans, itself, and other learning agents) to cooperate, but learns to defect against associates that are not apt to cooperate (Crandall and Goodrich 2004). However, despite these successes, the long development cycle meant that we received a low return on our investment. We spent significant amounts of time acquiring domain knowledge and analyzing and tweaking the learning algorithm.

We anticipate that an IAL approach could produce a similarly competent learning algorithm in significantly less time.

## Lessons for Interactive Artificial Learning

For an IAL algorithm to be success in the prisoner's dilemma maze game, we believe that it would need to provide human-machine interactions for the various steps of the IAL process. We briefly describe potential interactive methodologies in these steps.

*Configuring.* In developing SPaM, our initial selection of learning mechanisms and representations were insufficient to learn effective behavior in this particular maze game. Thus, we would have found it useful to have a set of learning algorithms to experiment with. Additionally, it would have been helpful to scaffold the problem initially, by teaching the agent to first navigate the world before teaching it whether to "defect" or "cooperate".

*Planning.* When designing our learning algorithm, we would have found it useful to have the agent construct and visualize potential plans as we selected different parameter values. For this game, one of the key parameters that affects strategic behavior is the first move – moving up indicates a willingness to cooperate but moving toward the door suggest the possibility of defection. Thus, during planning, we envision interactive tools in which the user can bias the learning agent toward one of these behaviors by modifying parameters specifying the agent's "aggressiveness."

*Acting.* The multi-agent learning problem is challenging since learning associates cause the environment to be non-stationary. This means that an agent's strategies must adapt to the changes in the environment. However, agent's strategies often change too slowly. In these situations, we envision a situation in which the human takes over acting until the agent's policy can "catch up" with the environment. This would allow the agent to continue to perform well in changing environments while potentially speeding up learning.

*Rewarding.* Similar to Thomaz and Breazeal's work on treating positive and negative feedback differently (Thomaz and Breazeal 2007), it would have been useful to reward agent-selected behaviors that displayed good strategic values, and penalize behaviors that showed shortsightedness.

*Updating.* From observing both humans and artificial agents play the maze game, it is clear that humans make substantially more changes to their utility estimates given experiences than do current artificial learning algorithms. Thus, a user could help the agent to update its estimates more quickly. To do this, the human would likely need to view a representation of the agent's utility estimates after an update is performed, and to allow the human to modify or influence these utilities to help the agent learn without waiting for reward signals to propagate through the system.

## Summary and Discussion

Interactive artificial learning is an exciting and relatively new research area with great potential. To date, several interesting and useful approaches have been proposed and evaluated. However, in analyzing conventional artificial learning and why it works, we believe that interactive artificial learning algorithms will need significantly more interactive power than they currently have if they are to allow end-users to program their own agents. In particular, our experience highlights three important aspects of the problem that future research in interactive artificial learning should address:

1. Future IAL algorithms should allow the user to view and understand the agent's internal state.
2. Future IAL algorithms should allow the user to interact with the agent's internal estimates and values.
3. Future IAL algorithms should allow users to alter the learning mechanisms and representations employed by the learning algorithm.

While challenging, we believe that these objectives can be met in many tasks by creating rich human-machine interactions for the various steps of the learning process, including configuration, planning, acting, observing and rewarding, and updating. Future work should evaluate what kinds of tasks are appropriate for these potential aspects of IAL.

## References

Argall, B.; Browning, B.; and Veloso, M. 2007. Learning by demonstration with critique of a human teacher. In *Proceedings of the Second ACM/IEEE International Conference on Human Robot Interaction*, 57–64.

Axelrod, R. M. 1984. *The Evolution of Cooperation*. Basic Books.

Chernova, S., and Veloso, M. 2008. Multi-thresholded approach to demonstration selection for interactive robot learning. In *Proceedings of the Third ACM/IEEE International Conference on Human-Robot Interaction*, 225–232.

Crandall, J. W., and Goodrich, M. A. 2004. Establishing reputation using social committment in repeated games. In *Proceedings of the AAMAS-2004 Workshop on Learning and Evolution in Agent Based Systems*.

Demiris, Y., and Meltzoff, A. 2008. The robot in the crib: A developmental analysis of imitation skills in infants and robots. *Infant and Child Development* 17:43–53.

Fails, J. A., and Olsen, Jr., D. R. 2003. Interactive machine learning. In *Proceedings of the Eighth International Conference on Intelligent User Interfaces*, 39–45. Miami, Florida, USA: ACM.

Goodrich, M. A., and Quigley, M. 2004. Satisficing Q-learning: Effcient learning in problems with dichotomous attributes. In *International Conference on Machine Learning and Applications*.

Goodrich, M. A.; McLain, T. W.; Anderson, J. D.; Sun, J.; and Crandall, J. W. 2007. Managing autonomy in robot teams: observations from four experiments. In *Proceedings of the Second ACM/IEEE international conference on Human-robot interaction*, 25–32.

Grollman, D., and Jenkins, O. 2007. Dogged learning for robots. In *IEEE International Conference on Robotics and Automation*, 2483–2488.

Kaber, D. B., and Endsley, M. R. 2004. The effects of level of automation and adaptive automation on human performance, situation awareness and workload in a dynamic control task. *Theoretical Issues in Ergonomics Science* 5(2):113–153.

Kaiser, D. M. 2007. Automatic feature extraction for autonomous general game playing agents. In *Proceedings of the $6^{th}$ International Conference on Autonomous Agents and Multiagent Systems*, 1–7.

Kohavi, R., and John, G. 1995. Automatic parameter selection by minimizing estimated error. In *Proceedings of the $12^{th}$ International Conference on Machine Learning*, 304–312.

Mitchell, T. 1997. *Machine Learning*. McGraw Hill.

Nicolescu, M. N., and Mataric, M. J. 2003. Natural methods for robot task learning: Instructive demonstration, generalization and practice. In *Proceedings of the Second International Conference on Autonomous Agents and Multi-Agent Systems*, 241–248.

Saunders, J.; Nehaniv, C. L.; and Dautenhahn, K. 2005. An examination of the static to dynamic imitation spectrum. In *Proceedings of the Third International Symposium on Imitation in Animals and Artifacts*, 109–118. Hatfield, UK: The Society for the Study of Artificial Intelligence and Simulation of Behavior.

Saunders, J.; Nehaniv, C. L.; and Dautenhahn, K. 2006. Teaching robots by moulding behavior and scaffolding the environment. In *First Annual Conference on Human-Robot Interaction*, 142–150.

Sheridan, T. B., and Verplank, W. L. 1978. Human and computer control of undersea teleoperators. Technical report, MIT Man-Machine Systems Laboratory.

Thomaz, A. L., and Breazeal, C. 2007. Asymmetric interpretations of positive and negative human feedback for a social learning agent. In *The 16th IEEE International Symposium on Robot and Human interactive Communication*, 720–725.

Thomaz, A. L., and Breazeal, C. 2008. Teachable robots: Understanding human teaching behavior to build more effective robot learners. *Artificial Intelligence* 172(6-7):716–737.

Thrun, S. B. 1992. Efficient exploration in reinforcement learning. Technical Report CMU-CS-92-102, Carnegie-Mellon University.

Woods, D. D.; Tittle, J.; Feil, M.; and Roesler, A. 2004. Envisioning human-robot coordination in future operations. *IEEE Transactions on Systems, Man, and Cybernetics – Part C: Systems and Humans* 34:210–218.

Yim, M. 2006. Astronauts must program. In *AAAI 2006 Spring Symposium: To Boldly Go Where No Human-Robot Team Has Gone Before*. Menlo Park, California, USA: AAAI.