Dealing with Trust and Reputation in Unreliable Multi-agent Trading Environments

Iraklis Tsekourakis and Andreas L. Symeonidis

Electrical and Computer Engineering Department, Aristotle University of Thessaloniki Thessaloniki, Greece
htsekourakis@gmail.com, asymeon@eng.auth.gr

Abstract. In shared competitive environments, where information comes from various sources, agents may interact with each other in a competitive manner in order to achieve their individual goals. Numerous research efforts exist, attempting to define protocols, rules and interfaces for agents to abide by and ensure trustworthy exchange of information. Auction environments and e-commerce platforms are such paradigms, where trust and reputation are vital factors determining agent strategy. And though the process is always secured with a number of safeguards, there is always the issue of unreliability. In this context, the Agent Reputation and Trust (ART) testbed has provided researchers with the ability to test different trust and reputation strategies, in various types of trust/reputation environments. Current work attempts to identify the most viable trust and reputation models stated in the literature, while it further elaborates on the issue by proposing a robust trust and reputation mechanism. This mechanism is incorporated in our agent, HerculAgent, and tested in a variety of environments against the top performing agents of the ART competition. The paper provides a thorough analysis of ART, presents HerculAgent s architecture and dis-cuss its performance.

1 Introduction

Agent Technology (AT) is constantly gaining ground in domains where continuous interaction is required. Software Agents may act in uncertain and dynamic environments, adapt illustrating various levels of autonomy and collaborate or compete in order to achieve their goals. Examples of such dynamic domains are Peer to Peer (P2P) networks, e-business and m-commerce solutions, autonomic and grid computing, as well as pervasive computing environments. [1]

It is more than obvious that interaction may entail malice, with agents (human or software) aiming to promote own interest while at the same time disserving others. In order to deal with this problem, the concepts of *trust* and *reputation* (T&R) are employed, providing agents with useful insight on which agents to trust and interact with.

Current work aims to analyze and discuss existing approaches on trust and reputation. Analysis is performed against the Agent Reputation and Trust testbed,

E. David et al. (Eds.): AMEC/TADA 2011, LNBIP 119, pp. 87–101, 2013.

a multi-parametric environment designed and developed for testing various trust strategies. Based on the analysis performed, a trust and reputation mechanism is developed and embedded into *HerculAgent* that is benchmarked against the top scoring agents of the ART competition. The paper is organized as follows: Section 2 discusses state-of-the-art on the available trust and reputation models, while Section 3 provides an overview of ART, discusses the winning T&R strategies and performs a preliminary analysis in order to identify the key factors affecting performance. Section 4 introduces the proposed T&R model and outlines the *HerculAgent* architecture. Finally, Section 5 discusses the performance of the agent on a set of experiments, while Section 6 proposes future directions and concludes the paper.

2 Trust and Reputation Models

There exists extensive literature related to trust and reputation, since it is strongly related to the application domain and the technologies used. Nevertheless, all approaches share a set of common factors, which are discussed within the context of this Section. Additionally, one should keep in mind that current work is focused on the T&R aspects in multi-agent trading environments, thus emphasis is given in that direction.

2.1 Specifying Trust and Reputation

Trust is the fundamental concern in open distributed systems. It lies at the core of all interactions between the entities that have to operate in uncertain and constantly changing environments. [2] In case of open multi-agent trading environments, trust pervades multi-agent interactions at all levels. In general, trust models are useful in spotting and marginalizing unreliable/malicious agents, in evaluating the outcome of an interaction and in leading to decisions on trustworthy agents to transact with.

Trust may be conceptualized in the following ways:

– **Individual-level trust**, whereby an agent has a set of beliefs about the honesty or reciprocate nature of the agents it interacts with;
– **System-level trust**, whereby agents operating in an environment are forced to be trustworthy by the rules of encounter (i.e. protocols and mechanisms) that regulate the system.

Should one discuss trust in the broader context where agents may act according to self-interest, and given that system-level trust mechanisms take a *de facto* approach on agent honesty, it is evident that individual-level trust issues are of great importance in contemporary trading environments. Research literature proposes three main approaches for specifying trust: the use of *A priori* evidence, the use of *Experienced* evidence, and the use of *Reputation*.

***A Priori* Evidence.** *A priori* evidence is evidence provided by specific protocols, policies and mechanisms which guarantee trust between participants. [3] In other words, when an agent acts following the rules that the protocols, policies, or mechanisms dictate ensures that this agent can be trusted.

***Experienced* Evidence.** As its name implies, experienced evidence is retrieved by agent interactions. This category is classified into two sub-categories: *direct* experience evidence [4] and *witness* evidence. [5]

Direct experience is the most relevant and reliable evidence source for trust management. It is the information an agent gains through the direct interactions with its partners. The trust reasoning efficiency of an agent is proportionate to the size of the interaction history saved by the agent. It is, though, disproportionate to the evidential effectiveness.

Witness evidence originates from the interactions of other agents in the community, which in turn may come from direct experience or witness evidence. Thus, the accuracy of evidence is strongly related to the source of evidence; due to its uncertainty, witness evidence is rarely exploited in existing trading environments.

Reputation Management. Reputation is the most exploited concept in trust management of multi-agent systems. Though the definition of reputation varies with respect to the context of the domain it is applied, one could argue that reputation is expressed as three levels of rating that may express the trustworthiness of an agent against other agents: *individual ratings, collective ratings* and the *rating transmissions*. In terms of rating, the techniques that manage the ratings could be divided to *rating retrieval* techniques, and *rating aggregation* techniques.

Rating retrieval is applied on distributed trading environments, where the topology is not known beforehand and network analysis techniques are employed in order to retrieve ratings.[6] [7] Having retrieved ratings, rating aggregation is performed in order to calculate reputation and define trust. In literature, rating aggregation may be performed in a number of manners, ranging from naive to more sophisticated ones.

2.2 Efficiency of Trust and Reputation Models

A successful trust and reputation model depends on the type of evidence the model provides to agents, the techniques used to get the above evidence, and the way an agent handles such evidence to extract trustworthiness for others. Efficiency of a T&R model is defined with respect to the following axes [8]:

- *Accuracy.* T&R models must provide good prediction on another agents future behavior. [9]
- *Adaptivity.* T&R models must be able to adapt in order to accommodate dynamic trustworthiness characteristics of other agents. [10]

- *Quick Convergence.* T&R modeling algorithms must quickly generate new models when unknown agents enter the system. [11]
- *Multidimensionality.* T&R models must differentiate between another agents varied trustworthiness characteristics across multiple categories. [12]
- *Efficiency.* T&R algorithms must generate models with minimal computational cost and in minimal time.

3 The ART Testbed

3.1 The ART Scenario and Architecture

As already discussed, a variety of approaches exist, aiming to model trust and reputation in multi-agent systems. The Agent Reputation and Trust (ART) testbed [13] provides an ideal framework for benchmarking different T&R strategies.

Within the context of ART, each agent represent an art appraiser, competing against all other agents (appraisers) in the system. Clients (handled by the ART server) request appraisals for paintings from different eras. In case an appraiser is an expert on paintings of the specific era, it is capable for providing an accurate appraisal, thus satisfying the client that will buy the painting and pay the appraiser. In case the agent is not an expert on paintings of the era, it may request paying a fee an evaluation (defined as opinion) by other appraisers. Appraisers may also transact with each other on reputation information on other appraisers. Based on their T&R strategy, agents must decide when and from whom to request opinions and reputation information, in order to generate accurate appraisals for clients. The more accurate the appraisals, the more the clients attracted and profit for the appraiser. Winner agent is declared the one with the highest bank account balance. Figure 1 illustrates the possible interactions and the type of information ex-changed between appraisers and clients. More information on ART can be found at [13].

The ART testbed comprises four basic modules [14]:

- *The Simulation Engine,* which is responsible for generat-ing controlled T&R environments by enforcing user-defined parameters.
- *The ART Database,* which stores all game information for reporting and later retrieval.
- *The ART GUI,* providing access to online game monitor-ing and result visualization.
- *The Agent Skeleton,* an agent wrapper for researchers to embed their T&R strategy, while ensuring unflustered communication with the other ART entities.

3.2 ART T&R Modeling and State-of-the-Art

Within the context of ART, an agent T&R strategy should span across three axes: (i) modeling of the other agents (environment), (ii) modeling request and,

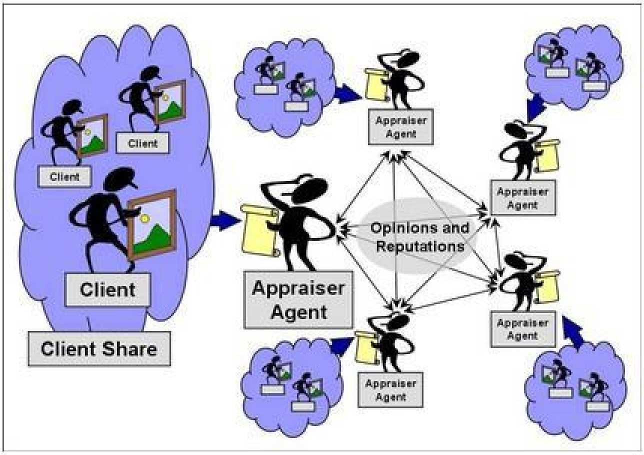

Fig. 1. ART entities and their interactions

(iii) modeling response. The response strategy directly affects its reputation, while the request and response strategies have an effect on environment modeling. [15]

The basic challenges in building an efficient T&R strategy have been identified by Costa et al. [2008] [16] and are: (i) the start game opinion requests, (ii) the identification of trustworthy agents, (iii) the appraisal definition policy and complexity, (iv) the reputation definition, request and response policies and, finally, the aggregation of all the above. Researchers may focus on one or more aspects of the strategy, in order to build an efficient agent.

May one go through related literature, one may identify several approaches that perform ART T&R modeling. Nevertheless, two are the ones that stand out: UNO and IAM.

UNO. Probably the most successful and efficient ART T&R model. Murillo and Munoz [17] focused on the request and response aspects of the T&R model. In both cases they exploit knowledge the agent builds on the other agents, based on:

- the error of own appraisals
- if an agent has responded to an UNO request and,
- the total number of requests an agent has made to UNO.

It is worth mentioning that UNO does not employ a reputation mechanism to extract trust values, since the UNO team considered that the number of participating agents is too small for generating trustworthy reputation estimations. Thus, they selected to work directly with the real appraisal error of each agent.

IAM. IAM is another successful ART agent paradigm. Teacy et al [18] describe IAM behavior as *Intelligent* (using statistic-al models for opponent modeling), *Abstemious* (spending its budget parsimoniously based on its trust model) and *Moral* (providing fair and honest feedback to those that request it). IAM decides based on the following information:

– Appraisal responses from the Simulator Engine and other appraisers
– Information on the behavior of other agents (e.g. reputation values).

The trust model of IAM comprises three parts: (a) the *lie detector*, which identifies malicious agents, (b) the *variation appraiser*, which estimates the variation of appraisal errors of the other agents and (c) the *weight estimator of the most accurate agents*. In contrast to UNO, IAM employs reputation in order to build its T&R model.

3.3 Preliminary Analysis of the ART Environment

In order to identify the factors that mostly affect appraiser performance, we performed an extensive set of experiments, where different game parameters and simple policies were tested. We employed the SimpleAgent (provided by the ART framework) and gradually tested it against the TestAgent (SimpleAgent equipped with a simple policy), the Cheatin-gAgent and HonestAgent, as well as the agents that participated in the 2008 ART competition. In the latter experiments, more elaborate policies where followed, based on the analysis performed. In all cases the game parameters were the same; these are defined in Table 1.

Table 1. Game parameters during experimentation

Time Epochs	50
Eras	10
Average number of clients per appraiser	20
Client fee	100
Appraisal cost	10
Certainty cost	1
Reputation cost	0.1
Appraisal messages	2
Certainty messages	20
Denial of personal opinion	True
Variable eras	2
Expertise value change	0.05

The types of policies investigated are discussed in Ta-ble 2, where the last column defines the number of discrete steps selected for each of the T&R factors. It should be mentioned that the following results are aggregates following monte carlo analysis.

Table 2. The various policies applied

Policy	T&R factor	Steps
Pol-1	Reputation definition policy	7
Pol-2	Reputation request policy (reputation only)	5
Pol-3	Reputation response policy	3
Pol-4	Honesty policy (based on agent trustworthiness)	3
Pol-5	Reputation request policy (reputation and certainty)	3
Pol-6	Appraisal cost for trustworthy agents	5
Pol-7	Appraisal cost for unreliable agents	5
Pol-8	Trustworthiness with respect to appraisal cost	3
Pol-9	Optimal reputation definition policy (Pol-1) Optimal trustworthiness policy (Pol-4) Honest response to trustworthy agents Dishonest response to unreliable agents	4
Pol-10	Optimal reputation definition policy (Pol-1) Honest response to trustworthy agents Dishonest response to unreliable agents Optimal appraisal cost policy (Pol-6)	3
Pol-11	Optimal reputation definition policy (Pol-1) Optimal reputation response policy (Pol-3) Optimal appraisal cost policy (Pol-6) Honest response to trustworthy agents Dishonest response to unreliable agents	3
Pol-12	Optimal reputation definition policy (Pol-1) Optimal reputation response policy (Pol-3) Optimal appraisal cost policy (Pol-6, Pol-7) Honest response to trustworthy agents Dishonest response to unreliable agents	3

TestAgent vs. SimpleAgent. An extensive set of tests was performed on the simple policies applied (Pol.1 Pol.8), in order to identify which of the T&R factors affect appraiser performance. Subfigures 2.1 2.8 illustrate the performance of the competing agents in various configurations (omitted due to space limitations). Block 1 denotes the set of basic rules identified and the optimal values of the most important factors:

BLOCK 1. BASIC RULES IDENTIFIED

```
Rule-1. IF ME > 0.5 THEN rep = rep-0.02 ELSE rep = rep+0.04
Rule-2. IF repAppraiser > 0.5 THEN it is considered trustworthy
Rule-3. IF an appraiser is trustworthy
        THEN provide accurate opinions
        ELSE provide falsified opinions
Rule-4. IF an appraiser is trustworthy
        THEN pay 0,7*AppCost to get an appraisal
        ELSE pay 0,15*AppCost
```

TestAgent vs. HonestAgent and CheatingAgent. The ART testbed provides two more agents for benchmarking: the *CheatingAgent* and *HonestAgent*. *TestAgent* equipped with the knowledge base generated during the first experimentation stage (Pol.1 Pol.8), was benchmarked against these agents. Subfigures 2.9 2.11 illustrate the performance of the competing agents in various configurations. Apart all other observations, one should also point out that in some cases, *SimpleAgent* stills outperforms *TestAgent*, since the strategy the latter follows is oriented towards more complex strategies.

TestAgent vs. ART 2008 Winning Agents. Finally, *TestAgent* was benchmarked against the top per-forming agents of the ART 2008 competition. Subfigure 2.12 illustrates the performance of the competing agents in various configurations. At this point *TestAgent* outperforms *SimpleAgent*, since the latter cannot cope with the complexity of the competitors strategies. Nevertheless, it is obvious that the approach *TestAgent* follows still lacks dynamicity and adjustability.

A number of useful observations were made through the analysis performed. First of all, computing reputation based on direct interactions with other agents proved more efficient than using information from reputation responses. Additionally, experimentation dictated that neither honesty, nor unreliability work alone. Competitors should be forced to play fair, so as to be rewarded with trust (or penalized with unreliability). Finally, the ranges of metric values were identified, where our agent increased efficiency. Outside of these ranges agent performance decreases. Based on these observations, *HerculAgent* was developed.

4 *HerculAgent* Architecture

HerculAgent follows a modular architecture, so as to meet each of the diverse needs imposed by the ART framework (Figure 3). Its behavior is expressed through nine strategy functions, which are aggregated into three behavior modules implementing three protocols: the *reputation protocol*, the *certainty protocol* and the *opinion protocol*.

4.1 *HerculAgent* Protocols

Reputation Protocol. The reputation protocol manages the reputation values Rep_ijt(i: $Agent_i$,, j: Era_j, t: $Epoch_t$), primarily based on previous direct interactions, and secondarily on indirect sources such as observation information. The *reputation* module actually deliberates on the reward or admonition strategy to follow, based on the appraisal estimates provided by other appraisers in the past. It also defines the number of reputation re-quests *HerculAgent* will make and the agents to request reputation information from. Finally, the *reputation* module determines the response strategy to reputation requests the agent receives from other agents.

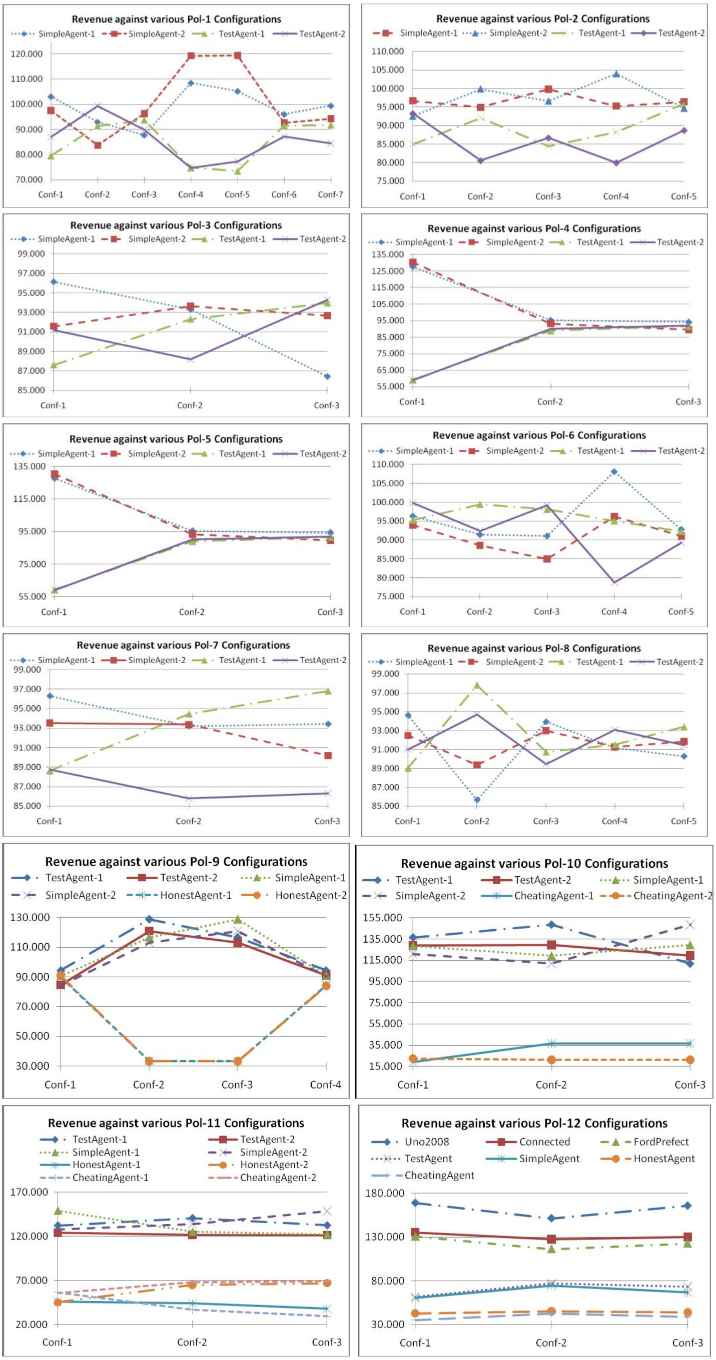

Fig. 2. Appraiser Revenue with respect to the various Policies applied and the configuration settings selected

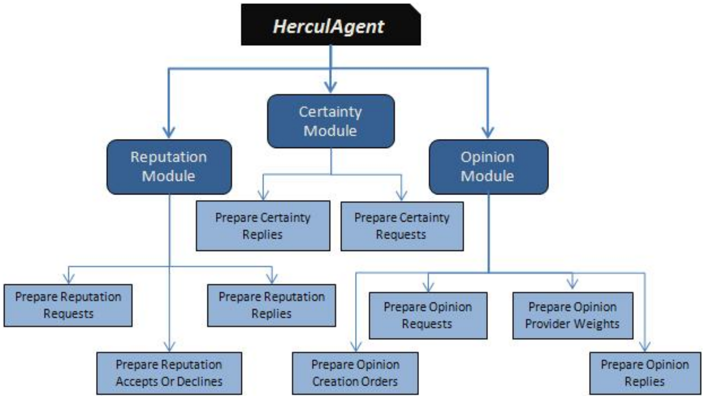

Fig. 3. The HerculAgent architecture

Certainty Protocol. The certainty protocol manages the certainty values Cer_ijt (i: $Agent_i$,, j: Era_j, t: $Epoch_t$) and refers to the confidence an agent has on the reputation of other appraisers. The *certainty* module implements the strategy *HerculAgent* has with respect to which agents to ask for their certainty values.

Opinion Protocol. Finally, the opinion protocol is the core business protocol of the agent, responsible for issuing accurate painting appraisals/opinions. The *opinion* module selects the agents to trust and request their opinion, while it also builds its own personal opinions base to respond to competition agents. It contemplates the strategy that the agent uses to define the combination of the other agents appraisals to create its own final appraisal and finally send it to the client.

4.2 *HerculAgent* Metrics

HerculAgent employs four metrics in order to design and develop its T&R strategy:

Mean error (ME(i,j,t)) is the weighted average of the appraisal relative errors an agent makes and varies for each era. ME depends on past interactions and is defined as:

$$ME(i,j,t) = \frac{((ME(i,j,t-1) * ErC(i,j,t-1) + Er(i,j,t-1)))}{(ErC(i,j,t-1)+1)} \quad (1)$$

where Er(i,j,t), is the relative appraisal error $Agent_i$, makes for a painting of Era_j at $Epoch_t$ of the game. ErC(i,j,t) is the respective number of the errors.

Reputation (Rep(i,j,t)) expresses the level of trustworthiness of $Agent_i$, for a painting of Era_j at $Epoch_t$ of the game. It ranges in the [0,1] interval.

Certainty (Cer(i,j,t)) indicates the certainty that an $Agent_i$ claims on its appraisal values for a painting of Era_j at epoch $Epoch_t$ of the game. It ranges in the [0,1] interval.

Self Confidence (SC(i,j,t)) expresses the certainty *HerculAgent* has on the other appraiser agents, as defined through past interactions. It denotes the possibility that the Rep(i,j,t) value computed for an appraiser is similar to its Cer(i,j,t) value. Block-2 denotes the pseudocode implementation of SC:

BLOCK 2. calculates() PSEUDOCODE

```
FUNCTION calculateSC(Agent i, Era j, Epoch t){
L1:  tempConf <- selfConfidence(Agenti(Appraisal(Eraj)));
L2:  IF (tempConf EQUALS 0) THEN tempConf = a
     ELSE tempConf = tempConf+((1-tempConf)*b);   {1}
L3:  selfConfidence(Agent_{i}(Appraisal(Era_{j}))) <- tempConf;
}
{1} After experimentation, a = 0.01, b = 0.005
```

Metrics are continuously calculated for each agent, era and epoch.

4.3 Dynamic Behavior Adaptation

HerculAgent employs two methods in order to adapt its behavior and strategy with respect to the data collected throughout the game:

setRepLimit() which adapts, for each $Agent_i$, for paintings of Era_j at each $Epoch_t$ the minimum Rep(i,j,t) value $Agent_i$ has to meet to be trusted.

setErrorsLimit() which defines, for each $Agent_i$, for paintings of Era_j at each $Epoch_t$ the error limit that is acceptable for an appraiser agent.

Block-3 denotes the pseudocode implementation of *setErrorsLimit()*. Function *setRepLimit()* is deployed in a similar manner.
BLOCK 3. setErrorsLimit() PSEUDOCODE

```
FUNCTION setErrorsLimit(Agent i, Era j, Epoch t){
L1:  FOR (1 TO numberOfEras){
L2:     min <- 1;
L3:     max <- 0;
L4:     FOR (1 TO numberOfAgents){
L5:         IF (errors(Agent_{i}, Era_{j}) < min
L6:         THEN min <- errors(Agent_{i}, Era_{j});
L7:         IF errors(Agent_{i}, Era_{j})> max
L8:         THEN max <- errors(Agent_{i}, Era_{j});
        }
L9:  ErrrorsLimit <- min + *max^2;  {1}
    }
}
{1} After experimentation,  = 0.2
```

4.4 *HerculAgent* Behavior

The core T&R model of *HerculAgent* focuses on trust, and the degree of trustworthiness we show to opponents. In order to decide, three behaviors were implemented: (i) the *typical* behavior (our basic strategy), (ii) the *optimistic behavior*, where opponents are given more credit and, (iii) the *pessimistic* behavior, where opponents are given less credit. In a similar manner, three behaviors were implemented in order to calculate the weighted average of the final appraisal: the *typical, aggressive*, and *submissive* behaviors. All behaviors are specified in the respective *HerculAgent.conf* file and behavior changes dynamically (upon game initiation).

5 Experiments

A number of experiments were performed with *HerculAgent* participating in all agent scenarios, as defined in Section 3. Various strategies were applied and interesting conclusions were drawn. The following results are aggregates following monte carlo analysis.

At first, *HerculAgent* was tested against the nave set of agents that were tested in the preliminary phase, and easily outperformed them. Figure 4 illustrates agent revenue (Bank balance) of an indicative game, as depicted by the ART Light Game Monitor Interface.

Consequently, *HerculAgent* was tested against the top performing agents of the ART 2008 competition. Figure 5 illustrates agent revenue (Bank balance) of an indicative game, while Figure 6 presents the aggregate results with respect to the different *HerculAgent* behaviors.

Through the numerous experiments performed in order to compare our strategy against the winning agents of the ART 2008 competition, we observe that our results are satisfactory but could be further improved. *HerculAgent* often succeeded in finished second third, nevertheless never succeeded in beating *Uno*.

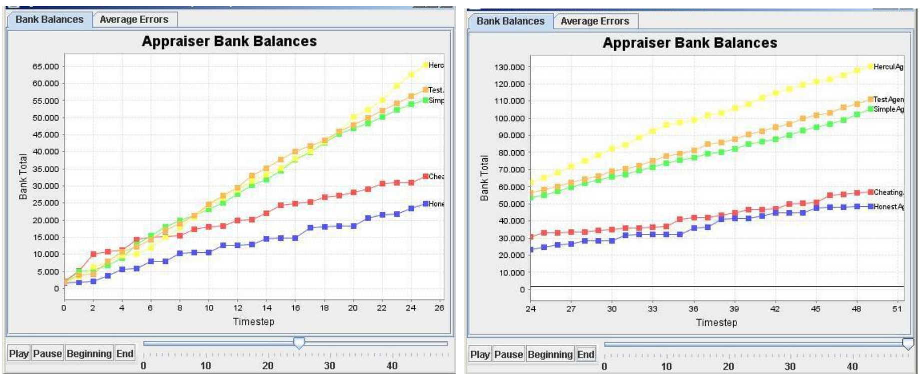

Fig. 4. HerculAgent against the naive set of agents

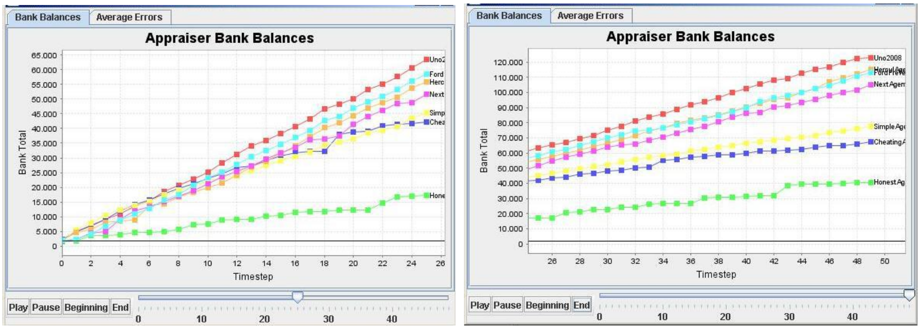

Fig. 5. HerculAgent against the ART 2008 top performing agents

The strategy of *HerculAgent* did not perform adequately at the first half of the game (start game effect). This can be justified based on the fact that *HerculAgent* computes trust mainly on information received through direct interactions. Thus, during the initial epochs there is not enough information available for reasoning. Whenever accurate initial appraisals are performed, the agent performs very well. In all cases, though, in the second half of the game, the bank total of *HerculAgent* improves significantly, at a rate even greater than the winner agent *Uno*.

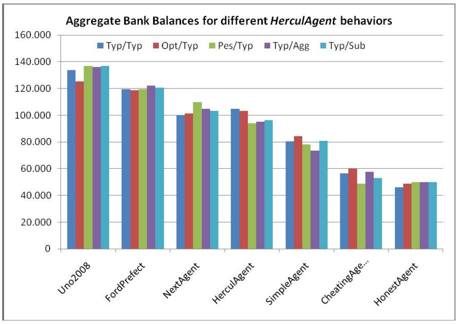

Fig. 6. Aggregate performance of HerculAgent against the ART 2008 top performing agents

6 Conclusions - Future Work

Current work discusses *HerculAgent*, an agent designed and developed for the analysis of various T&R models in dynamic trading environments. The ART platform was selected for experimentation, given that it provides a unique testbed for testing various game parameters. Preliminary analysis indicated the basic factors affecting performance, and a set of rules of thumb were identified, which where later embedded in our agent model.

Results show the strong points and drawbacks of *HerculAgent*.

Future enhancements include the development of an off-line mechanism that exploits regression techniques for estimating reputation values, as well as the improvement of the agent behavior during the first epochs of the game.

References

1. Simon, H.A.: The sciences of the artificial, 3rd edn. MIT Press, Cambridge (1996)
2. Ramchurn, S.D., Huynh, D., Jennings, N.R.: Trust in multi-agent systems. Knowl. Eng. Rev. 19, 1–25 (2004)
3. Huang, H., Zhu, G., Jin, S.: Revisiting trust and reputation in multi-agent systems. In: ISECS International Colloquium on Computing, Communication, Control and Management, vol. 1, pp. 424–429 (2008)
4. Ramchurn, S.D., Sierra, C., Godo, L., Jennings, N.R.: A computational trust model for multi-agent interactions based on confidence and reputation. In: Proceedings of 6th International Workshop of Deception, Fraud and Trust in Agent Societies, pp. 69–75 (2003)
5. Gómez, M., Carbó, J., Benac-Earle, C.: An Anticipatory Trust Model for Open Distributed Systems. In: Butz, M.V., Sigaud, O., Pezzulo, G., Baldassarre, G. (eds.) ABiALS 2006. LNCS (LNAI), vol. 4520, pp. 307–324. Springer, Heidelberg (2007)
6. Fullam, K.K., Barber, K.S.: Learning trust strategies in reputation exchange networks. In: Proceedings of the Fifth International Joint Conference on Autonomous Agents and Multiagent Systems, AAMAS 2006, pp. 1241–1248. ACM, New York (2006)
7. Sabater, J., Sierra, C.: Reputation and social network analysis in multi-agent systems. In: Proceedings of the First International Joint Conference on Autonomous Agents and Multiagent Systems: Part 1, AAMAS 2002, pp. 475–482. ACM, New York (2002)
8. Fullam, K.K., Klos, T.B., Muller, G., Sabater, J., Schlosser, A., Topol, Z., Barber, K.S., Rosenschein, J.S., Vercouter, L., Voss, M.: A specification of the agent reputation and trust (art) testbed: Experimentation and competition for trust in agent societies. In: The Fourth International Joint Conference on Autonomous Agents and Multiagent Systems, Utrecht, The Netherlands, pp. 512–518 (July 2005)
9. Fullam, K.K.: An expressive belief revision framework based on information valuation. Master's thesis, Dept. of EE, U. Texas (Austin) (2003)
10. Fullam, K.K., Barber, K.S.: A Temporal Policy for Trusting Information. In: Falcone, R., Barber, S.K., Sabater-Mir, J., Singh, M.P. (eds.) Trusting Agents. LNCS (LNAI), vol. 3577, pp. 75–94. Springer, Heidelberg (2005)
11. Ding, L., Kolari, P., Ganjugunte, S., Finin, T., Joshi, A.: Modeling and Evaluating Trust Network Inference. In: Seventh International Workshop on Trust in Agent Societies at AAMAS 2004 (July 2004)
12. Muller, G., Vercouter, L., Boissier, O.: Towards a general definition of trust and its application to openness in MAS. In: Falcone, R., Barber, K., Korba, L., Singh, M. (eds.) Proceedings of the Workshop on Deception, Fraud and Trust in Agent Societies at Autonomous Agents and Multi-Agent Systems, pp. 49–56 (July 2003)
13. ART testbed website: `http://www.ncbi.nlm.nih.gov/`

14. Fullam, K.K., Klos, T., Muller, G., Sabater-Mir, J., Barber, K.S., Vercouter, L.: The Agent Reputation and Trust (ART) Testbed. In: Stølen, K., Winsborough, W.H., Martinelli, F., Massacci, F. (eds.) iTrust 2006. LNCS, vol. 3986, pp. 439–442. Springer, Heidelberg (2006)
15. Kafalı, Ö., Yolum, P.: Trust strategies for ART Testbed. In: Ninth International Workshop on Trust in Agent Societies, AAMAS, pp. 43–49 (2006)
16. da Costa, A.D., de Lucena, C.J.P., da Silva, V.T., Azevedo, S.C., Soares, F.A.: Computing reputation in the art context: Agent design to handle negotiation challenges (2008)
17. Muñoz, V., Murillo, J.: Agent uno: Winner in the 2nd spanish art competition. Inteligencia Artificial, Revista Iberoamericana de Inteligencia Artificial 12(39), 19–27 (2008)
18. Teacy, W.T.L., Huynh, T.D., Dash, R.K., Jennings, N.R., Patel, J., Luck, M.: The art of iam: The winning strategy for the 2006 competition (2006)