Communicating Values to Autonomous Agents

Daniel Shapiro
Institute for the Study of Learning and Expertise
2164 Staunton Court, Palo Alto, CA 94306
shapiro@isle.org

Paul Collopy
DFM Consulting, Inc.
Post Office Box 247, Urbana, IL 61803
paul@dfmconsulting.com

Abstract

Although autonomous systems can support a wide variety of application goals, their perceived risk inhibits their deployment. While this problem is traditionally associated with an agent's skills, we take the novel perspective that the issue is *communication*. In this view, the agent simply needs a better representation of the user's interests so that its choices will not produce unintended effects. This paper proposes a methodology for constructing artificial agents that is rooted in this perspective. In particular, we show how the goal of *aligning* agent held objective functions with human utility can be transformed into a practical methodology for constructing, and then utilizing agents that provably act in their users' best interests.

Introduction

Autonomous agents are technological artifacts that perform tasks for people. They sense their environment, relate their perceptions to their tasks, and then select an appropriate response. In principle, such agents can perform complex functions, which give them the potential to greatly expand our reach. However, the risk increases as autonomy grows. Agents acting in critical applications can make catastrophic errors, while a complex artifact operating outside of human supervision seems almost guaranteed to drift from our intent over time. These concerns create a significant barrier to deployment. As a result, we need technology to enhance the trustworthiness of autonomous systems.

Current methods for supplying trust operate in one of several broad ways. First, we can develop skills that handle a wider variety of environmental conditions (a goal of reactive systems). Next, engineers can constrain the environment to make the available technology more robust, as with robotic assembly lines. However, this is infeasible in many circumstances (e.g., for robots on Mars). A third approach employs algorithms that support predictive analysis and/or formal guarantees, generally with the caveat that the environment must satisfy certain assumptions. Unfortunately, this approach restricts the complexity of the relevant tasks. In contrast, research that expands the scope of feasible tasks can negatively impact trust (e.g., work on agent architectures), as increased autonomy often elevates perceived risk.

While each of these methods associates trustworthiness with agent skills, we treat *communication* as the core issue. In this view, the agent is the user's delegate, and the problem is to provide it with a representation of its user's interests that will let it choose wisely in its user's stead. Thus, our goal is concordance. Whether the agent's capabilities are large or small, a good agent will make the same choices its user would make if the user were in the agent's shoes. In contrast, an undesirable agent lacks a sufficient model of its user, and can act in what appears to be an optimal fashion while making inefficient, surprising, and even dangerous decisions on its user's behalf.

We develop this perspective by adopting the framework for value alignment (Shapiro 2001), which provides a formal structure for relating agent reward functions to human utility, as well as two key theoretical results. The first guarantees that an *aligned* agent will maximize human utility as a consequence of maximizing its own reward (it will do everything it can for its user that lies within its skills). The second states that it is always possible to align any agent with any user. If this process proves practical in non-trivial domains, the concept of alignment offers a path towards a new class of uniquely trustworthy agents.

This paper outlines a practical methodology for establishing user-agent value alignment and for employing it to drive agent development. In particular, we show how to create alignment by separately considering the structural and numerical aspects of the agent's reward function in relation to user utility. This process can suggest agent-held sensors, and clarify the agent's options for action. Next, we show how to iteratively improve agent designs. This leads to several heuristics for predicting the performance of both aligned and unaligned agents based on an analysis of their reward functions, without recourse to expensive, behavioral tests. Finally, we show how to maintain alignment during operation through user-agent communication, in part by altering the agent's reward function to track changes in user utility that accrue over extend periods of time.

While this methodology is untested, it offers a novel suite of tools for designing, implementing, evaluating, and interacting with autonomous agents that focus on a more abstract plane. We emphasize a discourse about values above specific behavior, under the tenet that if agents carry the user's concerns at heart, trust necessarily follows.

Designing the 2009 Mars Rover

In order to make the need for alignment clear, we describe a real application task; the design of the NASA Mars rover scheduled for launch in 2009. We focus on the difficulties of the current design process that an alignment-based methodology can hope to reduce. The following text is adapted from a joint proposal with Dr. Marcel Schoppers, a member of the mission flight team.

At its inception, the 2009 Mars Mission was intended to land in the smooth center of a crater, then drive to the crater rim to analyze rock samples. The plan called for autonomous drives as long as 1 km, while the rover would need to survive 2 years to cover the distance. This required Radio-isotope Thermal Generators (RTGs) to provide power during the Martian winter (as the sun is too low on the horizon for solar panels), and a rover about the size of a Volkswagen beetle to carry the RTGs. As the mission evolved, budget constraints downsized the distance requirements by a factor of 10, to at most 50 meters per day. At this scale, the rover can be driven entirely from Earth, eliminating the need for autonomous navigation. Moreover, the rover's large size means that the mean hazard-free path will be ~150 meters, so the mission can afford to react to rock-hazards by causing the rover to fail-safe (freeze in place). Even software for veering around rocks is considered a low-probability optional extra. The rover's remaining mission is to reach, collect, and analyze approximately 30 rock samples.

Before it was clear how much the mission would have to shrink, the flight software team began to adapt a prior hazard-avoidance package by testing it in the most obvious way: we placed a huge rock directly in front of the rover. The rover drove straight into the rock. This illuminated several problems: the software designers believed such a situation would never arise; the proximity of the rock reduced the light level beyond what the vision system could handle, and it took lack of evidence to mean no danger; and the terrain-evaluation criteria had malfunctioned so the rover would not have avoided the rock even if it had been seen.

In addressing these problems, we replaced the hazard-detection and hazard-avoidance algorithms. To give ourselves confidence in the coherence and repeatability of the rover's behavior, we resorted to an A*-like search across a graph that is dynamically elaborated by simulating a set of possible moves. In our case the moves were a set of rover arcs and turns, and the evaluation function (the A* path-cost) was cast as the amount of energy each move would save off the expected energy-cost from "here" to the goal. We multiply this number by a factor that roughly corresponds to the probability the rover can safely complete the move. No move can be selected unless it is better than terrain-average. We note that this ranking function looks like an expected utility (although this was not intentional). The net effect is that the rover either moves safely towards its goal, with some leeway to drive around hazards when necessary, or it gives up quickly. We are now slightly confident of our software's behavior.

We remain acutely aware of several unresolved issues. (1) If the terrain is too rough or too sloped, our software won't know it. (2) Our algorithm relies on some 30 constants, which we have assigned through a combination of intuition and experiment. For example, the rover's path-safety evaluation is limited to a radius of 1.5 meters, flimsily justified by camera resolution, because arbitrarily-long paths will always become unsafe, if only by virtue of uncertainty. (3) We can justify each part of the move evaluation separately (energy-saved, move-safety) but cannot defend the product, other than by the maxim, "it works". (4) We included no reward for simply seeing new terrain. Although this might result in a preferred path to the goal, we don't want randomly exploring rovers and don't know how to make the resulting tradeoffs.

The current Mars '09 rover mission plan eliminates each of these issues by prohibiting hazard avoidance software from driving the rover. However, even if the software is limited to hazard detection/assessment, uncomfortable issues remain. The system cannot reliably detect rock clefts that will trap the rover's wheels, or rock configurations that let the wheels down with the rover's belly suspended. So far, our attempts to address these issues have produced hazard detectors that declare their fear so often they cost a significant fraction of the mission's operating time. In addition, the tuning required to achieve enough-but-not-too-much hazard detection cannot be carried out with flight hardware, and it requires many time-consuming tests to shrink the sample's standard deviation. In other words, merely declaring terrain hazardous is risky and expensive. However, we know of no other way to proceed, besides retreating to very short-range, completely manual driving.

This description makes it clear that the 2009 rover mission experiences the problems that value alignment is meant to address. First, its users are motivated by scientific return, but routinely sacrifice it by down-scoping the mission. Their willingness to deploy autonomy is limited by its perceived risk. Second, the users care about abstractions like safety and mission return (and are willing to measure them) while the engineering team struggles to give the rover the relevant perceptual data. Thus, they are trying to define and communicate a desired objective function across a gap in reference frames. Third, the engineering staff is very concerned with system validation. They want to know that the rover will behave well on Mars, and to understand how and why it might fail as an input to incremental design. They also find behavioral metrics extremely hard to obtain. Value alignment provides an alternate perspective on each of these issues.

We will use the Mars rover mission as a source of examples throughout this paper, and as a vehicle for illustrating the relevance, and the steps of our proposed methodology. We begin by introducing alignment theory.

User-Agent Value Alignment

The theory of alignment creates a bridge between the objectives of a user and those of an autonomous system. It addresses this question in the context of a decision theoretic problem frame that represents the user's concerns by a utility function, and the agent's by an analogous function called its reward. The theory identifies the conditions necessary to align the agent's reward with human utility in such a way that the agent is motivated to address the user's concerns.

Figure 1 illustrates the alignment problem via an influence diagram (Howard & Matheson, 1984). Here, the ovals are observed attributes, the rectangle identifies decision options, the arcs represent an influence between two quantities, and the absence of an arc represents conditional independence. The task is to align the agent's reward, **R**, with user utility, **U**, such that the agent's decisions, **D** (informed by the observations, **o**) maximize user utility. The task is difficult for two reasons. First, **R** and **U** can be based on different feature sets (denoted **x** for **R**, and **y** for **U**), implying that the agent cannot sense the user's concerns, and cannot necessarily represent **U**. As a result, its decisions can diverge from the ones the user would have it employ. Second, since the agent's behavior can impact utility via multiple pathways over time, the agent can adversely (and inadvertently) affect the user's utility in the process of pursuing its own goals.

The example in Figure 1 represents a lack of alignment. Here, we assume that the rover faces a decision to stop rather than navigate around the obstacle, where that choice has consequences for the user. In particular, if it stops in a communication shadow, the mission will fail. Unless the rover detects this condition (or an analog for it) and employs it to rank its action options, it cannot address the user's concern. From the rover's perspective, Stop will be its optimal action, yet it will have severe (if unintended) consequences for the user.

In order to establish alignment, we need to ensure that the agent is motivated by the user's concerns, and that its actions will not impact the user in adverse ways. The theory addresses these issues in two parts. The first defines a conditional independence relationship between the agent and user problem frames called *structural alignment.* In its presence, the agent can recognize all ways its actions affect user utility. Next, given structural alignment, we perform a numerical manipulation of agent reward to ensure *functional alignment*; a situation in which the agent always selects an action its user would have preferred.

Figure 2 illustrates structural alignment. This diagram indicates that the agent's action can only affect user utility via features that matter to the agent. In order to create this condition, we have introduced a surrogate for the user's concern with safety into the rover's reward function. Here we assume that the rover needs to know its orientation in order to be able to locate, and then communicate with earth. Given this modification, the consequences of the rover's choice to Stop are visible in its knowledge of orientation, which in turn influences the user's assessment of safety, and thus mission success. Said differently, the user's utility is conditionally independent of the rover's decisions and observations, given knowledge of its orientation. Equivalently, the user's utility is *caused* by the rover's orientation, with respect to the agent's decisions (Heckerman & Shachter, 1995).

If all interaction between the agent and the user passes through the agent's reward, we can motivate the agent to address user concerns. In particular, we can choose the agent's reward function so that the policy that produces its highest expected reward stream also produces the highest possible expected utility for the user. This condition defines functional alignment. Intuitively, we can produce functional alignment by setting the agent's reward for a given feature set equal to the expected utility for the user's corresponding observations, as this will cause the agent to increase expected utility whenever it increases reward.

The key theorem of value alignment states that structural alignment holds if and only if functional alignment can be satisfied. As a result, we *must* have the conditional independence relationship of Figure 2 if the agent is to act optimally for the user. This makes structural alignment a key target of design, and the centerpiece of a value-oriented design methodology.

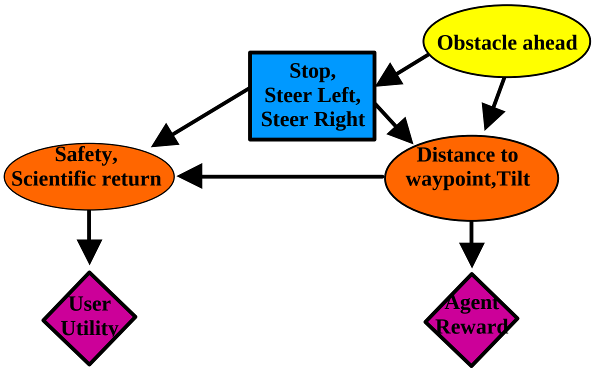

Figure 1. A joint user-agent problem frame.

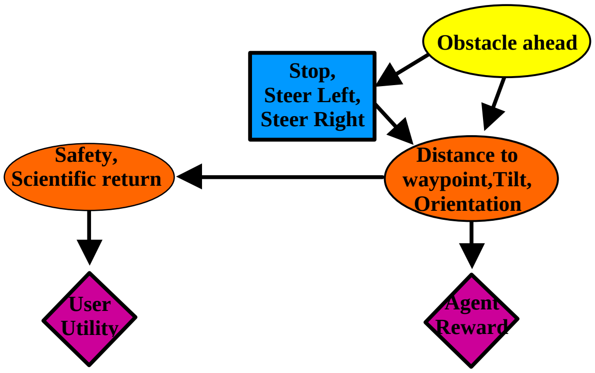

Figure 2. An example of structural value alignment.

Surprisingly, we can always establish structural, and therefore functional alignment between any agent and any user. While the process may be computationally complex (see below) it is important to notice that the relationship between the agent and the user can still be quite broad. The user can care about features outside the agent's ken (e.g., the geology of Mars rocks), and the agent can care about features that are irrelevant to the user. However, anything the user cares about that the agent can observe or affect must be visible in the agent's reward.

Using Alignment in Agent Design

We believe that user-agent value alignment can be employed as a principle to guide agent development. We elaborate on this theme by discussing each step of a proposed methodology, beginning with techniques for establishing alignment at design time, then iteratively improving an implementation, and finally maintaining alignment during agent operation.

Creating alignment

The most direct path for creating alignment follows the outline of the theory; we construct an agent-held reward function that establishes structural alignment, and then tune its free parameters to achieve functional alignment. We treat this process as an exercise in decision analysis, albeit it novel one. Here the goal is to refine user utility, expressed as a collection of preferences over a hierarchy of attributes that carry value (Keeney and Raiffa, 1976), until the terms obviously connect to the agent's problem frame. At the same time, we examine the existing agent design to see how it can inadvertently impact utility. This interplay drives changes to the agent's sensors, actions, and reward function, eventually producing a joint problem frame (expressed in terms of $\mathbf{x}$, $\mathbf{y}$, $\mathbf{o}$, $\mathbf{D}$, $\mathbf{R}$, and $\mathbf{U}$) that exhibits structural alignment. Given structural alignment, we employ several assessment techniques to produce functional alignment.

We have already illustrated an approach to structural alignment that reasons from agent design towards utility. This occurred in the transformation between Figures 1 and 2, when we recognized that the agent's decision to Stop could inadvertently sever communication; an element of $\mathbf{y}$ that carries indirect value to the user via its impact on safety. The fix was to introduce an orientation sensor into $\mathbf{x}$, so that the agent could be aware of, and motivated to address this concern. This line of reasoning can also suggest new agent actions. For example, we can avert the unintentional effect of Stop on communication by defining a recovery action that physically searches for earth.

We can also refine user utility until it suggests new agent sensors and actuators. For example, the user's interest in scientific return implies a utility for accurate locomotion so that the rover can efficiently reach interesting sites. This, in turn, suggests a role for landmark recognition software (a new virtual sensor), a means of steering relative to landmarks (a new action), and precision motors (new hardware) that will let the rover represent and address the user's desire. This line of reasoning is similar to the process of refining specifications.

A third method of creating structural alignment applies in general, and only requires minimal changes to the agent's design. Here, we simply include as many of the agent's actions and observations in its reward function as required. In the limit, this produces structural alignment through a degenerate path, since the agent's reward function will contain its entire problem frame (so the agent can only affect user utility through features it cares about as well.) Thus, we can always produce structural alignment between any agent and any user. However, this approach grows the agent's feature base and substantively complicates the task of producing functional alignment, as discussed below.

Note that we can detect the presence of structural alignment by asking subjective assessment questions. These all have a similar form: if you knew the agent was at a particular tilt, orientation, and location, would you care what it just did? If the user is indifferent to such new information, the agent can only affect $\mathbf{U}$ via $\mathbf{x}$, and structural alignment holds. Said more formally, these questions determine if $\mathbf{p(y|x) = p(y|x,D,o)}$ for all $\mathbf{D}$, and $\mathbf{o}$, satisfying the definition of structural alignment.

While the task of producing structural alignment involves an element of art, our approach to functional alignment has a more mechanical nature. It involves three steps:

1. model user utility, $\mathbf{U(y)}$,
2. assess the relation between user and agent problem frames, $\mathbf{p(y|x)}$, and
3. equate agent reward to expected utility, by setting $\mathbf{R(x) = EU(y|x)} = \Sigma_x\, \mathbf{U(y)\; p(y|x)}$

We have found that utility models have natural structure (Collopy 1999, 2001) driven primarily by microeconomic theory, which imposes constraints that insure transitivity and completeness in the ordering of alternatives in a specific domain. In the case of the rover, the user might place direct value on attributes for the rate of acquiring scientific data, mission risk, and power consumption, which can be decomposed into more detailed, measurable attributes (of indirect value). The supervening functional structure typically includes hyperbolic and logistic relationships, which satisfy constraints on the properties of utility functions (e.g., monotonicity, convexity, and the delta property in risk preference). Given this framework, we might have to assess ten parameters to fit a value model with a dozen attributes, verses the much larger number required by flat regression models. This simplification dramatically improves the practicality of the process, and generates structured models that tend to be transparent and easy to understand.

We suggest several methods for constructing **p(y|x)**. The first seeks deterministic relations, where some **y_i= g(x)**. For example, if the user and rover place value on the distance the rover will travel to reach the next waypoint, their estimates might have a fixed relation. Second, we can seek probabilistic relations with an analytic flavor, where **p(y_i) = g(x)**. For example, assume the user cares about mission risk, while the rover cares about tilt, measured as a derivate of altitude across grid cells. This derivative is plausibly connected to mission risk by a fixed, but probabilistic relation. Next, we can drive the attributes in **y** closer to **x** by further decomposing **U**, and thus simplifying the needed relation. Finally, we can resort to assessment techniques. For example, we can place the agent in a test domain such that it perceives the features of interest, **x**, and simultaneously ask the user to record his/her relevant perceptions, **y**. Repeated trials of this form will generate the desired distribution. In principle, we can automate this process by instrumenting a simulation to detect **y**, and recording data from multiple runs[1]. This task is simpler than communicating **y** to the agent, as it only concerns virtual sensors.

The final step of equating reward to expected utility is a problem in function fitting. Here, we are given **x**, **EU(y|x)**, and seek a functional form for **R(x)**, plus settings for its free parameters (generally interpreted as relative weights) that most closely approximate the target values. If we posit a convex **R**, many gradient descent techniques apply.

Iterative improvement

We can use the concept of alignment to iteratively improve agent designs. The theory currently provides two simple lemmas (Shapiro, 2002) that can be used to rank agents, where a more capable agent can make a superset of the same observations and decisions. They are: (1) a functionally aligned agent is weakly preferred to an equally capable unaligned agent, and (2) a functionally aligned agent is weakly preferred to a less capable, functionally aligned agent. The latter implies that you should always employ a more skilled individual who has your interests at heart. We believe we can extend the reach of these theorems to include more common cases. In particular, given two equally capable agents, we conjecture that the user should always prefer the one that (a) has a better hold on the truth (i.e., that has more reliable sensors), and (b) that possesses a more accurate perception of the user's expected utility (after accumulating sensor uncertainties into the reward function). This line of reasoning begins to transform the fundamentally structural theory of alignment into a more quantitative realm.

In addition to ranking lemmas, we believe we can create a more general capability for predicting (and evaluating) the performance of aligned and unaligned agents in numeric terms. This leads to several unabashed heuristics that predict utility without data from behavioral trajectories. Here we assume a probability distribution for **p(x)**, the states the agent will encounter in its domain, plus access to **p(y|x)** as before. Next, we consider three metrics:

1. Σ_x p(x) R(x)
2. Σ_x EU(y|x) p(x)
3. Σ_x p(x) (EU(y|x) – R(x))^2

The first calculates the expected reward for an agent's action in the domain, the second predicts the expected quality on a scale native to users, and the third calculates an error between user and agent perceptions of value. Each can be used to rank the behavior of alternate designs, and to identify regions of the attribute space where the agents most differ. This focuses attention in iterative design.

In actuality, of course, **R(x)** generates **p(x)** as the agent acts in its domain, so it is an act of will to assume a single distribution. However, these comparisons may make sense when contemplating sufficiently small design changes. In addition, we note that at least one software developer on the 2009 Mars rover project believes trajectory generation is prohibitively expensive, and he is quite willing to contemplate heuristics of this magnitude.

Maintaining alignment during operation

The concept of alignment can also play an active part in operational control. We have identified two main roles. The first follows from the realization that the user can supply some of the percepts required to establish alignment at run-time. For example, the rover is currently unable to detect some depth hazards (potholes) from binocular imagery. Rather than invest the time and energy to invent (and flight-certify) a special sensor, the user can simply tell the rover where such hazards exist, after viewing the same imagery. Although this data will be incomplete, the rover can rely on it where present, and otherwise employ a prior likelihood in its reward function. Assuming our conjectures regarding imperfect sensors hold, the rover will remain aligned in both cases and act to maximize human utility. It will just be more capable given better data.

We can also employ the concept of alignment to support extended operation. Here the goal is to accommodate changes in user interests by incrementally adjusting agent reward. This situation is not uncommon. For example, as the Pathfinder mission logged more successes, Sojourner's operators considered increasingly ambitious science goals at some risk to rover safety (e.g., by allowing higher levels of tilt and sensor noise). This concept was actually encoded in the operational plan as an allowable "risk level". In principle, we can model this shift in risk tolerance as an exogenous change to **U**, and derive an adjustment to **R** that preserves functional alignment.

[1] We can also record p(y|x,D,o) and detect structural alignment.

This results in a novel and principled way to control the rover during operation, which may be more flexible than current techniques. Instead of uploading a new action plan, we employ the rover's reward function as a control interface. As long as it remains aligned, the rover will make the best choices for us that it possibly can, even over prolonged periods of autonomous operation.

Discussion

The key idea in this paper is that a discourse over values can motivate agent design. As such, our goal has been to clarify a simple guiding principle for a complex enterprise: since agents make decisions for people, they should carry their user's values at heart.

The concept of value alignment to embodies this principle. However, its use raises a new issue, namely the need to bridge reference frames. This problem is endemic to all acts of delegation, but it is rarely addressed in explicit form. While principal-agent theory (Jensen and Meckling, 1976) employs monetary incentives to align the concerns of multiple (and generally non-cooperative) actors, the closest work with a methodological flavor comes from a very few authors in computer science. Wolpert, New, and Bell (1999) construct agent-held utility functions, but manipulate them as a vehicle for coordinating multiple agents, while Schoppers and Shapiro (1997) build an explicit, probabilistic relation between user and agent-held perceptions of state, and employ it to support design. They ascend the gradient of user utility with respect to decisions deep within their agent model.

Our work elaborates on these themes by employing value models to motivate many aspects of agent design. While our technical approach emphasizes the novel application of mostly mundane techniques (utility modeling, assessment and simulation), our methodology inherits one of the main benefits of decision analysis; the process itself increases clarity. We expect that users will learn a great deal about their own utility in the attempt to establish alignment.

We have shown that alignment is a highly desirable design objective, because it perfectly motivates agents to serve their users. However, we have deliberately ignored the question of *how* the agent optimizes **R**, and thus user utility, since our focus has been on values, not policies. Here, we note only that the benefits of alignment flow to *any* behavior generation technique that optimizes an objective, which is an extremely broad class of methods.

The key question, of course, is whether alignment is feasible in practice. As a form of constructive proof, we have offered a design methodology and suggested specific techniques to address each component problem. However, other solutions are also possible. For example, instead of working through **p(y|x)**, we can align **R(x)** directly with **U(y)** via a supervised learning method that employs numeric user feedback (or feedback about preferred action). Regardless of the approach, we note one caution: it will become more difficult to establish alignment as agent capabilities grow, because such systems can affect user utility in additional ways. That is, smarter agents are inherently harder to trust. .

In summary, we believe it is desirable and feasible to cast alignment as a design principle and place it at the core of a value-driven development methodology. We have shown that this approach can clarify early agent design, guide iterative improvement, and structure run-time interactions. More broadly, this strategy answers a yearning for an abstract appreciation of the agent design task that makes the specific implementation technology less central than the guiding intent: to supply value.

References

Collopy, Paul D. (1999). Joint Strike Fighter: Optimal Design through Contract Incentives. Pages 335-346 in 1999 Acquisition Reform Symposium Proceedings, Defense Systems Management College.

Collopy, Paul D. (2001). Economic-Based Distributed Optimal Design, AIAA 2001-4675. American Institute of Aeronautics and Astronautics, Reston, VA.

Heckerman, D., & Shachter, R. (1995). Decision-theoretic foundations for causal reasoning. Journal of Artificial Intelligence Research, 3, 405-430.

Howard, R., & Matheson, J. (1984). Readings on the principles and applications of decision analysis. Strategic Decisions Group, Menlo Park, CA.

Jensen, Michael and Meckling, William. "Theory of the Firm: Managerial Behavior, Agency Costs, and Ownership Structure". Pages 305-360 in The Journal of Financial Economics, Vol. 3, 1976.

Keeney, Ralph L., and Raiffa, Howard. Decisions with Multiple Objectives. John Wiley & Sons, New York, 1976.

Schoppers, M., & Shapiro, D. (1997). Designing embedded agents to optimize end-user objectives. Proceedings of the Fourth International Workshop on Agent Theories, Architectures and Languages. Providence, RI.

Shapiro, D, and Shachter, R. (2002). User-agent value alignment. Stanford Spring Symposium, Workshop on Safe Learning Agents.

Shapiro, D. (2001). Value-driven agents. PhD thesis, Department of Management Science and Engineering, Stanford University, Stanford, CA.

Wolpert, D., New, M., & Bell, A. (1999). Distorting reward functions to improve reinforcement learning. Tech. Report IC-99-71, NASA Ames Research Center, Mountain View, CA.