An Evaluation of Philosophical Agent Architectures for Mission Robustness

John R. Rose, William H. Turkett, Michael N. Huhns, and Soumik Sinha Roy

Department of Computer Science and Engineering
University of South Carolina
Columbia, SC 29208 USA
{rose,turkett,huhns,rssinha0}@cse.sc.edu
http://www.engr.sc.edu/research/CIT

Abstract. This paper reports on initial investigations of an agent architecture that embodies philosophical and social layers. 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 enables more robust decision-making and task execution. To evaluate the robustness, our investigations have included the effect of misinformation among cooperative agents in worth-oriented domains, and active countermeasures for dealing with the misinformation. We demonstrate that propagating misinformation is against the principles of ethical agents. Moreover, such agents are obligated to report on misbehavior, which minimizes its effects and furthers the progress of the agents and their society towards their goals. We also show how dedicating some agents to specialized tasks can improve the performance of a society

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, of 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.

For example, consider future NASA missions. As they involve longer durations, more remote locations, and more complex goals, 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 concurrent network-based systems, and 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. We will have no choice but to trust these crucial but complex software systems, so there should be a principled basis for our trust.

Abstraction is one technique we use to deal with complexity. What is the proper kind and level of abstraction for dealing with complex agent-based software? We think it will be reasonable to endow agents with a philosophy, and then describe their expected behavior in terms of their philosophy. By understanding their philosophies, we can use and interact with the agents more effectively. 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 fault tolerance, graceful degradation, recovery, and, ultimately, trust in our systems. Also, an explicit philosophy might help the agents understand and anticipate each other's behavior.

2 Philosophical Agents

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 [16]. Figure 1 illustrates such an agent architecture that can support both trust and coherence, where coherence is the absence of wasted effort and progress toward chosen goals [8]. 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. For agents to interact effectively, they will have to communicate their own principles and model the principles of others.

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. [18] 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.*

The lowest level of the architecture in Fig. 1 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.

Fig. 1. An architecture for a philosophical agent. The architecture defines layers of deliberation for enabling an agent to behave appropriately in society.

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 supported 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, because the agent did not intend to disrupt communications. Deontological theories can also legitimize inaction, even when inaction has predictably bad effects, if the agent did not intend those effects. For example, an agent could morally justify not turning off an overheating component, if it did not intend for the component to overheat.

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 instead 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 overvalues the success of its own 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.

As a simple example of how such principles might apply, distributed systems, *which most Internet applications are,* are susceptible to deadlocks and livelocks. However, if the components of the distributed system 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 still has to decide what basic *value system* to use under any ethical theory it might adopt.

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 theories 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 agent communication language that is available. We then chose an exploration type of scenario, in which a group of agents move through a two-dimensional domain trying to find and retrieve mineral samples.

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,
   - determine the closest mineral sample to the agent from memory
   - 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 current path.

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 picks up 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 its current path.

Our baseline agent, termed *self-interested,* 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, termed *cooperative,* is aware of other agents and, by estimating their behavior, attempts to avoid conflicts. It communicates its true intentions to other agents, thereby reducing conflicts even further. It also communicates opportunities by which other agents might benefit, thereby improving the overall societal performance towards a global mission.

A *prevaricating* agent pretends to be cooperative and instead provides misleading information to other agents. By this behavior it hopes to make more resources available for itself, but possibly at the expense of the other agents in its society.

A specialized *scout* agent can travel faster and farther because it does not gather any mineral samples, so its purpose is to aid the other agents in its society by searching for and reporting on the locations of mineral samples. The next section describes our experiments with these agents in different scenarios.

4 Evaluation

4.1 Scenario 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.
   - Communication between agents should not be required.
   - 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 or shared, and combinations of philosophical principles. The combinations are:

1. Independent agents; independent 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 Initial 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 agent's 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 lasted for 100 time steps. At each step, each agent chose and executed one action from its repertoire of capabilities.

We collected statistics for 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. The results produced by these experiments are documented [14] and include:

- **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. In this study, we examined an agent architecture that seeks to minimize the expenditure of resources in a distributed gathering task (the mineral sample collection scenario described above). We used two types of cooperating agents: an early finishing agent that terminates its activity after collecting its limit (8) of mineral samples, and a continuing cooperative agent that 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.

- **Detecting Misinformation from Agents**. In order to achieve mission robustness, the agent architecture must be able to handle misinformation. We considered three different agent architectures for addressing misinformation. In all three of these, agents keep track of the information they receive from other agents and which agent they receive the information from, as well as the originator of the information if it has been relayed.
  1. 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.
  2. The gullible-original agent architecture 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 from which it gets information directly and not those from which it receives information indirectly by relay through other agents.
  3. The skeptical agent architecture, in contrast to the gullible and gullible-original architectures, 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.
- **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.
- **Active Response to Misinformation.** Agents concluding that some agent is malfunctioning or malicious report that agent to a coordinating agent. Once the coordinating agent has received bad conduct reports from n distinct agents, it *terminates* the malicious agent.

4.3 Agent Test Scenarios for a Large World

Our investigations next focused on the impact of role-division. To this end we developed a scouting-agent architecture. The purpose of a scouting agent is to scout for mineral samples and pass this information on to collection agents, which then collect the samples. The collection agents are the same cooperative agents described above. The scouting agent architecture differs from the cooperative agent architecture in two principle areas. First, its mission only involves scouting for mineral samples and communicating its finding to other agents. It does not collect any samples. Second, it moves twice as fast as collection agents and is able to communicate its findings over a much greater distance.

In order to evaluate a society comprised of scouting and collection agents and make comparisons with self-interested agent societies and homogeneous cooperative agent societies, we had to employ a larger simulation world than that

described in the previous section. The test scenarios for our agent architectures are based on a simulated mineral specimen collection task on an unspecified planet. The test area is a 180x135 rectangular area. This was constructed by creating a 3x3 tiling of the 60x45 rectangular area world described in the previous section. The tests were run with n= 1800 total mineral samples. The samples were grouped in randomly placed clusters of size s=1800, 450, 200, 100, 50. We did not allow more than one mineral sample to occupy any given position.

All tests were run with m= 200 agents. The 200 agents were randomly distributed as 10 groups of 20 agents in the 9 tiles (recall the world is a 3x3 tiling of rectangles of size 60x45). An agent's field of view is defined as a v-by-v rectangle. All agents collecting mineral samples share the same size field of view, 11x11. Scout agents have a larger field of view of size 41x41. Scout agents are also able to communicate their findings to agents within their 41x41 field of view. Each simulation lasts for 300 time steps. At each step, an agent may take one action from its repertoire of capabilities.

The simulations involved a comparison of three types of agent societies: homogeneous self-interested agents, homogenous cooperative agents, and a mixture of scouting agents and cooperative agents. Each of the following figures shows the performance of the three types of societies under the same conditions. The parameter that is varied is the number of clusters in the world. In all cases, the sum of mineral samples in the clusters total 1800. Each of the simulations was repeated 5 times with different randomly chosen starting positions for the groups of agents. The same set of starting positions was used for each society simulation. Thus, in each figure, each data point is an average over 5 runs.

Figure 2 is a summary of the entire set of tests. It shows the percentage of mineral samples collected by each of the three architectures after the first 100 times steps. The parameter that is varied in this figure is the number of clusters in the environment, ranging from one cluster containing all 1800 mineral samples to 36 clusters each containing 50 mineral samples. The clusters are randomly distributed.

Of interest in this figure is the relative change in performance between the societies as the number of clusters increases. Starting with a single cluster, the society employing scout agents performs better than the homogeneous cooperative and homogeneous self-interested agent societies, this in spite of the fact that 20 of the 200 agents are scout agents and thus unable to collect mineral samples. As the number of clusters increases, the difference in performance of the three societies decreases.

This result suggests that the contribution of the scout agent with its enhanced mobility and field of vision more than makes up for its inability to collect mineral samples when considerable search is required, as is the case when there are few large clusters. However, as the number of randomly distributed clusters increases, the amount of search required decreases and the performance of the homogenous cooperative agent society gradually matches and eventually surpasses that of the heterogeneous society. In this setting the scout agent is less of an asset. If the

simulations are run long enough, the scout becomes a liability since it does not itself collect mineral samples.

Another observation that can be made is that the greater the number and distribution of clusters, the better the performance of the self-interested set of agents. This is not surprising, since self-interested agents are not penalized as heavily by their lack of cooperation when the mineral samples are distributed more uniformly and are thus easier to chance upon.

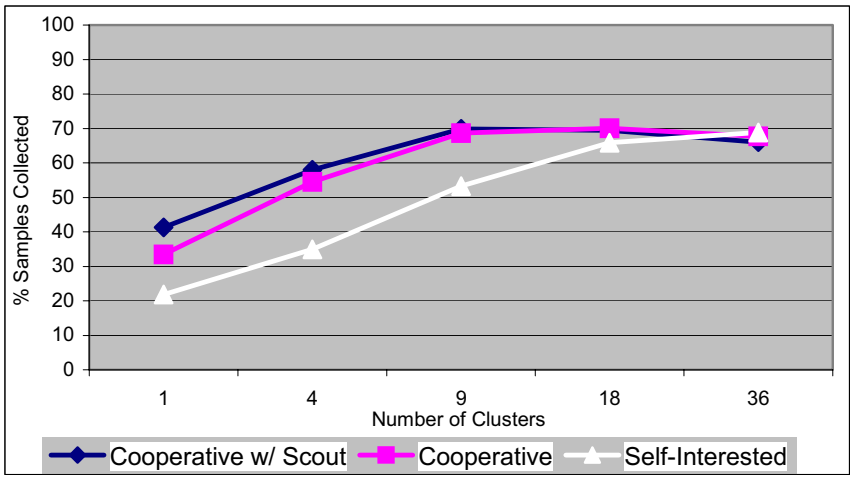

Fig. 2. Comparison of three architectures after 100 time steps according to number of sample clusters in a large-world environment.

In Fig. 3, the evolution of sample collection for each society is displayed over a period of 300 time steps. In this figure, all simulations occur in the same large-world environment in which there is a single large cluster containing all 1800 mineral samples. The data points at time step 100 are in fact the same data points shown for 1 cluster in Fig. 2. Figure 3 shows in detail how the scouting society develops an initial performance advantage and how this advantage with respect to the homogeneous cooperative society is gradually lost over time as information is propagated between agents in the homogeneous cooperative society. The initial advantage that the scouting society enjoys is due to the rapidity with which the scouts are able to investigate their world. While agents in the homogeneous cooperative society are not able to search as rapidly or directly communicate over as great a distance as scout agents, their steady cooperation allows them to reach the performance level of the scouting society at the end of 300 time steps. This figure also shows that cooperative behavior has a decided advantage compared to a self-interested approach when considerable search is required to locate a single large cluster.

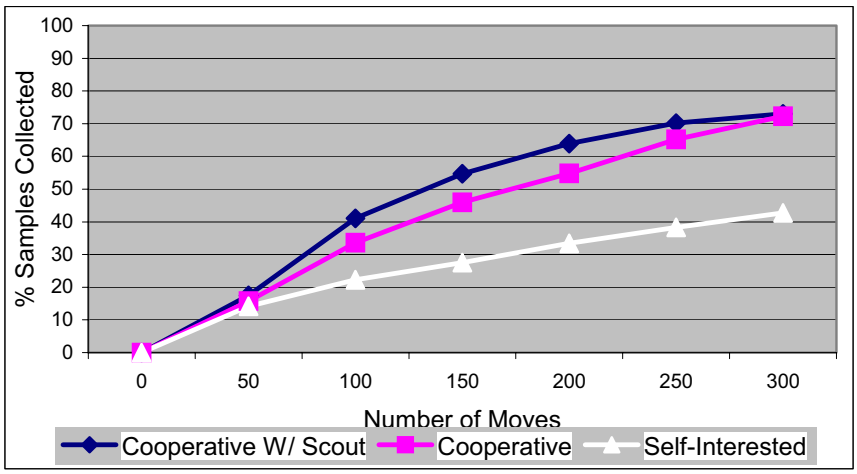

Fig. 3. Comparison of three agent architectures in a large-world environment containing a single large cluster of samples.

As the number of clusters is increased, the benefit provided by the scout agent's ability to rapidly investigate the world is diminished. Figure 4 shows the performance of the three agent societies in a large world environment where there are four randomly positioned clusters, each containing 450 mineral samples. Under these conditions, the scout agent society no longer has a performance edge over the homogeneous cooperative agent society. However, enough searching is required to find the clusters so that a cooperative approach significantly outperforms the non-cooperative approach of the self-interested agents. Another interesting observation is that for both cases of cooperative societies, the collection of mineral samples tapers off after the first 200 time steps. During the last 100 times steps of the simulation, very few additional samples are collected. We hypothesize that the agents have unevenly distributed themselves around the clusters, so that those agents that have not reached their carrying capacity are too far away from other clusters and most likely out of communication range to be able to locate any uncollected samples before the end of the simulation is reached.

Increasing the number of clusters even more reduces the benefit provided by cooperation. Figure 5 shows the performance of the three agent societies in a large world environment where there are nine randomly positioned clusters, each containing 200 mineral samples. Under these conditions, the scout agent society no longer has a performance edge over the homogeneous cooperative agent society. In fact, there are only 180 agents collecting samples in the scout society. As can be seen in this figure, after 100 times steps the homogeneous cooperative agent society with 200 agents collecting samples performs better.

More striking is the performance of the non-cooperating self-interested agent society. Its performance is now approaching that of the two cooperating agent

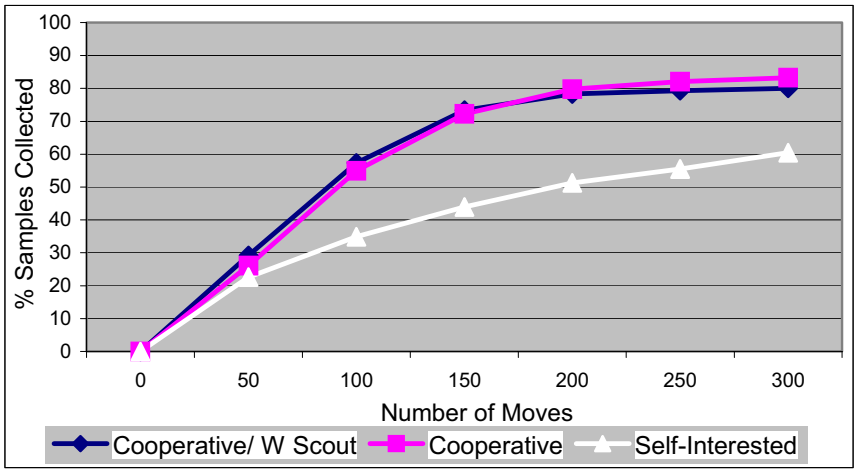

Fig. 4. Comparison of three agent architectures in a large-world environment containing four clusters.

societies. In simulations with even larger numbers of smaller clusters there is virtually no difference between the performance of the self-interested agent society and the homogeneous cooperative agent society. In contrast, the performance of the scout agent society is slightly lower, because it contains 20 fewer collection 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 applications go beyond traditional metaphors and models in terms of their dynamism, openness, and autonomy.

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

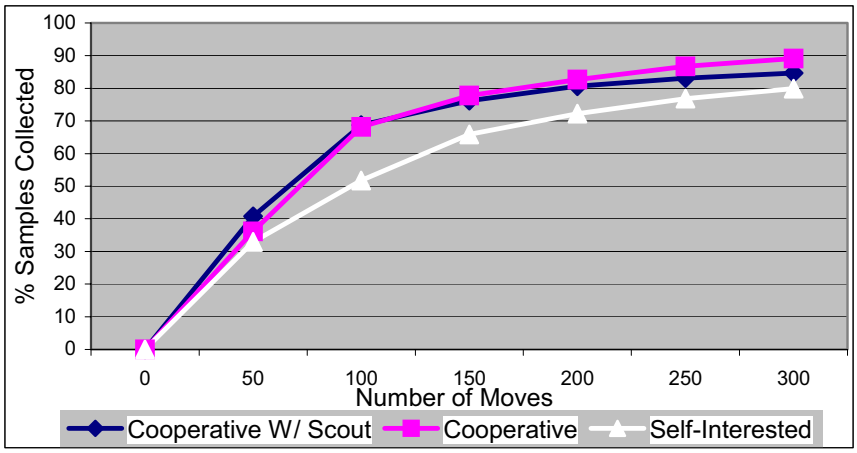

Fig. 5. Comparison of three agent architectures in a large-world environment containing nine clusters.

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.

Acknowledgements

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

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 **5** (March/April 2001) 90–92
4. Carnegie Mellon Center for the Advancement of Applied Ethics, http://www.lcl.cmu.edu/CAAE/index.htm
5. Castelfranchi, C.: Modeling Social Action for AI Agents. Artificial Intelligence **103** (1998) 157–182
6. Castelfranchi, C., Dignum, F., Jonker, C.M., and Treur, J: Deliberate Normative Agents: Principles and Architecture. In: 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 **C-36** (1987) 1275–1291
9. Gasser, L.: Social conceptions of knowledge and action: DAI foundations and open systems semantics. Artificial Intelligence **47** (1991) 107–138
10. Huhns, M.N. and Singh, M.P.: Cognitive Agents. IEEE Internet Computing **2** (November-December 1998) 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) 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) 473–484
14. Rose, J.R., Huhns, M.N., Sinha Roy, S., and Turkett Jr., W.H.: An Agent Architecture for Long-Term Robustness. In: Proceedings of the First Joint International Conference on Autonomous Agents and Multi-Agent Systems. ACM Press (2002)
15. 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
16. 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)
17. Sycara, K. and Zeng, D.: Coordination of multiple intelligent software agents. International Journal of Cooperative Information Systems **5** (1996) 181–212
18. Tambe, M., Pynadath, D.V., and Chauvat, N.: Building Dynamic Agent Organizations in Cyberspace. IEEE Internet Computing **4** (March-April 2000) 65–73
19. 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)
20. Wooldridge, M.J. and Jennings, N.R.: Software Engineering with Agents: Pitfalls and Pratfalls. IEEE Internet Computing (May/June 1999)