The Agent Reputation and Trust (ART) Testbed Architecture

Karen K. Fullam [a], Tomas B. Klos [b], Guillaume Muller [c], Jordi Sabater-Mir [d], Zvi Topol [e], K. Suzanne Barber [a], Jeffrey Rosenschein [e] and Laurent Vercouter [c]

[a] *Laboratory for Intelligent Processes and Systems, University of Texas at Austin, USA*
[b] *Center for Mathematics and Computer Science (CWI), Amsterdam, The Netherlands*
[c] *École Nationale Supérieure des Mines, Saint-Étienne, France*
[d] *Institute of Cognitive Science and Technology, CNR, Rome, Italy*
[e] *MAS Research Group—Critical MAS, Hebrew University, Jerusalem, Israel*

Abstract. The Agent Reputation and Trust (ART) Testbed initiative has been launched with the goal of establishing a testbed for agent reputation- and trust-related technologies. This testbed serves in two roles: (1) as a competition forum in which researchers can compare their technologies against objective metrics, and (2) as a suite of tools with flexible parameters, allowing researchers to perform customizable, easily-repeatable experiments. In the testbed's art appraisal domain, agents, who valuate paintings for clients, may gather opinions from other agents to produce accurate appraisals. This paper first gives a brief overview of the testbed domain problem to orient the reader to the game rules. A discussion of the ART Testbed implementation architecture is presented, explaining the functionality of the testbed's Game Server, Simulation Engine, Database, User Interfaces, and Agent Skeleton.

Keywords. Computational trust and reputation models, Multi-agent systems, Testbed,

1. Introduction

The Agent Reputation and Trust (ART) Testbed [12] initiative has been launched with the goal of establishing a testbed for agent reputation- and trust-related technologies. This international team of eight researchers from five countries has been formed to coordinate domain design, game specification, testbed development, and competition administration. The ART Testbed is designed to serve in two roles: (1) as a competition forum in which researchers can compare their technologies against objective metrics, and (2) as a suite of tools with flexible parameters, allowing researchers to perform customizable, easily-repeatable experiments.

A diverse collection of trust-modeling algorithms has been developed in recent years [3,8,10,11,13], resulting in significant breadth-wise growth. However, a unified research direction has yet to be established. Many experimental domains and metrics have been utilized, but unified performance benchmarks for comparing technologies, in spite of representational differences, have been neglected. In recent years, researchers [1,5,9] have recognized objective standards are necessary to justify successful trust modeling

systems and provide a baseline of certifiable strategies for future work. For trust algorithms and representations to crossover into application [2, 4], the public must be provided with system evaluations based on transparent, recognizable standards for measuring success. As a versatile, universal experimentation site, the ART Testbed will foster a cohesive exploration of trust research problems; researchers will be united toward a common challenge, out of which can come solutions to these problems via unified experimentation methods. Through objective, well-defined metrics, the testbed will provide researchers with tools for comparing and validating their approaches. The testbed will also serve as an objective means of presenting technology features—both advantages and disadvantages—to the research community. In addition, the ART Testbed will place trust research in the public spotlight, improving confidence in the technology and highlighting relevant applications.

In the testbed's art appraisal domain, agents, who valuate paintings for clients, may gather opinions from other agents to produce accurate appraisals. This paper first gives a brief overview of the testbed domain problem in Section 2 to orient the reader to the game rules. Then, in Section 3, a discussion of the ART Testbed implementation architecture is presented, explaining the functionality of the testbed's Game Server, Simulation Engine, Database, User Interfaces, and Agent Skeleton. Finally, Section 4 concludes, presenting plans for future work.

2. Testbed Domain Problem

The testbed operates in two modes: competition and experimentation. In competition mode, the testbed compares different researchers' strategies as they act in combination. Each participating researcher controls a single agent, which works in competition against every other agent in the system. Competition organizers can change parameters to permit the game structure to be adapted for subsequent competitions. To utilize the testbed's experimentation mode, the complete testbed will be downloadable for researcher use independent of the competition. Thus, results may be compared among researchers for benchmarking purposes, since the testbed provides a well-established environment for easily-repeatable experimentation. In experimentation mode, the researcher has the flexibility of complete control over all experiment parameters.

In the art appraisal domain, agents function as painting appraisers with varying levels of expertise in different artistic eras (e.g. classical, impressionist, postmodern). Clients request appraisals for paintings from different eras; if an appraiser does not have the expertise to complete the appraisal, it can purchase opinions from other appraisers. Other appraisers estimate the accuracy of opinions they send by the cost they choose to invest in generating an opinion; opinion providers may lie about the estimated accuracy of their opinions. Appraisers produce appraisals using opinions they receive from other appraisers; they receive more clients, and thus more profit, for producing more accurate appraisals. Appraisers may also purchase reputation information from one another, which helps appraisers know from which other appraisers to request opinions. More detail about the art appraisal domain rules can be found in the ART Testbed Competition Rules [6]. In [7], the authors detail today's most prominent trust research objectives, solutions to which the testbed must facilitate, and present a structured defense of the chosen art appraisal domain by delineating how the domain problem addresses each of the prominent research objectives.

Appraiser strategies are analyzed in terms of both agent- and system-based metrics. The agent perspective examines the utility of a strategy to a single appraiser without regard for the benefit to the overall agent system. In the testbed's appraisal scenario, appraisers attempt to accurately valuate their assigned paintings; their decisions about which opinion providers to trust directly impact the accuracy of their final appraisals. Therefore, in competition mode, the winning agent is selected as the appraiser with the highest bank account balance. In other words, the appraiser who is able to (1) estimate the value of its paintings most accurately and (2) purchase information most prudently, is deemed most successful. The testbed also provides functionality to compute the average accuracy of the appraiser's final appraisals and the consistency of that accuracy, represented as its final appraisal error mean and standard deviation, respectively. In addition, the quantities of each type of message passed between appraisers are recorded.

The system perspective employs metrics that emphasize social welfare, or benefit to the appraiser network as a whole. The testbed collects data on system metrics such as distribution of earnings among appraisers, number of messages passed (by type), number of transactions, and transaction distribution across appraisers. Finally, the testbed provides methods allowing researchers to define additional metrics and collect relevant data.

3. Implementation Architecture

As shown in Figure 1, the testbed architecture, implemented in Java, consists of five components:

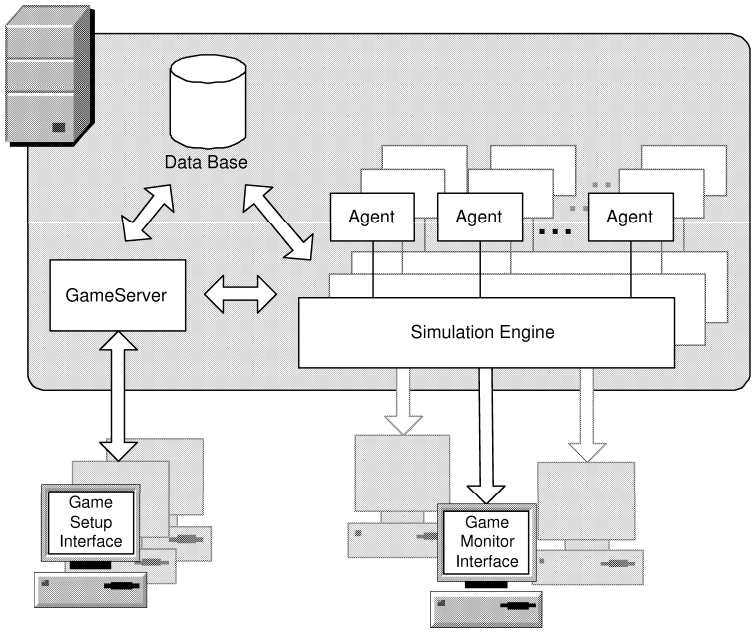

Figure 1. ART Testbed architecture.

1. Game Server, which allows setting up and running games (Section 3.1),
2. Simulation Engine (one per game), which controls the simulation environment (Section 3.2),
3. Agent Skeleton, designed to assist players in implementing their strategy code (Section 3.3),
4. Database, which stores data for Simulation Engine calculations and experiment analysis (Section 3.4), and
5. User Interfaces (Game Setup Interface and Game Monitor Interface), through which games are set up and viewed (Section 3.5).

In competition mode, the Game Server, Simulation Engines, and Database run on a central machine maintained by the ART Testbed Team, while the User Interfaces are locally accessible to players and observers. In experimentation mode, the researcher may run the Game Server, Simulation Engines, and Database locally. The following subsections provide a more detailed descriptions of each module.

3.1. Game Server

The Game Server manages the scheduling and starting of all games. A game is initiated by specifying parameters for new game through the Game Setup Interface (Section 3.5), either by the ART Testbed Team (in competition mode) or by the researcher (in experimentation mode); the Game Server records these game parameters in the Database and initiates a Simulation Engine (Section 3.2) for each new game. A properties file, input as the Game Server is instantiated, determines which game parameters are considered 'fixed' and which may be defined by the user when initiating a game. The Game Server also registers Agents (Section 3.3) for games, recording agent data in the Database as well.

3.2. Simulation Engine

A Simulation Engine is responsible for controlling a single game's simulation environment by enforcing chosen parameters. The Simulation Engine also assigns clients to appraisers and coordinates communication among appraisers. In each period, the Simulation Engine manages allocation of clients to appraisers, the conducting of opinion and reputation transactions, and calculation of final appraisals.

Specifically, a single game proceeds as a series of periods. In each period, the Simulation Engine coordinates the following actions in sequence:

1. At the start of each period, the Simulation Engine assigns clients (with paintings to be appraised) to each appraiser. Appraisers receive larger shares of clients if they have produced more accurate appraisals in the past, to simulate clients' preference for accurate appraisers. The Simulation Engine sends appropriate client appraisal request notifications to assigned appraisers to simulate clients requesting appraisals. To simulate the payment of up-front appraisal fees by clients, the Simulation Engine deposits fees into apppropriate appraiser bank accounts.
2. Next, the Simulation Engine facilitates reputation transactions among appraisers. Appraisers may conduct reputation transactions about third-party appraisers' expertise in given eras to decide from which appraisers to purchase appraisal

opinions. If a potential reputation provider chooses to accept an appraiser's reputation request, the requesting appraiser sends payment and the provider sends the reputation, which need not be truthful. Alternatively, the provider may reject the request, and the transaction is aborted. The Simulation Engine serves as a go-between for agent communication, handling the passing of transaction messages. In addition, the Simulation Engine handles payments between appraisers by withdrawing from requester bank accounts and depositing into provider bank accounts.

3. The Simulation Engine then faciliates opinion transactions among appraisers. When an appraiser requests an opinion, the potential provider may reject the request (thus the transaction is aborted), or send a (possible untruthful) certainty assessment about the opinion it claims it will send. In the latter case, the requester may either decline the potential opinion (and abort the transaction) or send payment. Upon receipt of payment, the provider sends the opinion, which need not be truthful. Again, the Simulation Engine handles the passing of transaction messages and payment withdrawals and deposits.
4. Finally, the Simulation Engine calculates the appraiser's final appraisal as a weighted average of the opinions the appraiser purchases. Opinion weights are real values between zero and one that an appraiser assigns, based on its trust model, to another's opinion. The Simulation Engine, not the appraiser itself, performs the appraisal calculation to ensure that the appraiser does not employ strategies for selecting opinions to use after receiving all purchased opinions.

While conducting each of the above tasks, the Simulation Engine passes all relevant data to the Database for recordkeeping, as described below in Section 3.4. The data is then available for retrieval to complete necessary calculations.

3.3. Agent Skeleton

The Agent Skeleton is designed to allow researchers to implant within their appraiser agent-customized trust representations and algorithms while permitting standardized communication protocols with other entities. All appraiser agents participating in the ART Testbed are descendants of the same abstract class `Agent`. This class defines a set of abstract methods to be coded by the researcher to define the behavior of his/her appraiser agent, as well as a set of methods to facilitate the communication with other appraiser agents. The `Agent` class also provides methods for interacting with the Simulation Engine (for tasks such as verifying bank balances).

3.4. Database

The Database stores all information about the games. First, it stores information about the participants (identification and classes provided) and the parameters for the current games. Then, during the course of these games, the Database collects, through the Simulation Engine, environment and appraiser data. These data are stored in a MySQL database and include:

1. For each appraiser for each timestep:
   - client share,

- fees paid to or from the agent (for reputations, opinions, or client appraisals),
- bank account balance,
- reputation weights associated with other appraisers,
- generated opinions,
- calculated final appraisals.

2. For the environment for each timestep:

   - true painting values,
   - all messages exchanged between appraisers.

In addition, the testbed is designed to provide researchers with the functionality to log additional data types in the Database during experimentation mode. With access tools for navigating Database logs, data sets are made available to researchers after each game session for game re-creation and experimental analysis. Information stored in the Database is extracted by the Simulation Engine to populate the User Interfaces, as detailed below in Section 3.5.

3.5. User Interfaces

User Interfaces permit the creation of games (according to user-specified parameters) and the joining of agents to games. The User Interfaces also allow observation of games in progress and access to information collected in the Database by graphically displaying details such as transactions between appraisers, accuracy of appraisers' final appraisals, and money earned by each appraiser. In competition mode, the ART Testbed Team (competition organizers) initiate official competition games with pre-specified parameters, while competition participants may join and view those official games. In experimentation mode, researchers may initiate games (with the parameters of their choosing), join games, and view games according to experiment needs.

To interact with the testbed, the user first launches the Game Setup Interface (GSI). This interface allows the user to initiate a new game, to join a game that has already been initiated (perhaps by another user) and to view any initiated game as a guest by simply selecting the game to view. By entering a login and password, a user is granted permission to view additional game results related to the agents owned by the user. From the GSI, the user can also launch a Game Monitor Interface and save the data of a completed game, including private data of agents owned by the user, as an XML file.

The Game Monitor Interface, shown in Figure 2, consists of a topology panel on the left, displaying the entire agent system, and a set of detail panels on the right. The user can view a detail panel of in-depth statistics for each agent owned by the user. In addition, a `System Data` detail panel displays additional system-wide statistics. The left-hand topology panel shows the current client share (proportionally represented by the number of client figures in the corresponding blue oval) and bank balance of each agent, updated after each simulation period. Arrows show the direction and quantity of each type of message passed between agents, including opinion requests, opinions, reputation requests, and reputations. Arrow thickness is proportional to message volume.

An agent's detail panel shows numerous statistics to assist the user in gauging the agent's performance. Among others, the user has available a `Bank Balance` chart (that measures the agent's total bank balance after each period), a `Transaction Fee` chart (that displays fees the agent earns or pays in each period such as appraisal fees earned

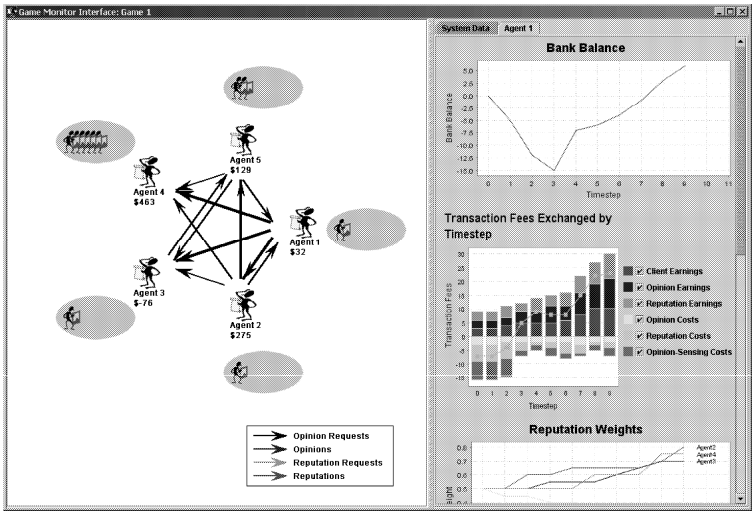

Figure 2. Game Monitor Interface.

from clients and opinion costs paid to opinion providers) or a `Reputation Weights` chart to see reputation weights the agent has estimated for each other agent in the system.

Finally, a `System Data` detail panel provides statistics about all agents in the system. The user can compare bank balances and average appraisal errors among all agents or view statistics comparing the number of completed transactions (both opinion and reputation transactions) between each pair of agents in the system.

4. Conclusions

This paper presents the implementation architecture for the ART Testbed, which provides researchers with easy access to a common experimentation environment and allows researchers to compete against one another to determine the most viable technology solutions. The testbed provides functionality through five modules—Game Server, Simulation Engine, Database, User Interfaces, and Agent Skeleton—to promote easy experimentation and agent development by participating researchers.

The testbed team plans to demonstrate the testbed in July of 2005, releasing a beta version of the testbed shortly after. Based on prototype testbed experimental review and feedback from the research community, the first testbed competition will be conducted in July of 2006. Development progress can be monitored through the ART Testbed website [12], where updates to testbed development are posted periodically. The website also hosts a discussion group through which the trust research community can provide feedback.

Acknowledgment

Jordi Sabater enjoys a Sixth Framework Programme Marie Curie Intra-European fellowship, contract No. MEIF-CT-2003-500573. Tomas Klos is working on the CIM-project

on Cybernetic Incident Management, sponsored by the Dutch government (SENTER) under project number TSIT2021. Research by Karen Fullam is sponsored in part by the Defense Advanced Research Project Agency (DARPA) Taskable Agent Software Kit (TASK) program, F30602-00-2-0588.

References

[1] K. Barber, K. Fullam, and J. Kim. Challenges for Trust, Fraud, and Deception Research in Multi-agent Systems. In *Trust, Reputation, and Security: Theories and Practice*, volume 2631 of *LNCS*, pages 8–14. Springer, 2003.

[2] BizRate. *BizRate*. http://www.bizrate.com, 2002.

[3] J. Carbo, J. Molina, and J. Davila. Comparing predictions of SPORAS vs. a Fuzzy Reputation Agent System. In *3rd International Conference on Fuzzy Sets and Fuzzy Systems*, pages 147–153, 2002.

[4] eBay. *eBay*. http://www.eBay.com, 2002.

[5] K. Fullam and K. S. Barber. Evaluating Approaches for Trust and Reputation Research: Exploring a Competition Testbed. In M. Paolucci, J. Sabater, R. Conte, and C. Sierra, editors, *Proc. IAT Workshop on Reputation in Agent Societies*, pages 20–23, 2004.

[6] K. Fullam, T. Klos, G. Muller, J. Sabater, A. Schlosser, Z. Topol, K. S. Barber, J. S. Rosenschein, L. Vercouter, and M. Voss. The Agent Reputation and Trust (ART) Testbed Competition Game Rules (version 1.0). Technical report, The University of Texas at Austin TR2004-UT-LIPS-028, 2004.

[7] K. Fullam, T. Klos, G. Muller, J. Sabater, A. Schlosser, Z. Topol, K. S. Barber, J. S. Rosenschein, L. Vercouter, and M. Voss. A Specification of the Agent Reputation and Trust (ART) Testbed. In *Proc. AAMAS*, 2005.

[8] D. Huynh, N. Jennings, and N. Shadbolt. Developing an Integrated Trust and Reputation Model for Open Multi-Agent Systems. In *Proceedings of The Workshop on Trust in Agent Societies at AAMAS-2004, New York, USA*, pages 65–74, 2004.

[9] J. Sabater. Toward a Test-Bed for Trust and Reputation Models. In R. Falcone, K. Barber, J. Sabater, and M. Singh, editors, *Proc. of the AAMAS-2004 Workshop on Trust in Agent Societies*, pages 101–105, 2004.

[10] J. Sabater and C. Sierra. Reputation and Social Network Analysis in Multi-Agent Systems. In *Proceedings of the first international joint conference on autonomous agents and multiagent systems (AAMAS-02), Bologna, Italy*, pages 475–482, 2002.

[11] M. Schillo, P. Funk, and M. Rovatsos. Using Trust for Detecting Deceitful Agents in Artificial Societites. *Applied Artificial Intelligence*, Special Issue on Trust, Deception and Fraud in Agent Societies:825–848, 2000.

[12] ART Testbed Team. *Agent Reputation and Trust Testbed*. http://www.lips.utexas.edu/~kfullam/competition/, 2005.

[13] B. Yu and M. P. Singh. An Evidential Model of Distributed Reputation Management. pages 294–301.