Computational Intelligence

An International Journal

Computational Intelligence, Volume 28, Number 4, 2012

DART: A DISTRIBUTED ANALYSIS OF REPUTATION AND TRUST FRAMEWORK

Amirali Salehi-Abari[1] and Tony White[2]

[1]*Department of Computer Science, University of Toronto, Toronto, Ontario, Canada*
[2]*School of Computer Science, Carleton University, Ottawa, Canada*

Artificial societies—distributed systems of autonomous agents—are becoming increasingly important in open distributed environments, especially in e-commerce. Agents require trust and reputation concepts to identify communities of agents with which to interact reliably. We have noted in real environments that adversaries tend to focus on exploitation of the trust and reputation model. These vulnerabilities reinforce the need for new evaluation criteria for trust and reputation models called exploitation resistance which reflects the ability of a trust model to be unaffected by agents who try to manipulate the trust model. To examine whether a given trust and reputation model is exploitation-resistant, the researchers require a flexible, easy-to-use, and general framework. This framework should provide the facility to specify heterogeneous agents with different trust models and behaviors.

This paper introduces a Distributed Analysis of Reputation and Trust (DART) framework. The environment of DART is decentralized and game-theoretic. Not only is the proposed environment model compatible with the characteristics of open distributed systems, but it also allows agents to have different types of interactions in this environment model. Besides direct, witness, and introduction interactions, agents in our environment model can have a type of interaction called a reporting interaction, which represents a decentralized reporting mechanism in distributed environments. The proposed environment model provides various metrics at both micro and macro levels for analyzing the implemented trust and reputation models. Using DART, researchers have empirically demonstrated the vulnerability of well-known trust models against both individual and group attacks.

Received 11 September 2009; Revised 13 May 2010; Accepted 18 November 2010; Published online 9 August 2012

Key words: multiagent systems, trust, reputation.

1. INTRODUCTION

Recently, many computer applications are open distributed systems in which the components are located on a large-scale network. These systems are decentralized and subject to change over the system's lifetime. E-business systems, peer-to-peer systems (Oram 2001), Web services (Mcilraith, Son, and Zeng 2001), the Semantic Web (Berners-Lee, Hendler, and Lassila 2001), and pervasive computing (Schmeck, Ungerer, and Wolf 2002) fall into the category of open distributed systems. With the growth of these open distributed systems through the Internet, artificial societies have been formed in these environments. Furthermore, communities of intelligent agents, which may eventually interact on the behalf of their users in e-commerce marketplaces (Chavez and Maes 1996), can be viewed as artificial societies.

The dynamics and evolution of these artificial societies are driven by information exchange and the actions that are based upon it. While cryptographic mechanisms can assist in assuring data integrity and authenticity by ensuring reliable data transfer they do little to prevent the creation of incorrect information. Although authentication protocols can associate the information with a particular entity, mechanisms beyond these are required to assess the correctness of the acquired information. While explicit verification of acquired information is a potential solution, this is often infeasible in real systems. Furthermore, subjective assessment of information by an agent is often required, making absolute correctness an indeterminate problem. It is the position of the paper that trust model and authentication components are two separate components of trust and reputation systems and the attacks on them should be analyzed separately. This is a commonly held view.

The focus of this paper is the assessment of information (or more generally service provision) function that is attributed to trust and reputation models. This assessment can be

used to decide upon the nature and frequency of interactions with other agents within an artificial society. According to Jarvenpaa, Tractinsky, and Vitale (2000), trust is an essential aspect of any relationship in which the trustor does not have direct control over the actions of a trustee, the decision is important, and the environment is uncertain. Trust management systems (Weeks 2001), employed as access control mechanisms to determine whether or not a request should be allowed, are not the focus of this paper. The scope of this paper is limited to decentralized computational trust and reputation models.

Trust and reputation have been studied and used in various fields from different perspectives. For instance, Nowak and Sigmund (1998) have explained why selfish individuals cooperate by using reputation concepts. The concept of trust in economics and business was discussed first by Akerlof (1970) when he introduced "the market of lemons problem." He identified certain severe problems of markets characterized by asymmetrical information. Economists have used trust and reputation to explain *irrational* behavior of players in repeated economic games (Kreps and Wilson 1982; Marimon, Nicolini, and Teles 2000). Computer scientists have used trust and reputation for modeling trustworthiness of entities and individuals in open distributed systems (e.g., online marketplaces, multiagent systems (MAS), and peer-to-peer systems) (Mui, Mohtashemi, and Halberstadt 2002b; Ramchurn, Huynh, and Jennings 2004). This paper concentrates on trust and reputation models for open distributed systems. Moreover, it is the view of this paper that complicated trust and reputation models are not universally implemented because of the possibility of exploitation of those models.

As reputation and trust have recently received considerable attention in different domains such as distributed artificial intelligence, computational economics, evolutionary biology, psychology, and sociology, there are many diverse definitions of trust and reputation available in these domains (Falcone and Castelfranchi 2001). Herein, we concentrate on several definitions of trust and reputation that have appeared mostly in e-commerce and computer science literature. Mui, Mohtashemi, and Halberstadt (2002a) define trust as "a subjective expectation an agent has about another's future behavior based on the history of their encounters." Grandison and Sloman (2000) state that trust is "the firm belief in the competence of an entity to act dependably, securely and reliably within a specified context." According to Dasgupta (2000), "Trust is a belief an agent has that the other party will do what it says it will or reciprocate, given an opportunity to defect to get higher payoffs." Gambetta (1988) defined trust to be "a particular level of subjective probability with which an agent assesses that another agent will perform a particular action, both before the assessing agent can monitor such an action and in a context in which it affects the assessing agents own action." According to Olmedilla et al. (2005), "Trust of party A to a party B for a service X is the measurable belief of A in that B behaves dependably for a specified period within a specified context (in relation to service X)." This paper adopts the term *trust* as defined by Mui et al. (2002a).

While trust definitions focus more on the history of agents' encounters and their beliefs, reputation is based on the aggregated information from other individuals. Sabater and Sierra (2001) declared that "Reputation can be defined as the opinion or view of someone about something" whereas reputation is the consequence of "word-of-mouth recommendations" from the perspective of Abdul-Rahman and Hailes (2000). Castelfranchi, Falcone, and Pezzulo (2003) consider reputation as a component of trust. This paper adopts the term *reputation* as defined by Sabater and Sierra (2001).

Amazon (http://www.amazon.com) and eBay (http://www.ebay.com) are important practical examples of reputation management systems. In these systems, the sellers list their items for sale and buyers bid for these items. Users are allowed to rate sellers and submit textual comments. The overall reputation of a seller is the average of the ratings obtained

from his customers. Several researchers have postulated that seller reputation has significant influence on prices, especially for high-valued products in the eBay market (Resnick and Zeckhauser 2002; Houser and Wooders 2006). Similarly, Brainov and Sandholm (1999) have studied the impact of trust on contracting in e-commerce marketplaces. Their approach shows the amount of trade and agents' utility functions are maximized when the seller's trust is equal to the buyer's trustworthiness. Moreover, they show that advanced payment contracts can eliminate inefficiency caused by asymmetric information about trust and improve the trustworthiness between sellers and buyers. These studies all imply the importance of trust and reputation models in open distributed systems, especially e-commerce marketplaces.

Similar to eBay, one approach to building a trust or reputation model is to have a central agency that keeps records of the recent activity of the users in the system, very much like the scoring systems of credit history agencies (e.g., Kasbah; Chavez and Maes (1996)). However, these centralized approaches require considerable overhead on behalf of the providers of the online community and failure of the agency causes failure in all parts of the system. Moreover, they are not compatible with most of the characteristics and limitations of open distributed system. Generally, since there is no central authority in a pure open distributed system, a centralized trust model is not suitable for these systems. For example, there is no trusted introducer service as is found in some distributed systems. That is why the scope of this paper is decentralized trust and reputation models.

The majority of open distributed computer systems can be modeled as MAS in which each component acts autonomously to achieve its objectives (Jennings 2001). An important class of these systems is one that is *open* in terms of joining and leaving the system. Huynh, Jennings, and Shadbolt (2006) pointed out three interesting features of these systems: (1) The agents are likely to be self-interested and may be unreliable; (2) No agent can know everything about its environment. In other words, there is no global perspective; and (3) No central authority can control all the agents due to different ownership. A key component of these open MAS is the interactions that certainly have to take place between agents. Furthermore, because agents have incomplete knowledge about their environments and other agents, trust and reputation plays crucial roles in these interactions.

To reach its goals, an agent usually requires resources that only other agents can provide. The agent benefits from choosing the agents with which it interacts such that they can provide those resources. In this light, the agent can minimize the risk of unsuccessful interactions and failure by predicting the outcome of interactions, and avoiding risky (unreliable) agents. Modeling the trustworthiness of the potential interaction partners enables the agent to make these predictions. Furthermore, analogous to the legal systems in which a rule violation generates a legal punishment for the offender, in the social world the penalty for a violator who violates a social norm is a bad reputation (Castelfranchi, Conte, and Paolucci 1998). Specifically, trust and reputation provide a form of social control in open distributed systems in which agents are likely to interact with unvisited and unknown agents.

Sabater and Sierra (2005) categorized computational trust and reputation models based on various intrinsic features. From their perspective, a trust and reputation model can be cognitive or game-theoretical in terms of its conceptual model. A cognitive model works based on beliefs and the mental states of individuals as opposed to game-theoretical models that rely on the result of pragmatic games and numerical aggregation of past interactions; the latter is used in this paper. Trust and reputation models might use different sources of information such as direct experiences, witness information, sociological information and prejudice. Witness information is the information that comes from other members of the community whereas sociological information is extracted from the social relations

between individuals and their roles in the community. Prejudice is connected to identifying characteristics of individuals (e.g., skin color or religious beliefs) and, being cognitive, is beyond the scope of the model proposed in this paper. Trust and reputation of an individual can be seen either as a global property available to all members of a society (centralized models) or as a subjective property assessed by each individual (decentralized models). Trust and reputation models vary in terms of individual behavior assumptions; in some models, cheating behaviors and malicious individuals are not considered at all whereas in others possible cheating behaviors are taken into account. There are many computational models of trust, a review of which can be found in Ramchurn et al. (2004) and Sabater and Sierra (2005).

In the other categorization of trust models presented by Ramchurn et al. (2004), trust models are classified into two main groups: Individual-level trust models and System-level trust models. In the former, an individual has some beliefs and understanding about the honesty of its interaction partners and the agent based on these beliefs will act. In the latter, the actors in the system have to be trustworthy by the rules defined in the system. There are three subcategories for individual-level trust models: (1) *Evolving and learning strategies*: The agents will learn about the other agents over a number of encounters and interactions; (2) *Reputation Models*: The agent can reason about the other agent based on the information gathered from the environment, especially by asking other agents; and (3) *Sociocognitive models of trust*: The agent can characterize the known motivations of the other agents. The model proposed here is individually based, and learns.

The main utility of trust and reputation models can be summarized as minimizing the risk associated with interacting with others. To reach this goal, an agent must be able to model trustworthiness of potential interaction partners and make decisions based on those models that assist agents in isolating the untrustworthy (undesirable and unreliable) agents from the society (Yu and Singh 2003).

Fullam et al. (2005) has defined the following set of criteria to evaluate trust and reputation models: (1) the model should be multidimensional; (2) converge quickly; (3) precisely model the agent's behavior; (4) be adaptive: the trust value should be adapted if the target's behavior changes; (5) be efficient in terms of computation. There is little consideration regarding the possibility that agents may attempt to exploit the trust and reputation model itself. In this regard, we believe that in addition to the criteria explained earlier, *exploitation resistance* is a crucial feature of trust models (Salehi-Abari and White 2009b). Exploitation resistance reflects the ability of a trust model to be impervious to agents who try to manipulate the trust model and who aim to abuse the presumption of trust. More precisely, exploitation resistance implies that adversaries cannot take advantage of the trust model and its associated systems parameters even when they are known or partially known to adversaries. It should be noted that these requirements do not cover authentication and authorization. Authentication and authorization are considered beyond the scope of this paper and the proposed framework.

To understand whether a given trust and reputation model is exploitation-resistant or design exploitation-resistant trust models, researchers require a framework (testbed) which provides the facility to specify heterogeneous agents employing various trust models and is flexible enough to accommodate a variety of adversarial behaviors (exploitation). This requirement for the framework motivates us to design and introduce a Decentralized Analysis of Reputation and Trust (DART) framework. By employing game-theoretical concepts and notions, DART is an abstract model and not restricted to any specific domain (e.g., e-commerce or peer-to-peer systems). Specially, DART is structured in a way that different trust models can be implemented and tested against different exploitations. Information assessment—a core facility of a trust and reputation model—is reduced to an assignment of cooperation or defection to an interaction.

Our contributions presented in this paper include:

- The formal description of an attack on a trust and reputation model and what it means for a model to be vulnerable to that attack.
- The design of a decentralized game-theoretic trust and reputation environment model (testbed). The proposed environment model is compatible with the characteristics of open distributed systems. Agents can have different types of interactions and consequently have access to different sources of information for assessment of other agents. Moreover, the proposed environment model provides the facility to define agents with various trust models and is flexible enough to accommodate a variety of adversarial behaviors.
- The assessment of several attacks on a variety of trust models using the testbed.

The remainder of this paper is structured as follows. Section 2 provides a number of working definitions that are important for the trust and reputation framework proposal. Sections 3 and 4 describe the details of the environmental model and agent model of DART, respectively. It should be noted that Section 4 includes information on *specific* trust and reputation models that have been analyzed using the framework and it should be stressed that the framework supports the inclusion of others. We discuss the state-of-the-art related work in Section 5. Section 6 presents a review of the research that has been undertaken using DART and discusses the framework specification in the context of Fullam's requirements introduced earlier. Finally, we conclude the paper by summarizing key messages and discussing potential future work in Section 7.

2. EXPLOITATION AND VULNERABILITY

2.1. Exploitation

In this section, we present the abstract exploitation models that one or more attackers might utilize to mount an attack on a trust and reputation model. We consider an agent's trust and reputation model, M, to be characterized by two attributes, R and P; R is the set of updating rules that are used to calculate changes in trust and reputation and P is the set of parameter values that are used to operate it.

Definition. Attack. An attack, A, is a sequence of cooperations and defections used by a set of malicious agents, $\{ma\}$, to achieve or maintain a trustworthy status as maintained by an agent *ta*, with which they are interacting, or to maximize their collective utility.

We put attacks into two categories: *individual attacks* and *collusion attacks*.

2.1.1. Individual Attacks. In individual attacks, an attacker usually takes the advantage of an existing vulnerability in the trust models to cheat other agents without the model preventing it. This type of attack is mounted by only one attacker against another agent or a set of other agents and usually takes place in direct interactions.

A con-man attack presented in Salehi-Abari and White (2009b) is an example of an individual attack. What the con-man attacker does is to build up trust from the victim's view point by being honest with him/her in several direct interactions. Then, when it comes to a high-risk interaction, the con-man will cheat on the victim. The con-man, by regaining the victim's trust, can again cheat on the victim. Consequently, behavior in which cycles of positive feedback followed by a single negative feedback results in untrustworthy agents remaining undetected in vulnerable trust models. Other known examples of individual attacks

are Proliferation, Reputation Lag, Re-entry, and Value Imbalance attacks (Kerr and Cohen 2009b) (see Section 5.3.2 for further details).

2.1.2. Collusion Attacks. Generally speaking, collusion can be defined as collaborative activity that gives to members of a colluding group benefits they would not be able to gain as individuals.

In contrast to individual attacks discussed earlier, collusion attacks are where a group of agents (at least two agents) conspire together to take advantage of breaches in trust models to defraud a specific agent or a set of agents. One or more of the colluding agents can sacrifice themselves in collusion attacks to maximize the utility of the colluding group.

Collusion attacks usually work based on the basic idea that one or more agents show themselves as trustworthy agents in one type of interaction (usually direct interaction). Afterward, they will be untrustworthy in other type of interaction (e.g., witness interaction) by providing false information in favor of other members of the colluding group. This false information usually encourages a victim to interact with members of the colluding group. The members of the colluding group will cheat the victim, if victim interacts with them. Witness-based Collusion Attack (Salehi-Abari and White 2009c) is an example of a collusion attack. In this attack, Enticer agents, which are trustworthy in their direct interactions, collude with malicious agents by providing a good rating for them. In this sense, they encourage the victim agent(s) to interact with malicious agents.

2.2. Vulnerability

We define more precisely *vulnerability* and the levels of vulnerability of trust models against an attack as follows (Salehi-Abari and White 2010):

Definition. Vulnerability. A trust model, M, is vulnerable to an attack, A, if a malicious agent, *ma*, adopting some strategy and with full or partial knowledge of an agent, *ta*, and its associated trust model, M, can be trustworthy as determined by *ta* and can gain higher utility than trustworthy agents.

We defined the following levels of vulnerability:

Definition. Low-level. A trust model, M, is vulnerable to an attack, A, with low-level risk, if it is vulnerable only for some specific model parameter settings and *ma* needs to be aware of the parameter values used by *ta* to mount a successful attack.

Definition. Medium-level. A trust model, M, is vulnerable to an attack, A, with medium-level risk, if it is vulnerable for any parameter settings and *ma* needs to be aware of the value of parameters used by *ta* to successfully mount an attack.

Definition. High-level. A trust model, M, is vulnerable to an attack, A, with high-level risk, if *ma* is able to successfully mount an attack under any conditions even when *ma* is not aware of the values of parameters.

2.3. Exploitation Resistant

As highlighted earlier, exploitation resistance reflects the ability of a trust model to be impervious to agents who try to manipulate the trust model and who aim to abuse the presumption of trust. Exploitation resistance implies that the agents attempting to exploit

another agent's trust model know both the details of the model and some or all of the model parameters. We formalize these concepts as follows:

Definition. Exploitation-Resistant. A trust and reputation model, M, exhibits exploitation resistance to an attack, A, if it is not vulnerable to that attack. We also refer to a trust and reputation model, M, as being exploitation-resistant when faced with an attack, A.

Definition. p-Exploitation Resistant. A trust and reputation model, M, exhibits partially exploitation resistance (p-exploitation resistance) to an attack, A, if it is vulnerable to that attack with low-level risk. We also refer to a trust and reputation model, M, as being p-exploitation resistant when faced with an attack, A.

3. THE ENVIRONMENT MODEL OF DART

As stated before, the majority of open distributed computer systems can be modeled as MAS in which each component acts autonomously to achieve its objectives (Jennings 2001). This section explains the main idea and principles behind our proposed multiagent environment (the environment model of DART) in which agents' interactions with their peers take place.

The environment model of DART is designed to be consistent with the nature and characteristics of open distributed systems. The proposed environment model follows three features of open distributed systems as described by Huynh et al. (2006). In the proposed model, heterogeneous agents with various perceptive and decision-making capabilities interact in a game-theoretic manner. This environment model can be viewed of as an undirected dynamic graph with nodes representing agents. An edge between two nodes (agents) in this graph indicates that these agents have interactions together and that they can communicate with each other. The motivation for this representation is that the social network of agents be explicitly represented.

3.1. Interactions

An agent interacts with the specific set of other agents that are the neighbors of the given agent. Two agents are *neighbors* if they interact with one another continuously. An agent maintains a *neighborhood* set which contains the name (unique ID) of its neighbors. Our assumption is that identity is unchanging and verified by an external agency providing an authentication service. The *neigborhood* set is a dynamic set, subject to changes over the agent's lifetime (i.e., the agent drops or adds connections). These changes are based on the results of the agent's interactions. Agents can have four types of interactions with their neighbors: Direct Interaction, Observed Interaction, Witness Interaction, and Introduction Interaction.

3.1.1. Direct Experience Interactions. Direct experience, incontrovertibly, is the most popular source of information for trust and reputation models (Ramchurn et al. 2004; Sabater and Sierra 2005). There are two types of direct experiences that an agent can infer agents' trustworthiness from: *Direct Interaction* and *Observed Interaction*.

Direct Interaction. Different fields have their own interpretation and understanding of direct interaction. In the context of e-commerce, direct interaction might be considered as buying or selling a product whereas in peer-to-peer systems (e.g., file sharing systems) direct

interaction is uploading or downloading files. Providing a service and consuming a service can be regarded as a direct interaction in the context of Web services while asking a question (sending a query) and answering that question (receiving the result of that query) is a direct interaction from the perspective of information retrieval.

Observed Interaction. Agents can judge the trustworthiness of another agent by relying on the observation of the given agent's interactions with other agents in the community. This source of information is not common in decentralized trust and reputation models because of the existing limitation for observing interaction in open distributed systems. For example, the observation of interactions in a peer-to-peer system, which is an open distributed system, for each peer is not straightforward.

Social learning theory leads us to hypothesize that observation of interactions can be considered to be one of the main information sources for learning of trustworthiness of other agents. Observational or social learning is based primarily on the work of Albert Bandur and his colleagues who showed that learning could occur through the simple process of observing someone else's activity. Consequently, people learn through observing others' behavior (Ormrod 2003).

To provide the observation of interactions for agents and resolve the aforementioned limitation in open distributed systems, the neighbors of an agent report their direct interactions with their own neighbors to their immediate neighbors. In this sense, DART has a decentralized system of news broadcasting that is consistent with its decentralized nature and provides the facility for agents to have social (observational) learning. Our motivation for including this type of interaction is the prevalence of information forwarding mechanisms in social network applications; e.g., Twitter and Facebook. From now on, we call this type of interaction by which agents report their direct interactions to their neighbors the ***Reporting Interaction***.

3.1.2. Witness Interaction. Witness information is information that comes from other members of the community regarding another agent. This information is provided in the form of a rating that can be based on observed interaction or direct interaction. An agent can ask for an assessment of the trustworthiness of a specific agent from its neighbors and then the neighbors send their ratings of that agent to the requesting agent. We call this asking for an opinion and receiving a rating, a ***Witness Interaction***.

3.1.3. Introduction Interaction. We are proposing a type of interaction for agents in DART called an *Introduction Interaction*. Agents can introduce or recommend one of their neighbors to the other one by using an Introduction Interaction. The Introduction Interaction allows trustworthy agents to become connected more quickly to each other. Using the Introduction Interaction, the diameter of the society of trustworthy agents will shrink and consequently the agents will have a higher number of trustworthy agents while being exposed to a smaller number of untrustworthy agents. Introduction interactions can be either *request-driven* or *asynchronous*.

In the request-driven scenario, an agent will make a request for a connection recommendation to one of its neighbors, and then, in the response, the neighbor introduces a new agent to the requester. In contrast, an asynchronous introduction is solely based on the decision of the recommender. For instance, after an agent is known as a trustworthy agent from the perspective of a neighbor as a consequence of direct interactions, the neighbor might introduce the agent to one of the other trustworthy agents. In this way, the trustworthy agent can extend its neighborhood by adding new trustworthy agents. This introduction can be an incentive for agents to be trustworthy to their peers to be introduced to more trustworthy agents in the

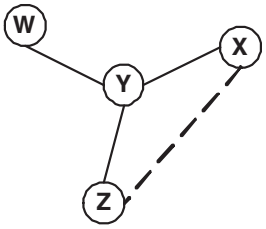


FIGURE 1. An example for introduction of an agent to another.

community. Our motivation for proposing this interaction is the existence of systems such as PGP.

For instance, as shown in Figure 1, after several direct interactions of agent X with agent Y, if agent X is known to be trustworthy from the viewpoint of agent Y, agent Y might introduce agent X to one of its other trustworthy neighbors; let us say agent Z.

There are several complex scenarios regarding this interaction that our described in detail in Sections 3.3.3 and 4.1.4.

3.2. Games: Iterated Prisoner's Dilemma (IPD) and Generalized Prisoner's Dilemma (GPD)

We have modeled interactions in the DART environment using two extensions of the Prisoners Dilemma: IPD and GPD.

The prisoner's dilemma, a problem in game theory, was originally formed by Merrill Flood and Melvin Dresher in 1950 while Albert W. Tucker formalized the game with prison sentence payoffs (Poundstone 1992). The Prisoner's Dilemma forms a nonzero-sum, noncooperative and simultaneous game in which two players may each cooperate with or defect from the other player. Similar to other games in game theory, the goal of each individual is maximizing his/her payoff, without any concern for other player's payoff. In this game, cooperating is strictly dominated by defecting since the game will only be played once between individuals.

In contrast, since the game is played repeatedly in the IPD (Axelrod 1984), each player has an opportunity to "punish" the other player for previous uncooperative play. As a result, cooperation might emerge as an equilibrium outcome. The IPD is closely related to the evolution of trust because if both players trust each other they can both cooperate and prevent mutual defection. Moreover, this trust can only build up in the environment where individuals have to interact with each other repeatedly.

We have modeled direct interactions using an IPD. Each agent plays one game with each of its neighbors in each cycle of simulation.

The GPD is a two-person game which specifies the general forms for an asymmetric payoff matrix that preserves the social dilemma. GPD is compatible with client/server structure where one player is the client and the other one is the server in each game. It is only the decision of the server which determines the ultimate outcome of the interaction. Note that a player can be a server in one game and a client in another (Feldman et al. 2004).

We used GPD to model witness, reporting, and introduction interactions because these interactions are compatible with the nature of client/server structure. For example, in a witness interaction, the asker agent is a client while the witness information provider is a server for that request and in a reporting interaction the reporter is the server whereas the listener is a client.

3.3. Cooperation and Defection

We define different kinds of **Cooperation** and **Defection** in the DART model. There are four types of cooperation and defection:

- Cooperation/Defection in Direct Interaction (CDI/DDI),
- Cooperation/Defection in Reporting Interaction (CRI/DRI),
- Cooperation/Defection in Witness Interaction (CWI/DWI),
- Cooperation/Defection in Introduction Interaction (CII/DII).

3.3.1. CDI/DDI. CDI/DDI have different interpretations depending on the context. In the context of e-commerce, defection in an interaction can be interpreted as that the agent does not satisfy the terms of a contract, sells poor quality goods, delivers late, or does not pay the requested amount of money to a seller depending on the role of the agent (Ramchurn et al. 2004). Therefore, defection could get higher payoffs for the agent defecting and cause some utility loss for the other agent. In contrast, if both interaction participants cooperate, they will get higher payoffs in the long term (Axelrod 1984).

Cooperation in peer-to-peer systems (e.g., file sharing) might mean allocating high bandwidth for uploading files while defection might be considered as low bandwidth allocation for uploading. In the context of information retrieval, defection in an interaction can be interpreted as that the queried agent returns irrelevant documents to the asking agent as the consequence of its query. In contrast, cooperation means that a proper answer is provided according to the query for the questioner.

Cooperation and defection may have their own interpretation in the domain of Web services. Generally, the cooperative service provider prepares a desirable service for a consumer, subject to the set of consumer constraints. By contrast, defection is the outcome of providing a low-quality and/or undesirable service.

3.3.2. CWI/DWI. As explained in Section 3.1, an agent can ask for an assessment of the trustworthiness of the specific agent from the perspective of other agents. In this sense, the witness agent can provide honest ratings of the agent or a false rating of the agent. Even a witness agent can hide its rating from an asking agent and might pretend not to have any relevant information. Therefore, the asking agent may encounter two types of response behavior from a witness agent: (1) cooperation or (2) defection. We define Cooperation/Defection in the context of Witness Interaction (CWI/DWI) as follows:

Definition: Cooperation in a witness interaction means that the witness agent will provide a reliable and honest rating for the asker agent regarding the queried agent. In contrast, defection in a witness interaction means that the witness agent does not provide a reliable and honest rating for the asker agent regarding the queried agent.

It is interesting to note that the defection in providing witness information can be based on malicious incentive, incompetence, or even noise. A witness agent might have an incentive to misrepresent its trust view of the trustee, which might result in a positive or a negative effect on a trustee's reputation. The witness agent may choose to overestimate the trust value of a trustee in the case of having a strong cooperative relationship with the trustee, whereas a competitive relationship may lead the rating agent to underestimate the trustee.

3.3.3. CRI/DRI. Agents might cooperate and defect in terms of reporting their news to their neighbors. We define cooperation and defection in a reporting interaction as follows:

Definition: Cooperation in a reporting interaction means that the agent will report important results of its interactions to the other party and it will not hide, lie about, or bias them. Similarly, defection in a reporting interaction means that the agent will hide, bias, or lie about the result of its own interactions with its other neighbors.

3.3.4. CII/DII. An agent with regard to introduction of one agent (a neighbor) to another one (another neighbor) can cooperate or defect. There are four cases regarding this cooperation/defection:

(1) An agent introduces two trustworthy agents to each other, which is considered cooperation of the agent with both other agents.
(2) An agent introduces one trustworthy agent to one untrustworthy agent, which is considered cooperation of the agent with the untrustworthy agent and defection for the trustworthy one.
(3) An agent prevents the introduction of two trustworthy agents to each other, while they are known as trustworthy agents from the perspective of the given agent. This is considered as defection for both of the trustworthy agents (optional).
(4) An agent introduces two untrustworthy agents to each other, which is considered defection of the agent with both of the other agents.

It is worth mentioning that the case 3 is optional and can be applied based on the policies of different open distributed systems. If the system put this obligation on the agents who join the system that they are responsible for introducing trustworthy agents to each other, then those agents who are reluctant to introduce their trustworthy neighbors to each other should be punished with low values of trust for their introduction interactions (case 3 should be used). However, if the introduction of the agents is not mandatory but is preferred, then there is no expectation from the agents to introduce their neighbors to each other (case 3 should not be used).

The general rule of thumb for understanding whether an introduction was cooperation or defection is based on the behavior of the introduced agents to each other. This assumption is compatible with the similar one for witness interaction in which the witness is responsible for the usefulness of its rating for the asking agent when this usefulness should be judged by the asking agent. For example, suppose that witness X has a good history of interactions with an agent Y and provides a good rating for it to the agent Z. Later on, the agent Z finds agent Y untrustworthy in based on its own interactions. In this sense, agent Z consider the high ratings provided by agent Y as a defection, although agent Y might not have an intention to harm agent Z.

Definition: Cooperation in introducing agents to each other (CII) means that the cooperative agent will introduce trustworthy agents to each other. In contrast, defection in introducing agents to each other (DII) means that the agent will not introduce trustworthy agents to each other or introduce an untrustworthy agent to the trustworthy one.

CII/DII can be perceived indirectly and directly. For indirect perception, when agent k is introduced to agent i by agent j, agent i, based on the CDI/DDI of agent k, can understand that this introduction was cooperation or defection. In other words, if the introduced agent k cooperates with agent i in the context of direct interactions, those cooperations also take into account for agent j's introduction interaction. Likewise, if agent k defects, this defection also will count for the agent j's introduction interaction. In this light, someone who introduces two agents to each other is responsible for the behavior of them, and will be punished or rewarded for this introduction.

Suppose that agents Y and Z are trustworthy to agent X, thus agent X decides to introduce Y and Z to each other. However, later on, Y does not find Z to be trustworthy to it, based on its direct interactions with Z. For this example, the introduction of Z to Y is first considered as cooperation (from the perspective of Y) because of direct perception. Then, it will be considered as defection in indirect perception because Z will defect in its direct interactions with Y. At the end, this introduction will be considered as defection (from the perspective of Y) after both agent Z and Y start to interact with each other although before the start of interactions, it is considered as cooperation (see numerical example in Section 4.1.4).

Suppose that Y is untrustworthy to X but Z is trustworthy to X in their direct interactions. X maliciously or mistakenly introduces Y and Z to each other. Nevertheless, Z, based on its interaction with Y, discovers that Y is a trustworthy agent. As a result of indirect perception, this introduction will be considered as cooperation for both Z and Y after both agents Z and Y start to interact with each other. However, before the start of direct interactions, it is considered as cooperation for Y and defection for Z because of their direct perception (for an expanded analysis, see numerical example in Section 4.1.4).

3.4. Protocols

We here explain protocols of the DART model which agents use for their different interactions and for connecting with each other. Corresponding to the four types of interactions explained in Section 3.1, there are four protocols: Direct Interaction Protocol, Reporting Interaction Protocol, Witness Interaction Protocol, and Introduction Interaction Protocol. All protocols use messages and each message is defined by the tuple of *<Type, Content, SenderID, DestinationID, TargetID>*, where *Type* shows the type of the message. The value of the $Content$ variable differs in each message type (i.e., it is type-dependent). $SenderID$ and $DestinationID$ represent the sender's ID (name) of the message and the destination's ID (name) of message, respectively. $TargetID$, which is used by some message types (not all), provides metadata information.

3.4.1. Direct Interaction Protocol. To model playing the prisoner's dilemma game for a direct interaction, each agent sends a direct interaction message (DIM) with the value of either cooperation or defection to each of its neighbors. As the neighbors will do the same, the agent will receive DIM from them as well. We denote a DIM as *<DIM,CDI/DDI, SenderID, DestinationID, nil>*, where nil means that no meta information is provided.

3.4.2. Witness Interaction Protocol. To simulate the witness interaction in our model, the agent looking for witness information about a target agent will send an *investigation* message, denoted by *<Inv, nil, SenderID, DestinationID, TargetID>*, to one or all of its neighbors. $TargetID$ includes the ID of the target agent and $SenderID$ is the id of the asker agent. The receiver agent will send its opinion in response to the investigation message using an *opinion* message denoted by *<Op, rating, SenderID, DestinationID, TargetID>*. The witness agent who sends out the *opinion* message will send a witness interaction message (WIM) to the asker agent after T_w cycles of simulation. The WIM denoted by *<WIM, CWI/DWI, SenderID, DestinationID, nil>* indicates that the previous *opinion* message was cooperation or defection. The WIM will be sent out after T_w cycles to simulate the fact that it takes some time to understand whether the witness agent was cooperative or noncooperative in the witness interaction. The intention behind the WIM is to simulate the perception of whether the corresponding opinion message was cooperation/defection, and it should not be mistaken as an agent's confession about its cooperation or defection.

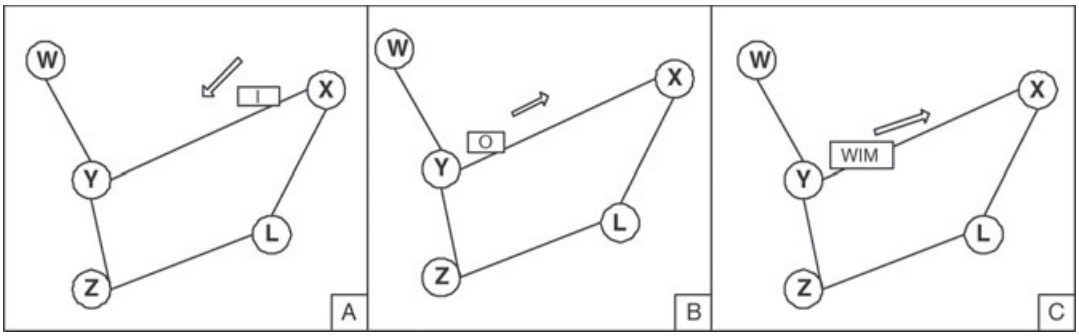


FIGURE 2. A scenario for demonstration of Witness Interaction Protocol.

To clarify the above explanation, consider the example illustrated in Figure 2, where *investigation* and *opinion* messages are depicted by boxes with the labels of *I* and *O*, respectively. Agent *X* intends to find out about the reputation of the target agent *Z* while never having interacted with it. Agent *X* thus sends an *investigation* message to agent *Y* asking about agent *Z* as shown in Figure 2(A). Upon receiving the *investigation* message, agent *Y* sends out its rating of agent *Z* to agent *X* as illustrated in Figure 2(B). After T_w cycles of simulation, agent *Y* will send *WIM* to agent *X* as shown in Figure 2(C).

It should be noted that the perception of whether provided witness information is a cooperation or defection is not a hard problem. This problem has been solved by other researchers and is beyond the scope of this paper. There are two basic approaches to achieving this perception that are proposed in the literature; these are referred to as endogenous and exogenous methods by Josang, Ismail, and Boyd (2007). The former tries to detect unreliable witness information (opinions) by using the statistical properties of the reported opinions (for example, Dellarocas (2000); Whitby, Josang, and Indulska (2004)). The latter rely on other information such as the reputation of the source or the relationship with the trustee such as used in the work of Yu and Singh (2003).

3.4.3. Reporting Interaction Protocol. In every cycle, or after specific number of cycles, each agent can send out all or a part of the results of its direct interactions with its neighbors in the format of a Report message, denoted by *<RM, Reports, ReporterID, DestinationID, nil>*. A report message includes the results of several direct interactions of the reporter with its neighbors stored in the *Reports* array. The result of each interaction is an element of the *Reports* array and is a tuple $<ID_1, ID_2, A_1, A_2>$, where A_1 and A_2 are the actions of the agent ID_1 and ID_2 in the reported interaction, respectively. Note that the value of A_1 and A_2 are either cooperation or defection (more precisely, either CDI or DDI).

According to the simulation of the perception of cooperation/defection in a reporting interaction, we first concentrate on how an agent can understand whether one neighbor is cooperative in reporting interactions or not. There are two scenarios, shown in Figure 3, which are:

- As shown in Figure 3(A), agent *X* is receiving the reports of interactions from *Y*. To understand whether *Y* is cooperating or not, agent *X* needs to hear the same report from another trustworthy source. Suppose *W* also reports its interactions to *X* and it is known as a cooperative peer in terms of its reporting interactions. Since *Y* and *W* are interacting with one another, thus their reports on interactions with each other should be compatible.

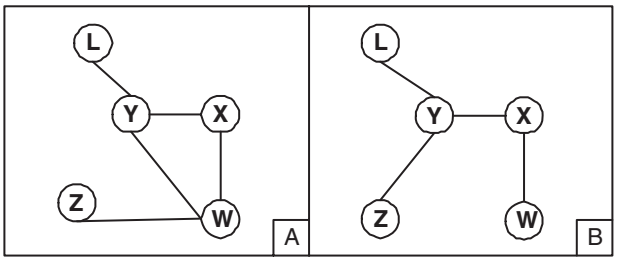


FIGURE 3. Scenarios for reporting interactions.

If incompatible, Y is not cooperative in reporting interactions given that W is already known as a trustworthy agent in terms of reporting interactions.

- As shown in Figure3(B), agent X is receiving reports of interactions of Y with L and Z from Y. To understand whether Y cooperates in reporting, agent X needs to hear the same report from another trustworthy source. However, X cannot hear about these interactions from other parties, thus it will consider that Y is cooperating in reporting unless proven otherwise. Agent X might connect to L, and Z in future and hear about their previous interactions with Y. Then, X will determine whether Y was cooperative previously or not.

To model this perception simply, the reporter agent (agent Y in the above example) will send a reporting interaction message (RIM) to the listener agent (agent X in the above example) indicating whether it was cooperative in reporting the interactions or not. A RIM is denoted by *<RIM, CRI/DRI, ReporterID, DestinationID, nil>*. The RIM will be sent out T_r cycles after the corresponding report was sent out. This simulates the fact that it takes time to understand whether a reporter was cooperative or noncooperative in the provided reports. The intention behind the RIM is to simulate the perception of whether the corresponding report message was cooperation/defection, and it should not be interpreted as an agent's statement about its cooperation or defection.

3.4.4. Introduction Interaction Protocol. As explained in Section 3.1.3, an introduction interaction can be either request-driven or asynchronous. For request-driven interactions, the agent which is looking for a recommendation sends an *AskingForRecommendation* Message (AFRM), denoted by *<AFRM, nil, SenderID, DestinationID, nil>*, to a neighbor. The neighbor has two response choices upon receiving this message:

- Recommending an agent by sending a *Recommendation* Message (REM) denoted by *<REM, nil, RecommenderID, DestinationID,TargetID>*, where *TargetID* maintains the ID of the introduced (recommended) agent.
- Not recommending any agents to the requester.

In the case of introducing an agent, the perception of whether this introduction was cooperation or defection is indirect which relies on the cooperation and defection of the introduced agent with the requester (recall Section 3.1.3). When there is no recommendation, the perception is direct which we simulate by sending an introduction interaction message (IIM) to the requester. IIM is denoted by *<IIM, CII/DII, IntroducerID, DestinationID, nil>* and can have the value of cooperation or defection (CII/DII). If the recommender agent

has no unknown-to-the-requester trustworthy neighbors then it will send IIM with the value of cooperation (CII) to the requester. However, if the recommender agent was reluctant to introduce a trustworthy agent to the requester, it will send an IIM with the value of defection (DII) to the requester (this is case 3 in Section 3.3.4 which is optional). It is important to note that IIMs simulate the perception of cooperation/defection while in a live system the cooperation/defection of an introduction interaction would be something that the agent itself would determine based on the received information.

All the above explanations are valid for an asynchronous introduction but the difference is that there is no AFRM in an asynchronous introduction. The recommender agent will send the REM to a neighbor at any time based on its introduction policy (see Section 4.3.4). Depending on the situation, the IIM will be forwarded to the neighbors.

3.5. Connection and Disconnection

One of the motivating factors for the framework proposal described in this paper is the capture and use of the agent social network. Agents only interact with their neighbors. The agent's neighborhood is dynamic and subject to change over the course of the simulation. An agent can disconnect from a neighbor and exclude it from the neighborhood set or it might make a new connection to other agents and include them in the neighborhood set. In Section 4.3, we will explain that these decisions are all made by the connection policy (CP) of agents.

Two agents can become connected to each other if and only if both agents agree to this relationship. Usually, one agent requests a connection to another agent by sending a connection request message (CRM) denoted by *<CRM, nil, SenderID, DestinationID, nil>* and the other one processes the request and based on its CP and perception variables decides whether to accept or reject this connection request. In the case of the acceptance of a connection request, the agent will send a connection acknowledge message (CAM) denoted by *<CAM, nil, SenderID, DestinationID, nil>* to the requester. Agents require the unique ID of the agent with which they intend to connect to make a request for connection. These IDs might be acquired either by referring to the registry list (see Section 3.6) or by introduction of another agents. Identity is unique and reported reliably by all agents.

For disconnection, the decision of one agent is enough. One agent can disconnect from a neighbor, and then the neighbor will be notified about this disconnection by a disconnection acknowledgment message (DCAM). DCAM is denoted by *<DCAM, nil, SenderID, DestinationID, nil>*, where senderID represents the agent which has decided to disconnect from DestinationID agent. An agent decides when to disconnect from a neighbor based upon its policies (see Section 4.3). For example, one agent can disconnect from a neighbor if the neighbor is known as untrustworthy, thus resulting in punishment of the untrustworthy agent by not interacting with it.

3.6. Registry List

There are situations in which agents have a tendency to connect to unknown and unvisited agents to interact with them. Two scenarios are modeled in this proposal. When an agent is isolated because of either the consequence of its previous interactions or its recent entrance to the system, it needs to make a connection request to some existing but unknown-to-it agents. The IDs of those agents are necessary to make this connection request. In the DART model, we introduce a component called the *registry list* in which the IDs of all existing agents are registered.

This registry list plays a role similar to a white page service in a distributed system. One of the roles of the registry list would be to authenticate agents when they register with the system; however authentication was not considered in the research reported in this paper. Those agents who are in need of a connection can acquire an agent ID by referring to this registry list. It should be noted that knowing the ID of an agent cannot guarantee the successful connection to the given agent (recall Section 3.5). Moreover, IDs appear in random order and the IDs are shuffled by each access to the registry list. This shuffling prevents the attack in which malicious agents try to register themselves at the top of the list to attract more isolated agents to themselves and consequently to have more opportunities for fraudulent interactions.

3.7. Agent Type and Initialization

The DART Framework provides the facility to define and to specify heterogeneous agents in terms of their perceptions and behaviors. Each agent type is defined as a tuple $<TypeId, Po_{id}, PM_{id}>$ where $TypeId$ is a unique identifier for that specific type of agent and Po_{id} is a set of policies for different interaction types and connection/disconnection policies. These policies make decisions based on the set of perception models PM_{id} consisting of trust and reputation models (see Section 4). As mentioned in Section 2.1, each trust model or reputation model, M, is characterized by two attributes, R and P; R is the set of updating rules that changes to trust and reputation is based on and P is the set of parameter values that are used to operate it

DART offers a facility to define different types of agents varying in their Po and PM sets. After definition and specification of Po and PM for each agent type, agents in the simulation environment will be initialized using one of these types. We define the $PT = \{p_1, p_2, \ldots, p_n\}$ vector which consists of the percentage of each type of agent in the simulation environment. The p_i in PT vector is the percentage of agents which will be initialized by Po_i and PM_i of agent type i. Note that $\sum_{i=1}^{n} p_i = 1$. In other words, each agent is initialized by the Po_i and PM_i with the probability of p_i.

3.8. Newcomers

Newcomers play crucial roles and have their own concerns in open distributed systems given that distributed entities might enter the system at any time. To model this characteristic, some agents can uniformly be inserted as isolated agents (nodes) in the environment over the course of the simulation. Suppose that the simulation period is 300 cycles and 50 agents are intended to be inserted over the simulation period. In this case, every six cycles (time step) one agent, an isolated node, will be inserted into the system. In general, it is possible to specify a model associated with the entrance and departure of agents in DART.

A newcomer adopts its type based on the same p_i probability explained in Section 3.7. The newcomer is not able to interact until it gets connected to at least one agent. To make a connection with an existing agent, the newcomer should acquire an agent ID and make a connection request. This ID acquisition can be accomplished by accessing the registry list as explained in Section 3.6.

3.9. Metrics

DART provides a collection of tools and metrics for researchers to experimentally analyze the agent types on both microscopic and macroscopic levels.

TABLE 1. Payoff Matrix of Iterated Prisoner's Dilemma.

P_1/P_2	Cooperate	Defect
Cooperate	3,3	0,5
Defect	5,0	1,1

On the macro level, the structure of agent society will be depicted in the form of an undirected graph over the course of the simulation. This visualization assists a researcher in studying how society structure will be changed over interactions.

On the micro level, we were interested in examining the internal properties of each agent type, such as utility of agents and the number of unsuccessful connections made by agents which is an indicator of the encounter risk of agents.

DART offers the following metrics for micro level analysis:

$\overline{U_{AT}(i)}$, the average of utilities for agents with the type of AT at time step i, is calculated by

$$\overline{U_{AT}(i)} = \frac{\sum_{a \in AT} U_{Avg}(a, i)}{N_{AT}} \tag{1}$$

where $U_{Avg}(a, i)$ is the average of utility of agent a over its interactions at time step i and N_{AT} is the total number of agents of society whose type is AT. The utility of each interaction is calculated based on the following payoff matrix (the well-known payoff matrix of the IPD (Axelrod 1984)):

According to Table 1, if agent P_1 defects and agent P_2 cooperates, agent P_1 gets the Temptation to Defect payoff of 5 points while agent P_2 receives the Suckers payoff of 0. If both cooperate each gets the Reward for Mutual Cooperation payoff of 3 points, while if both defect each gets the Punishment for Mutual Defection payoff of 1 point. We have chosen this well-known payoff matrix as the default but a time-evolving payoff matrix that respects the social dilemma can also be used based on requirements.

$\overline{D_{AT}(i)}$, the average of dropped connections for agents with the type of AT at time step i, is calculated by

$$\overline{D_{AT}(i)} = \frac{\sum_{a \in AT} D_{total}(a, i)}{N_{AT}} \tag{2}$$

where $D_{total}(a, i)$ is the total number of connections broken for agent a from the start time to time step i and N_{AT} is the total number of agents of society whose type is AT.

4. AGENT MODEL OF DART

The aim of each trust and reputation model is to guide an agent's decision making in deciding how, when and who to interact with in a specific context. We here explain our proposed agent model which provides mechanisms for deciding with whom, when, and how an agent will interact. This agent model is designed to perceive the behavior of other agents and consequently predict the trustworthiness of them to help an agent in making low-risk decisions.

The proposed agent model consists of a perception model and a set of policies. It is designed to be consistent with the definition provided in Section 2. The perception model, including trust models and reputation models, help agents in modeling the trustworthiness and reliability of other agents. On the other hand, the policies assist them in how they should behave with others considering the perceived trustworthiness of the other (possibly adversarial) agents.

It should be noted that Sections 4.1.1, 4.1.2, 4.1.3, and 4.1.4 describe example updating schemes which have been used in analysis using the framework (e.g., Salehi-Abari and White 2009a, 2009b). They are included to demonstrate the coupling between trust variables and perceptions (cooperation/defection); other models are possible.

4.1. Trust Models

To have multidimensional trust models, DART equips an agent (the truster) with four independent dimensions of trust (trust variables) while each trust variable corresponds to an interaction type. The motivation for having four trust variables is that trust in information received should be independently assessed. As stated earlier, an agent which is trustworthy in direct interactions is not necessarily trustworthy in reporting interactions or witness interactions. This subsection provides descriptions for these trust variables.

Each trust variable is defined by $T_{i,j}(t)$ indicating the trust rating assigned by agent i to agent j after t interactions between agent i and agent j, while $T_{i,j}(t) \in [-1, +1]$ and $T_{i,j}(0) = 0$. In DART, one agent in the view of the other agent can have one of the following levels of trustworthiness:

- *Trustworthy*,
- *Not Yet Known*,
- *Untrustworthy*.

Following Marsh (1994), we define for each agent an upper and a lower threshold to model different levels of trustworthiness. The agent i has its own upper threshold $-1 \leq \omega_i \leq 1$ and lower threshold $-1 \leq \Omega_i \leq \omega_i$. Agent j is trustworthy from the viewpoint of agent i after t times of interactions if and only if $T_{i,j}(t) \geq \omega_i$. Agent i sees agent j as a untrustworthy agent if $T_{i,j}(t) \leq \Omega_i$ and if $\Omega_i < T_{i,j}(t) < \omega_i$ then the agent j is in the state *Not Yet Known* in agent i's view.

4.1.1. Direct Interaction Trust (DIT). DIT is the result of the cooperation/defection that agents have in their direct interactions (CDI/DDI). Each agent maintains $DIT_{i,j}(t)$ variables for the agents having had direct interactions with them (its neighbors or ex-neighbors). The agent will update this variable based on the perception of CDI/DDI. Although various trust updating schemes can be employed for DIT in DART, We here present the following trust updating scheme motivated by that proposed in Yu and Singh (2000):

$$DIT_{i,j}(t+1) = \begin{cases} DIT_{i,j}(t) + \alpha_D(i)(1 - DIT_{i,j}(t)) & DIT_{i,j}(t) > 0, CDI \\ (DIT_{i,j}(t) + \alpha_D(i))/(1 - \min(|DIT_{i,j}(t)|, |\alpha_D(i)|)) & DIT_{i,j}(t) < 0, CDI \\ (DIT_{i,j}(t) + \beta_D(i))/(1 - \min(|DIT_{i,j}(t)|, |\beta_D(i)|)) & DIT_{i,j}(t) > 0, DDI \\ DIT_{i,j}(t) + \beta_D(i)(1 + DIT_{i,j}(t)) & DIT_{i,j}(t) < 0, DDI \end{cases} \quad (3)$$

where $1 > \alpha_D(i) > 0$ and $-1 < \beta_D(i) < 0$ are positive evidence and negative evidence weighting coefficients, respectively, for updating of the DIT variable of agent i. $\alpha_D(i)$ and $\beta_D(i)$ may be constant or may be updated based upon an updating scheme (Salehi-Abari and White 2009b). The values of $DIT_{i,j}(t)$, ω_i^{DIT}, and Ω_i^{DIT} determine that the agent j is either *Trustworthy*, *Not Yet Known*, or *Untrustworthy* in terms of direct interaction from the perspective of agent i.

4.1.2. Witness Interaction Trust (WIT). WIT is the result of the cooperation/defection that agents have regarding their witness interactions (CWI/DWI). Agent i maintains a $WIT_{i,j}(t)$ variable for the agent j from whom it has received witness information. Agent i will update this variable based on the perception of CWI/DWI from agent j.

$$WIT_{i,j}(t+1) = \begin{cases} WIT_{i,j}(t) + \alpha_W(i)(1 - WIT_{i,j}(t)) & WIT_{i,j}(t) > 0,\ CWI \\ (WIT_{i,j}(t) + \alpha_W(i))/(1 - \min(|WIT_{i,j}(t)|, |\alpha_W(i)|)) & WIT_{i,j}(t) < 0,\ CWI \\ (WIT_{i,j}(t) + \beta_W(i))/(1 - \min(|WIT_{i,j}(t)|, |\beta_W(i)|)) & WIT_{i,j}(t) > 0,\ DWI \\ WIT_{i,j}(t) + \beta_W(i)(1 + WIT_{i,j}(t)) & WIT_{i,j}(t) < 0,\ DWI \end{cases} \quad (4)$$

where $1 > \alpha_W(i) > 0$ and $-1 < \beta_W(i) < 0$ are positive evidence and negative evidence weighting coefficients, respectively, for updating of the WIT variable of agent i. The values of $WIT_{i,j}(t)$, ω_i^{WIT} and Ω_i^{WIT} determine that the agent j is either *Trustworthy*, *Not Yet Known*, or *Untrustworthy* in terms of witness interactions from the perspective of agent i.

4.1.3. Reporting Interaction Trust (RIT). RIT is the result of the cooperation/defection that agents have in reporting their interactions (CRI/DRI). Agent i maintains an $RIT_{i,j}(t)$ variable for the agent j from whom it has received report messages. Agent i will update this variable based on the perception of CRI/DRI from agent j.

$$RIT_{i,j}(t+1) = \begin{cases} RIT_{i,j}(t) + \alpha_R(i)(1 - RIT_{i,j}(t)) & RIT_{i,j}(t) > 0,\ CRI \\ (RIT_{i,j}(t) + \alpha_R(i))/(1 - \min(|RIT_{i,j}(t)|, |\alpha_R(i)|)) & RIT_{i,j}(t) < 0,\ CRI \\ (RIT_{i,j}(t) + \beta_R(i))/(1 - \min(|RIT_{i,j}(t)|, |\beta_R(i)|)) & RIT_{i,j}(t) > 0,\ DRI \\ RIT_{i,j}(t) + \beta_R(i)(1 + RIT_{i,j}(t)) & RIT_{i,j}(t) < 0,\ DRI \end{cases} \quad (5)$$

where $1 > \alpha_R(i) > 0$ and $-1 < \beta_R(i) < 0$ are positive evidence and negative evidence weighting coefficients, respectively, for updating of the RIT Trust variable. The values of $RIT_{i,j}(t)$, ω_i^{RIT}, and Ω_i^{RIT} demonstrate that the agent j is either *Trustworthy*, *Not Yet Known*, or *Untrustworthy* in terms of reporting interactions from the perspective of agent i.

4.1.4. Introduction Interaction Trust (IIT). IIT is the result of the cooperation/defection of the agents in introduction interactions (CII/DII). Agent i maintains an $IIT_{i,j}(t)$ variable for the agent j which has introduced agent k. Agent i will update this variable based on the perception of CII/DII of introducer agent j and CDI/DDI of the introduced agent k (direct and indirect perception, see Section 3.3.4).

$$IIT_{i,j}(t+1)$$

$$= \begin{cases} IIT_{i,j}(t) + \alpha_I(i)(1 - IIT_{i,j}(t)) & IIT_{i,j}(t) > 0, CII(j) \\ (IIT_{i,j}(t) + \alpha_I(i))/(1 - \min(|IIT_{i,j}(t)|, |\alpha_I(i)|)) & IIT_{i,j}(t) < 0, CII(j) \\ (IIT_{i,j}(t) + \beta_I(i))/(1 - \min(|IIT_{i,j}(t)|, |\beta_I(i)|)) & IIT_{i,j}(t) > 0, DII(j) \\ IIT_{i,j}(t) + \beta_I(i)(1 + IIT_{i,j}(t)) & IIT_{i,j}(t) < 0, DII(j) \\ IIT_{i,j}(t) + \alpha_I(i)(1 - IIT_{i,j}(t)) & IIT_{i,j}(t) > 0, CDI(k) \\ (IIT_{i,j}(t) + \alpha_I(i))/(1 - \min(|IIT_{i,j}(t)|, |\alpha_I(i)|)) & IIT_{i,j}(t) < 0, CDI(k) \\ (IIT_{i,j}(t) + \beta_I(i))/(1 - \min(|IIT_{i,j}(t)|, |\beta_I(i)|)) & IIT_{i,j}(t) > 0, DDI(k) \\ IIT_{i,j}(t) + \beta_I(i)(1 + IIT_{i,j}(t)) & IIT_{i,j}(t) < 0, DDI(k) \end{cases} \quad (6)$$

where $CDI(k)/DDI(k)$ is the cooperation/defection in direct interaction received by agent i from introduced agent k. Therefore, a cooperation/defection of the introduced agent k takes into account to update introducer agent j's IIT, $IIT_{i,j}$. This mechanism is the indirect perception of DII/CII which is explained in Section 3.3.4. CII_j/DII_j (Direct perception) is taken into account for calculation of IIT to give some consideration to the intention of the introducer.

Moreover, $1 > \alpha_I(i) > 0$ and $-1 < \beta_I(i) < 0$ are positive evidence and negative evidence weighting coefficients, respectively, for updating of the IIT variable. The values of $IIT_{i,j}(t)$, ω_i^{IIT}, and Ω_i^{IIT} demonstrate that the agent j is either *Trustworthy*, *Not Yet Known*, or *Untrustworthy* in terms of introduction interactions from the perspective of agent i.

To demonstrate how Equation (6) updates $IIT_{i,j}(t)$ based on direct and indirect perception of cooperation/defection, we present the two following scenarios and follow step-by-step the updating process of $IIT_{i,j}(t)$.

Scenario 1. Suppose that agents Y and Z are trustworthy to agent X, thus agent X decides to introduce Y and Z to each other. However, later on, Y does not find Z to be trustworthy to it, based on its direct interactions with Z.

Before the introduction, suppose that $DIT_{Z,Y} = DIT_{Y,Z} = 0$ as they have not interacted with each other yet and $IIT_{Y,X} = IIT_{Z,X} = 0$. Moreover, suppose that $DIT_{x,y} = 0.90$ and $DIT_{x,z} = 0.92$. As X consequently considers both Y and Z trustworthy to itself, X introduces these two agents to each other. As a result, based on the direct perception agents Y and Z considers this as an introduction cooperation and both $IIT_{Y,X}$ and $IIT_{Z,X}$ increase to 0.1. After that, Y and Z start interacting with each other, Z defects in its first direct interaction with Y and consequently $DIT_{Y,Z} = -0.1$. Moreover, this defection is taken into account in updating $IIT_{Y,X}$ as an indirect perception and consequently $IIT_{Y,X}$ decreases to 0. After the second defection, we have $DIT_{Y,Z} = -0.2$ and $IIT_{Y,X} = -0.1$. Therefore, after several defections, $IIT_{Y,X}$ has a low value indicating that X is not trustworthy in introduction interaction from the perspective of Y. Note that $IIT_{Z,X}$ is not necessary changed similar to $IIT_{Y,X}$ and it depends on whether Y cooperates with Z or not.

Scenario 2. Suppose that Y is untrustworthy to X but Z is trustworthy to X in their direct interactions. X maliciously or mistakenly introduces Y and Z to each other. Nevertheless, agent Z, based on its interaction with Y, discovers that Y is a trustworthy agent.

Before the introduction, suppose that $DIT_{Z,Y} = DIT_{Y,Z} = 0$ as they have not directly interacted with each other yet and $IIT_{Y,X} = IIT_{Z,X} = 0$. Moreover, suppose that $DIT_{X,Y} = -0.50$ and $DIT_{X,Z} = 0.92$, X introduces these two agents to each other. As a result, based on the direct perception, agent Y considers this introduction as cooperation whereas agent Z considers it as defection. As a result, $IIT_{Y,X}$ increases to 0.1 whereas $IIT_{Z,X}$ decreases to -0.1. After that, Y and Z start interacting with each other, both cooperate in the first interaction and consequently $DIT_{Y,Z} = 0.1$ and $DIT_{Z,Y} = 0.1$. Moreover, cooperations of these two agents together is taken into account for updating $IIT_{Y,X}$ and

$IIT_{Z,X}$ as the indirect perception. Thus, we have $IIT_{Y,X} = 0.2$ and $IIT_{Z,X} = 0.0$. After the second cooperation, we have $IIT_{Y,X} = 0.3$ and $IIT_{Z,X} = 0.1$. Therefore, after several cooperations, $IIT_{Y,X}$ and $IIT_{Z,X}$ have high values indicating that X is trustworthy in introduction interaction from the perspective of Y and Z. Note that the value of $IIT_{Z,X}$ will be slightly higher than the value of $IIT_{Y,X}$ because of direct perception. Direct perception is taken into account for calculation of IIT to give some consideration to the intention of the introducer.

4.2. Reputation Models

The four trust variables explained in Section 4.1 are the result of cooperation/defection of the neighbors of the agent in different aspects of direct, witness, reporting, and introduction interactions. These variables are used by the agent to model the trustworthiness of their neighbors to understand whether the given agent should maintain its connection with them or how much of the information received by the agent is reliable. On the other hand, agents need to predict the trustworthiness of those agents with whom they have never interacted. Therefore, we use reputation models for predicting the trustworthiness of these agents. These reputations are calculated based on the information (report or witness information) received from an agent's neighbors and the related trust variable.

A reputation variable is defined by $R_{i,j}(t)$ indicating the trust rating assigned by agent i to agent j after receiving t pieces of information (report or witness information), while $R_{i,j}(t) \in [-1, +1]$ and $R_{i,j}(0) = 0$.

We have defined two kinds of reputation: (1) Report-based Reputation (RR) and (2) Witness-based Reputation (WR). The former is calculated based on the reports received by an agent from its neighbors and the latter is computed relying on witness information received by the agent from its neighbors.

It should be noted that Sections 4.2.1 and 4.2.2 describe example updating schemes which have been used in analysis using the framework (e.g., Salehi-Abari and White 2009a). They are included to demonstrate the coupling between reputation variables and perceptions (cooperation/defection); others may be used.

4.2.1. Report-Based Reputation. RR is calculated based on report information. As explained in Section 3.1, a report conveys the result of an interaction of a neighbor with one of its own neighbors. An agent will store the information of a report in $rep_{i,j}(t)$, showing the result of the t^{th} reported direct interaction of agent i with agent j while the result can be cooperation or defection. For each report, two values of $rep_{i,j}(t)$ and $rep_{j,i}(t)$ will be stored by the report listener. Suppose that agent i has received a report message from agent j regarding the interactions of agent j and k, the agent i (recipient of the report) will calculate the Estimated DIT, $EDIT_{j,k}(t)$, of agent k from the perspective of agent j based on $rep_{k,j}(t)$. $EDIT_{j,k}(0) = 0$ and

$$EDIT_{j,k}(t+1) = \begin{cases} EDIT_{j,k}(t) + \alpha_E(i)(1 - EDIT_{j,k}(t)) & EDIT_{j,k}(t) > 0, rep_{k,j}(t+1) = CDI \\ (EDIT_{j,k}(t) + \alpha_E(i))/(1 - \min(|EDIT_{j,k}(t)|, & EDIT_{j,k}(t) < 0, rep_{k,j}(t+1) = CDI \\ \quad |\alpha_E(i)|)) & \\ (EDIT_{j,k}(t) + \beta_E(i))/(1 - \min(|EDIT_{j,k}(t)|, & EDIT_{j,k}(t) > 0, rep_{k,j}(t+1) = DDI \\ \quad |\beta_E(i)|)) & \\ EDIT_{j,k}(t) + \beta_E(i)(1 + EDIT_{j,k}(t)) & EDIT_{j,k}(t) < 0, rep_{k,j}(t+1) = DDI \end{cases} \quad (7)$$

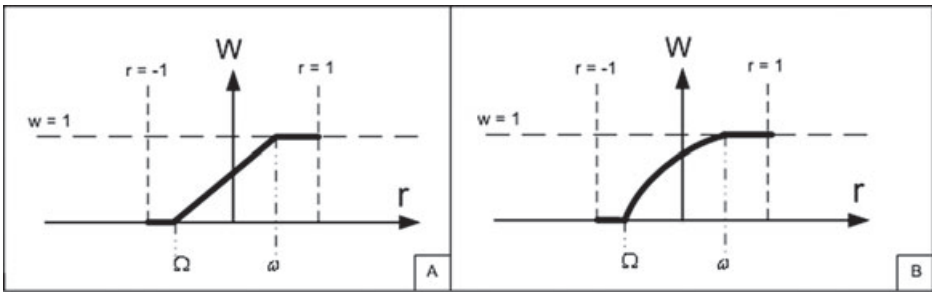


FIGURE 4. Demonstration of (A) $\phi_{Li}(t)$ and (B) $\phi_{Lo}(t)$ converter functions.

where $1 > \alpha_E(i) > 0$ and $-1 < \beta_E(i) < 0$ are positive evidence and negative evidence weighting coefficients, respectively. The RR of agent k from the perspective of agent i, $RR_{i,k}(t)$, is calculated by

$$RR_{i,k}(t) = \frac{\sum_{j \in neighbor(i)} (\phi(RIT_{i,j}(t)) \times EDIT_{j,k}(t))}{\sum_{j \in neighbor(i)} (\phi(RIT_{i,j}(t))} \tag{8}$$

where $neighbor(i)$ includes the neighbors of agent i and $RIT_{i,j}(t)$ is the RIT of agent j from the perspective of agent i after receiving t reports. $\phi(r)$ is the converter function that maps the values of a trust variable to the weights in the range of [0, 1] with regard to the related ω and Ω. We present two variants of converter functions, *Linear* and *Logarithmic* denoted by $\phi_{Li}(r)$ and $\phi_{Lo}(r)$, respectively. Figure 4 shows these two different types of converter function.

$$\phi_{Li}(r) = \begin{cases} 0 & -1 \le r < \Omega \\ \dfrac{r - \Omega}{\omega - \Omega} & \Omega \le r \le \omega \\ 1 & \omega < r \le 1 \end{cases} \qquad \phi_{Lo}(r) = \begin{cases} 0 & -1 \le r < \Omega \\ \dfrac{\log(r - \Omega + 1)}{\log(\omega - \Omega + 1)} & \Omega \le r \le \omega \\ 1 & \omega < r \le 1. \end{cases} \tag{9}$$

4.2.2. Witness-Based Reputation. WR for a specific agent is calculated based on the ratings of other agents. As explained in Section 3.1.2, this rating can be based on the observed interactions or direct interactions. Note that if one agent rates another agent based on the direct interaction, other agents should also rate based on the direct interaction. The type of rating is a convention between agents of the same community. The asking agent stores the ratings of other agents in an $Opinion$ variable. $Opinion_{j,k}$ shows the rating issued by agent j regarding agent k. The value of this variable is in the range of $[-1, 1]$. WR of agent k from the perspective of agent i after reception of t opinions (ratings) is denoted by $WR_{i,k}(t)$ and calculated by

$$WR_{i,k}(t) = \frac{\sum_{j \in OpinionSenders} (\phi(WIT_{i,j}) \times Opinion_{j,k})}{\sum_{j \in OpinionSenders} \phi(WIT_{i,j})} \tag{10}$$

where the $OpinionSenders$ variable includes indices of the neighbors of agent i who sent their ratings about agent k and $WIT_{i,j}$ is the current value of WIT variable of agent j from the perspective of agent i. Note that $\phi(r)$ is a converter function as previously explained in Section 4.2.1.

4.3. Policies and Strategies

The required perception model (trust and reputation variables) in the DART framework have now been introduced; these variables help agents perceive the cooperation/defection of other agents situated in their environment in four dimensions and consequently to determine the trustworthiness of other agents. Incontrovertibly, this perception provides agents with the foundation for making decisions as with whom they should interact. This perception is absolutely necessary but not sufficient for a trust model since agents need to decide how and when they should interact with other agents. In this sense, each agent requires policies to help them in making decisions for their interactions with the other agents. Different types of policies are introduced and explained in the following subsections.

4.3.1. Direct Interaction Policy (DIP). This type of policy assists an agent in making decisions regarding its direct interactions while this decision might be made based on the trust perception of the agent. For example, *malicious* agents might have an *unconditional defection* policy which means they defect in interactions with any other agents regardless of its trustworthiness level. They might have a more complicated policy while cooperating with a group of agents and defecting in interactions with the remainder of an agent society (i.e., colluding).

4.3.2. Witness Interaction Policy (WIP). This type of policy exists to aid an agent in making three categories of decisions related to its witness interactions. First, agents should decide how to provide the witness information for another agent on receiving a witness request. Should they manipulate the real information and forward false witness information to the requester (an example of defection) or should they tell the truth? The second decision made by the WIP is related to when and from whom the agent should ask for witness information. Should the agents ask for the witness information when it has a connection request from an unknown party? Should the agents ask for witness information from a subset or all of its neighbors? The third decision is on how agents should aggregate the received ratings. For example, should the agent calculate the simple average of ratings or a weighted average of ratings?

We defined three subwitness interaction policies: answering policy (AP), querying policy (QP), and information-gathering policy (IGP). AP intends to cover the first category of decisions mentioned earlier while QP and IGP apply to the second and third categories, respectively.

4.3.3. Reporting Interaction Policy (RIP). This type of policy controls the flow of reports to neighbors. An agent, based on this policy, decides how to select important news and broadcast it to their neighbors. Moreover, the agents decide whether or not to report the interactions honestly or not or whether to hide the news from its neighbors. As with other policies, each agent can cooperate or defect in reporting. We defined two subreporting interaction policies: reporting policy (RP) and report-gathering policy (RGP). RP intends to cover how the agent should select the news to report and how the agent should report the past interactions (e.g., being honest or lying). On the other hand, the RGP focuses on how to aggregate the reports and how to make decisions based on the reports.

4.3.4. Introduction Interaction Policy (IIP). Agents can have different behaviors for Introduction Interaction; for example, an agent might introduce two trustworthy agents or try to introduce trustworthy agents to untrustworthy ones. Furthermore, agents need to decide if

they should connect to the agents recently introduced. These types of decisions are all made by an IIP. Note that for making this type of decision different trust or reputation variables might be taken into account.

Algorithm 1 An Example for Introduction Interaction Policy

```
{Suppose that the agent i is executing this code}
for all j, k ∈ Neighborhood do
  if DIT_{i,j}(t) ≥ ω_i^{DIT} and DIT_{i,k}(t) ≥ ω_i^{DIT} and shouldBeIntroduced(j, k) then
    Introduce k to j
    Introduce j to k
  end if
end for
if i receives introduction of agent k from agent j then
  if IIT_{i,j}(t) ≤ Ω_i^{IIT} then
    Reject the introduction
  else
    ConnectTo(k)
  end if
end if
```

An example of a simple IIP is demonstrated in Algorithm 1. This policy attempts to connect trustworthy agents (in direct interactions) if the primitive *ShouldBeIntroduced(j,k)* returns a *true* value.

ShouldBeIntroduced(*j,k*) can be easily implemented relying on whether agent *j* and agent *k* have been previously introduced to each other or not; if yes, the primitive returns false otherwise a true value will be returned. The second if statement of this policy deals with the introductions received by the agent. If the introducer agent is *Untrustworthy* in terms of introduction interactions ($IIT_{i,j}(t) \leq \Omega_i^{IIT}$) then the introduction will be rejected; otherwise the agent will connect to the introduced agent.

4.3.5. Connection Policy. This type of policy assists an agent in making decisions regarding whether it should make a request for a connection to other agents and whether the agents should accept/reject a request for a connection. There is an internal property for this kind of policy called *Socializing Tendency* (*ST*) which dramatically affects decisions for making a connection request and the acceptance of the connection request. Those agents with higher ST values tend to connect with a higher number of agents as opposed to the conservative agents with lower ST values.

4.3.6. Disconnection Policy (DP). DP aids an agent in deciding whether or not it should drop a connection to a neighbor. For example, this policy can decide to whether or not to disconnect from a given agent based on the agent's trustworthiness in direct interactions.

5. RELATED WORK

This section reviews related work in three ways. First, testbeds are described indicating where important characteristics provided by DART are missing. Second, several important trust and reputation models are reviewed. Several of these models are subsequently assessed for vulnerabilities in the research discussed in Section 6. Finally, attacks and resistance to

them are discussed. Results of aspects of the attack resistance described are also discussed in Section 6.

5.1. Trust and Reputation Environment Models (Testbeds)

Open distributed systems can be modeled in open MAS that are composed of autonomous agents that interact with one another using defined policies. We here describe some existing testbed environments in which agents' interactions with their peers take place. It is worth mentioning that despite the fact that many trust models have been recently proposed, a general trust evaluation testbed does not exist. In this sense, we are interested in the testbeds which do not have any barriers for real-world implementation and still are generic.

5.1.1. IPD. The IPD (Axelrod 1984) has been used as a testbed for evaluation of trust and reputation models and strategies (Axelrod 1984; Schillo, Funk, and Rovatsos 2000; Mui et al. 2002a,b). Utility is a metric used for the comparative assessment of trust and reputation models. Although IPD is suitable for direct interaction modeling, it has several shortcomings for trust and reputation modeling: First, as agents evaluate one aspect of opponents' behavior, multidimensional trust modeling is not encouraged. Second, agents cannot separate untrustworthy agents because they have to interact with all other agents. Third, it suffers from the lack of system-level metrics and only focuses on total utility of the agent. In Section 6, we will discuss how our proposed game-theoretic testbed overcomes these drawbacks while maintaining the simplicity of the IPD.

5.1.2. SPORAS Testbed. The SPORAS experiments (Zacharia 1999) have been widely used for the evaluation of trust and reputation models. For instance, Regret (Sabater and Sierra 2001), AFRAS (Carbo, Molina, and Davila 2002; Rubiera, Molina, and Muro 2003) have used this set of experiments. The SPORAS experiments evaluate reputation models by measuring the time taken for them in electronic marketplaces to converge to true reputations. However, these experiments suffer from the following shortcomings. First, the experiments only employ one single-agent metric and ignore the system-level metrics. Second, they do not consider multidimensional trust models. Third, while this set of experiment focuses on trust model accuracy, and adaptivity, it ignores the agent's capability in making decisions based on trust, such as determining whether or not to disconnect from untrustworthy agents. The consideration of disconnection/connection for agents is important as human society possesses this capability; i.e., SPORAS fails to take account of the social network present in artificial societies.

5.1.3. Agent Reputation and Trust (ART). Fullam et al. (2005) introduced the ART Testbed which serves two roles: (1) as a competition platform in which researchers can compare their trust and reputation models against objective metrics, and (2) as a suite of flexible tools, allowing researchers to perform experiments. In ART, agents use trust strategies to exchange expertise with others to appraise paintings. Agents make money by appraising the paintings while more accurate appraisal results in better income. Furthermore, those agents that appraise a painting more accurately will receive more paintings to appraise in the future. However, agents' expertise is limited to appraising a subset of the paintings. They are required to exchange expertise with other trustworthy agents. Moreover, agents can also exchange trust values with others (Witness Interaction). The agent who has the most money at the end wins the game.

ART does not provide modeling of a Reporting Interaction and an Introduction Interaction and the corresponding dimensions of trust for agents. Moreover, modeling certain

exploitation classes (e.g., collusion) is hard in ART because each trust strategy controls a single agent, which works in competition against every other agent in the system. Furthermore, ART suffers from a lack of tools and metrics for demonstrating the structure of the social network. There is no limitation in ART regarding agent communication (i.e., all agents can communicate with one another) as opposed to our environment model in which only neighbor agents are allowed to interact and communicate with each other. Once again, the social network is ignored.

5.1.4. Kerr and Cohen. Most recently, Kerr and Cohen (2009a) have introduced an experimental testbed for evaluation of trust and reputation systems used in e-commerce marketplaces. Using the proposed testbed, the researcher can examine the performance of agents in different e-commerce marketplaces while each agent can employ a specific trust and reputation system. Profit and number of sales are two metrics that can be used for the comparative assessment of trust and reputation models. Although the proposed testbed is flexible enough to model different e-commerce marketplaces with heterogeneous buyers and sellers using various trust and reputation models, the testbed is not generic enough for other domains (e.g., peer-to-peer systems). Because of specialization in marketplaces, the proposed testbed has many environmental parameters (e.g., number of sellers and buyers, the number of products, etc.) that influence the performance of agents in marketplaces. As a result, the diagnosis of the vulnerabilities that exist in the trust and reputation models is not straightforward since the vulnerabilities can be connected to the marketplace (environment model) or the trust and reputation model used. Being a market-oriented model, the social network created and maintained by agents is considered secondary. The importance of the social network is considered fundamental to the proposal presented in this paper.

Reporting Interaction and Introducing Interaction and the corresponding dimensions of trust are not considered in this testbed. Moreover, the proposed testbed does not consider the agent social network, connection and disconnection of agents which is important as human society and most of open distributed systems (e.g., peer-to-peer systems) possesses this capability. In their testbed, there is no limitation on agent communication (as any agent may communicate with any other) as opposed to our environment model in which only neighbor agents are allowed to interact and communicate with each other.

5.2. Trust and Reputation Models

The body of research on trust and reputation models is large; a review of which can be found in Ramchurn et al. (2004), Artz and Gil (2007), Sabater and Sierra (2005), and Josang et al. (2007). In this section, we limit our discussion to some popular centralized models and concentrate on the decentralized trust models incorporating multiple information sources which are the main focus of this paper.

Steve Marsh was among the first to introduce a computational trust model for a distributed artificial intelligent society. His model attempted to integrate aspects of trust taken from sociology and psychology. His model takes into account only an agent's own experiences (direct interaction) while differentiating three types of trust: Basic Trust, General Trust, and Situational Trust (Marsh 1994). Because Marsh's model is based on sociological foundations, the model is too complex to be easily used in today's MAS. Moreover, the model only considers an agent's own experiences and does not involve any social mechanisms. Hence, a group of agents cannot collectively build up a reputation for others. All agents can interact with each other without any constraint. We consider this lack of a neighborhood to be significant limitation of this work.

Zacharia and colleagues have suggested *Sporas* and *Histos* systems for reputation management (Zacharia, Moukas, and Maes 1999; Zacharia and Maes 2000). Reputation in Sporas extends the reputation management systems used in eBay and Amazon by introducing a new method for rating aggregation. Briefly, once a rating is received it updates the reputation of the involved party instead of storing all ratings and calculating the average. However, Sporas is not suitable for open distributed systems because of its centralized design. Moreover, the Sporas experiments neither account for multidimensional trust, nor do they measure an agent's ability to make trust-based decisions, leading to isolation of untrustworthy agents.

Histos was developed to compensate for the lack of personalization that Sporas reputation values deal with. This model covers direct interaction and witness information where a value for reputation is assigned by each individual. The main weakness of this model is the simultaneous use of the reputation value of an individual also as reliability of the provided information by that agent. If the agent is reliable in direct interaction, it does not mean that it has to be also a trustworthy witness. As a result, Sporas is vulnerable to collusion attacks (see Section 6 for a discussion on this point).

The Beta Reputation System (BRS) (Josang and Ismail 2002) is a probabilistic trust model that works based on the beta distribution. The system is centralized and designed to meet the requirements of online communities. In BRS, users rate the performance of other users by providing either negative or positive feedback. The feedback values are then used to calculate shape parameters of the user's reputation. In other words, the beta distribution takes two parameters: a count of past honest (positive) interactions (feedback) and a number of past dishonest (negative) interactions (feedback). However, BRS does not show how it is able to cope with misleading (inaccurate) information. Whitby et al. (2004) extend BRS and show how it can be used to filter out unfair or inaccurate ratings by using an endogenous method, where the agent's ratings are discarded if they are statistical outliers. However, their approach is only effective when a significant majority of available reputation sources are fair and accurate. BRS and its variants are not suitable for open distributed systems because of their centralized designs.

Tran and Cohen (2004) proposed a marketplace model and learning algorithms for buying and selling agents in electronic marketplaces. By considering the possible existence of dishonest selling agents in the market, learning agents employed a trust model to distinguish untrustworthy agents and prevent from interacting with them. This work is representative of a direct experience model: agents make use only of their own experience in evaluating the trustworthiness of others. Each agent maintains pairs of expected outcomes and possible actions and then selects an action among the possible actions to maximize the expected value. After an action is taken, the real outcome is used to update the expected outcome for that action. Gradually, the buyer will learn which agents are trustworthy to interact with for a given product. However, this work only focuses on DIT and does not consider any reputation mechanism.

Mui et al. (2002b) discuss the strength of the various notions of reputation using a simple simulation working based on evolutionary game theory. This work focuses on the strategies of each agent only for direct interaction, and does not consider gathering reputation information from other parties in the network or any other social mechanism. Moreover, they review existing works on reputation among diverse domains such as distributed artificial intelligence, economics, and evolutionary biology.

In another work, Mui et al. (2002a) have proposed probabilistic models for reputation which use Bayesian statistics. Reputation for an agent is inferred based on propagated ratings from an agent's neighbors. In their probabilistic trust model, they show that if the number of interactions is too low then trust cannot be built. They calculate the probability of an agent being trustworthy on the next interaction by considering the frequency of positive and

negative direct impressions gathered from the social network. This work does not take into account witness-based collusion and is vulnerable to the con-man attack (Salehi-Abari and White 2009b).

Regret (Sabater and Sierra 2001) is a decentralized trust and reputation system designed for e-commerce environments. The system takes into account three different sources of information: direct experiences, information from third-party agents, and social structures. The direct trust, witness reputation, neighborhood reputation, and system reputation are introduced in Regret where each trust and reputation value can have an associated reliability measure. This measure tells the agent how confident the system is regarding that value according to how it has been calculated. The reliability value is calculated from the number of ratings taken into account in producing the trust values and the deviation of these ratings. However, this model still suffers from the malicious witness providing false reputation and as a result is vulnerable to collusion attacks. Moreover, apart from the direct trust component, the remaining model is not readily applicable because it is not obvious how each agent can build the social network on which Regret depends.

Yu and Singh developed an approach for social reputation management in which they represented an agent's ratings regarding another agent as a scalar and combined them with testimonies using combination schemes similar to certainty factors (Yu and Singh 2000). The drawbacks of the combination model led Yu and Singh to consider an alternative approach (Yu and Singh 2002); specifically, an evidential model of reputation management based on the Dempster–Shafer theory.[1] This model represents the agent's belief (probability) that a partner will cheat, and the probability that it will not cheat. Moreover, the model also explicitly represents the agent's lack of belief in those outcomes. In this model, an agent relies on its own experience if it is sufficient. If not, it asks for the opinions of others using a "TrustNet." An agent can solicit information from its neighbors when needed. If the neighbor cannot provide information, it may refer the agent to one of its own neighbors. Actually, they use direct information and witness information while not combining these two types together. In this model, there are two kinds of information that a witness can provide when it is asked about another agent: (1) rating about the queried agent if it is the neighbor of the given agent, or (2) referral to another agent. However, malicious witnesses and collusion attacks are not considered in either of these two proposed models.

Yu, Singh, and Sycara (2004) have proposed a trust model in large-scale peer-to-peer systems in which each peer has its own a set of acquaintances. A subset of these acquaintances is identified as its neighbors. A peer maintains a model of each acquaintance. The acquaintance's reliability and credibility are included in this model. Reliability is used for providing high-quality services whereas credibility is used for providing trustworthy ratings to other peers. The weighted majority algorithm (WMA) is adapted to predict the trustworthiness of an agent based on the set of testimonies from the witnesses. The focus of this work is more in peer-to-peer systems and it does not model the reporting interaction and introduction interaction which are presented in this paper.

Jurca and Faltings (2002) introduce a reputation mechanism in which agents report truthfully about their interactions' results to the set of broker agents called R-agents. R-agents specialize in buying and aggregating reports from other agents and selling back reputation information to them when they need it. The reputation for a specific agent is simply calculated by averaging the reports related to that agent. In spite of a distribution of R-agents in the system, the reputation mechanisms should not be regarded as completely decentralized

[1] Dempster – Shafer Theory (Kyburg 1987) is founded on the fact that there is no causal relationship between a hypothesis and its negation. In this light, lack of belief does not mean disbelief and reflects a state of uncertainty.

mechanisms because regular agents are still dependent on R-agents for acquisition of a specific agent's reputation. Although a payment scheme for reputation reports is proposed, motivating agents to share their reports truthfully, this method does not work if most agents lie about the reports or if they collude in giving false reports. In these scenarios, the reputation score will be incorrect since the trustworthiness (reliability) of the reporter is not taken into account where the reputation is calculated by simple averaging of reports in this model. As a result, the model is vulnerable to collusion attacks. Moreover, newcomers are not modeled when there is an assumption that information agents already store some reputation information. This assumption is the result of one rule of the system, allowing agents to sell a report for an agent when they have previously bought reputation information for that agent. Direct interaction and witness interaction are also not addressed in this model.

Huynh, Jennings, and Shadbolt (2004, 2006) proposed a trust and reputation model called FIRE that integrates a number of information sources to estimate the trustworthiness of an agent. Specifically, FIRE incorporates interaction trust, role-based trust, witness reputation, and certified reputation to provide a trust metric. The interaction trust and witness reputation are the result of the past experience of direct interaction and reports of witnesses about an agent's behavior, respectively. Role-based trust is defined by different role-based relationships between agents whereas certified reputation is built from the third-party references which are provided by agents themselves. There are two assumptions in FIRE which makes it vulnerable to collusion attacks: (1) Agents have a tendency to share their experiences with one another, and (2) Agents are honest in exchanging information. In other words, FIRE does not consider the existence of malicious witnesses or reporters in its environments and consequently colluding is not considered. Moreover, the interactions between agents are not confined to an agent's neighborhood as any agent can interact with any other.

In the Social Interaction Framework (SIF) (Schillo et al. 2000), agents are playing a Prisoner's Dilemma set of games with a partner selection phase. Each agent receives the results of the game it has played plus the information about the games played by a subset of all players (its neighbors). An agent evaluates the reputation of another agent based on observations as well through other witnesses. However, the reporting component of this work is completely centralized as opposed to our requirement for a decentralized reporting component. Moreover, there is an assumption that reported interactions are not manipulated or spurious. As a result, it is vulnerable to collusion attacks. The SIF does not describe how to find witnesses, whereas in electronic communities deals are broken among people who often would never have met each other.

Sen and Sajja (2002) model reputation using both direct interaction and observed interaction. Observations are noisy with noise modeled using a Gaussian distribution and may differ from the actual performance. One trust variable is considered for both sources of information and reinforcement learning is used to update the value of that variable. Due to the noise in observations, the rule used to update the reputation value for direct interaction has a greater effect than the rule used to update the value for an observation. Agents can query other agents about the performance of a given agent and the response is a Boolean value that says if the partner is good or not. The subset of agents to be queried is selected randomly from the set of possible witnesses. In this model, although the existence of liars is assumed, the liar should lie consistently and the number of them should be less than half of the population of agents. Agents only use witness information to make decisions while direct experiences are only used as pieces of information to be communicated to others. However, this model is vulnerable against collusion attacks because it uses the same trust variable for observations and direct interactions. The reporting mechanism is centralized and the amount of observational noise is not defined clearly. Moreover, each agent should have a priori knowledge of the percentage of liars in the population to calculate the necessary numbers

of witness queries. This work focuses more on calculation of the number of witnesses to get rid of liars' information instead of aggregation of witness information with the existence of liars.

5.3. Cheaters, Inaccurate Witnesses, and Exploitation

Most recently, researchers have become aware of the existence of exploitation in the artificial societies employing trust and reputation models (Kerr and Cohen 2009b; Salehi-Abari and White 2009b,c), and the existence of inaccurate witnesses (Dellarocas 2000; Yu and Singh 2003; Whitby et al. 2004) and naive agents (Salehi-Abari and White 2009a). It is this research that has motivated the framework proposal described in this paper. We will review these works later in this section.

5.3.1. Con-Man Attack and the Con-Resistant Model. The con-man attack introduced and modeled by Salehi-Abari and White (2009b) has been applied to direct trust components of trust models. In the con-man attack, a con-man usually takes advantage of someone else and attempts to defraud that person by gaining their confidence. Salehi-Abari and White (2009b) have demonstrated how a con-man can exploit three well-known trust models Yu and Singh (2000), Regret, and FIRE such that he/she is still known as trustworthy after repeated cycles of interaction while conning others. They utilize DART to demonstrate these vulnerabilities. They introduced two characteristics of con-resistant trust models: first, cautiously increment trust after having seen any defection and second, larger punishments after each defection. Based on the proposed features, they proposed a con-resistant scheme, called AER, and empirically demonstrated its utility using DART.

In other related work, Salehi-Abari and White (2010) mathematically analyze the interaction of the con-man in five trust models: Yu and Singh, Regret, FIRE, Probabilistic trust models, and AER. They have proven that simple con-man agents (CAs) can exploit Yu and Singh's trust model, Regret, FIRE, and probabilistic trust models regardless of the model's parameters. However, AER has been shown to be exploitation resistant to the con-man attack using the definitions provided in Section 2. Furthermore, CAs will have to increase the number of cooperations in each and every cycle with AER to achieve a specific trust value.

5.3.2. Smart Cheaters in Marketplaces. Kerr and Cohen (2009b) examined the security of several e-commerce marketplaces each of which employs a specific trust and reputation system. To this end, they first proposed Proliferation, Reputation Lag, Re-entry, and Value Imbalance attacks and then evaluated them on each marketplace. In the proliferation attack, the seller simply opens a number of accounts, and tries to sell the same product through each of them. As a consequence, the attacker will have more opportunities to sell her product. The Reputation Lag attacker is the seller who behaves honestly for 45 days and cheats for 15 days (the lag before an act of cheating impacts reputation) and then leaves the marketplace. A Re-entry attacker simply opens an account to cheat other agents, then leaves the account to open another. A Value Imbalance attacker is a trustworthy seller on small transactions to gain reputation, but a cheater on the large ones to gain extra profit.

This work measures the vulnerability of a specific marketplace against each of the proposed attacks based on the percentage of monetary profit that strategic cheaters make when compared to honest sellers' profits. Apparently, the higher this percentage is, the lower the security of that specific marketplace. However, it is not straightforward to conclude that these vulnerabilities are connected to the environment model and marketplace or the trust and reputation model used. This is mainly because the attacks were not mounted directly against the trust and reputation model but rather mounted against the marketplace embracing

the agent policies (behaviors), trust models, and rules for participation in that marketplace. Moreover, some proposed attacks (e.g., Proliferation and Re-entry) cannot be classified as attacks against trust and reputation models as they deal with issues of agent identity and authentication; issues outside of the scope of this paper.

5.3.3. Witness-Based Collusion Attacks. Witness-based collusion attacks (Salehi-Abari and White 2009c) degrade the value of DIT in trust-aware agent (TAA) societies. In these attacks, Enticer agents which are trustworthy in their direct interactions, collude with malicious agents by providing a good rating for them. After formalizing the Witness-based Collusion Attack in DART, Salehi-Abari and White (2009c) experimentally show how a unidimensional trust model is vulnerable against witness-based collusion attacks. This vulnerability results in TA_w agents, which use a unidimensional trust model to weight the ratings, exposing themselves to a higher level of encounter risk. Furthermore, TA_w^+ agents, by using WIT, weight the rating of Enticer agents and decrease the impact of them in their final assessment. This results in exposing themselves to a lower level of risk in their interactions. Finally, they conclude multidimensionality can significantly increase resistance against witness-based collusion attacks.

5.3.4. Coping with Inaccurate Witness Information. The general solution to dealing with inaccurate witness information is to ignore or reduce the effect of unreliable opinions. There are two basic approaches to judging the accuracy of opinions. These are referred to as endogenous and exogenous methods by Josang et al. (2007). The former tries to detect unreliable witness information (opinions) by using the statistical properties of the reported opinions; for example, Whitby et al. (2004) and Dellarocas (2000). The latter relies on other information such as the reputation of the source or the relationship with the trustee such as described in the work of Yu and Singh (2003).

5.3.5. Naive Agents. Salehi-Abari and White (2009a) introduced the concept of a naive agent. Naive agents are naive in terms of deciding how, when and with whom to interact while always cooperating with other agents. They analyzed the impact of this agent class on agent societies using a game-theoretic model built on the DART platform. Their results demonstrate that naive agents help malicious agents survive by cooperating with them directly (by providing good services) and indirectly (by giving a good rating for them).

6. DISCUSSION

This section presents a highlight of the research which are conducted by DART and reviews the desirable properties of DART.

6.1. Research Highlights Using DART

6.1.1. Witness-Based Collusion Attack. As mentioned in Section 5.3.3, Salehi-Abari and White have modeled the witness-based collusion attacks by using Enticer and malicious agents (Salehi-Abari and White 2009c). In addition to these two types of agents, the agent society includes TAAs which are equipped with perception model (trust and reputation variables) and with policies. They have defined two classes of TAAs: Trust-Aware (TA_w) and Trust-Aware$^+$ (TA_w^+) where TA_w uses a unidimensional trust model as opposed to TA_w^+ which uses a multidimensional trust model (see Salehi-Abari and White (2009c) for the details of specification and implementation of these four agents in DART).

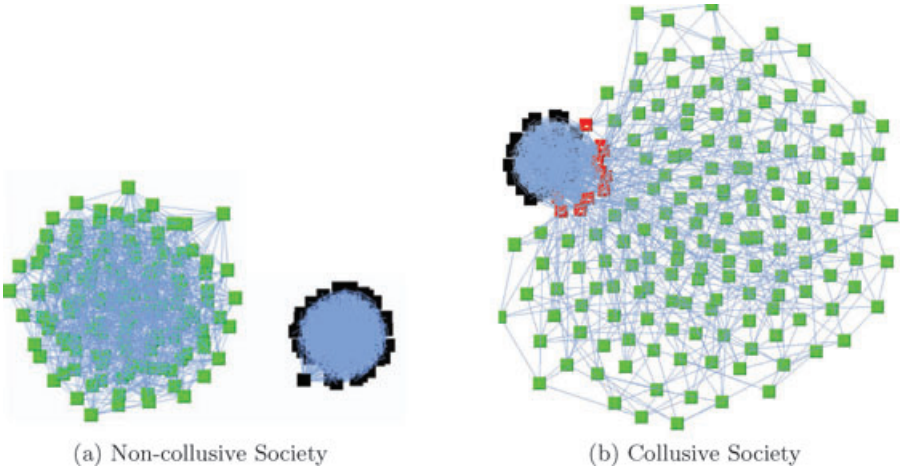


FIGURE 5. The final society structure in Experiment 1.

Experiment 1. They have run two simulations of 200 agents for this experiment. In the first simulation, which models a noncollusive society, TA_w agents comprise 75% of the population and the rest are malicious agents (for our convenience, we refer to this simulation as Sim1). The second simulation represents the witness-based collusive society in which Enticer and malicious agents comprise 5% and 20% of the populations, respectively, and the rest are TA_w agents (for our convenience, we refer to this simulation as Sim2). The objective of this experiment is to understand the effect of a witness-based collusion attack on the structure of agent society and on the level of encounter risk. Encounter risk is defined to be linearly related to the average number of dropped connections.

The structures of the agent society after 400 time steps for Sim1 and Sim2 are presented in Figure 5 where TA_w agents and malicious agents are in green (light gray in white–black print) and in black, respectively. Red squares with white "-" represent Enticer agents. In noncollusive societies as shown in Figure 5(a), we have two isolated groups of TA_w and malicious agents. In the witness-based collusive society (see Figure 5b), we could not achieve separation of malicious and TA_w agents seen in Sim1. Because TA_w agents perceived Enticer agents as trustworthy agents in direct interaction thus they maintain their connections with Enticer agents.

Figure 6 illustrates $\overline{D}$ of TA_w over the course of two simulations of Sim1 and Sim2. TA_w agents in Sim1 (noncollusive society) have considerably fewer dropped connections when compared to the TA_w agents in Sim2 (witness-based collusive society). In this sense, TA_w agents expose themselves to higher level of risk of being exploited by malicious agents in Sim2 as a result of ongoing witness-based attacks, when compared to Sim1. This high level of risk is due to the fact that each TA_w agent is surrounded by Enticer agents, resulting in receiving more manipulated opinions about other malicious agents while the senders of all opinions are trustworthy in terms of direct interactions.

Experiment 2. We have run two simulations of 200 agents for this experiment, in each of which Enticer, malicious agents are 5% and 20% of the populations, respectively. The remainder of the population (75%) is either TA_w or TA_w^+. Both TA_w and TA_w^+ benefit from using the Conservative CP and WIPs for inquiring about the trustworthiness of the connection requester from neighbors. Note that these two types employ various witness

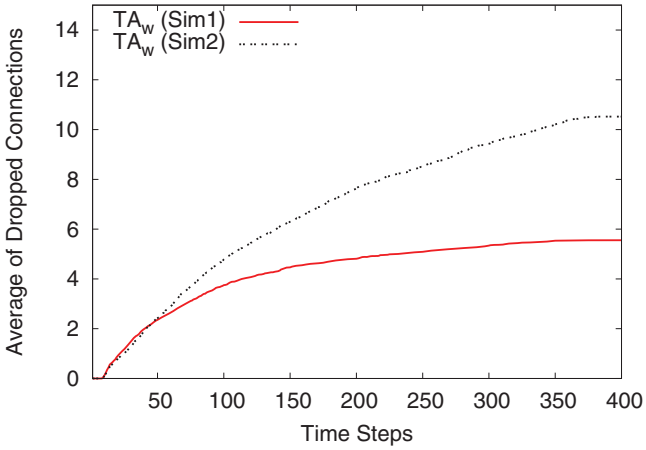


FIGURE 6. $\overline{D}$ of agent types over simulation.

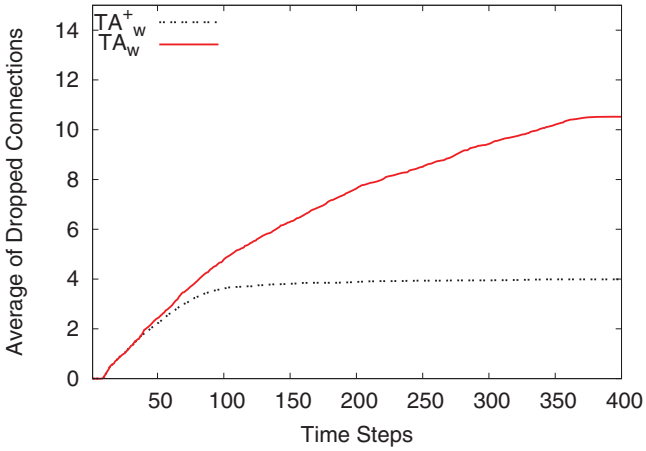
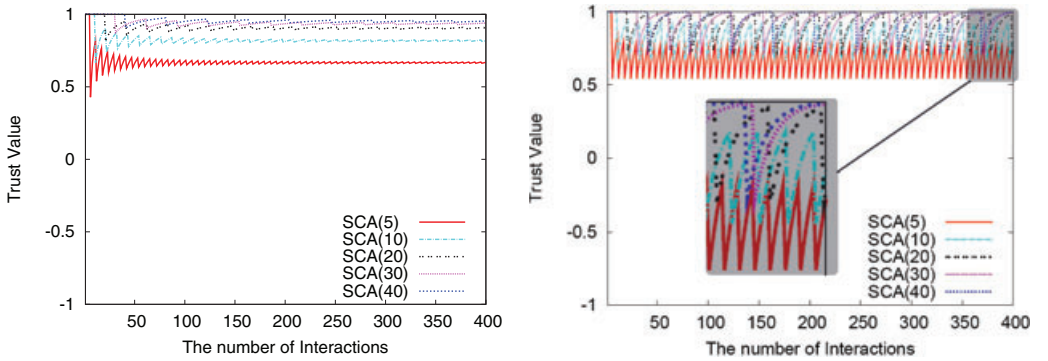


FIGURE 7. $\overline{D}$ of agent types over the simulation.

information-gathering policies and different trust models. TA_w utilizes a unidimensional trust model (i.e., DIT), whereas TA_w^+ utilizes a multidimensional trust model (i.e., DIT and WIT).

This experiment is intended to demonstrate the benefit of using multidimensional trust where there are witness-based collusion attacks. More precisely, the intention behind this experiment is to show that TA_w^+ agents by using multidimensional trust and appropriate WIPs (e.g., WTW) can decrease the impact of Enticer and malicious agents (colluding groups) on aggregating the reputation ratings. As a result, the TA_w^+ agents can decide more reliably regarding the trustworthiness of other agents and expose themselves to a lower level of risk. As shown in Figure 7, TA_w^+ agents have considerably fewer dropped connections when compared to TA_w. Policies used by this agent type result in successful acceptance/rejection of connection requests. In this sense, TA_w^+ agents expose themselves to smaller numbers of untrustworthy agents and consequently lower the level of risk of being exploited by these agents.



(a) Exploitation of Regret (b) Exploitation of FIRE

FIGURE 8. Exploitation of Regret and FIRE by a con-man agent.

6.1.2. CA and Trust Models. As mentioned in Section 5.3.1, the con-man attack introduced and modeled in Salehi-Abari and White (2009b) has been applied to direct trust components of trust models. All simulations were run with two agents in DART, one TAA which utilizes a specific trust model (e.g., Regret, FIRE, and Yu and Singh) and a CA. The direct interaction policy of TAAs is tit-for-tat which starts by cooperation and then imitates the opponent's last move. The direct interaction policy of CAs follows the formal language presented in Salehi-Abari and White (2009b) which is solely dependent on the parameter θ. The strategy of a CA is denoted by SCA(θ).

The experiments support the hypothesis on how a con-man can exploit three well-known trust models Yu and Singh (2000), Regret, and FIRE such that he/she is still known as trustworthy after repeated cycles of interaction while conning others. The vulnerability of regret and FIRE are demonstrated in Figure 8.

Figure 8(a) shows the trust value variation of TAA over the 400 interactions. It is clear that the con-man with SCA(5) can stabilize its trust value at 0.66. Moreover, by increasing θ to 10, 20, 30 and 40, the CA can reach a trust value of 0.81, 0.90, 0.93, and 0.95, which are high values of trust for a con-man; i.e., the agent is considered trustworthy. Figure 8(b) shows the variation of trust when FIRE modeled is exploited. Although FIRE is more sensitive to defection when compared to Regret as a result of its enhanced rating recency function, it is still vulnerable to the con-man attack.

As mentioned in Section 5.3.1, Salehi-Abari and White (2009b, 2010) proposed the con-resistant trust model, called AER. Using DART, they ran the simulations with the same settings as explained above with the difference that the TAA used AER. Figure 9 shows the trust value variation of the TAA over the 400 interactions. Interestingly, regardless of the value of θ for SCA(θ), the con-man was recognized by the trust model and achieved a low value of trust. It is worth noting that the con-man still has a chance to be forgiven but with a very large number of cooperations and a change in its pattern of interaction.

6.1.3. Naive Agent. As mentioned in Section 5.3.5, Salehi-Abari and White (2009a) introduced the concept of a naive agent. They analyzed the impact of this agent class on agent societies including malicious agents, naive agents, and TAAs (see Salehi-Abari and White (2009a) for the details of specification and implementation of these four agents in DART). We here present the experiment and results regarding the impact of the proportion of naive agents on the utility of other agents (see the other experiments and results on Naive

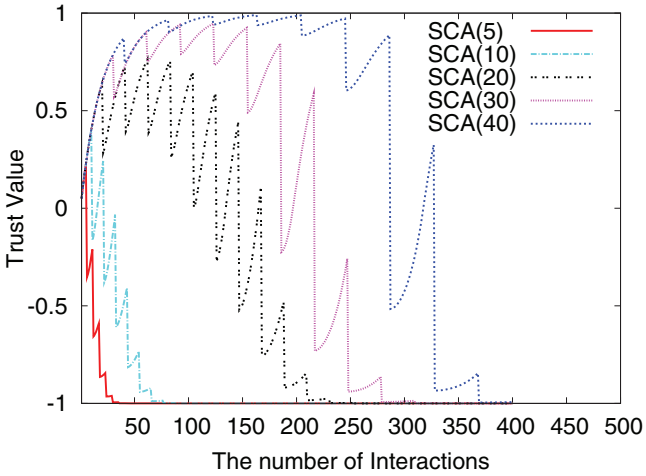


FIGURE 9. The reaction of AER to the con-man attack.

TABLE 2. Population Distributions of Experiment 3.

	Population				
Agent Type	Pop1	Pop2	Pop3	Pop4	Pop5
Malicious	34%	34%	34%	34%	34%
Naive	0%	11%	22%	33%	44%
Trust-Aware	66%	55%	44%	33%	22%

agents in Salehi-Abari and White (2009a)). We have run five simulations of 200 agents with different proportions of Naive and TAAs while maintaining malicious agents unchanged as shown in Table 2.

Figure 10 presents $\overline{U}$ of each agent type at time step 400 for each of the runs. By increasing the proportion of Naive agents, $\overline{U}_{Malicious}$ increases considerably although the proportion of malicious agents is unchanged. $\overline{U}_{TA}$ in all runs stays at 3 indicating that the proportion of Naive agents does not influence $\overline{U}_{TA}$. $\overline{U}_{Naive}$ increases slightly because malicious agents have more choices to connect to Naive agents and to satisfy their ST threshold. For Pop5, the $\overline{U}_{Malicious}$ exceeds that of TA agents. In such societies, where malicious agents are unbounded in terms of their ability to exploit other agents, there is no incentive to be a TAA since malicious agents have better utility. This is the outcome of having a high proportion of Naive agents in the artificial society.

6.2. Properties of DART

Generally, there is no guarantee that a trust model implemented within DART is exploitation-resistant to one attack or not. As mentioned in Section 1, DART provides the facility for researchers to specify heterogeneous agents employing various trust models and is flexible enough to accommodate a variety of adversarial behaviors (exploitation). Specifically, DART is a tool and framework for researchers, which helps them to define

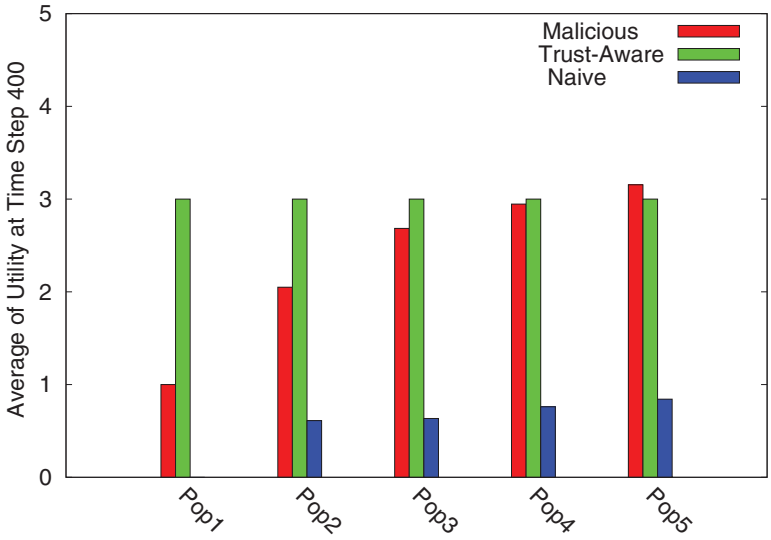


FIGURE 10. $\overline{U}$ for five runs.

their own interested strategic exploitations and agent types easily and test their focused trust models on the specified agent society. This is completely the responsibility of the researchers to define their exploitation and check whether their examined trust models are exploitation resistant to that specific model or not. However, the research outlined in the previous section has demonstrated the utility of the framework.

The proposed testbed provides the desirable features and properties for an effective trust and reputation environment which are presented by Fullam et al. (2005):

- **Modularity:** The proposed testbed provides a wide range of capabilities through adjustable environment and agent parameters. Parameterization allows the researcher flexibility while conducting a wide variety of experimental scenarios. For example, when the researchers design an experiment in which all the agents should be able to interact with each other, the complete graph is generated for the initial graph. In terms of interaction types, the researchers are able to design their agents in such a way that they interact in all or some of the possible types of interaction. For example, experimental agents in Salehi-Abari and White (2009b) employed only direct interaction whereas the heterogeneous agent society in Salehi-Abari and White (2009a) had two types of direct interaction and witness interaction.
- **Multipurpose Design:** Similar to the ART testbed (Fullam et al. 2005), our testbed can be used in different modes such as experiments and competitions. In competition mode, the average utility metric presented in Section 3.9 can be used to rank the agent types.
- **Accessibility:** Various types of trust and reputation can be tested in DART. The proposed environment model is completely independent of an agent's architecture and model. This model provides the opportunity to define different agents types as explained in Section 3.7. For example, Salehi-Abari and White (2009b) utilized the DART framework to specify three types of TAAs varying in trust models. Each TAA possesses either of Regret (Sabater and Sierra 2001), FIRE (Huynh et al. 2004) or Yu and Singh's (2000) trust model. Similarly, two classes of TAAs have been specified in Salehi-Abari and

White (2009c): Trust-Aware (TA_w) and Trust-Aware$^+$ (TA_w^+) where TA_w uses a unidimensional trust model as opposed to TA_w^+ which uses a multidimensional trust model. Not only can the agent society in DART be heterogeneous in terms of used perception model (trust and reputation models) but also the heterogeneous agents in terms of behaviors are able to be defined in DART. For example, naive, malicious, TAAs (Salehi-Abari and White 2009a) are specified in DART.

- **Objective Metrics:** The metrics in DART, as explained in Section 3.9, capture single-agent (microscopic) and system-wide (macroscopic) perspectives. The researcher can select appropriate metrics based on their research objectives and analyze those metrics over the course of simulation. For example, the structure of agent society and the average of dropped connections were two metrics chosen in Salehi-Abari and White (2009c) whereas the structure of agent society and the average of utility describe was selected for experiments in Salehi-Abari and White (2009a).
- **Problem Focus:** By employing game-theoretical concepts and notions (recall Section 3.2), our testbed is not restricted to domains such as e-commerce. It is an abstract model and structured in a way that related trust problems are addressed while ignoring out-of-scope research areas such as identity management (authentication), belief revision, and domain knowledge.

Regarding scalability, as DART defines agent neighborhoods, indicating the nature of the social network, the complexity of simulation scales as the size of the neighborhood.

The environment model of DART does not suffer any of the shortcomings of the IPD (mentioned in Section 5.1) in spite of maintaining the simplicity of game-theoretic models. First, agents can evaluate different aspects of opponents' behavior and consequently multidimensional trust is encouraged. Second, agents can separate untrustworthy agents because they do not have to interact with all other agents and only have to interact with their neighbors. Third, it is equipped with system-level (macroscopic) metrics.

While existing testbeds such as the IPD and ART (recall Section 5.1) have focused on either or both of direct interactions and witness interactions, agents can have four types of interactions in the proposed model: Direct Interaction, Witness Interaction, Reporting Interaction, and Introduction Interaction. Reporting interactions are the localized decentralized reporting mechanism which let an agent inform its neighbors regarding the result of its current interactions. The introduction interaction, which can be request-driven or asynchronous, provides an incentive for agents to be trustworthy to extend their trustworthy neighborhood.

7. CONCLUSION AND FUTURE WORK

Artificial societies and open distributed environments, especially e-commerce need trust and reputation models. While reviewing important existing trust and reputation models from the literature, embracing centralized and decentralized models, we have noted a tendency to focus on exploitation of trust and reputation models. These vulnerabilities reinforce the need for new evaluation criteria for trust and reputation models, which we have called exploitation resistance, that reflects the ability of a trust model to be unaffected by agents who try to manipulate the trust model. To analyze and understand how the trust models are exploitation-resistant, we see the need for an appropriate framework (testbed). This observation has motivated this paper to propose a DART model which consists of environment and agent models as explained in Sections 3 and 4, respectively.

The proposed environment model is compatible with the characteristics of open distributed systems. Agents can have different types of interactions and consequently access to different sources of information for assessment of other agents. The proposed environment model provides the facility to define agents with various behaviors and is flexible enough to accommodate a variety of adversarial behaviors. The proposed environment model does not suffer from the shortcomings of existing testbeds in spite of maintaining the simplicity of game-theoretic models.

Besides direct, witness interaction, and introduction interactions, agents in our environment model can have a type of interaction called the reporting interaction as explained in Section 3.1. This interaction type and its associated trust dimension facilitates the decentralized reporting mechanism in distributed environments. We proposed the agent model of DART in Section 4 which incorporates multiple sources of information to assess the trustworthiness of agents.

Future work is planned in the following two areas: (1) the extension of environment and agent models of DART by introducing new types of interactions and information sources, and (2) further exploration of reporting interactions and introduction interaction.

We plan to extend the DART agent and environment models by using other sources of information, such as sociological information. In this sense, the agents possess different roles in society, and will be judged based on their roles and relationships with others. Note that this extension should still be consistent with open distributed characteristics as explained in Section 1 which makes it a significant practical and research challenge. Indubitably, designing a decentralized mechanism for providing sociological information is the main challenge. Furthermore, in a practical system trust modification would include a consideration of the value of a transaction.

In the design of the environment and agent models of DART proposed in this paper, we ignore the existence of noise in the perception of interactions. For example, in our model, when an agent perceives a direct interaction of another agent (target agent) as a defection, the target agent is penalized by the receiver agent regardless of the fact that whether this defection was the result of noise. We are interested in considering the effect of noise in our model in future research.

REFERENCES

ABDUL-RAHMAN, A., and S. HAILES. 2000. Supporting trust in virtual communities. *In* HICSS '00: Proceedings of the 33rd Hawaii International Conference on System Sciences, Vol. 6. IEEE Computer Society: Washington, DC, p. 6007.

AKERLOF, G. A. 1970. The market for lemons: Quality uncertainty and the market mechanism. The Quarterly Journal of Economics, **84**(3):488–500.

ARTZ, D., and Y. GIL. 2007. A survey of trust in computer science and the semantic web. Web Semant., **5**(2):58–71.

AXELROD, R. 1984. The Evolution of Cooperation. Basic Books: New York.

BERNERS-LEE, T., J. HENDLER, and O. LASSILA. 2001. The semantic web. Scientific American, **284**(5):29–37.

BRAINOV, S., and T. SANDHOLM. 1999. Contracting with uncertain level of trust. *In* EC '99: Proceedings of the 1st ACM conference on Electronic commerce. ACM: New York, pp. 15–21.

CARBO, J., J. MOLINA, and J. DAVILA. 2002. Comparing predictions of sporas vs. a fuzzy reputation agent system. *In* Proceedings of the International Conference on Fuzzy Sets and Fuzzy Systems, Interlaken, Switzerland, pp. 147–153.

CASTELFRANCHI, C., R. CONTE, and M. PAOLUCCI. 1998. Normative reputation and the costs of compliance. Journal of Artificial Societies and Social Simulation, **1**(3). Available at http://jasss.soc.surrey.ac.uk/1/3/3.html. Accessed July 28, 2012.

CASTELFRANCHI, C., R. FALCONE, and G. PEZZULO. 2003. Trust in information sources as a source for trust: A fuzzy approach. *In* AAMAS '03: Proceedings of the second international joint conference on Autonomous agents and multiagent systems. ACM: New York, pp. 89–96.

CHAVEZ, A., and P. MAES. 1996. Kasbah: An agent marketplace for buying and selling goods. *In* First International Conference on the Practical Application of Intelligent Agents and Multi-Agent Technology (PAAM'96), Practical Application Company, London, UK, pp. 75–90.

DASGUPTA, P. 2000. Trust as a commodity. *In* Trust: Making and Breaking Cooperative Relations. Department of Sociology, University of Oxford: Oxford, UK, pp. 49–72.

DELLAROCAS, C. 2000. Mechanisms for coping with unfair ratings and discriminatory behavior in online reputation reporting systems. *In* ICIS, Brisbane, Australia, pp. 520–525.

FALCONE, R., and C. CASTELFRANCHI. 2001. Social trust: A cognitive approach. *In* Trust and Deception in Virtual Societies. Kluwer Academic Publishers: Berlin Heidelberg, pp. 55–90.

FELDMAN, M., K. LAI, I. STOICA, and J. CHUANG. 2004. Robust incentive techniques for peer-to-peer networks. *In* EC '04: Proceedings of the 5th ACM Conference on Electronic Commerce. ACM: New York, pp. 102–111.

FULLAM, K. K., T. B. KLOS, G. MULLER, J. SABATER, A. SCHLOSSER, Z. TOPOL, K. S. BARBER, J. S. ROSENSCHEIN, L. VERCOUTER, and M. VOSS. 2005. A specification of the agent reputation and trust (ART) testbed: Experimentation and competition for trust in agent societies. *In* AAMAS '05: Proceedings of the Fourth International Joint Conference on Autonomous Agents and Multiagent Systems. ACM: New York, pp. 512–518.

GAMBETTA, D. 1988. Can we trust trust. *In* Trust: Making and Breaking Cooperative Relations. Basil Blackwell: Oxford, pp. 213–237.

GRANDISON, T., and M. SLOMAN. 2000. A survey of trust in internet applications. IEEE Communications Surveys and Tutorials, **3**(4):2–16.

HOUSER, D., and J. WOODERS. 2006. Reputation in auctions: Theory, and evidence from eBay. Journal of Economics & Management Strategy, **15**(2):353–369.

HUYNH, T. D., N. R. JENNINGS, and N. SHADBOLT. 2004. Developing an integrated trust and reputation model for open multi-agent systems. *In* 7th International Workshop on Trust in Agent Societies, New York, pp. 65–74.

HUYNH, T. D., N. R. JENNINGS, and N. R. SHADBOLT. 2006. An integrated trust and reputation model for open multi-agent systems. Autonomous Agents and Multi-Agent Systems, **13**(2):119–154.

JARVENPAA, S. L., N. TRACTINSKY, and M. VITALE. 2000. Consumer trust in an internet store. Information Technology and Management, **1**(1–2):45–71.

JENNINGS, N. R. 2001. An agent-based approach for building complex software systems. Communications of the ACM, **44**(4):35–41.

JOSANG, A., and R. ISMAIL. 2002. The beta reputation system. *In* Proceedings of the 15th Bled Electronic Commerce Conference, Bled, Slovenia.

JOSANG, A., R. ISMAIL, and C. BOYD. 2007. A survey of trust and reputation systems for online service provision. Decision Support Systems, **43**(2):618–644.

JURCA, R., and B. FALTINGS. 2002. Towards incentive-compatible reputation management. *In* Proceedings of the AAMAS 2002 Workshop on Deception, Fraud and Trust in Agent Societies. ACM Press: New York, pp. 92–100.

KERR, R., and R. COHEN. 2009a. An experimental testbed for evaluation of trust and reputation systems. *In* Proceedings of the Third IFIP WG 11.11 International Conference on Trust Management (IFIPTM'09), Purdue University, West Lafayette, IN, pp. 252–266.

KERR, R., and R. COHEN. 2009b. Smart cheaters do prosper: Defeating trust and reputation systems. *In* AAMAS '09: Proceedings of the Eighth International Joint Conference on Autonomous Agents and Multiagent Systems, ACM, Budapest, Hungary, pp. 993–1000.

KREPS, D. M., and R. WILSON. 1982. Reputation and imperfect information. Journal of Economic Theory, **27**(2):253–279.

KYBURG, Jr., H. E. 1987. Bayesian and non-bayesian evidential updating. Artificial Intelligence, **31**(3):271–293.

MARIMON, R., J. P. NICOLINI, and P. TELES. 2000. Competition and reputation. *In* Proceedings of the World Conference Econometric Society, Seattle, WA.

MARSH, S. 1994. Formalising Trust as a Computational Concept. Ph.D. thesis, University of Stirling, Stirling, UK.

MCILRAITH, S. A., T. Cao SON, and H. ZENG. 2001. Semantic web services. IEEE Intelligent Systems, **16**:46–53.

MUI, L., M. MOHTASHEMI, and A. HALBERSTADT. 2002a. A computational model of trust and reputation for e-businesses. *In* Proceedings of the 35th Annual Hawaii International Conference on System Sciences (HICSS'02), Vol. 7. IEEE Computer Society: Washington, DC, p. 188.

MUI, L., M. MOHTASHEMI, and A. HALBERSTADT. 2002b. Notions of reputation in multi-agents systems: A review. *In* AAMAS '02: Proceedings of the First International Joint Conference on Autonomous Agents and Multiagent Systems. ACM: New York, pp. 280–287.

NOWAK, M. A., and K. SIGMUND. 1998. Evolution of indirect reciprocity by image scoring. Nature, **393**:573–577.

OLMEDILLA, D., O. F. RANA, B. MATTHEWS, and W. NEJDL. 2005. Security and trust issues in semantic grids. *In* Semantic Grid: The Convergence of Technologies, Volume 05271 of *Dagstuhl Seminar Proceedings*, Internationales Begegnungs- und Forschungszentrum (IBFI), Schloss Dagstuhl, Germany. 2005/2005_dagstuhl_trust_grid.pdf.

ORAM, A. editor. 2001. Peer-to-Peer: Harnessing the Power of Disruptive Technologies. O'Reilly & Associates, Inc.: Sebastopol, CA.

ORMROD, J. E. 2003. Human Learning (4th ed.). Prentice Hall: Upper Saddle River, NJ.

POUNDSTONE, W. 1992. Prisoner's Dilemma: John Von Neumann, Game Theory and the Puzzle of the Bomb. Doubleday: New York.

RAMCHURN, S. D., D. HUYNH, and N. R. JENNINGS. 2004. Trust in multi-agent systems. Knowledge Engineering Review, **19**(1):1–25.

RESNICK, P., and R. ZECKHAUSER. 2002. Trust among strangers in internet transactions: Empirical analysis of eBay's reputation system. The Economics of the Internet and E-Commerce, **11**:127–157.

RUBIERA, J. I. C, J. M. MOLINA, and J. D. MURO. 2003. Trust management through fuzzy reputation. International Journal of Cooperative Information Systems, **12**(1):135–155.

SABATER, J., and C. SIERRA. 2001. Regret: A reputation model for gregarious societies. *In* Fourth Workshop on Deception Fraud and Trust in Agent Societies, Montreal, Canada, pp. 61–70.

SABATER, J., and C. SIERRA. 2005. Review on computational trust and reputation models. Artificial Intelligence Review, **24**(1):33–60.

SALEHI-ABARI, A., and T. WHITE. 2009a. Detecting and dealing with naive agents in trust-aware societies. *In* Trust '09: Proceedings of the 12th International Workshop on Trust in Agent Societies, Budapest, Hungary, pp. 117–128.

SALEHI-ABARI, A., and T. WHITE. 2009b. Towards con-resistant trust models for distributed agent systems. *In* IJCAI '09: Proceedings of the Twenty-First International Joint Conference on Artificial Intelligence, Pasadena, CA, pp. 272–277.

SALEHI-ABARI, A., and T. WHITE. 2009c. Witness-based collusion and trust-aware societies. *In* SPOSN-09: Proceedings of the Workshop on Security and Privacy in Online Social Networking Vancouver, Canada, pp. 1008–1014.

SALEHI-ABARI, A., and T. WHITE. 2010. Trust models and con-man agents: From mathematical to empirical analysis. *In* AAAI '10: Proceedings of the Twenty-Fourth Conference on Artificial Intelligence, Atlanta, GA, pp. 842–847.

SCHILLO, M., P. FUNK, and M. ROVATSOS. 2000. Using trust for detecting deceitful agents in artificial societies. Applied Artificial Intelligence, Special Issue on Trust, Deception and Fraud in Agent Societies, **14**(8): 825–848.

SCHMECK, H., T. UNGERER, and L. C. WOLF, editors. 2002. Trends in Network and Pervasive Computing - ARCS 2002, *In* Proceedings of International Conference on Architecture of Computing Systems, Karlsruhe,

Germany, April 8–12, 2002, Volume 2299 of Lecture Notes in Computer Science. Springer: Heidelberg Berlin.

SEN, S., and N. SAJJA. 2002. Robustness of reputation-based trust: Boolean case. *In* AAMAS 02: Proceedings of the First International Joint Conference on Autonomous Agents and Multiagent systems, Bologna, Italy, pp. 288–293.

TRAN, T., and R. COHEN. 2004. Improving user satisfaction in agent-based electronic marketplaces by reputation modelling and adjustable product quality. *In* AAMAS '04: Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems, IEEE Computer Society, Washington, DC, pp. 828–835.

WEEKS, S. 2001. Understanding trust management systems. *In* SP '01: Proceedings of the 2001 IEEE Symposium on Security and Privacy, IEEE Computer Society, Washington, DC, p. 94.

WHITBY, A., A. JOSANG, and J. INDULSKA. 2004. Filtering out unfair ratings in bayesian reputation systems. *In* Proceedings of 7th International Workshop on Trust in Agent Societies, New York.

YU, B., and M. P. SINGH. 2000. A social mechanism of reputation management in electronic communities. *In* CIA '00: Proceedings of the 4th International Workshop on Cooperative Information Agents IV, The Future of Information Agents in Cyberspace, London, UK, pp. 154–165.

YU, B., and M. P. SINGH. 2002. An evidential model of distributed reputation management. *In* AAMAS '02: Proceedings of the First International Joint Conference on Autonomous Agents and Multiagent Systems. ACM: New York, pp. 294–301.

YU, B., and M. P. SINGH. 2003. Detecting deception in reputation management. *In* AAMAS '03: Proceedings of the Second International Joint Conference on Autonomous Agents and Multiagent Systems. ACM: New York, pp. 73–80.

YU, B., M. P. SINGH, and K. SYCARA. 2004. Developing trust in large-scale peer-to-peer systems. *In* 2004 IEEE First Symposium on Multi-Agent Security and Survivability, Drexel University, Philadelphia, PA, pp. 1–10.

Zacharia , Giorgos. September 1999. Collaborative reputation mechanisms for online communities. *In* Masters thesis, Massachusetts Institute of Technology Cambridge, MA.

ZACHARIA, G., and P. MAES. 2000. Trust management through reputation mechanisms. *In* Applied Artificial Intelligence, **14**:881–907.

ZACHARIA, G., A. MOUKAS, and P. MAES. 1999. Collaborative reputation mechanisms in electronic marketplaces. *In* HICSS '99: Proceedings of the Thirty-Second Annual Hawaii International Conference on System Sciences, Vol. 8. IEEE Computer Society: Washington, DC, p. 8026.