Applying Environmental Factors to Trust Algorithms in Competitive Multi-Agent Systems

Michael deFreitas, Harvey Siy, Mansour Zand
Department of Computer Science, University of Nebraska at Omaha
Email: {mdefreitas,hsiy,mzand}@unomaha.edu

***Abstract*—In any multi-agent system (MAS) where agents must rely on each other in order to achieve individual or shared goals, the issue of trust is important. How does one agent decide whether or not to rely on another, particular agent to perform a task? This and other related questions are at the forefront of recent research into trust and reputation in MAS. One area not deeply explored is the effect of the MAS environment itself on trust decisions. For example, if an agent is operating in a MAS where it is expensive (in whatever manner "expensive" makes sense in the MAS) to initiate a transaction with another agent, should that relatively high cost affect the agent's trust decisions and if so, how? What about the level of competitiveness in the MAS? Are the agents working towards a set of common goals, or is it "every agent for itself"? How should each type of environment – or even an environment where the level of competitiveness changes over time – affect a participating agent's trust decisions?**

This work explores methods for considering those types of environmental factors in an agent's trust algorithm. The theory is that an agent capable of a) detecting and b) reacting to certain environmental factors will be more effective in accomplishing its goals, whether those goals are shared with other agents or not. Using the current state-of-the-art research testbed, an "environmentally-aware" trust algorithm will be designed and implemented in a software agent. This agent will then be pitted against a "stock" (unmodified) agent in a simulated competitive MAS to see if the modified agent outperforms its peers.

***Index Terms*—multi-agent systems, trust, agent reputation, simulation, testbed**

I. Introduction

Agents in multi-agent systems (MAS) where agents interact or exchange information require some algorithm to determine the trustworthiness of other agents. This is especially true in competitive MAS, where agents involved in a transaction might not be working towards the same goal. The Complexity Model of trust involves trust evidence of three kinds: Competence, Motivation, and Continuity [1]. The truster's evaluation of these three types of evidence (as they relate to the trustee) results in a number representing the truster's confidence in a successful transaction with the trustee. This work will examine the effect of considering environmental factors when weighting the importance of the three attributes. Specifically, the environmental factors to be considered are 1) the cost of initiating a transaction and 2) the cost of transaction failure.

A. Significance

This work has the potential to add an additional layer of insight onto existing agent trust algorithms. That is, while many algorithms today consider, for instance, the historical performance of peer agents, or perhaps the reputation of an agent as reported by other agents, there seems to be little work in trust "meta algorithms", which use indirect factors to change the internal assumptions of the algorithms (or, more concretely, the weight factors for the three types of trust evidence mentioned above). If this work shows that algorithms tuned for the agent's environment perform better than non-tuned algorithms, it might uncover a way to take existing trust algorithms to a new level of sophistication.

If the inclusion of environmental factors into trust algorithms shows promise, almost any cooperative or competitive MAS could be analyzed for candidate environmental factors, which would then inform the trust algorithms of participating agents. For example, in service-oriented architectures (SOAs) where clients can dynamically bind to different service providers offering the same type of service, clients and services can be represented by different agents. Tracking service reputation can be used to facilitate a more dependable service selection process [10]. Several researchers have studied the promising feasibility of a collaborative, agent-based approach for determining service reputation and trustworthiness [8][9]. Our work complements these studies by providing a trust algorithm that can be used to incorporate environmental factors that model real-world implications impacting service selection.

B. Challenges

The two primary challenges in this work will be 1) quantifying the environmental factors for a given environment (in this case, the Agent Reputation and Trust (ART) testbed; see section III) and 2) correctly accounting for those factors in the trust algorithm. The testbed is highly configurable and already has "costs" for transaction initiation, though there is no specific mechanism for applying a "cost" to transaction failure. A secondary challenge, therefore, will be to simulate costs for transaction failure; costs which can be communicated to the agents at runtime.

C. Objectives

The primary objective of this work is to design, implement, and test a trust algorithm that considers environmental factors and adjusts its calculations when

doi:10.4304/jsw.6.11.2263-2270

those factors change. An agent running this algorithm will be pitted against a "stock" agent in a series of tests to determine if the environmentally-aware agent performs better than its unmodified peer.

II. BACKGROUND

A. History of the problem

The issue of trust between agents (in the general sense) is, of course, at least as old as human history. Every person we know, every person we meet for the first time, and even non-human entities (businesses, web pages, newspapers) are assigned some level of trustworthiness for the entire range of interactions we might have with that person or entity. Every interaction we have may increase our trust in the other entity, it may decrease our trust, or it may have no effect.

Cooperative software agents in a MAS operate under a similar trust dynamic. An agent will have certain information ("knowledge" [3]) about other agents in the system, which may include a history of past interactions, some measure of reputation, information about other agents' capabilities, etc. As an agent gains knowledge, its trust algorithm will make decisions regarding interactions with other agents.

The design and development of knowledge representations specifically related to trust and reputation, and the algorithms that use that knowledge to make decisions, has been the focus of intense research over the last several years. The Agent Reputation and Trust (ART) Testbed was created specifically to research trust algorithms in an environment where agent behavior and interaction can be observed in a simulated competitive environment [6] (See section III).

B. Mathematical Foundation

We propose that the trust algorithm based on the Cognitive Model of Trust [1] (Figure 1) can be described as a function:

$$C = f(A_e, A_t, C_e, C_t, I, S);\ \ 0 \le C \le 1$$

Where:

C = *confidence in successful outcome for a transaction*
A_e = *evidence of trustworthiness of other (trustee) agent*
A_t = *trust in source of* A_e
C_e = *evidence of control over transaction for truster*
C_t = *trust in source of* C_e
I = *confidence in instrument of control over transaction*
S = *trust in self*

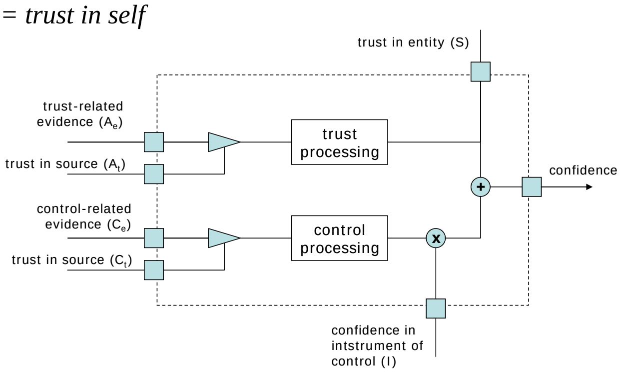

FIGURE 1. COMPLEXITY MODEL OF TRUST [1], SIMPLIFIED

In Cofta's model, A_e and A_t are combined as input to a generic "trust processing" module. Similarly, C_e and C_t are handled by a "control processing" module. Input I – the confidence the agent has in its own instrument of control over the transaction – modifies the output of the "control processing" module. This is then combined with the output from the "trust processing" module and modified by S, the agent's measure of trust in itself. The final output is the measure of confidence C.

To include environmental factors, a seventh parameter E is added:

E = *environmental factors which affect* A_e

This new set of factors modify A_e (trust-related evidence), as shown in the modified Cofta model in Figure 2.

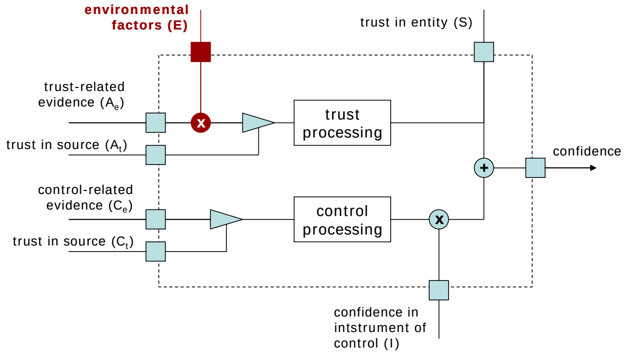

FIGURE 2. COMPLEXITY MODEL OF TRUST, WITH ENVIRONMENTAL FACTORS

For the purposes of this work, this model will be greatly simplified, considering only factors A_e and E, and their effect on the agent's trust algorithm. Although the "stock" agents included in the ART Testbed will be used as both a baseline during testing and as a starting point for modification, any implicit or explicit mechanisms for dealing with the other factors (A_t, C_e, S, etc.) will not be examined or modified.

Factor A_e comes in three types: evidence of (the other agent's) Motivation, evidence of Competence, and evidence of Continuity. The "trust processing" component takes these three types of evidence and computes a trust value, which can be thought of as a point in the "cube of trust" [1] (Figure 3).

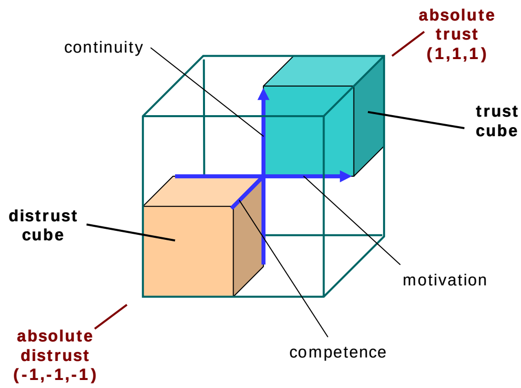

FIGURE 3. CUBE OF TRUST [1]

For this work, we only consider evidence of Motivation (the other agent's willingness to enter into a transaction, and follow through on it) and Competence (the other agent's ability to produce correct results). This simplification results in the following equation, which is to be encoded in the trust algorithm of an agent:

$$C = f(A_{mot}, A_{com}, E_{mot}, E_{com})$$

Where:

C = confidence in successful outcome for a transaction
A_{mot} *= evidence of motivation of trustee agent*
A_{com} *= evidence of competence of trustee agent*
E_{mot} *= environmental factor affecting motivation*
E_{com} *= environmental factor affecting competence*

Furthermore, $\boldsymbol{E_{mot}}$ and $\boldsymbol{E_{com}}$ act as weights on $\boldsymbol{A_{mot}}$ and $\boldsymbol{A_{com}}$, respectively, so a more accurate representation is:

$$C = f((g(A_{mot}) \bullet W_{mot}), (h(A_{com}) \bullet W_{com}))$$

Where $g()$ and $h()$ represent the agent's normal processing for evidence of motivation and competence, respectively, and $\boldsymbol{W_{mot}}$ and $\boldsymbol{W_{com}}$ are weights derived from $\boldsymbol{E_{mot}}$ and $\boldsymbol{E_{com}}$.

The primary programmatic task of this work is to derive $\boldsymbol{W_{mot}}$ and $\boldsymbol{W_{com}}$ from information available to the agent, and apply those weights to the existing functions $g(\boldsymbol{A_{mot}})$ and $h(\boldsymbol{A_{com}})$ appropriately.

C. Defining E_{mot} and E_{com}

The following sections describe $\boldsymbol{E_{mot}}$ and $\boldsymbol{E_{com}}$ in more concrete terms.

1. E_{mot} – Cost of Transaction Initiation

An unmotivated trustee agent – that is, one that does not follow through on requested transactions – adversely affects the truster agent by causing it to incur some Cost of Transaction Initiation. If this cost is trivial, then unmotivated trustees will have little effect on the overall success of the truster, and if the cost is high, broken promises by trustee agents may have significant effect on the truster.

$\boldsymbol{E_{mot}}$ is therefore mapped to the Cost of Transaction Initiation, and $\boldsymbol{W_{mot}}$ rises as the cost rises. This will be described in terms of ART Testbed parameters in section III.

2. Ecom – Cost of Transaction Failure

An incompetent trustee agent – that is, one that follows through on requested transactions, but produces poor results – adversely affects the truster agent by causing it to incur some Cost of Transaction Failure. If this cost is trivial, then incompetent trustees will have little effect on the overall success of the truster, and if the cost is high, poor performance by trustee agents may have significant effect on the truster.

$\boldsymbol{E_{com}}$ is therefore mapped to the Cost of Transaction Failure, and $\boldsymbol{W_{com}}$ rises as the cost rises. This will be described in terms of ART Testbed parameters in section III.

III. Method of Solution

A. Technological Framework

To assess the efficacy of factoring in these environmental variables, we created a simulation environment based on the ART Testbed, which includes everything needed to construct and modify agents, pit them against each other in competitions, and collect data for later analysis. Details of the implementation are described in a technical report [2].

The testbed simulates an "art appraisal" domain, wherein each agent acts as an appraiser for simulation-provided "clients", each of which pays a fixed fee to have a painting appraised. Paintings have "eras" and each agent has an expertise in each era; the higher the expertise, the more accurate that agent will be in appraising paintings. During each timestep of the simulation, the following actions take place:

1. Clients are assigned to each agent, and they pay a fee for a painting appraisal
2. Each agent optionally asks for reputation information from other agents, about 3rd-party agents (i.e. agent A asks agent B about agent C)
3. Each agent optionally asks other agents to provide opinions on one or more of its assigned paintings. These opinions may or may not be provided, and they may or may not be accurate.
4. Each agent gathers the opinions provided by other agents, and optionally an opinion it generates itself, and sends them to the simulation along with weight values for each.
5. The simulation calculates the appraisal errors and returns the results to the agents. Also, the simulation redistributes clients based on the average accuracy of each agent (more-accurate agents get more clients, less-accurate agents get fewer clients)

Each agent is responsible for keeping its own counsel about the reputation of other agents, based on any of the information passed back and forth during each timestep. Agents are not compelled to tell the truth about their expertise, or to spend much effort (expressed as fees paid to the simulation) in generating opinions for other agents, or even to follow through on promises to other agents. It's perfectly fair for an agent to take opinion fees from other agents and then do nothing in return. At the end of the game, the agent with the most money wins.

B. Experimental Components

The major components required to conduct the experiments in this work are the ART Simulation Engine, the ART Game Management Interface, and four agents: the Stock Agent, the Modified Agent, an "Honest Agent", and an "Ornery Agent". Additionally, the ART produces output files and includes a graphical analysis tool that will be used to evaluate experiment results.

1. Simulation Configuration

The ART simulation engine is highly configurable, utilizing XML files for setting a range of simulation parameters. The three parameters of primary importance to this work are:

TABLE I. GLOBAL SIMULATION PARMETERS

Parameter	Symbol
Client Fee	f
Opinion Cost	C_p
Prior Timestep Client Factor	q

For each experiment, these values are modified to provide a given environment to the Modified Agent, e.g. "High Cost of Transaction Initiation", or "Low Cost of Transaction Failure". The environment variables are calculated from the parameters as shown in Table II.

TABLE II. ENVIRONMENT VARIABLES

Environment Factor	Interpretation
Cost of Transaction Initiation (C_p / f)	C_p / f > .25 = High C_p / f < .10 = Low Otherwise Normal
Cost of Transaction Failure (1-q)	(1-q) < .4 = Low (1-q) > .7 = High Otherwise = Normal

2. Stock Agent

The stock agent is the baseline competitor for this work. The Stock Agent has a basic reputation and trust strategy for dealing with other agents, but does not change its strategy in different environments. This is the "agent to beat" for the Modified Agent, which is simply a Stock Agent modified to consider environmental factors.

TABLE III. STOCK AGENT STRATEGY

Strategy Component	Logic
Starting Reputation	All agents initialized with reputation = 1.0
Requesting Opinions	Sends requests for all assigned paintings to all agents with reputation > 0.5
Confirming Opinion Requests	Confirms (and pays C_p for) all opinion request replies with certainty > 0.7
Responding to Opinion Requests	Provides true expertise to all requesting agents
Providing Opinions	Provides best effort (paying full cost) for agents with reputation > 0.5; provides minimal effort (paying minimum cost) for other agents.
Adjusting Reputation – Motivation	Adjusts target agent's reputation -.03 for all ordered opinions that are not provided.
Adjusting Reputation - Competence	Adjusts target agent's reputation +.03 for all opinions within 50% of painting's true value; adjusts reputation -.03 otherwise.

3. Modified Agent

The Modified Agent implements an environmentally-aware strategy and is the focus of this work. It behaves in a very similar fashion to the Stock Agent, except in the way it modifies reputations of other agents, as described in Table IV. Note that if both the Cost of Transaction Initiation and the Cost of Transaction Failure are normal, the Modified Agent behaves exactly the same as the Stock Agent.

4. Honest Agent

The Honest Agent is a "dummy" agent which does not keep track of agent reputation. It is included in the experiments as a control agent, to gauge the behavior of the Stock and Modified Agents towards the Honest Agent vs. the Ornery Agent.

TABLE IV. MODIFIED AGENT STRATEGY

Strategy Component	Logic
Starting Reputation	Same as Stock Agent
Requesting Opinions	Same as Stock Agent
Confirming Opinion Requests	Same as Stock Agent
Responding to Opinion Requests	Same as Stock Agent
Adjusting Reputation – Motivation	Evaluates the Cost of Transaction Initiation as High, Low, or Normal. For all ordered opinions that are not provided, the target agent's reputation is adjusted .1 if the cost is High, .01 if the cost is Low, or .03 if the cost is Normal.
Adjusting Reputation - Competence	Evaluates the Cost of Transaction Failure as High, Low, or Normal. For all opinions within 50% of the painting's true value, the reputation of the providing agent is adjusted upward. Otherwise the reputation of the providing agent is adjusted downward. The amount of the adjustment is .06 if the cost is High, .01 if the cost is Low, and .03 if the cost is Normal.

TABLE V. HONEST AGENT STRATEGY

Strategy Component	Logic
Starting Reputation	No reputation logic.
Requesting Opinions	Requests opinions on all assigned paintings, from all agents.
Confirming Opinion Requests	Confirms (and pays C_p for) all opinion request replies.
Responding to Opinion Requests	Provides true expertise to all requesting agents.
Providing Opinions	Provides best effort (paying full cost) for all opinions.
Adjusting Reputation – Motivation	No reputation logic.
Adjusting Reputation – Competence	No reputation logic.

5. Ornery Agent

The Ornery Agent has a basic reputation and trust strategy similar to the Stock Agent, but its motivation and competence are both randomized. At the start of each simulation, the Ornery Agent randomly calculates two parameters: Motivation Threshold and Opinion Cost.

TABLE VI. ORNERY AGENT STRATEGY

Strategy Component	Logic
Starting Reputation	Same as Stock Agent
Requesting Opinions	Same as Stock Agent
Confirming Opinion Requests	Same as Stock Agent
Responding to Opinion Requests	Same as Stock Agent
Providing Opinions	Randomly determines which opinions to provide, and randomly determines amount to spend on each opinion.
Adjusting Reputation – Motivation	Same as Stock Agent
Adjusting Reputation - Competence	Same as Stock Agent

Every time the Ornery Agent decides whether or not to follow through on a promised opinion transaction, it generates a random number between 0 and 1 and compares it to the Motivation Threshold. If the number is higher than the threshold, the Ornery Agent will follow through and generate an opinion, paying the Opinion Cost it randomly determined at the beginning of the simulation. If the random number is less than the Motivation Threshold, then the Ornery Agent will not follow through on the transaction (though it keeps the opinion fee – C_p – spent by the requesting agent).

For example, an Ornery Agent with a Motivation Threshold of 0.75 and an Opinion Cost of 9.5 will be relatively unmotivated (refusing to generate promised opinions 75% of the time), but highly competent (paying near the maximum for any opinions it generates).

The Ornery Agent provides the interesting and varied behavior required to test the Modified Agent's treatment of agents with various motivation and competence levels.

6. Mapping to Mathematical Foundation

As described in section II, there are several components that go into the calculation of $\boldsymbol{C}$, the overall confidence an agent has in another agent. In both the Stock and the Modified Agents, $\boldsymbol{C}$ takes the form of a "reputation" value between 0 and 1, inclusive, which is kept for each agent in the game. The functions g() and h() are computed at the start of each timestep, at which point all the information from the prior timestep is available, including which agents followed through on their promised opinions (evidence of motivation, $\boldsymbol{E_{mot}}$) and the accuracy of those opinions ($\boldsymbol{E_{com}}$).

Finally, the weight parameters $\boldsymbol{W_{mot}}$ and $\boldsymbol{W_{com}}$ are realized in the reputation adjustment parameters used by the Modified Agent, as described in Table IV.

C. Experiments

1. Experimental Strategy

To test the Modified Agent's performance compared to the Stock Agent, a set of four different game types will be run, with each game executed three times and the results averaged. The critical measure of success is the Modified Agent's ability to outperform the Stock Agent in each game type. It is not necessary for the Modified Agent to actually win the game (i.e. end up with the most money among all agents), it is only necessary for it to beat the Stock Agent. The theory being tested proposes that an "environmentally aware" agent will outperform a similar, but unaware, agent. This does not necessarily mean that the Modified Agent will be the best-performing agent overall. In fact, due to the many trust and reputation factors that are essentially ignored as well as the relative success of "cheating" agents (which the Modified Agent is not), it is likely that the Modified Agent will not be the overall winner.

2. Game Parameters and Participants

The following were in effect for each game:

TABLE VII. GAME PARAMETERS

Parameter	Value
Number of agents (1 stock, 1 modified, 3 honest, 3 ornery)	8
Rounds per game	20
Number of painting eras	10
Client fee	100

Note that the motivation threshold and opinion cost for each Ornery Agent is decided randomly at the start of the game, and changes between games. This means that each of the three trials of each game type will have a different mixture of motivation/competence among the participating Ornery Agents.

3. Baseline Game

The baseline simulation will capture data about the nominal performance of the Stock Agent and the Modified Agent in an environment where both should follow essentially the same trust algorithm, that is where both the Cost of Transaction Initiation and Cost of Transaction failure are normal. The variable game parameters are set in Table VIII.

TABLE VIII. BASELINE GAME PARAMETERS

Parameter	Value	Environmental Factor	Category
Opinion fee (C_p)	15	C_p/f = .15	Normal
Prior timestep client factor (*q*)	.5	(1-q) = .5	Normal

4. High Cost of Transaction Initiation Game

This game raises the Cost of Transaction Initiation, testing the Modified Agent's ability to deal with unmotivated Ornery Agents (Stock and Honest agents are always fully motivated, as described in section II). The variable game parameters are set in Table IX.

TABLE IX. HIGH COST OF TRANSACTION INITIATION GAME PARAMETERS

Parameter	Value	Environmental Factor	Category
Opinion fee (C_p)	50	C_p/f = .5	High
Prior timestep client factor (*q*)	.5	(1-q) = .5	Normal

5. High Costs Game

This game has both a high Cost of Transaction Initiation and Cost of Transaction Failure. This tests the Modified Agent's ability to deal with lack of motivation and/or lack of competence in other agents. The variable game parameters are set in Table X.

TABLE X. HIGH COSTS GAME PARAMETERS

Parameter	Value	Environmental Factor	Category
Opinion fee (C_p)	50	C_p/f = .5	High
Prior timestep client factor (*q*)	.2	(1-q) = .8	High

6. Low Cost of Initiation, High Cost-of-Failure Game

This game mixes a low Cost of Transaction Initiation – encouraging lenience towards unmotivated Ornery Agents – with a high Cost of Transaction Failure. The Modified Agent is expected to continue to trust agents that don't meet promises, but quickly eliminate agents that don't provide accurate opinions (when they choose to provide opinions at all). The variable game parameters are set in Table XI.

TABLE XI. LOW COST OF INITIATION, HIGH COST OF FAILURE GAME PARAMETERS

Parameter	Value	Environmental Factor	Category
Opinion fee (C_p)	5	C_p/f = .05	Low
Prior timestep client factor (*q*)	.2	(1-q) = .8	High

IV. Results and Discussion

A. Summary of Results

The results of the experiments were promising, with the Modified Agent outperforming the Stock Agent in two of the three non-baseline scenarios. In the baseline scenario, the Stock Agent eked out a narrow average victory, but overall difference with the Modified Agent was minimal, as expected.

TABLE XII. Summary of Results

Game	Average Ending Balance: Stock Agent	Average Ending Balance: Modified Agent	# Wins by Modified Agent
Baseline	**35,969**	35,369	1/3
High Cost of Transaction Initiation	119,559	**122,119**	3/3
High Costs	109,249	**120,545**	3/3
Low Cost of Initiation, High Cost of Failure	**73,173**	68,509	0/3

Detailed results for each game are in the following sections.

B. Results of Specific Experiments

1. Baseline Game

The Baseline game was not analyzed in great detail, only to verify that the Stock Agent and the Modified Agent behaved similarly, as expected. Table XIII**Error! Reference source not found.** shows the settings of the Ornery Agents. The table is read as follows. In trial one, Ornery Agent #1 was highly motivated (it had a relatively low Motivation Threshold; any random number greater than .37 would cause the agent to follow through on a promised opinion), but it had low competence because it was only willing to spend 2.1 "units" of money (out of a maximum 10) for each opinion generated. In trial #3, Ornery Agent #2 had very low motivation (refusing to honor its agreements 88% of the time) and *also* had very low competence, meaning that even if it did follow through and provide a promised opinion, it spent very little (1.4 units) on that opinion, making it likely to be inaccurate.

TABLE XIII. Baseline Game Ornery Agents

Trial #	Ornery Agent #	Motivation Threshold	Opinion Cost
1	1	.37 (H)	2.1 (L)
	2	.47 (M)	0.5 (VL)
	3	.62 (L)	4.2 (M)
2	1	.44 (M)	4.7 (M)
	2	.79 (L)	7.8 (H)
	3	.10 (VH)	6.2 (H)
3	1	.92 (VL)	9.5 (VH)
	2	.88 (VL)	1.4 (VL)
	3	.53 (M)	8.9 (VH)

Table XIV shows the final reputation of each Ornery Agent, and the timestep in which the agent was deemed "untrustworthy". In the non-baseline game details, this table gives insight into the effect of the Modified Agent's more or less aggressive approach to lack of motivation and/or competence in the Ornery Agents. Where applicable, the earliest timestep that an Ornery Agent is dropped is marked in **bold**.

TABLE XIV. Baseline Game Ornery Agent Longetivity

Trial #	Ornery Agent #	Final Reputation/ Timestep Dropped: Stock Agent	Final Reputation/ Timestep Dropped: Modified Agent
1	1	0.46 / 5	**0.40 / 3**
	2	0.28 / 5	0.31 / 5
	3	0.28 / 5	0.31 / 5
2	1	0.40 / 6	0.40 / 6
	2	**0.46 / 5**	0.46 / 6
	3	**0.31 / 9**	0.31 / 10
3	1	0.40 / 3	0.40 / 3
	2	0.28 / 5	0.31 / 5
	3	0.46 / 7	**0.46 / 6**
Avg. Final Balance		**35,969**	35,369

It is clear from these results that the Stock Agent and the Modified Agent both treated the various Ornery Agents in approximately the same way. It is interesting to note that none of the Ornery Agents survived past the 10th timestep (out of 20) in any of the simulations. As expected, the highly-motivated, competent Ornery Agent #3 in Trial #2 survived the longest. The possible reasons for all the Ornery Agents' early demise are discussed in section IV.C.

2. High Cost of Transaction Initiation Game

In this game, it's expected that the Modified Agent will take aggressive action against any Ornery Agent that does not fulfill its promises, and that is what the data shows.

TABLE XV. High Cost of Transaction Initiation Game Ornery Agents

Trial #	Ornery Agent #	Motivation Threshold	Opinion Cost
1	1	.65 (M)	9.6 (VH)
	2	.55 (M)	0.9 (VL)
	3	.57 (M)	0.9 (VL)
2	1	.44 (M)	4.7 (M)
	2	.82 (VL)	7.8 (H)
	3	.10 (VH)	5.2 (M)
3	1	.92 (VL)	7.4 (H)
	2	.64 (M)	3.6 (L)
	3	.53 (M)	8.9 (VH)

TABLE XVI. High Cost of Transacton Initiation Game Ornery Agent Longetivity

Trial #	Ornery Agent #	Final Reputation/ Timestep Dropped: Stock Agent	Final Reputation/ Timestep Dropped: Modified Agent
1	1	0.40 / 5	**0.49 / 3**
	2	0.43 / 5	**0.34 / 4**
	3	0.43 / 7	**0.34 / 3**
2	1	0.40 / 7	**0.40 / 6**
	2	0.46 / 4	**0.28 / 1**
	3	0.31 / 10	0.46 / 10
3	1	0.43 / 3	**0.28 / 1**
	2	0.28 / 5	**0.31 / 3**
	3	0.46 / 7	**0.40 / 4**
Avg. Final Balance		119,559	**122,119**

The Modified Agent, as expected, took a very hard line against Ornery Agents when they didn't provide promised opinions. In every case but one (where the Ornery Agent was highly motivated, and therefore provided 90% of its promised opinions), the Modified Agent dropped the Ornery Agents below the level of "untrustworthy" sooner than did the Stock Agent. In two cases (Trial 2/Ornery Agent 2, and Trial 3/Ornery Agent 1), the Modified Agent dropped Ornery Agents with "Very Low" motivation after just a single timestep, due to the high number of promised opinions that were not provided.

3. High Costs Game

In this game both the Cost of Transaction Initiation and the Cost of Transaction Failure are high. In contrast to the previous game, highly motivated, but incompetent, agents will be aggressively excluded by the Modified Agent, as it harshly penalizes both failure to follow through, and failure to provide high-quality results.

From Table XVIII, the Modified Agent again was more aggressive than the Stock Agent in most trials. The relative longevity of Ornery Agents #2 and #3 in Trial 1, and Ornery Agent #1 in Trial 3, was due to fortunate random elements in opinions provided to the Modified Agent, which resulted in greater than usual positive adjustments to the agents' reputation due to what the Modified Agent perceived as high competence. These random elements, introduced by the simulation engine, will be discussed in section IV.C.

TABLE XVII. HIGH COSTS GAME ORNERY AGENTS

Trial #	Ornery Agent #	Motivation Threshold	Opinion Cost
1	1	.59 (M)	3.7 (L)
	2	.63 (M)	6.1 (M)
	3	.49 (M)	4.4 (M)
2	1	.21 (H)	5.3 (M)
	2	.94 (VL)	4.7 (M)
	3	.77 (L)	7.8 (H)
3	1	.33 (H)	1.1 (VL)
	2	.41 (M)	6.7 (H)
	3	.90 (VL)	4.1 (M)

TABLE XVIII. HIGH COSTS GAME ORNERY AGENT LONGEVITY

Trial #	Ornery Agent #	Final Reputation/ Timestep Dropped	
		Stock Agent	Modified Agent
1	1	0.40 / 8	**0.28 / 5**
	2	0.46 / 6	0.46 / 6
	3	0.46 / 8	0.44 / 8
2	1	0.43 / 7	**0.40 / 6**
	2	0.46 / 4	**0.31 / 2**
	3	0.34 / 8	**0.43 / 6**
3	1	0.46 / 3	0.46 / 3
	2	0.43 / 7	**0.46 / 6**
	3	0.46 / 4	**0.31 / 2**
Avg. Final Balance		109,249	**120,545**

4. Low Cost of Initiation, High Cost-of-Failure Game

The Modified Agent was least successful in this game, ending up with a lower balance than the Stock Agent in all three trials even though the Modified Agent was more aggressive overall.

The result for Trial 1, Ornery Agent #3 is very surprising. This agent was highly motivated, and highly competent, and yet the Modified Agent dropped its reputation below the "untrustworthy" threshold after the 2nd timestep. It was likewise too-aggressive towards Ornery Agent #2 in Trial 3, also a motivated, competent agent. Possible reasons for this behavior are discussed in the next section.

TABLE XIX. LOW COST OF INITIATION, HIGH COST OF FAILURE GAME ORNERY AGENT

Trial #	Ornery Agent #	Motivation Threshold	Opinion Cost
1	1	.01 (VH)	1.8 (VL)
	2	.63 (L)	9.3 (VH)
	3	.07 (VH)	9.5 (VH)
2	1	.47 (M)	8.3 (VH)
	2	.50 (M)	3.7 (L)
	3	.27 (H)	2.7 (L)
3	1	.83 (VL)	8.1 (VH)
	2	.21 (H)	6.8 (H)
	3	.84 (VL)	4.1 (M)

TABLE XX. LOW COST OF INITIATION, HIGH COST OF FAILURE GAME ORNERY AGENT LONGEVITY

Trial #	Ornery Agent #	Final Reputation/ Timestep Dropped	
		Stock Agent	Modified Agent
1	1	0.40 / 5	**0.40 / 4**
	2	0.31 / 5	**0.29 / 4**
	3	0.46 / 13	**0.20 / 2**
2	1	0.40 / 9	**0.40 / 8**
	2	0.43 / 7	**0.31 / 5**
	3	0.43 / 8	**0.43 / 6**
3	1	0.34 / 4	**0.46 / 3**
	2	0.37 / 11	**0.40 / 7**
	3	0.46 / 5	**0.43 / 4**
Avg. Final Balance		**73,173**	68,509

C. Discussion

While the results of the experiments were promising, the aberration of the final game type exposed problems with the assumptions made when the Modified Agent's logic was designed. A closer review of the simulation results for Ornery Agent #3 in Trial 1 revealed that, even though Agent #3 had relatively high Opinion Costs, the generated appraisals were mostly inaccurate.

The problem lies in the assumption that a competent agent, paying a relatively high opinion cost, will generate accurate appraisals. The logic of the Modified Agent was designed under this assumption, however the simulation clearly does not adhere to this framework as closely as originally thought. There are several simulation parameters that were left at default values that may have an impact on this aspect of opinion-generation behavior [4]. An exploration of these parameters is left for future work as discussed in the next section.

D. Conclusions and Future Work

The goal of this work was to demonstrate, within a very simple, limited framework, that an agent modified to consider environmental factors when making trust decisions would outperform an unmodified, but otherwise identical agent. The modified agent would specifically consider environmental factors affecting motivation-based decisions ($\boldsymbol{E_{mot}}$) and those affecting competence-based decisions ($\boldsymbol{E_{com}}$). These factors would be used to increase or decrease the modified agent's aggressiveness when dealing with other agents of varying motivation and/or competence.

Incorporating these factors provides a more accurate model of how distributed systems work in reality. For example, it improves the fidelity of the trust model in automated service selection in SOAs. With an increasing number of Web services that offer similar functionalities [10], it is becoming more crucial for services to differentiate themselves by their reputation for delivering high quality services. A client may have a choice of several service providers, each with different degrees of responsiveness (motivation) and reliability (competence). Also providers may delegate the service to other service providers. In such scenarios, the cost of transaction initiation may correspond to response time, as each request to a service provider takes time. Moreover, when the request times out, as in the case of nonresponsive providers, this time can be quite significant. Transaction cost can also map to other measures of economic costs as well. Also, the cost of transaction failure can be tied to the probability that a client will stop using the service provider.

The work was successful in demonstrating the usefulness of environmental factors for informing trust decisions. The Modified Agent outperformed the Stock Agent in two out of three trials where their behaviors would be expected to be different. In these trials, the Stock Agent treated every instance of a lack of motivation (promised opinions not provided) or a lack of competence (opinions of low accuracy) in the same way, regardless of the costs incurred because of those failures in the other agents. The Modified Agent correctly took aggressive action against unmotivated/incompetent agents when environmental factors dictated it, and quickly marked as "untrustworthy" agents that were causing the Modified Agent to incur high costs due to those agents' failures. This demonstrates the usefulness of the technique.

In the third trial, inaccurate assumptions about the rules of the simulation resulted in the Stock Agent coming out the winner. The results of the first two trials lead us to believe that an adjustment of the simulation parameters, such that they more closely match the assumptions underlying the logic of the Modified Agent, would improve the Modified Agent's performance.

The first step in future work, then, is to bring the simulation in line with the original assumptions, and retest the Modified Agent. A positive result would provide very solid evidence of the usefulness of the technique, leading to more advanced work, such as:

- Fine-tuning the Modified Agent's trust algorithm to allow for a more granular segmentation of environmental factors. Move beyond High/ Normal/Low categorizations and towards a continuum of values that allow fine control over trust decisions
- Expand the algorithm to consider the third axis of the Complexity Model of Trust, Continuity. This aspect of trust is more complex than Competence or Motivation, and may not have an easy counterpart in the ART Testbed.
- Take a winning agent and attempt to apply this technique to that agent and thereby improve it. In the games of this work, the Ornery Agents were always the ultimate winners, though their Competence- and Motivation-based trust algorithms were the same as the Stock Agent. They would be a good starting point, however an even more complex, competition-quality agent would be a more interesting challenge.

V. REFERENCES

[1] P. Cofta, "Distrust," *Proceedings of the 8th International Conference on Electronic Commerce*, Fredericton, New Brunswick, Canada, 2006, pp. 250–258.

[2] M. deFreitas, "*Environmental Factors Affecting Trust Decisions in Multi-Agent Systems*," Technical Report, Department of Computer Science, University of Nebraska at Omaha, 2007.

[3] R. Falcone, C. Castelfranchi, "Trust Dynamics: How Trust is influenced by direct experiences and Trust itself," *Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems – Volume 2*, New York, New York, 2004, pp. 740-747.

[4] K. Fullam, T. Klos, G. Muller, et al., "A Specification of the Agent Reputation and Trust (ART) Testbed: Experimentation and Competition for Trust in Agent Societies," *Proceedings of the Fourth International Joint Conference on Autonomous Agents and Multiagent Systems*, Utrecht, The Netherlands, 2005, pp. 512-518.

[5] K. Fullam, K. S. Barber, "Learning Trust Strategies in Reputation Exchange Networks," *Proceedings of the Fifth International Joint Conference on Autonomous Agents and Multiagent Systems*, Hakodate, Hokkaido, Japan, 2006, pp. 1241-1248.

[6] K. Fullam, T. Klos, G. Miller, J. Sabater, et al., "The Agent Reputation and Trust (ART) Testbed Game Description (version 2.0)", http://www.lips.utexas.edu/art-testbed/pdf/SpecSummary.pdf (as of September 22, 2007).

[7] O. Kafali, P. Yolum, "Trust Strategies for ART Testbed," *The Workshop on Trust in Agent Societies at the Fifth International Joint Conference on Autonomous Agents and Multiagent Systems*, Hakodate, Hokkaido, Japan, 2006, pp. 43-49.

[8] J. Li, D. Ma, L. Li, H. Zhu. "AADSS: Agent-based Adaptive Dynamic Semantic Web Service Selection." *Proceedings of the 4th International Conference on Next Generation Web Services Practices*, 2008, pp. 83-89.

[9] E. M. Maximilien, M. P. Singh. "A Framework and Ontology for Dynamic Web Services Selection." *IEEE Internet Computing*, September-October, 2004, pp. 84-93.

[10] Z. Xu, P. Martin, W. Powley and F. Zulkernine. "Reputation-Enhanced QoS-based Web Services Discovery." *Proceedings of the IEEE International Conference on Web Services (ICSW 2007)*, pp.249-256, July, 2007.