COOPERATIVE TEAM FORMATION AMONG AGENTS WITH SOCIAL-RELATION AWARENESS

Xiaoqin Zhang and Aliasgar Dahodwala
University of Massachusetts at Dartmouth
North Dartmouth, MA, USA

Ping Xuan
Clark Univeristy
Worcestor, MA, USA

ABSTRACT

In this paper, we bring social relation awareness into agent's decision-making process. An agent decides whether to offer help or not to another agent based on both utility gain and also social attachment consideration toward the other agent. We experiment with heterogeneous agents that use different combinations of utility and social relation to make decisions. We study the development of cooperative teams among agents as they help each other to accomplish tasks. We also evaluate the effect of social relationship development on individual agent's and the system's overall performance.

KEYWORDS

Cooperation, social interactions.

1. INTRODUCTION

Agent-based systems provide a useful framework for developing real world application like electronic commerce, recommender systems and personal assistants. Agents deployed in these applications often interact in an open environment with other agents or humans (D, 1989; Huhns and Singh., 1997). The interactions involve cooperation, collaboration, or competition for resources to achieve the specified. When the complexity of agent interactions increases, the behavioral characteristics of agents acting in a group should be studied, and suitable interaction strategies should be developed in order to improve or optimize system performance. We assume that typical real-world environments abound in cooperation possibilities: situations where one agent can help another agent by sharing work. The development of cooperative relationships leading to exchanges of help can improve both local and system-level performances.

In this work we study the usage of social relationship in agent's decision-making process and it's impact on forming cooperative teams among agents. Some researchers such as Sandip Sen (Sen, 1996) have developed a reciprocity mechanism that encourages cooperation among self-interested agents. Building upon the reciprocity mechanism, we introduce social relation measurement (liking/disliking) in agent's decision-making mechanism. The social relation measurements are developed between agents based on their previous interactions, which can be used to help the agent's decision making in its future interaction with other agents. Additionally, the expressed social response from the other agent can be used as a feedback to evaluate its previous interaction with the agent. Also, it can be used to predict the other agent's future responses toward itself and hence help it to choose with whom to cooperate. We have designed suitable interaction strategies that take advantage of cooperation possibilities in the environment. Additionally, we have observed that reciprocal cooperation among agents for different tasks generated stable, mutually beneficial agent groups.

The rest of this paper is organized as the following. Section 2 lists some related work. Section 3 describes the setup used to perform all our experiments. Section 4 describes a set of experiments using this setup, where we study the implications of using social relation consideration with utility reasoning. Lastly, Section 5 presents our conclusion.

2. RELATED WORK

A fair number of approaches have been developed to use reasoning mechanisms that consider exchange of help or social reputations when deciding how to interact with the other agents. The work in (Castelfranchi et al., 1998) uses normative reputation to enhance the performance of agents that comply with social norms. In the SPIRE framework (Glass and Grosz, 2000), performance in a group is improved when agents reason about the effect of withdrawing from social commitments to their reputation. There has also been some research done in creating an analytical model for predicting the mix of different strategy distributions under an evolutionary scheme (Saha, 2005). It predicts both the population dynamics and the evolutionarily dominant strategy given the environmental conditions. In contrast, our work uses a combination of social relation measurement and utility to decide whether to offer help for another agent.

In the majority of multi-agent systems, agents have related but also distinct problem-solving expertise, so they have to coordinate when solving problems. Some research in the area of coalition formation in agent societies has focused on cooperative agents (Shehory and Kraus, 1998), where the agents group together to perform an allocated task. In our approach the agents only cooperate in order to complete a task assigned to them if they find they are unable to do so themselves. A related work on coalition in agent societies takes self-interested agents into account (Lerman and Shehory, 2000) when forming trust relationships in order to maximize their final utility. Learning of cooperative behavior has also been addressed in (Denzinger and Kordt, 2000), where an off-line learning module is used to generate situation-action pairs. Our work uses online classification and agents do not have to store a priori models of the other agents.

The idea of using other agents' options to build a reputation is not new. The work described in (Schillo et al., 2000) and (Yu and Singh, 2000) are good examples of this. In both cases they use a trust-net for weighing the other agents' opinions. Our structure to evaluate the helping decision can be considered a trust-net. In (Saha et al., 2003), a decision mechanism is presented that compares current helping cost with expected future savings from interaction with the agent requesting help. The experiment uses heterogeneous agents that have varying expertise for different job types. In our work, however, all the agents are equally capable of doing the job. When making helping decisions, they also consider social relation with the requesting agents based on the previous experiences.

The work presented in (Sen, 1996) uses a probabilistic reciprocity mechanism as opposed to a deterministic reciprocity mechanism to generate stable and cooperative behavior among a group of self-interested agents. In our work, we base agent reciprocity on the utility value of the job and the social relations developed over the past interactions. In (Sen, 2002), the effects of believing other agent's opinions when deciding to help an agent is studied. An experience based trusting mechanism is introduced for reciprocative agents that are able to withstand individual and group level exploits.

3. ENVIRONMENT SETUP

In this work, we set up a package delivery scenario to study the social relationship among agents using MASS simulator (Horling et al., 2000). There is a special agent (Agent X) that generates the package requests and awards a reward to the delivering agent for successful completion of the task. The delivery job consists of picking up the package from the indicated location and delivering it to the destination location. These locations are marked on a square grid of locations identified by their X & Y grid values. This two dimensional grid of cells is called a map. The agents can move over adjacent cells to get from one cell location to another. The map is implemented as a 2D array with cell values ranging from 1-4. The cell value represents the number of simulated time pulses needed for an agent to move to the next cell from the current cell. A value of 1 indicates a normal cell, and higher values such 2, 3 & 4 indicate cells having traffic like congestion or hurdle. An example of a 10x10 map is shown in Figure 1. In order to simulate the dynamic nature of traffic in reality, the grid cell values are updated after an interval of every 50-100 simulated time pulses. All agents receives the map update information.

We have created Agent X and multiple regular agent instances using JAF(Vincent et al., 2001) . Each package to be delivered has a source location on the map, a destination location and reward money associated with it. A message with the package details is sent to all the regular agents in the system. Each of the regular agents then send back their responses to Agent X indicating their availability and an estimate of time they will take to complete the task. Based on the responses from all the agents, Agent X then chooses the best agent for the current delivery job. Our work is focused in studying and analyzing the different variations of those regular agents.

1	1	1	1	1	1	1	3	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	3	1	1	1
1	1	4	1	1	1	1	1	1	1	1	1	1	1	2
1	1	1	1	1	1	1	4	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	2	1	1	1
1	1	1	3	1	1	1	1	1	1	1	1	1	1	1
1	3	1	1	3	1	1	1	1	1	1	1	1	1	1
1	1	2	1	1	1	1	2	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	2	1	1
2	1	1	2	1	1	1	1	1	1	1	1	1	3	1
3	1	1	1	3	1	1	1	1	1	3	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	4	1	1	1	1	3	1	1	1

Figure 1. A sample map grid

3.1 Agent Needs

Regular agents have three levels of needs:

• Physical: this refers to the energy required for the agent to survive and perform its work. Energy is consumed when the agent moves over the map. To accomplish delivery tasks the agent needs to have enough energy. The agent pays to recharge itself by going to the charging dock at the center of the map. When the energy falls below a threshold value, the agent prioritizes its goal to recharge itself.

• Monetary: the agent receives money for delivering packages. The agent constantly tries to make money by delivering packets. If the agent fails to deliver an assigned task it pays a percentage of the reward money as penalty. So an agent always tries to pick up only those tasks that it thinks it can accomplish based on its current location and energy. The agent also needs the money to be able to pay for recharging.

• Social: the agent needs to interact with other agents whenever it needs help. Usually the agent only accepts tasks that it thinks it can accomplish with the level of energy it has. However due to changing traffic conditions over time, sometimes the agent might not have enough energy to complete the current task. Under such circumstances it has to pay a penalty if it drops the package. In order to avoid such penalty, it tries to find another agent to take over this delivery task.

3.2 Agent Tasks

To accomplish each of the below mentioned tasks, the agents needs to perform a series of subtasks. These tasks have been described using TAEMS structure (Decker, 1996) as shown in Figure 2.

Recharge Task - Once the agent decides that it needs to recharge, it first gets the location of the charging dock. Next it calculates a path to get to the charging dock. Next it moves towards it and once it reaches the dock, it pays to refill its energy. The period of subtask *Goto Location* is dynamic depending on the agent's initial location, whereas the others are fixed. The period of the subtask *Recharge* is fixed as 10 time pulses.

Packet Delivery Task - First the agent gets the pickup location from the delivery job description. Then it decides on a path from its current location to the source and follows that path. Next it picks up the packet and generates a path to the destination. Following that path it delivers the packet to the final destination. In return the agent gets money for the job. Due to the changing map traffic condition, if at any point the agent realizes that it would not be able to deliver the packet on time due to insufficient resources, it calls upon another agent for help. The agent chooses one agent to call for help according to the ordering in its preference list, until a positive response is obtained or the entire list is exhausted, which causes the delivery task to fail. The agent's preference list includes all other agents currently available in the environment. At the start of the experiment, each agent begins with a random ordering of agents in the preference list. As the agent interacts with other agents and develops relations with them, the list is then sorted as per *liking/disliking* value with the agent having the highest liking value placed at the top of the list. This ordering simulates human tendency to

ISBN: 978-972-8939-23-6 © 2010 IADIS

first ask help from someone who has helped them in the past. Each of the subtasks *GetPickupLocation, DecidePathtoPickup, PickUpPacket, GetDeliveryLocation, DecidePathtoDelivery, DropPacket* takes around two time pulses. Whereas, the sub-tasks *GotoPickupLocation* and *GotoDeliveryLocation* are dynamic, their times depend on the current path length and are updated whenever the path is recalculated.

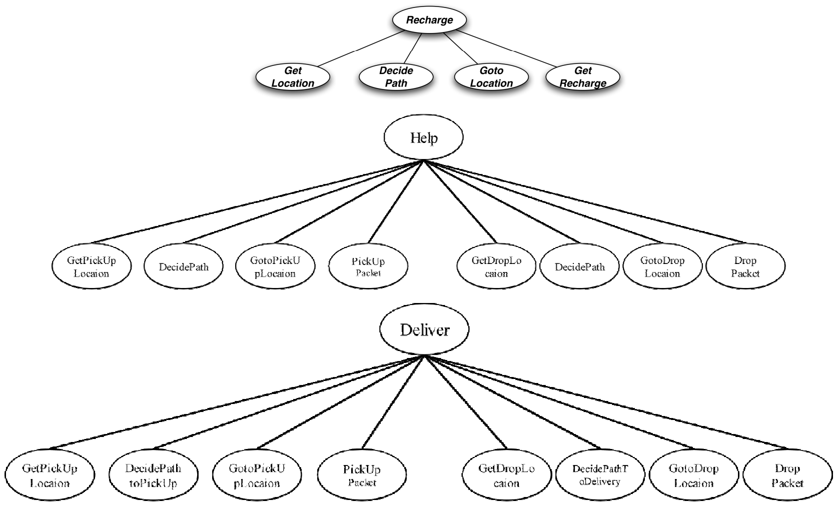

Figure 2. Task representations.

Help Task - The helping task is similar to the packet delivery task, with the only difference that this time the packet information comes from another regular agent rather than from AgentX. When an agent decides to help out another agent, it gets the pickup and drop-off location from the asking agent. It then calculates a path to the pickup location and follows that path. Upon reaching the pickup location it takes the packet and then calculates the path to the drop-off location. It follows the path till it reaches the drop off location. Upon successful delivery the agent informs the asking agent of the successful delivery.

An agent can only perform one task at a time. Multiple delivery tasks cannot be combined together. Every delivery task comes with a deadline, the agent needs to accomplish it by the deadline to avoid paying penalty.

4. STUDY OF SOCIAL RELATIONS

We designed a set of experiments to study the behavior of regular agents. We hypothesize that developing social relations amongst agents are beneficial to the overall agent community and to the individuals as well. For this experiment we used a total of six regular agents that move and interact on a 25 x 25 size grid map. Each of the six agents starts with a fixed amount of energy and money. The experiments were run for 10,000 simulation time pulses. New packages are generated every 30 to 35 time pulses with a random reward in the range of [40, 80].

4.1 Various Decision Mechanisms for Helping

To study the effect of social relations on agent behavior we came up with different mechanisms for making decisions whether to help another agent. One of the decision factors is the money being offered by the asking agent. The other factor is the liking of this agent has towards the asking agent. Initially all agents start with no liking amongst themselves. During the course of the experiment as the agents interact with each other the liking value increases for granted helping requests and decreases for declined helping requests. In the future work, an explanation will be added when an agent declines a help request, which may improve the measurement of the liking value. With these two decision factors we have constructed three variations of helping decision mechanisms based on the liking value this agent has toward the requesting agent and the agent's expectation towards the other agent (money value).

Mechanism 1: **Money and liking.** The agent cares about both money and also being nice to the other agent. If the money offered is really high (*money value >= MoneyHighThreshold*) or if the two agents share a strong friendship (*liking value >= RelationHighThreshold*) , then it helps the other agent with the delivery job. It also helps the other agent when neither the money offered or the liking score between them is too low (*money value >= MoneyLowThreshold* and *liking value >= RelationLowThreshold*).

Mechanism 2: **Liking only**. The agent does not care about the value of the money being offered at all. If it likes the other agent (*liking value >= RelationThreshold*) then it helps the agent regardless of the amount of money being offered.

Mechanism 3: **Money only**. The agent cares only about the money. It makes its decision solely based on the monetary value of the delivery job being offered, without any consideration to the relationship. It helps when the amount of money being offered is high (*money value >= MoneyThreshold*)

Parameter	Set 1-1	Set 2	Set 3	Set 1-2	Set 1-3	Set 1-4
Helping Mechanism	Money+ Liking	Liking	Money	Money+ Liking	Money+ Liking	Money+ Liking
MoneyLowTsd.	20			35	20	35
MoneyHighTsd.	40			40	45	45
MoneyTsd.			20			
RelationLowTsd.	-2			-2	-2	-2
RelationHighTsd.	15			15	15	15
RelatinTsd		-2				

Figure 3. Experiment setting for six different sets

4.2 Experimental Results

Three groups of experiments were performed by using different values for the experiment parameters. Each of these three groups consists of six individual sets of experiments that use the same overall experiment parameters but differ in agent help mechanisms and parameter values in help mechanisms. Each of these sets consists of six regular agents. The help parameters and help mechanisms used for each of the six sets is given in Figure 3. Notice that *RelationLowThreshold* and *RelationThreshold* have negative value as *-2*, in order to encourage agents to initialize helping behavior in the beginning phase of the experiments when the liking values for other agents are 0.

The behavior of each of the agents is analyzed and tabulated under various heading notations. These notations are used in the experiment analysis table, and are explained in Figure 4. All three group experiments use 70% as 'Help Money Ratio', which is the percentage of the total package utility that is offered to the helping agent in return for the delivering of the package. The same map changing pattern and the same package lists are used for all experiment sets to ensure the comparison's fairness.

Group 1: Dynamic map, random generated tasks

In this group of experiment, new packages are generated every 30 to 35 time pulses with a random reward in the range of [40,80]. Figure 5 presents the result, which shows the set 1-3 has the highest success percentage followed by set 1-1 and set 2. In set 1-1, set 2 and set 1-3 we can see that there is formation of strong inter-agent relations, with the agents pairing up and helping each other out in delivering packages. Agent pair is identified based on the significance of the number of mutual help actions between two agents. It is observed that one agent can only pair with one another agent in a given experiment run. The reason can be that the more positive response it receives from another agent, the more likely it will request help from that agent again before asking other agents. In set 3 we do not find any such pairing up of agents. In set 1-2 and set 1-4 we can see that not all the agents have formed pairs. From these results we find that using help mechanism 1 that combining utility and social relation consideration delivers a better success ratio as compared to help mechanism 2 and 3. We also find that set 1-3 performs the best as it has the maximum range for the money range.

Group 2: Dynamic map, random generated burst tasks

In this group of experiments we employ a different strategy for package generation. The packages are generated in a burst of 1-4 packets with an interval of 5 pulses between them. The next burst of packages then happens after a random amount of time between the values of 30 and 65. Using this package generation method we have more packages appearing in bursts, which would make the agents busier and probably increase the chances of inter agent interaction.

Heading	Explanation
No. of Tasks Delivered	The total number of directly-assigned packages that were successfully delivered by the agent
No. of Tasks Delivery Failed	The total number of directly-assigned packages that could not be delivered to the destination
No. of Tasks Helped	The total number of tasks that were delivered by an agent to help another agent.
No. of Tasks Help failed	The total number of packages that could not be delivered to help another agent
Success %	The ratio of the total number of successfully delivered directly-assigned packages to the total number of directly-assigned packages.
Utility (fp=0.5)	Sum of the money that the agent made by delivering packages and helping agents less the penalties the agent paid for failure, if failure penalty is 50%
Utility (fp=0.9)	Utility that the agent would make if failure penalty was 90%
Utility (fp=0.2)	Utility that the agent would make if failure penalty was 20%
Agent Pairs	Lists the mutually cooperative pairs formed among the agents. The agent pairs were determined based on the number of times that these two agents help each other.
Social Relations	Based on the number of mutually cooperative agent pairs formed within the system, the system is classified as having strong, weak or none social relations. Strong relation means all agents formed pairs, and none indicates no agent pairing.

Figure 4. Notation used in result analysis

Sets		1-1	2	3	1-2	1-3	1-4
No. of Tasks	Delivered	167	144	158	157	151	138
	Del. Failed	0	0	25	28	0	24
	Helped	89	105	75	79	112	96
	Help Failed	54	59	51	46	47	50
Success %		60.24	58.59	54.82	55.53	61.88	55.06
Utility	fp = 0.5	13387	12806	12020	12048	13891	11663
	fp = 0.9	12153.4	11454	10434.4	10417.6	12811	9991.8
	fp = 0.2	14312.2	13820	13209.2	13270.8	14701	12916.4
Agent Pairs		A1-A3	A1-A4		A3-A5	A1-A3	A2-A4
		A2-A6	A2-A3		A4-A6	A2-A6	A3-A5
		A4-A5	A5-A6			A4-A5	
Social Relations		strong	strong	none	weak	strong	weak

Figure 5. Group 1 results (fp: failure penalty ratio).

In this group of results shown in Figure 6, set 1-1 has the highest success percentage followed by set 1-3 and set 2. In set 1-3 and set 1-4 we can see that there is formation of strong inter-agent relations, with the agents pairing up and helping each other out in delivering packages. In set 3 we do not find any such pairing up of agents and formations of inter-agent relations. In set 1-1, set2 and set 1-2 we can see that not all the agents have formed pairs.

From these results, we again find that using help mechanism 1 delivers a better success ratio as compared to help mechanism 2 and 3. We also observe that as compared to group 1, set 2 that using only social relation for decision-making performs worse here. This can be explained as the following: since the agents become much busier in this experiment, agreeing to help other agents solely based on social relation consideration may result in over-committing itself and not able to finish the delivery tasks it intends to.

Group 3: Mixed group of agents

In this group of experiments we use the same settings as from group 2 with the exception that we have a mix of agents with different help mechanism together.

- Agent A1 and Agent A2 use mechanism 1 (money + liking).
- Agent B1 and Agent B2 use mechanism 2 (only liking).
- Agent C1and Agent C2 use mechanism 3 (only money).

Sets		1-1	2	3	1-2	1-3	1-4
No Of Tasks	Delivered	173	138	156	135	157	151
	Del Failed	0	0	24	27	0	33
	Helped	95	110	83	102	109	95
	Help Failed	48	67	57	58	55	43
Success %		60.77	56.24	54.20	53.74	60.32	55.78
Utility	fp = 0.5	14665	13243	12305	11839	14209	12809
	fp = 0.9	13614.6	12077.4	10573	9984.6	12969	11194.6
	fp = 0.2	15452.8	14117.2	13604	13229.8	15139	14019.8
Agent Pairs		A1-A2	A1-A5		A1-A2	A1-A6	A1-A5
		A3-A4	A3-A4		A4-A5	A2-A3	A2-A6
						A4-A5	A3-A4
Social Relations		weak	weak	none	weak	strong	strong

Figure 6. Group 2 results

In this group results shown in Figure 7, we find that the agent pairs are formed between those agents that use liking/disliking factors in their help mechanism. Whereas, the Agents C1 and C2 that only use money as their decision factor do not form any pairs. Overall the success ratio is in between a set of agents all using mechanism 1 and a set of agents all using mechanism 3. We had expected such a performance since we are using agents of different help mechanisms.

Sets		Mix set
No Of Tasks	Delivered	137
	Del Failed	2
	Helped	122
	Help Failed	55
Success %		58.73
Utility	fp = 0.5	14741
	fp = 0.9	14183
	fp = 0.2	15160
Agent Pairs		A1-B2
		A2-B1
Social Relations		weak

Figure 7. Group 3 results.

4.3 Result Summary

Based on our observation of the results from the experiment groups above we can see that sets 1-1 and 1-3 with a hybrid decision-making mechanism have better performance than all the other sets. Set 3 with utility-only decision mechanism has the worst performance on average and set 2 with social-relation-only mechanism has a median performance. We can attribute the success of type 1 agent community over the type 2 and type 3 communities to the fact that type 1 agents use social relation measurement liking/disliking to influence their decision making process. They form cooperative teams with other agents and help each other out whenever possible. Using social relation measurement the agents are able to make better helping decisions. The community performance is better with a higher percentage of packages being delivered successfully and the amount of money made is higher. Thus we conclude that having social relations among agents proves to be beneficial in making better decisions and working together in this experimental setting.

5. CONCLUSION

In this paper we hypothesized that cooperative behavior based on agent liking/disliking model provides a robust decision mechanism for agents to develop stable, mutually beneficial groups. This type of decision mechanism adds to the agent's rational decision-making capabilities. We simulated artificial package delivery environment with different combinations of agents employing money and liking/disliking mechanism to make helping decisions. Our experiments established that agents using hybrid utility and social relation consideration for decision making performed significantly better compared to agent using only utility reasoning or only social relation consideration in decision making under similar simulation environments.

Evaluating scenarios where agents have the freedom of using different decision mechanisms can produce interesting insight about evolving agent strategies. An adaptive agent can chooses which decision mechanism to use depending on the type of the other agent that is in the current interaction. We also plan to investigate the effects of introducing multiple adaptive agents into the system and the effects on the performance and the stability of agent relations. We believe that creating artificial agents using more complex social behavioral models than what we have used in this work can help building better agents to be deployed in applications such as electronic commerce, recommender systems and personal assistants.

REFERENCES

Castelfranchi, C., Conte, R., and Palucci, M. (1998). Normative reputation and the costs of compliance. *Journal of Artificial Societies and Social Simulation.*

D, G. (1989). *Emotional Intelligence*. New York: Aldine de Gruyter.

Decker, K. (1996). TAEMS: A Framework for Environment Centered Analysis & Design of Coordination Mechanisms. *In Foundations of Distributed Artificial Intelligence*, Chapter 16, pages 429–448. G. O'Hare and N. Jennings (eds.), Wiley Inter-Science.

Denzinger, J. and Kordt, M. (2000). Evolutionary online learning of cooperative behavior with situation-action pairs. *In Proceedings. of the Fourth International Conference on Multi- Agent Systems*, 2000, pages 103–110.

Glass, A. and Grosz, B. (2000). Socially conscious decision-making. *In Proceedings of Agents 2000 Conference*, pages 217 – 224, Barcelona, Spain.

Horling, B., Vincent, R., and Lesser, V. (2000). Multi-agent system simulation framework. *In 16th IMACS World Congress 2000 on Scientific Computation, Applied Mathematics and Simulation*. EPFL.

Huhns, M. N. and Singh., M. P. (1997). *Readings in Agents*. Morgan Kauffman.

Lerman, K. and Shehory, O. (2000). Coalition formation for large-scale electronic markets. *In Proceedings. of the Fourth International Conference on Multi- Agent Systems*, pages 167–174.

Saha, S. (2005). Predicting agent strategy mix of evolving populations. *In Proceedings of the Fourth International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 1075–1082.

Saha, S., Sen, S., and Dutta, P. S. (2003). Helping based on future expectations. *In Proceedings of the Second International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 289–296, Melbourne, Australia. ACM Press.

Schillo, M., Funk, P., and Rovatsos, M. (2000). Using trust for detecting deceitful agents in artificial societites. *In Applied Artificial Intelligence, Special Issue on Trust, Deception and Fraud in Agent Societies*.

Sen, S. (1996). Reciprocity: a foundational principle for promoting cooperative behavior among self-interested agents. *In Proceedings of the Second International Conference on Multiagent Systems*, pages 322–329, Menlo Park, CA. AAAI Press.

Sen, S. (2002). Believing others: Pros and cons. *In Artificial Intelligence*, pages 142(2):179–203.

Shehory, O. and Kraus, S. (1998). Methods for task allocation via agent coalition formation. *Artificial Intelligence*, 101(1-2):165–200.

Vincent, R., Horling, B., and Lesser, V. (2001). An Agent Infrastructure to Build and Evaluate Multi-Agent Systems: The Java Agent Framework and Multi-Agent System Simulator. *Lecture Notes in Artificial Intelligence: Infrastructure for Agents, Multi-Agent Systems, and Scalable Multi-Agent Systems*, 1887.

Yu, B. and Singh, M. P. (2000). A social mechanism of reputation management in electronic communities. *In Proceedings of Fourth International Workshop on Cooperative Information Agents*, pages 154–165.