**Paper No. 499**

# An Agent Architecture for Long-Term Robustness

John R. Rose
University of South Carolina
Swearingen Engr. Center
Columbia, SC 29208 USA
+1-803-777-2405
rose@cse.sc.edu

Michael N. Huhns
University of South Carolina
Swearingen Engr. Center
Columbia, SC 29208 USA
+1-803-777-5921
huhns@sc.edu

Soumik Sinha Roy
University of South Carolina
Swearingen Engr. Center
Columbia, SC 29208 USA
+1-803-777-5622
soumik@sc.edu

William H. Turkett Jr.
University of South Carolina
Swearingen Engr. Center
Columbia, SC 29208 USA
+1-803-777-5622
turkett@cse.sc.edu

## ABSTRACT

This paper describes an architecture for enabling robust autonomous decision making and task execution. A key feature of the architecture is that agent behavior is constrained by sets of agent societal laws similar to Asimov's laws of robotics. In accordance with embedded philosophical principles, agents use decision theory in their negotiations to evaluate the expected utility of proposed actions and use of resources. This results in planning and task execution that is dynamic, rational, distributed, occurs at multiple levels of granularity, and can be trusted. We report on our initial investigations of agent architectures that embody philosophical and social layers. Our investigations have included the effect of misinformation among cooperative agents in worth-oriented domains, and active countermeasures for dealing with the misinformation. We examine the agents' use of philosophical principles for mission preeminence and rational progress towards goals.

## Categories and Subject Descriptors

I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence – *multiagent systems.*

## General Terms

Reliability.

## Keywords

Philosophy for agents, agent architectures, robust systems.

## 1. INTRODUCTION

The improvements in Internet-based software agents that are underway at many laboratories and corporations are fulfilling the promise of personalized, friendly Web services. The improvements come at a cost, however—greater implementation complexity. Thus, as we gradually rely more on the improved capabilities of these agents to assist us in networked activities such as e-commerce and information retrieval, we also understand less about how they operate.

Abstraction is the technique we use to deal with complexity. What is the proper kind and level of abstraction for complex software agents? We think it will be reasonable to endow agents with a philosophy. Then, by understanding their philosophies, we can use them more effectively.

For example, consider future NASA missions. As they become longer, more complicated, and farther away, the software systems controlling them will of necessity become larger, more intricate, and increasingly autonomous. Moreover, the missions must succeed in the face of uncertainties, errors, failures, and serendipitous opportunities. While small, well-specified systems with limited types of known external interactions can be proved correct, consistent, and deadlock-free via formal verification, such conditions do not hold for network-based systems. We will basically have to trust the systems, so there should be a principled basis for our trust.

Unfortunately, constructing large error-free software systems appears not to be achievable by current means. Additionally, the large size of the systems and the unknowns to which they will be subjected cause them to be untestable to even find out if, when, or where they might fail. A new paradigm and architecture for software development are thus needed [19], and we are investigating ones based on the premise that errors will always be present in software systems, and we should try to not only compensate for them, but also take advantage of them. Furthermore, an agent-based architecture, with the agents having explicit philosophies, is a promising foundation for engendering trust. We can trust the agents to act autonomously if they embrace ethical standards that we understand and with which we agree. We expect that this will lead to robustness, fault tolerance, recovery, graceful degradation, and, ultimately, trust in our systems.

## 2. PHILOSOPHICAL AGENTS

An agent-based approach is inherently distributed and autonomous, but when the communication channels that link the agents are bandwidth-constrained or noisy, the agents will have to make decisions locally, which we hope will be coherent globally, as well as worthy of trust. We can trust the agents to act locally (autonomously), if we understand and agree with their principles.

To endow agents with ethical principles, we as developers need an architecture that supports explicit goals, principles, and capabilities (such as how to negotiate), as well as laws and ways to sanction miscreants [15]. Figure 1 illustrates such an agent architecture that can support both trust and coherence, where

*AAMAS'02*, July 15-19, 2002, Bologna, Italy.

coherence is the absence of wasted effort and progress toward chosen goals [8].

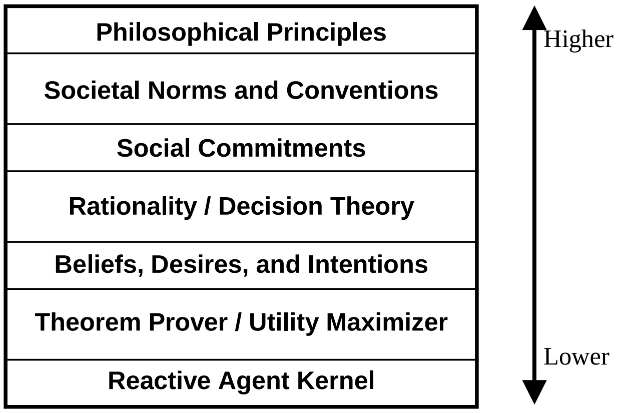
Philosophical Principles
Societal Norms and Conventions
Social Commitments
Rationality / Decision Theory
Beliefs, Desires, and Intentions
Theorem Prover / Utility Maximizer
Reactive Agent Kernel
Higher
Lower

**Figure 1: Architecture for a philosophical agent. The architecture defines layers of deliberation for enabling an agent to behave appropriately in society.**

The lowest level of the architecture enables an agent to react to immediate events [12]. The middle layers are concerned with an agent's interactions with others [5,6,7,13], while the highest level enables the agent to consider the long-term effects of its behavior on the rest of its society [11]. Agents are typically constructed starting at the bottom of this architecture, with increasingly more abstract reasoning abilities layered on top.

Awareness of other agents and of one's own role in a society, which are implicit at the social commitment level and above, can enable agents to behave coherently [9]. Tambe et al. [17] have shown how a team of agents flying helicopters will continue to function as a coherent team after their leader has crashed, because another agent will assume the leadership role. More precisely, the agents will adjust their individual intentions in order to fulfill the commitments made by the team.

If the agents have sufficient time, they can negotiate about or vote on which agent should become the new leader. When time is short or communication is not allowed, the agents can follow mutually understood social conventions, such as “the agent with the most seniority becomes the new leader.”

## 2.1 Ethical Abstractions

Ethics is concerned with codes and principles of moral behavior [4]. Most ethical theories distinguish between the concepts of *right* and *good*:

- **Right** is that which is right in itself
- **Good** is that which is good or valuable for someone or for some end.

Should software agents favor right or good? *Deontological theories* emphasize right before good. They oppose the idea that the ends can justify the means, and they place the locus of right and wrong in autonomous adherence to moral laws or duties. A proponent of these, the German philosopher Immanuel Kant (1724-1804), defined his “categorical imperative” as an absolute and universal moral law based entirely on reason. For right action by an agent, the categorical imperative would be: “Agents should act as they think all other agents should act.” For example, breaking a promise would not be right, because if all agents did it, the system they support would not function.

Kant's categorical imperative does not contain a way to resolve conflicts of duty. Also, an action is not wrong unless the agent explicitly intends for it to do wrong. For example, an agent on a NASA deep space probe who is responsible for managing communications with ground control would not be wrong to shut down the communications link for diagnostics, even if that might leave other agents on the probe unable to communicate. These theories can also legitimize inaction, even when inaction has predictable, but unintended effects.

In contrast, *teleological theories* choose good before right: something is right only if it maximizes the good; in this case, the ends can justify the means. In teleological theories, the correctness of actions is based on how the actions satisfy various goals, not the intrinsic rightness of the actions. Choices of actions can maximize either individual or societal good, where good may be pleasure, preference satisfaction, interest satisfaction, or aesthetic ideals.

What agents need to decide actions are not just universal principles (each can be stretched) and not just consequences, but also a regard for their promises and duties. Agents have prima facie duties to keep promises, help others, repay kindness, etc. [11]. In the context of a NASA mission, an agent could repay a kindness to another agent by offering, without being asked, to donate a resource such as excess battery power. While agents have such duties, there is no ranking among the duties, which are defeasible. For example, an agent on a NASA deep space probe might find it acceptable to monopolize a communication channel to ground control to the detriment of other agents because it values the success of its task without regard to the consequences for other agents.

## 2.2 Machine Ethics

Isaac Asimov proposed a moral philosophy for intelligent machines in a *Handbook of Robotics* [1] that defined three Laws of Robotics. These were subsequently augmented by the “Zeroth Law” [2]. An adaptation of these laws for a collection of agents sent on a NASA mission might be:

**Principle 1:** An agent shall not harm the mission through its actions or inactions.

**Principle 2:** Except where it conflicts with Principle 1, an agent shall not harm the participants in the mission.

**Principle 3:** Except where it conflicts with the previous principles, an agent shall not harm itself.

**Principle 4:** Except where it conflicts with the previous principles, an agent shall make rational progress toward mission goals.

**Principle 5:** Except where it conflicts with the previous principles, an agent shall follow established conventions.

**Principle 6:** Except where it conflicts with the previous principles, an agent shall make rational progress toward its own goals.

**Principle 7:** Except where it conflicts with the previous principles, an agent shall operate efficiently.

Distributed systems—which most Internet applications are—are susceptible to deadlocks and livelocks. However, if the system components obey these seven philosophical principles, then the susceptibilities would disappear, because deadlock and livelock would violate Principle 6.

## 2.3 Applying Ethics

A philosophical approach to distributed system design presupposes that the components, or agents, can

- enter into social commitments to collaborate with others,
- change their mind about their results, and
- negotiate with others.

However, the ethical theories above are theories of justification, not of deliberation. An agent can decide what basic "value system" to use under any approach.

The deontological theories are narrower and ignore practical considerations, but they are only meant as incomplete constraints—that is, the agent can choose any of the right actions to perform. The teleological theories are broader and include practical considerations, but they leave the agent fewer options for choosing the best available alternative. All of these ethical approaches are single-agent in orientation and encode other agents implicitly. An explicitly multiagent ethics would be an interesting topic for study.

# 3. Methodology

The goal of our research is to evaluate the utility of different combinations and precedence orderings of behavior-guiding principles. To make the most progress toward our goal, we chose to use an agent-development toolkit (ZEUS) to provide most of the low-level functionality we need. We also selected the FIPA ACL, because it is the closest to a standard that is available.

We developed an initial set of four agent architectures. All agent architectures use the same two algorithms for checking memory for previous mineral samples and controlling the actual movement of the agents. The decision of which mineral sample to move towards is defined separately for each agent.

**Checking Memory**

1. Check to see if there are mineral samples the agent remembers and has not picked up that are currently out of the viewing area
2. If there are mineral samples in memory,
    a. determine the closest mineral sample to the agent from memory
    b. make a move of one space towards that mineral sample
3. Else if there are no mineral samples in memory, then make a random move of one space along the same path it was on.

**Movement**

1. The agent moves one position either in a random direction if it has chosen to move randomly or in the direction of its chosen mineral sample
2. If the agent reaches the same position as the mineral sample it is searching for, it retrieves the mineral sample and then senses again.
3. If the agent reaches an empty spot, it senses again.
4. If the agent cannot move into a spot because there is another agent already there, the agent attempts to make a random move of one space along the same path it was on.

Our baseline agent, Agent 0, is purely self-interested and unaware of other agents. Conflicts and inefficiencies arise as agents of this type attempt to pick up the same samples.

A more capable agent, Agent 1, is aware of other agents and, by estimating their behavior, attempts to avoid conflicts. Agent 2 is cooperative, and communicates its true intentions to other agents, thereby reducing conflicts even further. Agent 3 is more cooperative, in that it communicates not only its intentions, but also opportunities by which other agents might benefit, thereby improving the overall societal performance towards a global mission. The next section describes our experiments with these agents in different scenarios.

# 4. Evaluation

## 4.1 Evaluation Considerations

We require a test scenario that will allow us to make clear comparisons between the performances of agent architectures with different combinations and precedence orderings of philosophical principles. There are several features that we considered in selecting a scenario:

1. The scenario must justify multiple simultaneous tasks.
2. The tasks must be uniform to simplify performance evaluation.
3. It should be possible to carry out the tasks without explicit cooperation.
    a. Communication between agents should not be required.
    b. Global knowledge of the task scenario should not be required.

Based on these features, we considered abstract tasks such as:

1. Exploring (rover-type exploration)
2. Inspecting (inspecting a space station for damage from space debris)
3. Gathering (collecting mineral specimens on a Mars)
4. Building (space station construction)
5. Delivering (transporting supplies to appropriate destinations)

We then considered these abstract tasks in the context of future NASA scenarios involving unmanned probes, such as sample collection on Mars, evaluation of asteroids, and exploration of the hypothesized liquid ocean beneath the icy crust of Europa. This analysis indicated a large overlap between abstract gathering and inspecting tasks and moderate overlap with exploring and delivering tasks.

Next, we considered a matrix of types of test cases. Essentially, the test matrix is an enumeration of goal types, i.e., independent vs. shared, and combinations of philosophical principles. The combinations are:

1. Independent agents and goals; various combinations of philosophical principles

2. Independent agents, shared goals; various combinations of philosophical principles
3. Flat agent confederations, shared goals; various combinations of philosophical principles
4. Hierarchically organized agents, shared goals; various combinations of philosophical principles

Metrics that we considered for evaluating performance include:

1. Measure of independent goals accomplished
2. Measure of shared goals accomplished
3. Time required for goal accomplishment
4. Communication cost
5. Resource usage
6. Number of collaborative actions pursued.

## 4.2 Agent Test Scenarios

We developed several test scenarios for our agent architectures based on a simulated mineral specimen collection task on an unspecified planet. The test area is a 60x45 rectangular area. The tests were run with *n*= {50, 100, 150, 200} mineral samples with varying degrees of clustering. The degree of clustering ranged from a random distribution of mineral samples at one end of the spectrum to a single cluster of all *n* samples (the mother lode) at the other end. We did not allow more than one mineral sample to occupy any given position.

Tests were run with *m*= {6, 12, 24} randomly distributed agents, each sharing the same architecture. We also varied the size of the agents' field of view, defined as a *v*-by-*v* rectangle. All agents share the same size field of view in a given test run. The value of *v* was varied in the range (*v*= {7, 9, 11}). Each simulation lasts for 100 time steps. At each step, an agent may take one action from its repertoire of capabilities. Each simulation was repeated 10 times with a different random distribution of agents and samples.

We collected a number of statistics such as the total number of samples collected, the number of samples collected per agent, and the number of cooperative actions taken per agent, as well as averages and standard deviations. An example of this is shown in Figure 2, which also shows that when the samples and agents are distributed randomly, very little opportunity for cooperation arises. Consequently, a myopic self-interested agent performs as well as a socially aware, cooperative agent.

The single metric that we used to summarize the results of any given test run was the percentage of mineral samples collected in 100 time steps. Two major variants of the tests are determined by whether or not there is an upper limit placed on the number of samples that an agent could collect (Figure 3 shows a case where agents are limited to 8 samples). All agent architectures share the common goal of gathering as many mineral specimens as possible.

A definition of what constitutes cooperation or competition depends on the overall *societal* goals that the group of agents believes they are trying to achieve [16]. For example, if the goal is to balance the load carried by each agent, then cooperation means an agent with a heavier load gives in to others and the standard deviation among loads is reduced. If the goal is to maximize the percentage of samples collected, then cooperation means an agent should retrieve its closest samples and announce its intentions to do so, in order to deter other agents from wasting time on the same quests. Greater cooperation can be arrived at by communicating to nearby agents the location and estimated size of clusters of samples. The benefit of this to the society of agents is shown in Figure 3.

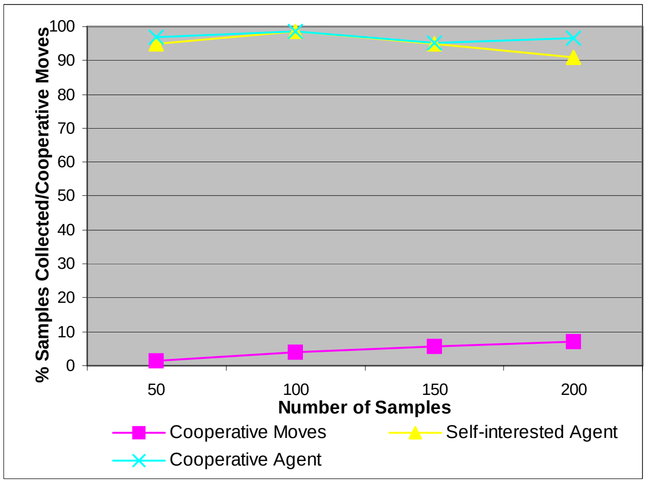

**Figure 2: There is little opportunity for cooperation, and therefore little benefit to cooperative search, when agents and samples are distributed randomly.**

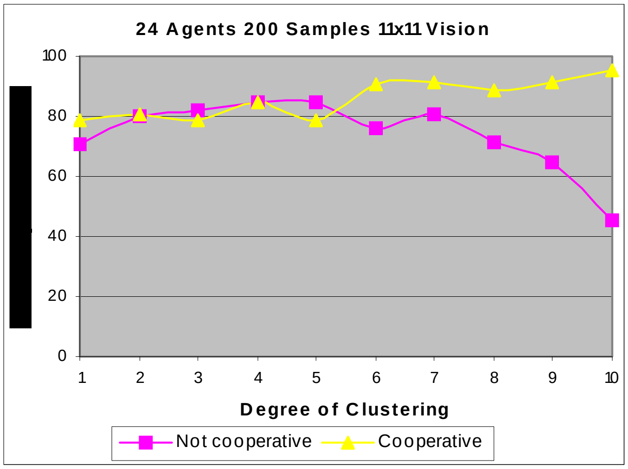

**Figure 3: Benefits of communicating the locations of clusters to other agents.**

We next investigated the same core scenario based on a simulated collection of mineral specimens on an unspecified planet. There are a total of 50 mineral samples and 24 agents. In the worth-oriented tests, agents have a sample carrying capacity of 8. In subsequent tests, agents have a sample carrying capacity of 2. The test area is a 60x45 rectangular area. Under these scenarios, the goal is to collect as many of the mineral samples as possible. Many of the tests are evaluated over different degrees of clustering of the 50 samples. There is one configuration for each of the ten degrees of sample clustering. The interpretation of the clustering values on the X-axis are 1=10 clusters, 2=9 clusters, 3=8 clusters, …, 10=1 cluster. Each simulation is run for 100 time steps.

## 4.3 Worth-Oriented Evaluation of Mission Success

On long-term missions, overall success will depend on the ability to conserve resources in order to meet long-term objectives. This has prompted us to examine an agent architecture that seeks to minimize the expenditure of resources in the mineral sample collection scenario that we have been using. This architecture, dubbed ‘EarlyFinisher’, is an extension of the cooperative agent architecture described above. The essential difference between our standard cooperative agent and an early finishing agent is that the early finishing agent terminates activity after collecting its limit (8) of mineral samples. In contrast, the standard cooperative agent continues to cooperate with other agents after having collected its limit of samples by communicating to other agents the existence of samples that it finds or relaying messages that it receives.

Figure 4 shows that the early finishing architecture consistently uses fewer moves per sample on average compared to the standard cooperative architecture and the noncooperative architecture. This figure shows that this efficiency holds over all degrees of data clustering, ranging from 10 clusters (1 on the X-axis) to the aggregation of the samples in a single large cluster (10 on the X-axis). The bumps in the curve at 3 and 8 on the X-axis are probably an artifact of the configuration of the sample clusters.

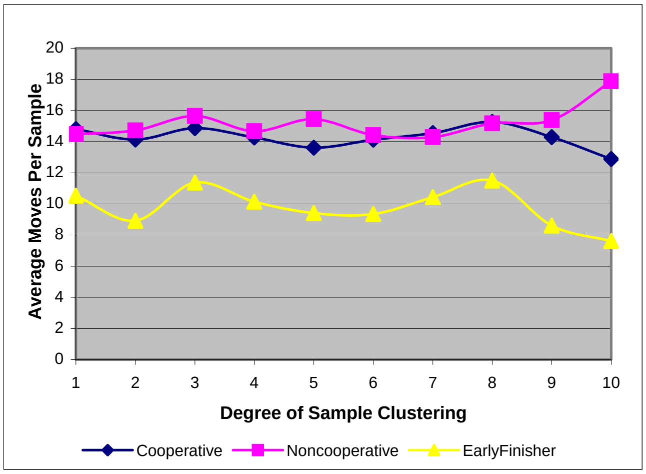

**Figure 4: The early finishing agent architecture consistently uses fewer moves per mineral sample in comparison with either cooperating or noncooperating agent architectures.**

In Figure 5 we see that the early finishing architecture is competitive with the cooperative and noncooperative architectures, only slightly under-performing overall. We note that when there is little clustering of samples, there is little opportunity for cooperation. In such situations there is very little difference in performance (1-8 on the X-axis of Figure 5). However, as the degree of clustering increases, cooperative behavior improves performance as can be seen for both the Cooperative and EarlyFinisher (8-10 on the X-axis of Figure 5).

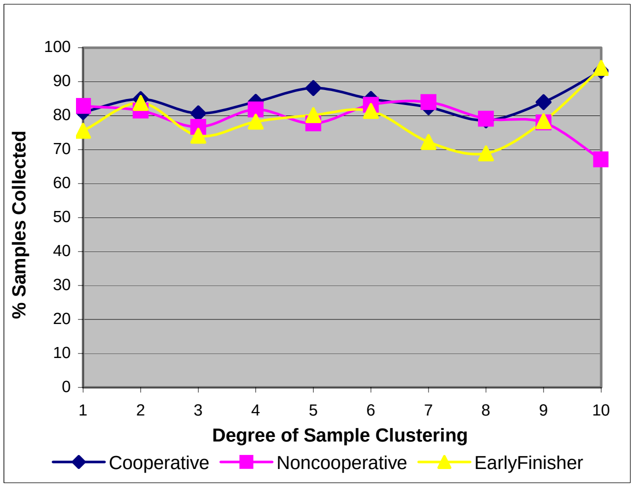

**Figure 5: The performance of the early finishing agent architecture is marginally poorer than the cooperative and noncooperative architectures.**

## 4.4 Misinformation from Agents

In order to achieve mission robustness, the agent architecture must be able to handle misinformation. In the context of cooperating agents, misinformation could well originate from malfunctioning agents and, in the less than halcyon world of the twenty-first century, malicious agents. In a set of experiments we examined the effects of misinformation when agents assume that all information they receive from other agents is correct. In this section agents have a mineral sample carrying capacity of 2.

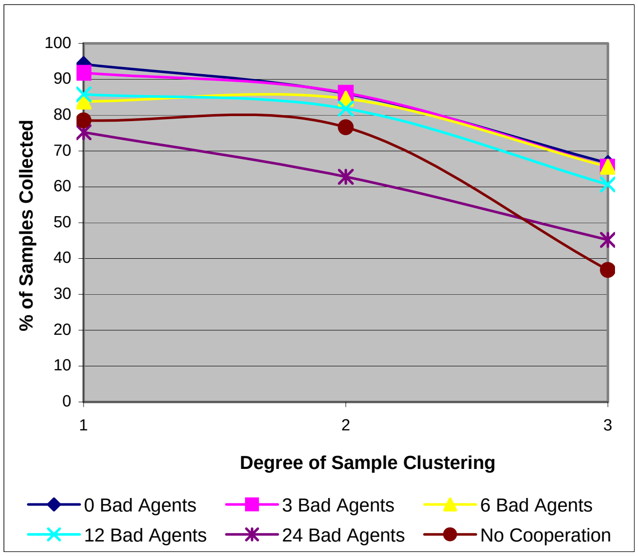

**Figure 6: As the number of malfunctioning/malicious agents increases there is a corresponding decrease in the number of mineral samples collected. In this figure, the “bad agents” provide misinformation only when they detect a cluster of mineral samples.**

In Figure 6 there are two general trends that can be seen. First, as the number of agents providing misinformation increases, the number of samples collected decreases uniformly, although the decrease is very small when there is only one cluster. The reason for this is that the agents that are providing misinformation only do so in this experiment when they observe a cluster of samples. When observing a cluster, such ‘bad agents’ communicate an erroneous position for it. If there is only one cluster, then there is very little opportunity for such agents to dispense misinformation. The second trend to be observed is that performance also decreases as a function of degree of clustering even as the number of ‘bad agents’ is held constant. This set of experiments was carried out with three clusters (X=1), two clusters (X=2) and one cluster (X=3). The two reference curves in this figure are the non-cooperative architecture and the cooperative architecture with no bad agents. Note that the non-cooperative curve drops dramatically from X=2 to X=3. This is a result of the fruitless search for mineral samples that the noncooperating agents are conducting over the entire search area while the samples are grouped in a single large cluster.

In a second set of experiments, the ‘bad agents’ were designed to provide misinformation both when they discover sample clusters and when no cluster is in sight. This increase in overall misinformation magnified the performance degradation observed in the previous set of experiments. However, the same general trends seen in Figure 6 are also seen in Figure 7.

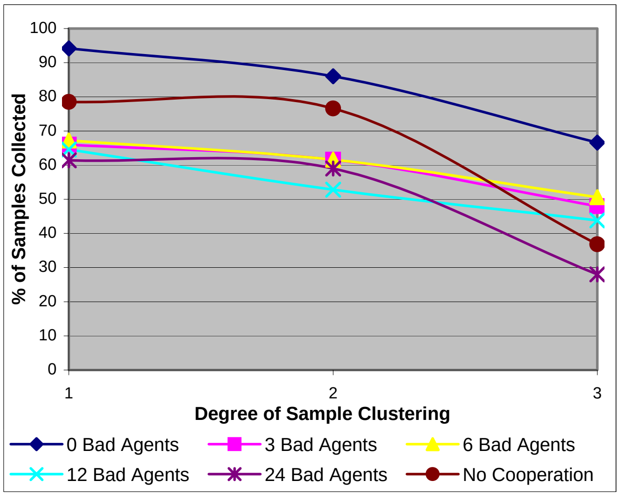

**Figure 7: As the number of malfunctioning/malicious agents increases there is a corresponding decrease in the number of mineral samples collected. In this figure, the “bad agents” consistently provide misinformation regardless of whether or not they perceive a cluster of mineral samples.**

## 4.5 Detecting Misinformation from Agents

The previous section describes the effect of misinformation in a cooperative agent environment where agents naively assume that all information they receive from other agents is correct. We considered three agent architectures for addressing misinformation. All three of these architectures keep track of information that they receive from other agents and which agent they received the information from, as well as the originator of the information if it has been relayed.

### *4.5.1 Gullible Agents*

The gullible agent architecture is an extension of the cooperative architecture in which agents assume that all agents provide correct information. When an agent determines that information it receives does not match its own direct observations, it classifies the agent from which it received the information as malicious and ignores future information provided by that agent.

### *4.5.2 Gullible-Original Agents*

This is a refinement of the gullible agent architecture. The difference is that while the gullible agent disbelieves all agents that it perceives to have proffered misinformation, the gullible-original agent only discredits the agents that it gets information directly from and not those that it received information by relay via other agents.

### *4.5.3 Skeptical Agents*

In contrast to the gullible and gullible-original architectures, the skeptical architecture disbelieves all agents until it is able to verify through observation that the information it receives is correct. In other respects it conforms to the cooperative agent architecture.

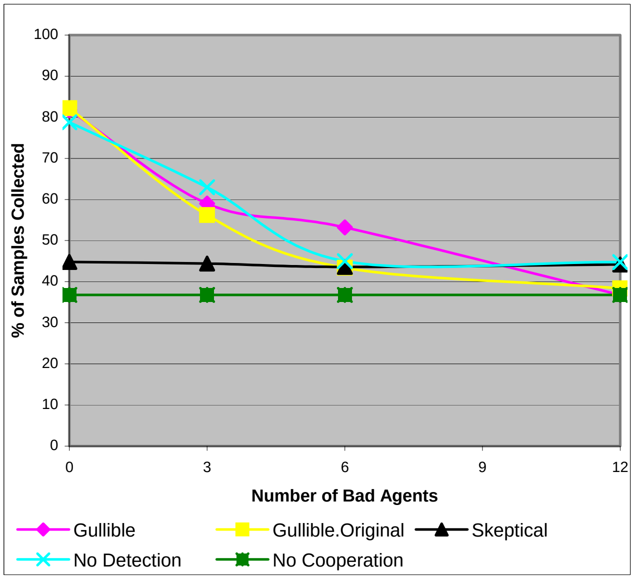

**Figure 8: Performance of misinformation-detecting architectures in tests involving a single cluster of samples.**

## 4.6 Passive Response to Misinformation

Once an agent determines that another agent is responsible for misinformation the passive response taken is simply to ignore the agent that is perceived to be malfunctioning or malicious. Figures 8 and 9 show the performance of these architectures in comparison to the standard cooperative architecture labeled as ‘No Detection’ and the noncooperative architecture. In the case of a single large cluster (Figure 8) where there is a clear advantage for cooperation, all cooperative architectures perform better than the noncooperative architecture, even when up to 50% of the agents are spewing misinformation. However, the skeptical approach is at a disadvantage when most of the agents are providing correct information. We hypothesize that since the simulations are run for only 100 time steps and there is only one cluster, skeptical agents are not able to overcome their skepticism due to insufficient opportunity to verify information by direct observation.

Simulations with three clusters provide greater opportunity for agents to confirm by direct observation the accuracy of the information that they receive from other agents. In Figure 9, all of the architectures for detecting and ignoring misinformation perform better than the standard cooperative architecture, which assumes that all information received from other agents is correct. In addition, these architectures outperform the noncooperative architecture.

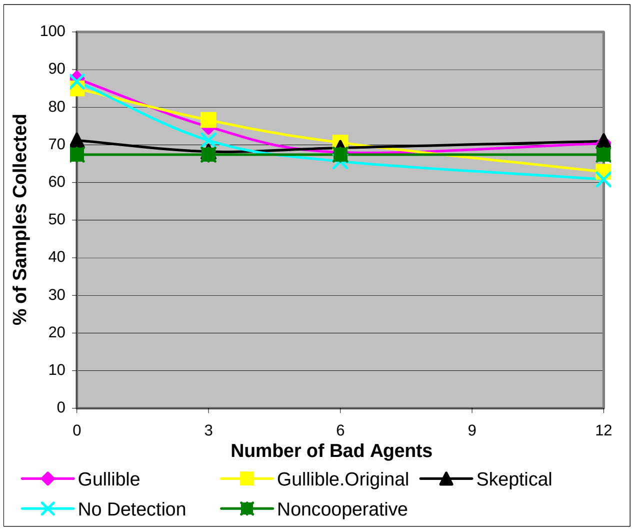

**Figure 9: Performance of misinformation-detecting architectures in scenarios involving three clusters of mineral samples.**

## 4.7 Active Response to Misinformation

Simply ignoring agents that are perceived to be malfunctioning or malicious on the basis of misinformation is quite often an inadequate response. The first principle that we propose in section 2.2 states that an agent should not harm the mission through its actions or inactions. Arguably, simply ignoring malfunctioning or malicious agents allows them to continue to harm the mission. Consequently, an agent that concludes that some other agent is malfunctioning or malicious must act to keep the bad agent from harming the mission. Balanced against this requirement is also the concern that the ability of a single agent to restrain the behavior of another agent must be limited. There is the potential for bad agents to cause chaos by acting to restrain good agents. The approach that we have investigated requires the agreement of ***n*** agents to restrain an agent. Agents concluding that some agent is bad report the "bad agent" to a coordinating agent. Once the coordinating agent has received bad conduct reports from ***n*** distinct agents, it "terminates" the bad agent. While we have not exhaustively analyzed conditions to characterize optimal values of ***n***, we have investigated values of ***n*** = 1, 2, and 3 and in our simulations the best results are obtained with ***n*** = 3.

In Figures 10 and 11, the results of actively restraining faulty agents are shown. In this set of experiments, the skeptical approach to recognizing bad agents results in the greatest reduction of mineral samples collected by bad agents (Figure 10). The uncollected samples are thus available for future collection by non-faulty agents. As can be seen in Figure 11, restraining faulty agents does not significantly degrade the performance of the gullible-original and skeptical agents. In fact, there is a small performance improvement for skeptical agents.

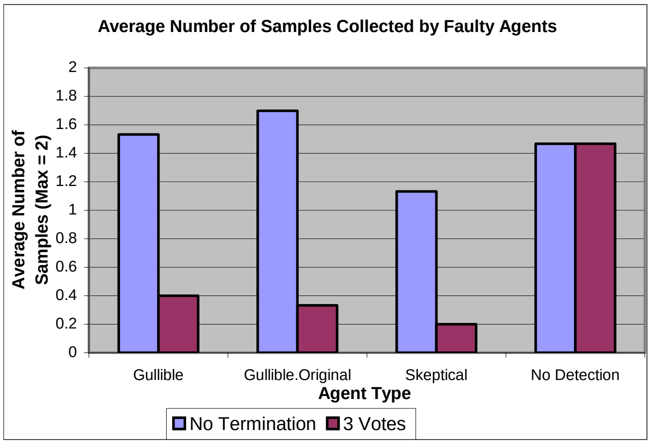

**Figure 10: The average number of samples collected by faulty agents is significantly reduced by an active response.**

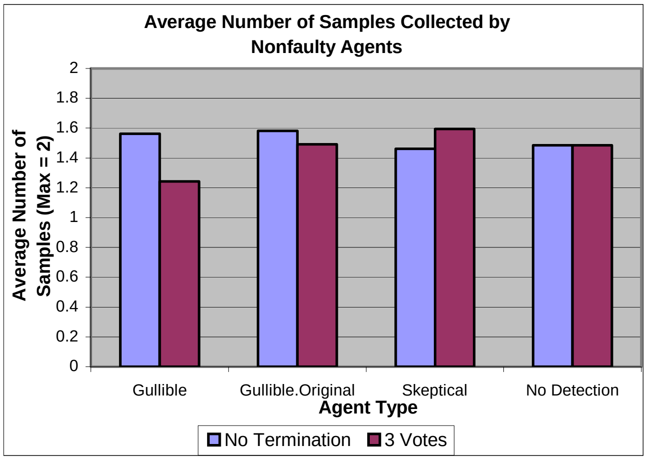

**Figure 11: Actively restraining of faulty agents does not significantly alter the average number of samples collected by the Gullible-original and Skeptical agents.**

# 5. CONCLUSIONS

The agents we construct—and the systems they implement, manage, and enact—must be trustworthy, ethical, parsimonious of resources, efficient, and—failing all else—rational. What we are investigating differs from current work in software agents in that:

- We are not researching new agent capabilities *per se*
- We are not developing an agent-based system for a new application domain
- We are investigating how agents can be the fundamental building blocks for the construction of general-purpose software systems, with the expected benefits of robustness and autonomy
- We are characterizing agents in terms of mental abstractions, and multiple agents in terms of their interactions. These abstractions matter because anticipated missions go beyond traditional metaphors and models in terms of their dynamism, openness, and trustworthiness.

The benefit of this architecture to complex missions such as future NASA planetary and deep space missions is fourfold: (1) it will support missions of much greater complexity than are possible under the current model of earth-based control, (2) it will reduce costs by minimizing the amount of earth-based support required for missions, (3) it will eliminate communication time lag as a significant factor in local task execution, providing the ability to react to and take advantage of serendipitous events, and (4) it will significantly enhance mission robustness. The development of the proposed architecture builds on developments in decision theory, agent societies, trusted systems, and ubiquitous computing.

## 6. ACKNOWLEDGMENTS

This work was supported by the National Science Foundation under grant no. IIS-0083362 and by NASA under grant no. NAS5-98051.

## 7. REFERENCES

[1] Asimov, I., *I, Robot*. Gnome Books, 1950.

[2] Asimov, I., *Foundation and Empire*. Gnome Books, 1952.

[3] Buhler, P.A. and Huhns, M.N., “Trust and Persistence,” *IEEE Internet Computing* vol. 5, no. 2, March/April 2001, pp. 90--92.

[4] Carnegie Mellon Center for the Advancement of Applied Ethics, http://www.lcl.cmu.edu/CAAE/index.htm.

[5] Castelfranchi, C., “Modelling Social Action for AI Agents,” *Artificial Intelligence*, vol. 103, 1998, pp. 157-182.

[6] Castelfranchi, C., Dignum, F., Jonker, C.M. and Treur, J., “Deliberate Normative Agents: Principles and Architecture,” *Proceedings of The Sixth International Workshop on Agent Theories, Architectures, and Languages (ATAL-99)*, Orlando, FL, July 1999.

[7] Cohen, P.R. and Levesque, H.J., “Persistence, Intention, and Commitment,” In: Cohen, P.R., Morgan, J., and Pollack, M.E. (eds.), *Intentions in Communication*. MIT Press, 1990.

[8] Durfee, E.H., Lesser, V.R., and Corkill, D.D., “Coherent cooperation among communicating problem solvers,” *IEEE Transactions on Computers*, vol. C-36, 1987, pp. 1275—1291.

[9] Gasser, L., “Social conceptions of knowledge and action: DAI foundations and open systems semantics,” *Artificial Intelligence*, vol. 47, 1991, pp. 107—138.

[10] Huhns, M.N. and Singh, M.P., “Cognitive Agents,” *IEEE Internet Computing*, vol. 2, November-December 1998, pp. 87—89.

[11] Mohamed, A.M. and Huhns, M.N., “Multiagent Benevolence as a Societal Norm,” In: Conte, R. and Dellarocas, C. (eds.), *Social Order in Multiagent Systems*, Kluwer Academic Publishers, Boston, MA, 2001.

[12] Muller, J.P., Pischel, M., and Thiel, M., “Modeling Reactive Behavior in Vertically Layered Agent Architectures,” in M.J. Wooldridge and N.R. Jennings (eds.), Intelligent Agents, LNAI 890, Springer-Verlag, Berlin, 1994, pp. 261—276.

[13] Rao, A.S. and Georgeff, M.P., “Modeling rational agents within a BDI-architecture,” In *Proceedings of the International Conference on Principles of Knowledge Representation and Reasoning*, 1991, pp. 473—484.

[14] Rose, J.R., Sengupta, A., Singh, S., and Valtorta, M., “Dynamic Decision Support for Command, Control, and Communication in the Context of Tactical Defense,” ONR Grant No. N00014-97-1-0806.

[15] Singh, M.P. and Huhns, M.N., “Social Abstractions for Information Agents,” in Klusch, M. (ed.) *Intelligent Information Agents*, Kluwer Academic Publishers, Boston, MA, 1999.

[16] Sycara, K. and Zeng, D., “Coordination of multiple intelligent software agents,” *International Journal of Cooperative Information Systems*, vol. 5, 1996, pp. 181—212.

[17] Tambe, M., D.V. Pynadath, D.V., and Chauvat, N., “Building Dynamic Agent Organizations in Cyberspace,” *IEEE Internet Computing*, vol. 4, March-April 2000, pp. 65—73.

[18] Vidal, J.M. and Durfee, E.H., “Building Agent Models in Economic Societies of Agents, in *AAAI-96 Workshop on Agent Modeling*, Portland, OR, July 1996.

[19] Wooldridge, M.J. and Jennings, N.R., “Software Engineering with Agents: Pitfalls and Pratfalls,” *IEEE Internet Computing*, May/June 1999.