TRUST IN AGENT SOCIETIES
(with Agent Reputation and Trust (ART) Testbed Special Session)

Organized by:

Rino Falcone - ISTC-CNR - Italy,
Suzanne Barber - The University of Texas - USA
Jordi Sabater-Mir - IIIA-CSIC - Spain
Munindar Singh - North Carolina State University - USA

DESCRIPTION OF THE WORKSHOP

The aim of the workshop is to bring together researchers who can contribute to a better understanding of trust and reputation in agent societies. Most agent models assume secure and reliable communication to exist between agents. However, this ideal situation is seldom met in reality. In fact, many techniques (e.g. contracts, signatures, long-term personnel relationships, reputation) have been evolved over time to detect and prevent deception and fraud in human communication, exchanges and relations, and hence to assure trust between agents. Artificial societies will need analogous techniques.

Trust is more than secure communication, e.g., via public key cryptography techniques. For example, the reliability of information about the status of your trade partner has little to do with secure communication. With the growing impact of electronic societies, trust and privacy become more and more important. Trust is important in applications such as human-computer interaction to model the relationship between users and their personal assistants. Different kinds of trust are needed: trust in the environment and in the infrastructure (the socio-technical system) including trust in your personal agent and in other mediating agents; trust in the potential partners;

trust in the warrantors and authorities (if any). Another growing trend is the use of reputation mechanisms, and in particular the interesting link between trust and reputation. Many computational and theoretical models and approaches to reputation have been developed in the last years.

Trust appears to be foundational for the notion of "agency" and for its defining relation of acting "on behalf of". It is also critical for modeling and supporting groups and teams, organizations, co-ordination, negotiation, with the related trade-off between individual utility and collective interest; or in modeling distributed knowledge and its circulation. In several cases the electronic medium seems to weaken the usual bonds in social control: and the habit or disposition to cheat grow stronger. In experiments of cooperation supported by computers it has been found that people are more leaning to defeat than in face-to-face interaction, and a preliminary direct acquaintance reduces this effect. So, computer technology can even break trust relationships already held in human organizations and relations, and favor additional problems of deception and trust.

We encourage an interdisciplinary focus of the workshop - although focused on virtual environments and artificial agents - as well as presentations of a wide range of models of deception, fraud, reputation and trust building.

Suggested topics include, but are not restricted to, the following:

* Models of trust and of its functions;
* Models of deception and fraud; approaches for detection and prevention;
* Models and mechanisms of reputation;
* Role of control and guaranties mechanisms;
* Models and mechanisms for privacy and access control;
* Theoretical aspects, e.g., autonomy, delegation, ownership;
* Integration of conventional and agent-based mechanisms;
* Policies, interoperability, protocols, ontologies, and standards;
* Scalability and distribution across multiple domains or within the global domain;
* Test-beds and frameworks for computational trust and reputation models;
* Legal aspects;
* Application studies (e.g., e-commerce, e-health, e-government) of the above.

PROGRAM COMMITTEE

Suzanne Barber - Computer Science, The University of Texas - USA

Javier Carbó - Universidad Carlos III de Madrid - Spain

Cristiano Castelfranchi - Cognitive Science, ISTC National Research Council - Italy

Rosaria Conte - Cognitive Science, ISTC National Research Council - Italy

Robert Demolombe - Computer Science, CERT/ONERA - France

Rino Falcone - Cognitive Science, ISTC National Research Council - Italy

Catholijn Jonker - Computer Science, Vrije Universiteit Amsterdam - The Netherlands

Stephane Lo Presti - Royal Holloway, University of London

Brendan Neville - Computer Science, Imperial College London - UK

Andrea Omicini - Computer Science, Università di Bologna, Italy

Jeremy Pitt - Computer Science, Imperial College London - UK

Sarvapali D. Ramchurn - Computer Science, University of Southampton, UK

Jordi Sabater-Mir - Computer Science, IIIA-CSIC - Spain

Giovanni Sartor - Law and sociology, University of Bologna - Italy

Sandip Sen - Computer Science, University of Tulsa - USA

Munindar Singh - Computer Science- North Carolina State University - USA

Chris Snijders - Sociology, Utrecht University - The Netherlands

Luke Teacy - Electronics and Computer Science - University of Southampton, UK

Yao-Hua Tan -Economics and Business Administration, Free University Amsterdam, The Netherlands

TABLE OF CONTENTS

pp.	Title / authors
1	**Trust Attributes, Methods, and Uses** Gabriel Becerra and Jason Heard and Rob Kremer and Jörg Denzinger
7	**Extending the ART Testbed to Deal with Heterogeneous Agent Reputation Models** Anarosa A. F. Brandão, Laurent Vercouter, Sara Casare and Jaime Sichman
14	**Reputation-based Security for Agent Communities in a Rapidly Evolving Environment** Suratna Budalakoti, Daiqian Zhan, K. Suzanne Barber
22	**On the convenience of using similarità between tasks in a trust and reputation model for contracting web services** Alberto Caballero, Juan A Botìa, Antonio F. Gòmez-Skarmeta
31	**Cognitive Model of Trust as Relational Capital** Cristiano Castelfranchi, Rino Falcone and Francesca Marzo
39	**Applying Trust to a Secure Message Passing Domain** David De Angelis and K. Suzanne Barber
44	**Beyond Trust. Fuzzy Contextual Corrective Filters for Reliability Assessment in MAS** Esteve Del Acebo, Nicolás Hormazábal and Peplluís de la Rosa
49	**Expectations Enhanced Trust Model** Adel ElMessiry
54	**Learning TaskSpecific Trust Decisions** Ikpeme Erete, Sandip Sen
62	**Dynamic Trust Formation in Multi-Agent Systems** Dr Robert Ghanea-Hercock
68	**Trust dynamics, motivational attitudes and epistemic actions** Mario Gómez, Clara Benac Earle and Javier Carbó
80	**Guaranteed Security in Trust and Reputation Systems** Reid Kerr and Robin Cohen
90	**How to talk about reputation using a common ontology: from definition to implementation** Isaac Pinyol, Jordi Sabater-Mir and Guifré Cuní
102	**The ART of IAM: The Winning Strategy for the 2006 Competition** W. T. Luke Teacy, Trung Dong Huynh, Rajdeep K. Dash, Nicholas R. Jennings, Jigar Patel and Michael Luck

112 **Secure Migration of Mobile Agents based on Distributed Trust**
Martijn Warnier, Michel Oey, Reinier Timmer and Frances Brazier

117 **Towards More Effective E-Marketplaces: A Novel Incentive Mechanism**
Jie Zhang, Robin Cohen

Trust Attributes, Methods, and Uses

Gabriel Becerra and Jason Heard and Rob Kremer and Jörg Denzinger
Department of Computer Science
University of Calgary
Calgary, AB, Canada
{ayala, heard, kremer, denzinge}@cpsc.ucalgary.ca

ABSTRACT
In the literature, several different definitions of trust and models have been presented to aid researchers as they apply the concepts of trust within their systems. The multiplicity of trust definitions creates a high level of ambiguity. This paper takes a step away from models of trust and attempts to define and categorize several attributes, methods of discovery, and methods of evaluation that can be used to describe and discuss trust and trust models. This research leads to a vocabulary that enables researchers to communicate and compare investigations on trust more easily. To show how the terminology can be useful for describing models and systems that use trust, five case studies are presented. Each case study describes research using the new terminology presented in this paper.

Categories and Subject Descriptors
I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence—*Multiagent systems*

General Terms
theory, design, standardization

Keywords
Trust, Terms, Survey, Multi Agent Systems

1. INTRODUCTION
A multi-agent system (MAS) is a collection of multiple, possibly independent entities (called agents) working in a common environment. One goal of MAS studies is to achieve synergy, or to achieve more through cooperation or coordination between agents than would otherwise be possible [3]. Open systems are one type of MAS in which agents from untrusted sources may operate [11]. Within an open system, there must exist either some form of control over the behaviour of individual agents or some method for agents to evaluate which other agents can be trusted. The issue of trust itself is not new, however, and has been discussed extensively by many researchers in psychology, management, communications, sociology, economics, and political science well before trust became useful to the study of agents [17].

Up to the year 2001 there were at least 65 cited articles and monographs defining the concept of trust. Out of those 65 articles 23 pertain to the field of psychology; 23 were within the disciplines of management and communications; and 19 are from areas such as sociology, economics, and political science [17]. The large number of definitions of trust contributes to the ambiguity that surrounds this concept.

Mayer et al. set the groundwork for the next generation of trust research and its goal: define trust models that accommodate a specific definition of trust [16]. As a response to this call to create models of trust, researchers begun constructing models based on the properties they considered important or relevant to their context, be it sociology or economics. While this work succeeded in creating several models helpful to several different specific situations, it failed to generate models that were neutral.[1]

This paper will attempt to distill the many definitions and models of trust into several key portions similar to the way that Jennings et al. aided the discussion of agents in general by outlining a set of possible descriptors for agents, including reactive, social, and rational [13]. Once completed, this paper will then use these new terms to describe several systems and models of trust.

1.1 Agents and Multi-Agent Systems
As described by Denzinger and Hamdan an agent is a 3-tuple *(Situation, Action, Data)* and a function $Situation \times Data \rightarrow Action$ [4]. $Situation$ is the set of all situations that the agent can find itself, $Action$ is the set of all actions that the agent can perform, and $Data$ is the set of all possible configurations of the agent's internal data. In other words an agent is a function that takes the situation and the agent's internal state and outputs some action. To further refine the type of agent being discussed, agents may be described as having one or more of the following properties: social, fair, semi-autonomous, proactive, reactive, knowledge-based, and rational [13]. A multi agent system (MAS) is simply a set of agents (Ag) and an environment that they share (Env).

2. THE TERMS OF TRUST
If Alice trusts Bob to perform some task, she must have determined that his attributes meet some set of requirements. This paper will

[1]A neutral model in this instance is a model that is as useful to a sociologist or a computer agent researcher as it would be to an economist. In other words, the scope of the model would not be constrained to any particular area of expertise.

divide the trust terminology into three categories. The first is the set of *attributes* of Bob that she may consider. The second is the set of *methods of discovery* that Alice can use to determine those attributes. The third is the set of *methods of evaluation* Alice will use to evaluate the attributes as she has determined them.

This section will outline each of the terms that are used repeatedly in the area of trust, and will list a selection of authors that have used these terms (or synonymous terms to date).

2.1 Attributes

Four main attributes must be considered to evaluate whether Bob can be trusted to perform the required task. The first is his *integrity*, the second is his *motivation*, the third is his *predictability* and the fourth is his *competence*. Integrity defines the level of trustworthiness of Bob, while motivation defines how motivated he is to perform the given task. Predictability describes how regular Bob's actions are. Finally, competence describes Bob's ability to perform the required task.

2.1.1 Integrity

The attribute of integrity describes how ethical Bob is in general. This can describe how ethical, honest or moral an agent is. This can also be called Bob's willingness to act (and behave) properly. Instances of this attribute are found in the following papers:

Hartman considers this property as a *type* of trust whereas in this work we have classified it as an attribute which Alice seeks to determine in order to measure the trustworthiness of Bob [9].

Mayer et al. consider that the relationship between integrity and trust involves Alice's perception that Bob adheres to a set of principles that Alice finds acceptable [16].

McKnight and Chervany consider that when Alice securely believes that Bob makes good faith agreements with other agents, tells the truth, and fulfills his promises, then Bob is a an ethical agent that truly cares about Alice's interests [17]. This property is attached to a belief condition that leads toward the proper foundation of intention.

2.1.2 Motivation

An area that is less well studied includes the motivation of Bob to complete the task. This may be an area of future study, because as Bob's motivation increases, Alice relies less on Bob's integrity. For instance, if Alice knows about Bob's goals and she thinks that the task at hand is thought of by Bob as important for his goals, then Alice will perceive Bob as a highly motivated agent.

2.1.3 Predictability

Both prediction and trust are means of uncertainty reduction. To be meaningful, trust must go beyond predictability. In other words, Bob's predictability is insufficient to make Alice take a risk and put herself in a vulnerable situation. It is, however, helpful for Alice to determine the predictability of Bob. Instances of this property are found in the following papers:

Mayer et al. consider predictability as a factor that influences the cooperation between two agents [16]. If Alice expects that Bob will predictably behave positively, Alice will have the disposition to cooperate with Bob.

McKnight and Chervany argue that Alice is willing to depend on, or intends to depend on, Bob in a given task or situation with a "feeling of relative security", even though negative consequences are possible [17]. This property is composed of five elements: (1) Alice may have to deal with negative consequences or risk in unfamiliar or uncertain situations, (2) Alice's readiness to depend or rely on Bob is central to trusting intentions, (3) Alice feels safe, assured, and comfortable about the prospect of depending on Bob, (4) trusting intention is situation and person specific, and (5) trusting intention involves *willingness*, which is not the same as having control over Bob.

Van Witteloostuijn explores the conditions under which trustworthy behaviour can be expected [22]. His study is based on game theory and aims to the comprehension of how trust and cooperation are closely related. Moreover, van Witteloostuijn considers that cooperation is a consequence of the unobservable cognition of trust.

2.1.4 Competence

Competence is an attribute that describes how skilled Bob is for the task that Alice wishes him to perform. It is possible that Bob only has a *specific competence*, which indicates that Bob can only perform the selected task in a certain set of situations ($Sit' \subset Sit$). Instances of this attribute are found in the following papers:

Falcone and Castelfranchi argue that this attribute is present when Alice believes that there exists an agent, say Bob, that has the power to achieve the task [5]. The authors differentiate between two types of *delegation*: weak and strong. The first type refers to cases where there is no explicit delegation of the task from Alice to Bob. The specific task is accomplished by the exploitation of an autonomous action of Bob. The second type of delegation exists when Alice explicitly delegates the task to Bob. In this case, there is a willingness to depend on Bob's competence in such area.

Hartman identifies competence as another type of trust [9]. As with integrity, we classify competence as an attribute and not as a type of trust. Hartman considers that this property allows Alice to assess the extent to which its interests will be cared for and be protected. Moreover, this question will also help in evaluating the consistency of Bob's behaviour while performing the task.

McKnight and Chervany attach competence to a belief condition that serves as the foundation of the *intention* property [17]. For instance, Alice believes that Bob is skilled enough to perform the task she needs to be done or executed.

Mayer et al. developed the competence property under the term "ability" [16]. Ability is that group of skills, competencies, and characteristics that enable Bob to have influence within some specific domain. The domain of the ability is specific because Bob may be highly competent in some technical area. However, such competence may be limited by the context or the specificity of the task.

2.2 Methods of Discovery

There are four main methods that can be used to discover Bob's attributes. They are *intuition*, *experience*, *hearsay*, and *records*. Intuition includes all methods that Alice can use to determine Bob's attributes without considering Bob specifically. Generally this focuses on how agents in general would react to the current situation. Experience focuses on personal experience with Bob. Hearsay

is the opinions of other agents who are not necessarily trusted. Records are any institutionally supported data about Bob.

2.2.1 Intuition

The concept of intuition is hard to adapt to multi agent systems. It is hard to think of the "gut" feelings that Alice may have towards Bob. However, to clearly define the meaning of this method, we will define an intuition as any computation that Alice can perform before her first conversations with Bob. This can include game theoretic approaches to the situation not specific to the agent Bob. In general, this method can help Alice determine Bob's motivation to complete the task. Instances of this property are found in the following papers:

Romahn and Hartman consider this as their third type of trust in their model [20]. As with their previous classifications, we have to disagree with such classification. However, we agree on the description of the attribute and its sub-division (rational and irrational intuition). The description of this question can be summarized as an aid for Alice when measuring how *right* a situation, deal, or relationship with Bob feels. This question may lead to complications while implementing an agent-based system if we consider that *intuition* depends too much on the so-called gut-feelings. The latter is what the authors considered *irrational intuition* or the chemistry between two parties. On the other hand, the *rational* part of this property can be considered in an agent-based system by applying some heuristics to Bob's competence level.

2.2.2 Experience

Alice, as a cooperative agent, may have had previous experiences with Bob. In addition, Alice may have observed Bob as he worked with another agent. These interactions may have been all related to the actual task or another set of tasks that Bob may also be capable of doing. This method can help Alice assess Bob's integrity and competence. Instances of this property are found in the following papers:

Sierra et al. make use of this method in order to asses trustworthiness under the concept of *confidence*, which they describe as a measure of certainty, based on evidence from past direct interactions with Bob [19].

Denzinger and Hamdan investigate tentative stereotype models based on agents' situation-action pairs [4]. The results of this investigation allows agents to predict an agent's behaviour even after few observations. Their models can be reevaluated in order to switch from the original impression to other type of stereotype.

Bower et al. suggest and analyze uncertainty and learning about the populations of agents [1]. The analysis shows how the degrees of trust by Alice, and cooperation by Bob, can depend on the past behaviour of Bob. That is, such analysis depends on the experience Alice has had with Bob.

2.2.3 Hearsay

As Alice communicates with more agents, she may be able to obtain data on Bob's integrity and competence through third-party sources, without direct interaction [16]. Instances of this property are found in the following papers:

Falcone and Castelfranchi deal with this matter by applying transitivity concepts [5]. Let $\prec$ mean trust and $George \in Ag$. If $Alice \prec Bob$, and $Bob \prec George$, then $Alice \prec George$.

Sierra et al. make use of this method in order to asses the trustworthiness under the concept of *reputation*, which is described as Alice's measure of certainty (leading to trust), based on the aggregation of confidence measures provided to it by other agents that have previously interacted with Bob [19].

McKnight and Chervany treat this property under the concept of *influence* by stating that when Alice allows Bob to influence her, Alice is depending on that opinion to be correct given the fact that bad consequences may follow if it is misdirected on another party's opinion [17].

2.2.4 Records

Another method of obtaining information about Bob is through records from an institution or authority. In this case, Alice still obtains information about Bob, from an agent that represents an institution or an authority. And, usually such an institution or authority has methods to deal with their own agents reporting something that is false.

While Carter and Ghorbani consider the case where Alice should be aware of the trust level she has towards an *institution*, and in this way, records could be considered a special case of hearsay, we beleive that this method deserves it's own separate method [2].

2.3 Methods of Evaluation

Once the attributes of Bob have been discovered as much as is possible, they must then be evaluated to determine whether Bob can be trusted. To do this, Alice must consider two factors. How much does Alice *risk* to trust Bob, and how *willing* is Alice to take that risk.

2.3.1 Level of Risk

Risk is considered to be an essential component of trust. It is important for researchers to understand that risk is inherent in the behavioural manifestation of the willingness to be *vulnerable*. Trust will lead to risk taking in a relationship, and the form of the risk taking depends on the situation.

We emphasize the fact that vulnerability is a consequence of taking the risk of trusting other agents. Instances of this property are found in the following papers:

Gambetta considers trust as a particular level of the subjective probability with which Alice assesses that Bob will perform an action [8]. For trust to be present there must be the possibility for disappointment or betrayal.

Mayer et al. provide one of the most used definitions of trust based vulnerability, which is defined as "the mutual confidence that no party to an exchange will exploit another's vulnerabilities" [16].

Ford while investigating the concept of *knowledge sharing* took into consideration this particular question in the sense that trusting other parties with valuable information involves a high-level risk [6].

Bower et al. develop the concept of risk within the area of game theory [1]. They consider that "trusting is risky for Alice because there are two types of agents and Alice is uncertain about the type it is confronting. When a rational (opportunistic) agent is allowed discretion, Bob's immediate payoff is higher if it chooses to betray

the trust of Alice, but Alice's immediate payoff is higher if Bob chooses to cooperate".

2.3.2 Willingness to Trust

According to Mayer et al., the disposition to trust is an internal factor that will affect the likelihood of Alice to trust other agents [16]. Such a disposition to trust might be thought of as the general *willingness to trust* others.

Given a situation and the internal data of Alice, she will have a disposition or willingness to either depend on Bob or perform certain action towards a specified task herself [16]. Based on this duality, we identify the following subdivision: the willingness to depend on another agent, and the willingness to act or achieve a specified task. Instances of this property are found in the following papers:

McKnight and Chervany argue This question is linked to a belief condition that leads toward the proper foundation of intention [17].

Ford by using the term *behavioural trust* is able to identify that the central question is whether Alice is willing to depend on Bob or not [6]. The willingness of Alice to be vulnerable to Bob's actions, are based on the expectation that Bob will perform a particular action important to Alice, irrespective to the ability to monitor or control Bob. The same definition is used by Carter and Ghorbani [2] while elaborating a new trust model based on reputation.

Hartman relies on the existence of legal systems in the absence of integrity [9]. Hartman considers that legal systems (institutions or authorities) decrease Alice's risk in any relationship.

3. CASE STUDIES

This section will show how the terms defined above can be helpful in describing and comparing several trust systems.

3.1 The ART Testbed

The Agent Reputation and Trust (ART) Testbed is a multi agent game designed to be a testbed for trust issues [7]. The testbed simulates an open multi agent system where art appraisal agents communicate with one another to increase the accuracy of their appraisals. Each competitor in the system controls one art appraisal agent. Each agent is assigned capabilities for appraising various categories of art work. The system provides clients who will make appraisal requests of each agent. When an agent Alice doesn't possess the ability to appraise a piece of artwork (or possesses only a limited ability for that type of artwork), she can request an opinion from another appraisal agent Bob. Bob has the option of being dishonest in both the expression of its ability to appraise the artwork and in the actual opinion sent back to Alice. To aid Alice in determining which agents are trustworthy, she can ask agents other than Bob for their evaluation of Bob's abilities.

A successful agent in the ART testbed must take both *integrity* and *competence* into account. Integrity comes from the knowledge of how well other agents self-report their capabilities when they have been queried for an opinion. Competence is important since each agent has its own level of knowledge for various types of artwork. *Predictability* is also a factor, as it relates to competence. An agent in the ART framework is capable of using *intuition*, *experience* and *hearsay*. Intuition could be used to consider how agents' behaviours might change over the course of the game. Experience and hearsay are the main mechanisms supported and encouraged by the testbed's design. The exclusion of *reports* from an authority is probably necessary to maintain the focus on reputations and opinions.

3.2 CASA (Cooperative Agent System Architecture) and Social Commitments

This section will introduce and evaluate the CASA system as a basis of trust and agents [15]. CASA is a communication-based multi agent system written in Java. CASA has several unique features that could be useful in the exploration of social commitments and trust.

A *cooperation domain* (CD) is an agent within CASA designed to aid the communication of many agents in large groups. The CD allows agents to communicate with one another without knowing about every other agent. Agents register with the CD, and as a result they receive all non-private communications that are sent to the CD. In addition, they can receive a list of all agents that have joined the CD and choose to communicate directly with those agents. CASA agents are free to communicate with each other directly (they can bypass the CD), but they will potentially lose the use of services offered through the CD and, relating to trust, the service of identifying or sanctioning anti-social agents.

CASA provides a useful tool for researching social commitments and trust: the CASA system implements several conversation *policies*. These policies are used by CASA agents to define the communication protocols used. Conversation policies map messages with certain performatives to the creation, cancellation, and fulfillment of social commitments. Kremer and Flores describe how the CASA system defines social commitments and bases the conversation policies on the performatives in CASA from FIPA [14].

Because the communication system in CASA is based upon social commitments, it is easy for an agent to monitor communications of itself (or in some special cases, all agents in a CD) and determine which agents are breaking social commitments. Agents who monitor their own communication can determine other agents *integrity* and *predictability* based on personal *experience*. In addition, if an agent communicates within a cooperation domain, it is possible to include a system sanctioning agent which removes agents that pass some level of anti-social behaviour—this may increase an agent's *willingness to trust* agents within the system [10]. Another option is a BBB (better business bureau) agent which will, upon request, provide *records* on the integrity of agents within that cooperation domain. These services are provided by trusted agents which are privy to all communications through the cooperation domain.

At this point, the areas that CASA does not provide assistance with are competence, motivation, intuition and hearsay. These areas could be implemented in individual agents, but have not been at this point.

3.3 Adaptive Trust and Cooperation: An Agent-Based Simulation Approach

This project focuses on applying agent-based technologies both to simulate inter-firm relations and to determine how cooperation, trust and loyalty emerge from agent-based computational economics model instead of following a transaction cost economics model [18]. Briefly, this market-driven project aimed at identifying how trust and loyalty affect each other under several circumstances.

In this model, the attributes described above (e.g. integrity and competence) are determined by applying three methods of discovery: intuition (rational), personal experience, and hearsay. Rational intuition is used by applying an economic selection on a set of possible partners. The agent asks itself: how profitable will this partnership be? In addition to this internal computation, the agent also gathers information from its previous interactions with other agents. Externally, the agent will gather information from transaction partners. This because it is cheaper to gather information from other sources than is to be extremely cautious in events where cooperation from both sides will increase profits. Consequently, if profits are increased, then trust and loyalties toward each other will also increase. The latter brings up the concept of predictability since agents in this simulation wonder about what the behaviour of their partner would be in cases where the partnership has not been as effective as desired. Finally, after applying the methods of discovery and determining the proper values for the attributes, the agent then decides on its willingness to cooperate. This is reflected by its risk factor, which affects its behaviour given certain circumstances. In other words, the agent computes values that serve as thresholds that indicate the best moment to defect and terminate its commitments.

The model used by Nooteboom et al does not present any reference to methods of discovery such as records, motivation, and did not specify how hearsay exactly affect the trust level of one agent toward another agent. It is important to mention that the authors never make use of integrity, but it is easy to determine that an agent's integrity in this evaluation is not very important. Because agents, using this model, defect as soon as the profits or other offers are close to their thresholds (either lower or upper bounds).

3.4 Certified Reputation

A typical problem with using *hearsay* in multi agent systems is that of finding witnesses which can and are willing to report on agents' that are of interest. To help alleviate this problem, Huynh et al. introduce what they have termed certified reputation [12]. These certified reputations are held by the agent that they describe. Security is assumed to be accomplished by some public-key cryptographic system, and is not the focus of the paper. Since each agent holds its own certified reputations, an agent Alice who wishes to evaluate an agent Bob can simply ask him for his set of certified ratings. Since Bob holds the ratings, and chooses which to forward to Alice, there is a tendency for these ratings to be skewed toward positive. To help deal with this situation, Huynh et al. have incorporated *experience* into the model to deal with the fact that certain referees may not be as trustworthy as other referees. Agents keep track of the perceived precision of referees and use this to aid in their evaluations of the certified ratings. This was found to be an efficient method for evaluating the *competence* and *integrity* of agents within the system.

3.5 Trust and Honor

Sierra and Debenham [21] introduce the concepts of honor and reliability to argumentative dialogs (negotiations) that lead to the creation of social commitments. While the authors treat trust as a measurement of the expected deviations from a commitment, honor is treated as a measurement of the expected integrity of the arguments used in the negotiation. An integral part of this research are the predicates used throughout the article. For instance, Build(Alice, Bob, r) means "Alice considers Bob to be a potential trading partner for deals in a relationship r".

The attributes that Alice is trying to evaluate are: capable (competence), honor (integrity), and predictability. The predicate Capable(Alice, Bob, d) is used to mean "Bob's ability to do what it says it can do (for Alice) in deals of type d". It is important to point out that the predicates Capable(Alice, Bob, d) and Reliable(Alice, Bob, c) (where c is a context) determine Bob's competence in a specific area. For instance, Bob's results may vary based on task he is requested to perform (e.g. provide information about the weather, plan and schedule the allocation of resources).

Alice uses reputation (hearsay) as a method of discovery while measuring trust. Sierra and Debenham acknowledge the importance of Bob's reputation (which is provided to Alice by other agents). Alice uses *willingness to trust* and *level of risk* as her methods of evaluation. For example, Alice determines whether or not she wants to trust Bob over another competitor (Charles). The latter is determined by comparing the values provided by her probability function P (e.g. P(Build(Alice, Bob, r)) $\leq$ P(Build(Alice, Bob, r))).

The model used by Sierra and Debenham does not present any reference to methods of discovery such as records, motivation, and did not specify how hearsay exactly affect the trust level of Alice toward Bob.

4. CONCLUSION

While much research has been conducted into trust, including research outside the areas of multi agent systems and computer science, discussing trust is still difficult. As research into trust began, new definitions were created for each instance that was researched. Then, various fields of research popularized models of trust. Because of the proliferation of trust research in many separate fields of study, it is difficult to discuss or compare models of trust because a uniform vocabulary has not been introduced.

This paper hopes to aid in the discussion of trust with the set of descriptive terms that it provides to describe the attributes and methods used in active trust systems. The terms outlined in this paper will allow researchers to easily identify and discuss the attributes and methods used in various models and projects.

5. ACKNOWLEDGMENTS

We are thankful for support from the National Science and Engineering Research Council (NSERC) of Canada. We would like to acknowledge the entire *Knowledge Science Group* (directed by Dr. Rob Kremer) for their ideas and points of view during a couple of brainstorming sessions. In addition, we have to mention Dr. Francis Hartman for *trusting* us with his findings on the concept of trust, specially with his latest model.

6. REFERENCES

[1] A. Bower, S. Garber, and J. Watson. Learning about a population of agents and the evolution of trust and cooperation. *International Journal of Industrial Organization*, 15(2):165–190, April 1997.

[2] J. Carter and A. A. Ghorbani. Towards a formalization of value-centric trust in agent societies. In *Web Intelligence and Agent Systems*, volume 1, pages 167–184. IOS Press, 2004.

[3] J. Denzinger. Knowledge-based distributed search using teamwork. In *Proceedings of the First International Conference on Multi-Agent Systems*, pages 81–88, San Francisco, CA, USA, 1995.

[4] J. Denzinger and J. Hamdan. Improving modeling of other agents using tentative stereotypes and compactification of observations. In *IAT 2004*, pages 106–112. Intelligent Agent Technology, September 2004.

[5] R. Falcone and C. Castelfranchi. The socio-cognitive dynamics of trust: Does trust create trust? In R. Falcone, M. P. Singh, and Y.-H. Tan, editors, *Trust in Cyber-Societies: Integrating the Human and Artificial Perspectives*, number 2246 in Lecture Notes in Artificial Intelligence, pages 54–72. Springer, 2001.

[6] D. Ford. *Handbook of Knowledge Management*, volume 1 of *International Handbooks on Information Systems*, chapter 29 - Trust and Knowledge Management: The Seeds of Success. Springer-Verlag, 2003.

[7] K. Fullam, T. Klos, G. Mueller, J. Sabater, A. Schlosser, Z. Topol, K. S. Barber, J. Rosenschein, L. Vercouter, and M. Voss. A specification of the agent reputation and trust (ART) testbed: Experimentation and competition for trust in agent societies. In *The Fourth International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS-2005)*, pages 512–518, 2005.

[8] D. Gambetta. *Trust: Making and Breaking Cooperative Rea*, chapter Can we Trust Trust?, pages 213–237. University of Oxford, 2000.

[9] F. Hartman. The role of trust in project management. In Kahkonen and Koskinen, editors, *Managing Business by Projects*, volume 1. Project Management Association Finland and Nordnet, 1999.

[10] J. Heard. Detecting broken social commitments: An implementation and investigation. Master's thesis, University of Calgary, Calgary, AB, Canada, August 2005. http://www.cpsc.ucalgary.ca/ heard/msc-thesis.pdf.

[11] C. E. Hewitt. The challenge of open systems. *Byte*, 10(4):223–242, April 1985.

[12] T. D. Huynh, N. R. Jennings, and N. R. Shadbolt. Certified reputation: How an agent can trust a stranger. In *The Fifth International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS-2006)*, pages 1217–1224, May 2006.

[13] N. R. Jennings, K. Sycara, and M. Wooldridge. A roadmap of agent research and development. *Journal of Autonomous Agents and Multi-Agent Systems*, 1(1):7–38, 1998.

[14] R. Kremer and R. Flores. Using a performative subsumption lattice to support commitment-based conversations. In *Proceedings of the* 4th *International Conference on Autonomous Agents and Multi Agent Systems (AAMAS 2005)*, pages 114–121, July 2005.

[15] R. Kremer, R. Flores, and C. L. Fournie. A performative type hierarchy and other interesting considerations in the design of the CASA agent architecture. In *Advances in Agent Communication*, volume 2922 of *Lecture Notes in Artificial Intelligence*, pages 59–74. Springer-Verlag, Heidelberg, Germany, 2004.

[16] R. C. Mayer, J. H. Davis, and D. F. Schoorman. An integration model of organizational trust. *Academy of Management. The Academy of Management Review*, 20(3):709–734, July 1995.

[17] H. McKnight and N. Chervany. Trust and distrust definitions: One bite at a time. In R. Falcone, M. P. Singh, and Y.-H. Tan, editors, *Trust in Cyber-Societies: Integrating the Human and Artificial Perspectives*, number 2246 in Lecture Notes in Artificial Intelligence, pages 27–54. Springer, 2001.

[18] B. Nooteboom, T. B. Klos, and R. J. Jorna. Adaptive trust and co-operation: An agent-based simulation approach. In R. Falcone, M. P. Singh, and Y.-H. Tan, editors, *Trust in Cyber-Societies: Integrating the Human and Artificial Perspectives*, volume 2246 of *Lecture Notes in Artificial Intelligence*, pages 83–109. Springer, 2001.

[19] S. Ramchurn, C. Sierra, L. Godo, and N. R. Jennings. Devising a trust model for multi-agent interactions using confidence and reputation. *International Journal of Applied Artificial Intelligence*, 18:833–852, 2004.

[20] E. Romahn and F. Hartman. Trust: A new tool for project managers. *Project Management Institute 1999 Annual Symposium and Seminar*, pages 85:1–85:6, October 1999.

[21] C. Sierra and J. Debenham. Trust and honour in information-based agency. In *AAMAS '06: Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems*, pages 1225–1232, New York, NY, USA, 2006. ACM Press.

[22] A. van Witteloostuijn. A game-theoretic framework of trust. *International Studies of Management & Organization*, 33(3):53–71, 2003.

Extending the ART Testbed to Deal with Heterogeneous Agent Reputation Models

Anarosa A. F. Brandão[1], Laurent Vercouter[2], Sara Casare[1] and Jaime Sichman[1]
[1]Laboratório de Técnicas Inteligentes – EP/USP
Av. Prof. Luciano Gualberto, 158, trav. 3, Cidade Universitária
05508-970 – São Paulo – Brazil
{anarosa.brandao,sara.casare,jaime.sichman}@poli.usp.br
[2]Ecole Nationale Supérieure des Mines de Saint-Etienne
158, cours Fauriel, 42023 Saint-Etienne Cedex 2, France
Laurent.Vercouter@emse.fr

ABSTRACT
In open MAS it is often a problem to achieve agents' interoperability. The heterogeneity of its components turns the establishment of interaction or cooperation among them into a non trivial task, since agents may use different internal models and the decision about trust other agents is a crucial condition to the formation of agents' cooperation. In this paper we propose the use of an ontology to deal with this issue. We experiment this idea by enhancing the ART reputation model with semantic data obtained from this ontology. This semantic data is used during interaction among heterogeneous agents when exchanging reputation values and these data may be used as input for reputation reasoning modules of agents that use different reputation models.

Categories and Subject Descriptors
I.2.11 [**Distributed Artificial Intelligence**]: Multiagent systems

General Terms
Design, Experimentation, Standardization.

Keywords
Reputation, trust, ontology, ART testbed.

1. INTRODUCTION
Multiagent systems (MAS) are gaining wide acceptance in both industry and academia as a powerful paradigm for developing software systems. A great number of them have been constructed in the last 10 years, with different aims and architectures. In particular, regarding the homogeneity of the agents, in some systems all the agents have the same architecture (homogeneous) and in others agents have different internal models (heterogeneous). However, both of them present agents sharing a common communication infrastructure (middleware), like KQML [10][14]. The MAS composed of heterogeneous agents mostly present the characteristic of openness. Open MAS are composed of autonomous distributed agents that may enter and leave the agent society at their will because open systems have no centralized control over the development of its parts [2]. Multiagent auction systems and virtual enterprises are examples of such open and distributed applications [26].

Whenever we consider the domain of open MAS, this implies that some agents, potentially implemented by different developers, can enter or leave dynamically the system, and that these agents may be involved in collective tasks and will cooperate with other agents that rely on them. Furthermore, since agents are considered as autonomous entities, we cannot assume that there is a way to control their internal behavior. These features are interesting to obtain flexible and adaptive systems but they also create new risks about the reliability and the robustness of the system. The risk that an agent does not behave well (intentionally or not) is increased. This bad behavior can cause the failure of some tasks performed by the agent but also of collective tasks in which it is involved.

The existence of autonomous agents in an open system implies that there is a risk that agents do not behave as expected and may harm other agents or the overall system. Solutions to this problem have been proposed by the use of trust models where agents are endowed with a model of other agents that allows them to decide if they can or cannot trust another agent. Such trust decision is very important because it is an essential condition for cooperation among truly autonomous agents. The trust decision process has been the subject of several propositions from different researchers. Most of them use the concept of reputation as the basis of a decision. Reputation is a subject that has been studied in several works [1][8][9][16][20] with different approaches but also with different semantics attached to the reputation concept. Nevertheless, the heterogeneity of such reputation models may prevent the trust decision process' success due to the aforementioned differences. Casare and Sichman [6] [7] proposed a Functional Ontology of Reputation to serve as a common vocabulary among the different agent reputation models and some directions about how it could be done.

In this paper we propose the use of Casare and Sichman approach from [7] combined with the general agent architecture proposed on [23] to extend the ART testbed [11] to deal with heterogeneous agent reputation models. ART (Agent Reputation and Trust) is a testbed for experimentation and comparison of trust models. To do so, we briefly describes the functional ontology of reputation and three agent reputation models in order to show how the ontology can be applied to allow interoperability among agents that have different reputation models. Moreover, an outline of this approach is sketched in the context of ART testbed through the description of some scenarios.

The paper is organized as follows: Section 2 presents an outline of the functional ontology of reputation; Section 3 briefly describes three different reputation models; Section 4 presents direct mappings between the agent reputation models and the functional ontology of reputation. In Section 5 we describe a scenario for using the mappings in the context of the ART testbed and in Section 6 we present some related work. Finally, Section 7 presents our conclusions about what was made so far, as well as some ideas for future work.

2. THE FUNCTIONAL ONTOLOGY OF REPUTATION (FORe)

The Functional Ontology of Reputation (FORe) aims at providing a functional perspective on representing and analyzing reputation as a social control mechanism for societies of agents. It includes, as its kernel, the following concepts: reputation nature, roles involved in reputation formation and propagation, information sources for reputation, evaluation of reputation, and reputation maintenance.

The nature of the reputation distinguishes a reputation according to the kind of entity it is associated to. Therefore, the ontology concept ReputationNature is composed of concepts such as IndividualReputation, GroupReputation and ProductReputation.

Reputation formation and propagation involves several roles, played by the entities or agents that participate in those processes. The ontology defines the concepts ReputationProcess and ReputationRole which are composed of concepts such as ReputationEvaluationProcess and TargetRole, respectively.

Moreover, reputation can be classified according to the origin of beliefs and opinions that can derive from several sources. The ontology defines the concept ReputationType which can be PrimaryReputation or SecondaryReputation. Moreover, the concept PrimaryReputation is composed of concepts such as ObservedReputation and DirectReputation and the concept SecondaryReputation is composed of concepts such as PropagatedReputation and CollectiveReputation. In general, a reputation can be evaluated through several types of measurements and different factors [5]. To deal with this, the ontology concept ReputationEvaluationComponent is composed of concepts such as ReputationEvaluationValue and ReputationFinalValue. The reputation maintenance process deals with the modifications that occur in the content and structure of a reputation over time. Examples of concepts related to this process are ReputationMaintenanceProcess and MaintainerRole. More details about the FORe can be found on [6] and [7].

3. AGENT REPUTATION MODELS

According the Oxford Dictionary, reputation is the opinion that people in general have about what somebody or something is. For example, a school may have an excellent reputation and a person could have a reputation of being lazy.

Reputation is an indispensable condition for the social conviviality in human societies. The emergence of Internet based virtual societies has caused the migration of reputation related concepts from the world of human interactions to the world of virtual interactions. Reputation and trust systems are key factors for successful electronic commerce environment. They are used as mechanisms in order to search trustful partners as well as an incentive to avoid cheaters and frauds [22].

In the last years several computational models of reputation have been proposed [15][22][25][26]. As an example of research produced in the MAS field we describe three of these models in this section: a cognitive reputation model, a typology of reputation and the reputation model used in the ReGret system. These models were chosen because their specifications are in different levels of abstraction. The cognitive reputation model consists of a high level conceptual model, described in natural language. The typology of reputation is a taxonomy regarding reputation sources of information, while the ReGret system is a computational mechanism that involves a reputation model in its kernel.

3.1 Cognitive Reputation Model

The Cognitive Reputation Model proposed by Conte and Paolucci [9] treats the several aspects associated with the reputation transmission in the social sphere. Image and reputation are two central concepts in this model. Image is an evaluative belief, as “good” or “bad”, formed using information acquired by agent experience and reflects the target agent behavior. Reputation is a belief about others’ minds, a meta-belief that results from beliefs transmission. It represents the process as well as the effect of transmission of a target image.

Besides, the model describes the concepts related to the set of agents involved in image and reputation formation and transmission. Target agents are individuals, groups, or even artifacts that play the role of the evaluation object. Evaluator agents are those entities able to develop an evaluative belief about others as an effect of their social interactions and perceptions. Propagators, or third parties, are those entities able to transmit reputation information about a target to another entity beneficiary agent. Beneficiaries are individuals, groups, or organizations for which the evaluation of the target brings some benefit. As the authors emphasize, there is a non-empty intersection between these four sets of agents. Therefore, an agent can be at the same time a member of the evaluators set as well as a member of the propagator and the beneficiaries’ sets.

3.2 Typology of Reputation

The Typology of Reputation proposed by Mui et al. [15] is based on the reputation research done by several areas of interest, such as Economy, Computer Science and Biology. This typology distinguishes reputation notions in three classification levels. The first level distinguishes reputation according to the target entity nature, while the second and third levels distinguish individual reputation according to the information source.

At the first level there are two concepts: *individual reputation*, which is the reputation that can be used to describe an individual; and *group reputation*, which is reputation that can be used to describe a group of individuals.

The second and third level contains the following concepts: *direct reputation*, which is the reputation derived from direct encounter or observation; and *indirect reputation* is the reputation based on second-hand evidence. The direct reputation concept has two sub concepts: *interaction derived reputation*, the one based on actual encounters between a target agent and its evaluating agent; and

observed reputation, that one resulting from observations made about another agent's encounters. Also, the indirect reputation concept has three sub concepts: *group derived reputation* that is estimates for agents in social groups; *prior derived reputation*, based on prior beliefs about strangers; *propagated reputation*, based on information gathered from others.

3.3 Regret System

The ReGret system [20][22] is a trust and reputation system developed for complex e-commerce environments where social relationships play an important role. The idea is to consider the social aspects involved in the reputation and trust notions, taking into account the social structures among the members of a society in the evaluation of these notions. This model deals with four kinds of reputation evaluation: *direct trust*, *witness reputation*, *neighborhood reputation* and *system reputation*.

Direct Trust refers to the evaluation that is built from direct interaction, including both direct experience and direct observation. Witness Reputation is calculated using the information gathered from other agents of the community that had direct experience with the target agent. Neighborhood Reputation is the reputation based on social prejudice. System Reputation is the reputation based on agent membership to a certain group. Additionally, the Regret system proposes the *credibility* notion, a kind of rating associated to the received information about reputation that evaluates the truthfulness of information received from a witness. The trust evaluation is the result of the reputation evaluation.

4. MAPPING THE AGENT REPUTATION MODELS TO THE FORe

Visser et al [24] suggest three different ways to support semantic integration of different sources of information: (i) a centralized approach, where each source of information is related to one common domain ontology; (ii) a decentralized approach, where every source of information is related to its own ontology; and (iii) a hybrid approach, where every source of information has its own ontology and the vocabulary of these ontologies are related to a common ontology. That ontology organizes the common global vocabulary in order to support the source ontologies comparison. Casare and Sichman [7] used the hybrid approach to show that the FORe serves as a common ontology for several reputation models. Therefore, considering the ontologies which describe the agent reputation models we can define a mapping between these ontologies and the FORe whenever the ontologies use a common vocabulary.

Table 1. Mapping the Cognitive Reputation Model to FORe

Cognitive Model of Reputation	FORe concept
Image	PrimaryReputation
Reputation	SecondaryReputation
Target agents	TargetRole
Propagators	TransmitterRole
Beneficiary	RecipientRole

Since the models described on section 3 use different levels of abstraction, we show that FORe can be mapped to any of them. Nevertheless, it seems that FORe provides a flexible way of representing reputation and its associated concepts as a common vocabulary. A mapping between the Cognitive Reputation Model ontology and the FORe is presented on table 1.

Table 2. Mapping the Typology of Reputation to FORe

Typology of Reputation	FORe concept
Individual reputation	IndividualReputation
Group reputation	GroupReputation
Direct reputation	PrimaryReputation
Indirect reputation	SecondaryReputation
Interaction derived reputation	DirectReputation
Observed reputation	ObservedReputation
Prior derived reputation	StereotypedReputation
Group derived reputation	CollectiveReputation
Propagated reputation	PropagatedReputation

Other models, such as the Typology of Reputation and the ReGret System Model are also mapped to the FORe, see tables 2 and 3, respectively.

Table 3. Mapping between ReGret System and the FORe

ReGret System	FORe concept
Direct Trust	PrimaryReputation
Witness Reputation	PropagatedReputation
Neighborhood Reputation	StereotypedReputation
System Reputation	CollectiveReputation
Credibility	EvaluatorFactor

All the information concerning the mappings between the agent reputation models and the FORe can be directly inferred if both use a common vocabulary by simply classifying the resulting ontology from the integration of a given reputation model ontology and the FORe in a tool such as Protégé [17] with an inference engine such as RACER [18].

5. EXPERIMENTAL SCENARIOS USING THE ART TESTBED

In order to exemplify the use of the mappings presented in section 4, we define a scenario where several agents are implemented using different agent reputation models. This scenario includes the agents' interaction during the simulation of the game defined by ART testbed [11], and it describes possible ways for interoperability between different trust models using the FORe. As we mentioned in the introduction of the paper, ART is a testbed for experimentation and comparison of trust models.

The scenario definition begins with the description of the ART testbed and a general agent architecture that deals with interactions involving reputation.

5.1 The ART testbed

During the International Workshop on Trust in Agent Societies at AAMAS'04 it was decided that a working group named ART (Agent Reputation and Trust) would be created to develop a testbed for the experimentation and comparison of trust models. During AAMAS'05 it was presented a demo of the ART testbed and a competition between trust models using ART took place at AAMAS'06.

This first version of ART provides a simulation engine on which several agents, using different trust models, may run. The simulation consists in a game where the agents have to decide to trust or not other agents. This game is an art appraisal game in which agents are required to evaluate the value of paintings. In fact, agents act as appraisers who are hired by clients to deliver appraisals about paintings for a fixed fee [12]. The game can be summarized as follows: (i) the ART simulation server gives to each agent a given number of clients; (ii) each client gives to its agent the description of a painting. The client expects that the agent evaluates the painting and gives back to him an accurate value; (iii) a painting is mainly characterized by an era. According to this era, an agent is more or less accurate in its evaluations; (iv) each agent can ask other agents to help it in its evaluation of a painting if it thinks that these other agents are more accurate than itself for the corresponding era. This exchange of information is called an opinion transaction and to get this evaluation the requester pays a fixed amount of money to the provider; (vi) agents also interact by the way of reputation transactions whenever the information that is bought is about the reputation of another agent (a target) for a given era. This reputation represents the estimated accuracy of the target in its evaluation of paintings of the given era; (vii) finally, each agent calculates a value for the paintings of its clients and receives money for this evaluation. The game is played during several iterations. At the next iteration, the amount of clients associated with a particular agent depends on the accuracy of its evaluation in the previous iteration. More details about the ART testbed can be found in [11].

During most of this game, the incompatibility of trust models is not important because they are used locally by their agents' decision processes. However, the interaction which involves the reputation transaction is a moment where agents have to exchange information from their trust models, meaning that interoperability among trust models is required. In the current version of the ART testbed, interoperability is obtained by asking the developers of each agent to provide a way to map their trust model into a very simple common reputation model. This simple model defines that: (i) an agent associates with each of its acquaintances a single value named ``Reputation weight''; (ii) this reputation weight is a value in the domain [0:1]. Given the simplicity of the common model, the reputation building process needs a considerable number of interactions between agents until reach stable reputation values.

The addition of semantic data to this common model may improve the agent performance during the process of reputation building and allow interoperability between different trust models. The scenario we are defining includes enhancing the common reputation model of ART with the addition of the FORe. In order to do it, we sketch in the following a general agent architecture for reputation interaction considering this new common model.

5.2 An agent architecture for interoperability

Since the common reputation model includes semantic data related to reputation, agents must have an internal ontological representation of its reputation model in order to integrate these data with the FORe data. This is done using the hybrid approach proposed by Visser et al [24] and gives the desired interoperability between different trust/reputation models through the mappings described on section 4. Moreover, agents must be able to reason about the information received from reputation transactions according to their own reputation models to take decisions. Figure 1 shows a view of the agent general architecture for reputation interaction.

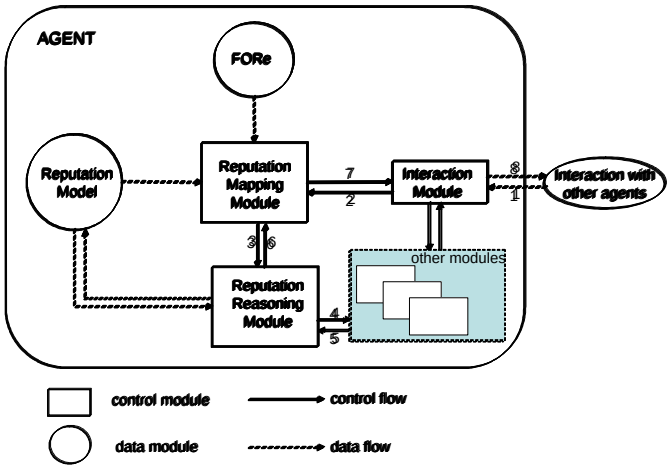

Figure 1. Agent architecture for reputation interaction

This general architecture describes the way interactions must be treated by agents in our scenario: the interaction module receives a message (1). If the message is about reputation, it is transmitted to the reputation mapping module (2), else it goes to other modules. Here we are just considering reputation interactions. The message content is described using the enriched common model and must be translated to the agent internal reputation model. The reputation mapping module analyzes the content of the message and transforms it to comply with the agent internal reputation model. This is made using the mappings described on section 4. The translated message is forwarded to the reputation reasoning module (3). Since the message is according to the agent reputation model, it can be handled by the reasoning module itself or in conjunction with other modules (4 and 5). Whenever the agent needs to interact about reputation with other agents, the message is formulated according to its own reputation model in the reputation reasoning module and is sent to the reputation mapping module (6) to be translated to the enriched common model. Therefore, the message is forwarded to the interaction module (7) to be sent to other agents (8). A simple application of such architecture is found in [23].

5.3 Reputation transaction scenarios

While including the FORe to the ART common reputation model, we have incremented it to allow richer interactions that involve reputation transaction. The enriched common model was called FOReART. Moreover, given the ART application domain (art appraisal) it is also included semantics to data such as skill on specific painting eras and so far. For instance, during interactions about reputation the agents may send a value in the interval [0:1] plus the semantics concerning the way it builds this value. In this section we describe two scenarios concerning reputation transactions. Both are described in the context of ART testbed, but the first is valid for any kind of reputation transaction and the second is specific for the art appraisal domain.

5.3.1 General scenario

Suppose that agents A, B and C are implemented according to the architecture presented on section 5.2, each of them using a different reputation model and the FOReART common reputation model. Agent A uses the Typology of Reputation model, agent B uses the Cognitive Reputation Model and agent C uses the ReGret System model.

Consider the interaction about reputation where agents A and B receive from agent C information about the reputation of agent Y. A big picture of this interaction is showed in Figure 2.

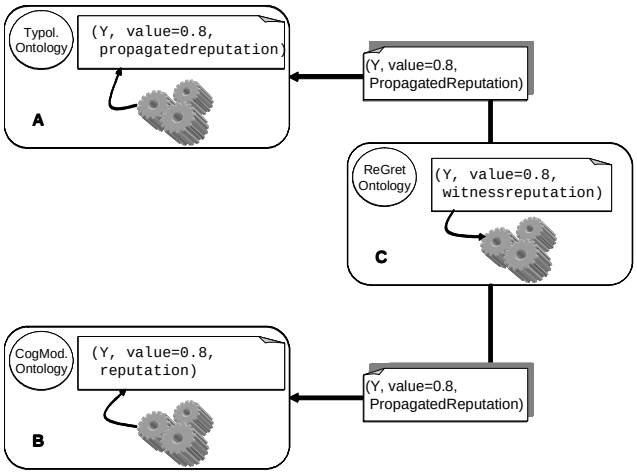

Figure 2. Interaction about reputation

Agent C internal reputation model (ReGret model), through its Reputation Reasoning Module, provides information of type witness reputation, which is associated with the value 0.8. This information is forwarded to the agent C Reputation Mapping Module to be compliant to the FOReART, yielding the information PropagatedReputation (Other models, such as the Typology of Reputation and the ReGret System Model are also mapped to the FORe, see tables 2 and 3, respectively.

Table 3) associated with the value 0.8. Both, agent A and B, receive this information that is treated by their Reputation Mapping Module and is translated to their agent reputation model, respectively. Therefore, the corresponding information in agent A reputation model is *propagated reputation* (Since the models described on section 3 use different levels of abstraction, we show that FORe can be mapped to any of them. Nevertheless, it seems that FORe provides a flexible way of representing reputation and its associated concepts as a common vocabulary. A mapping between the Cognitive Reputation Model ontology and the FORe is presented on table 1.

Table 2) and in agent B reputation model is *reputation*. Although there isn't a direct mapping between PropagatedReputation and *reputation* (Table 1), the agent B reputation model translation is based on the fact that PropagatedReputation is classified as SecondaryReputation (Section **Erro! Fonte de referência não encontrada.**) in the FORe and that SecondaryReputation is mapped to the *reputation* (Table 1) concept.

The way agents A and B make use of the information depends on their internal reputation model and their Reputation Reasoning Module implementation. This means that the same information may lead to different decisions for agent A and B. For instance, agent A reputation model is the Typology of Reputation Model, where the propagated reputation represents the reputation that is derived from third parties information and agent B reputation model is the Cognitive Reputation Model, where reputation represents a belief about others' minds, a meta-belief that results from beliefs transmission. Therefore, the value each agent assigns to the information received depends on the weight their reputation reasoning module gives to "second-hand" information and on their previous knowledge about agent Y reputation. For example, agent A Reputation Reasoning Module may be implemented to incorporate reputation values greater than 0.9 whenever their associated semantics were *indirect reputation* (subsection 3.2). The semantics of indirect reputation in agent A reputation model (Typology of Reputation) allows the inference of *propagated reputation* as *indirect reputation* (subsection 3.2), which leads to agent A decision of not incorporating the information received from agent C in its internal reputation building process concerning agent Y. On the other side, agent B Reputation Reasoning Module may be implemented to incorporate reputation values greater then 0.75 whenever their associated semantics were *reputation*. This leads to agent B decision of incorporating the information received from agent C to its reputation building process concerning agent Y.

5.3.2 Art appraisal scenario

Considering the same agents A and B from the previous scenario and the art appraisal domain, another interesting scenario describes the following situation: agent A asks to agent B information about agents it knows that have expertise on some specific painting era. In this case agent A wants information concerning the *direct reputation* agent B has about agents that have skill on an specific era, such as cubism. Following the same steps of the previous scenario, a message is prepared by agent A in its Reputation Reasoning Module using information from its internal model. Therefore, the message ((`agent` = ?, `value`= ?, `skill` = `cubism`, `reputation` = *directreputation*)) is forwarded to its Reputation Mapping Module to be translated to the FOReART and then is sent to agent B. Agent B receives this message which contains the query ((`agent` = ?, `value` = ?, `skill` = cubism, `reputation` = PrimaryReputation)). The message is

translated to agent B reputation model in its Reputation Mapping Module which results on ((`agent` = ?, `value` = ?, `skill` = cubism, `reputation` = *image*)). The translated message is forwarded to agent B's Reputation Reasoning Module. In the Reputation Reasoning Module the message is analyzed and the response is prepared to be sent to agent A. A big picture of this interaction is showed in Figure 3.

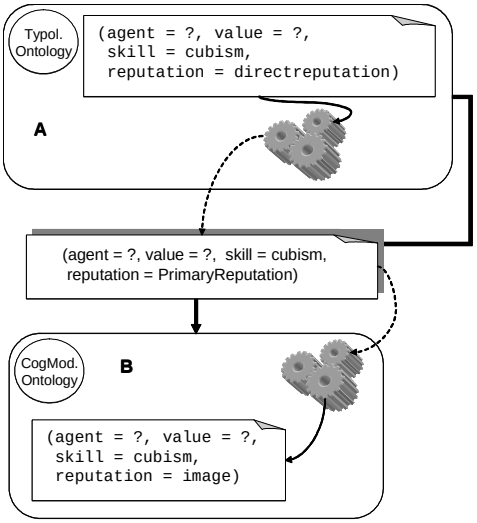

Figure 3. Interaction about specific types of reputation values

Agent B response to agent A is processed in its Reputation Reasoning Module and it is composed of tuples (`agent`, `value`, `cubism`, `image`) , where the pair (`agent`, `value`) is composed of all agents and associated reputation values whose agent B knows their expertise about cubism by its own opinion. At this point all the information is based on Agent B reputation model, but it is ready to be translated to the common model. An example of such response can be seen in Figure 4. This response is forwarded to the Reputation Mapping Module and is translated to the FOReART (Figure 5) to be sent to agent A.

```
( (agent = X, value = 0.3, era = cubism,
   reputation = image);
  (agent = Y, value = 0.7, era = cubism,
   reputation = image);
  (agent = Z, value = 0.9, era = cubism,
   reputation = image) )
```

Figure 4. Agent B response before the mapping

```
( (agent = X, value = 0.3, era = cubism,
   reputation = PrimaryReputation);
  (agent = Y, value = 0.7, era = cubism,
   reputation = PrimaryReputation);
  (agent = Z, value = 0.9, era = cubism,
   reputation = PrimaryReputation) )
```

Figure 5. Agent B response after the mapping

After receiving the information sent by agent B, agent A processed it in its Reputation Mapping Module and translated it to its own reputation model to be analyzed by its Reputation Reasoning Module. Figure 6 shows the translation of the message presented on Figure 5 to agent A reputation model. Again, the way agent A makes use of this information depends on its internal reputation model and the implementation of its Reputation Reasoning Module. For example, agent A Reputation Reasoning Module may be implemented to incorporate reputation values greater than 0.5 whenever their associated semantics were direct reputation for the cubism painting era. In this case, agent A will incorporate the reputation values associated to agents Y and Z.

```
( (agent = X, value = 0.3, era = cubism,
   reputation = directreputation);
  (agent = Y, value = 0.7, era = cubism,
   reputation = directreputation);
  (agent = Z, value = 0.9, era = cubism,
   reputation = directreputation) )
```

Figure 6. Agent B response translated to agent A reputation model

6. CONCLUSIONS AND FUTURE WORK

In this paper we present a proposal for reducing the incompatibility between reputation models by using a general agent architecture for reputation interaction. This architecture relies on a Functional Ontology of Reputation (FORe), used as a globally shared reputation model. A reputation mapping module allows agents to translate information from their internal reputation model into the shared model and vice versa. The ART testbed has been enriched to use the ontology during agent transactions. Some scenarios were described to illustrate our proposal and they seem to be a promising way to improve the process of building reputation just using existing technologies such as FORe, the reasoning engines that are available (Racer and Jena [13]), and the ART testbed.

The next step is the implementation of the scenarios described here to execute on ART testbed. Furthermore, we intend to enhance the first version of ART testbed to allow agents that follow the proposed architecture to execute/interact on it.

7. ACKNOWLEDGMENTS

This work is supported by a USP-COFECUB project 98/04 and by individual grants. Anarosa A. F. Brandão is supported by CNPq/Brazil under grant 310087/2006-6, Jaime Sichman is partially supported by CNPq/Brazil under grants 304605/2004-2, 482019/2004-2 and 506881/2004-0. Laurent Vercouter was partially supported by FAPESP/Brazil under grant 2005/02902-5.

8. REFERENCES

[1] Abdulrahman, A. *A framework for decentralized trust reasoning*, PhD Thesis, University of London, London, UK, 2004.

[2] Agha, G. A. Abstracting Interaction Patterns: A Programming Paradigm for Open Distributed Systems, In (Eds) E. Najm and J.-B. Stefani, *Formal Methods for Open Object-based Distributed Systems* IFIP Transactions, 1997, Chapman Hall.

[3] Boissier, O. and Demazeau, Y. An architecture for social and individual control and its application to computer vision. In:

Y. Demazeau; J.-P. Muller; J. Perram. (Ed.). *Proceedings of the 6th European Workshop on Modelling Autonomous Agents in a Multi-Agent World*, Odense, Denmark: 1994, p. 107—118.

[4] Bordini, R.H. and Hübner, J. F. Jason: *A Java-based agent speak interpreter used with saci for multi-agent distribution over the net*, 2004, Available:http://jason.sourceforge.net/, last visited on 10/03/2006.

[5] Bromley, D.B. *Reputation, Image and Impression Management*. John Wiley & Sons Ltd. England, 1993.

[6] Casare,S. and Sichman, J.S. Towards a Functional Ontology of Reputation, In *Proceedings of the 4th Intl Joint Conference on Autonomous Agents and Multi Agent Systems (AAMAS'05)*, Utrecht, The Netherlands, 2005, v.2, pp. 505-511.

[7] Casare, S. and Sichman, J.S. Using a Functional Ontology of Reputation to Interoperate Different Agent Reputation Models, *Journal of the Brazilian Computer Society*, (2005), 11(2), pp. 79-94.

[8] Castelfranchi, C. and Falcone, R. Principles of trust in MAS: cognitive anatomy, social importance and quantification. In *Proceedings of ICMAS'98*, Paris, 1998, pp. 72-79.

[9] Conte, R. and Paolucci, M. *Reputation in Artificial Societies: Social Beliefs for Social Order*, Kluwer Academic Publ., 2002.

[10] Ferris, G; Blass, F; Douglas, C.; Kolodinsky, R; Treadway, D. Personal Reputation in Organizations. In: Greenberg, Jerald, *Organizational Behavior– The State of the Science.* LEA. New Jersey, 2003, p. 211-246.

[11] Fullam, K.; Klos, T.; Muller, G.; Sabater, J.; Topol, Z.; Barber, S.;Rosenchein, J.; Vercouter, L. and Voss, M. A specification of the agent reputation and trust (art) testbed: experimentation and competition for trust in agent societies. In *Proceedings of the 4th Intl. Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS'05)*, ACM-Press, 2005, pp. 512-158.

[12] Fullam,K. and Barber, S. Learning Trust Strategies in Reputation Exchange Networks, In: *Proceedings of the 5th International Joint Conference on Autonomous Agents and Multiagent Systems, (AAMAS'06)*, Hakodate, Japan, ACM-Press, 2006, pp. 1241-1248.

[13] Jena: A semantic web framework for Java, available at http://jena.sourceforge.net acessed at 04/02/2007

[14] Labrou, Y; Finin, T.; Peng, Y. Agent communication languages: the current landscape. In*: IEEE Intelligent Systems*, 14, (2), (March/April, 1999), p. 45-52.

[15] Mui, L.; Halberstadt, A.; Mohtashemi, M. Notions of Reputation in Multi-Agents Systems: A Review. In: *Proceedings of 1st Intl. Joint Conference on Autonomous Agents and Multi-agent Systems (AAMAS 2002)*, Bologna, Italy, 2002, part 1, pp. 280-287.

[16] Muller, G. and Vercouter, L. Decentralized monitoring of agent communication with a reputation model. In *Trusting Agents for Trusting Electronic Societies*, LNCS (3577), 2005, pp. 144-161.

[17] Protegé, available at http://protege.stanford.edu

[18] RacerPro, available at http://www.racer-systems.com/

[19] Rao, A.S.and Georgeff, M. P. BDI agents: from theory to practice. In: Lesser, V. (Ed.*). Proceedings of the International Conference on MultiAgent Systems (ICMAS'95), 1*, San Francisco, USA, 1995, p. 312–319.

[20] Sabater, J. and Sierra, C. ReGret: Reputation in gregarious societies. In Müller, J. et al (Eds) Proceedings of the 5th International Conference on Autonomous Agents, Montreal, Canada, 2001, ACM Press, pp. 194-195.

[21] Sabater, J. and Sierra, C. Reputation and Social Net-work Analysis in Multi-Agent Systems. In: *Proceedings of the 1st Intl Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS-02)*, Bologna, Italy, 2002, part 1, pp.475-482.

[22] Sabater, J. and Sierra, C. Review on Computational Trust and Reputation Models. In: *Artificial Intelligence Review*, Kluwer Acad. Publ., (2005), v. 24, n. 1, pp. 33 – 60.

[23] Vercouter,L, Casare, S., Sichman, J. and Brandão, A. An experience on reputation models interoperability based on a functional ontology In *Proc. of the 20th IJCAI*, Hyderabad, India, 2007, pp.617-622.

[24] Visser, U.; Stuckenschmidt, H.; Wache, H. and Vogele, T. Enabling technologies for inter-operability. In: In U. Visser and H. Pundt, Eds, *Workshop on the 14th International Symposium of Computer Science for Environmental Protection*, Bonn, Germany, 2000, pp. 35-46.

[25] Yu, B. and Singh, M.P. An Evidential Model of Distributed Reputation Management. In: *Proceedings of the 1st Intl Joint Conference on Autonomous Agents and Multi-agent Systems (AAMAS 2002)*, Bologna, Italy, 2002, part 1, pp. 294 - 301.

[26] Zacharia, G. and Maes, P. Trust Management Through Reputation Mechanisms. In: Applied Artificial Intelligence, 14(9), 2000, pp. 881-907.

[27] Zambonelli, F, Jennings, N; Wooldridge, M. Developing multiagent systems: The gaia methodology, ACM Trans. Softw. Eng. Methodol. 12 (3) (2003), 317-370.

Reputation-based Security for Agent Communities in a Rapidly Evolving Environment

Suratna Budalakoti, Daiqian Zhan, K. Suzanne Barber
The University of Texas at Austin
1 University Station Stop C5000, Austin, TX 78712
+1-512-471-5350
{*suratna, daisyzhan, barber*}*@mail.utexas.edu*

***Abstract*—**

The mobile agent paradigm is a flexible and powerful model that allows an autonomous program, known as an agent, to move independently in an environment. Unfortunately, this flexibility may also make agents vulnerable to various kinds of attacks from malicious entities. Distributed soft approaches to security, where agents cooperate with reputation exchange about other entities in the network, are highly suitable for such scenarios. However if the environment is changing rapidly, reputation and other information may become obsolete very fast. In this research we investigate two online learning algorithms, randomized weighted majority and Winnow, based on the mixture-of-experts and the learning-from-examples frameworks respectively, to see how well they can respond to a constantly evolving environment. We also investigate conditions under which these algorithms are preferable to less sophisticated approaches. Though we simulate these algorithms in a setup consisting of mobile agents moving over a network of hosts, our conclusions may be seen as generally applicable for agents in rapidly evolving conditions.

I. Introduction

An autonomous agent can generally be defined as 'a system situated within and a part of an environment that senses that environment and acts on it, over time, in pursuit of its own agenda and so as to effect what it senses in the future' [1]. While this definition is sufficiently broad to include all types of agents such as biological, robotic or computational, in the context of software, an agent may reasonably be defined as any entity composed of code and data that is not bound that is not bound to the system where it originated [2][3]. A mobile agent moves around the environment (such as a network), performing required tasks, accessing resources, or collecting data as required. These characteristics make it essential that agents be capable of significantly more autonomy in comparison to traditional software.

Some domains where software agents can be extremely useful are e-commerce, distributed information retrieval, and secure brokering applications where two agents may need to travel to a neutral third-party host to complete a transaction. A mobile agent may function alone, but often a number of similar agents co-ordinate with each other agents to achieve a common goal. Such a group may be referred to as a mobile agent community. A common example of a mobile agent community is a community of agents searching over a distributed market over a network, such as the Internet, to purchase various products [4].

For our discussions in this paper, we focus on a similar specific scenario, though our conclusions are applicable to more general scenarios. Suppose we have a network of hosts, and a community of agents that travel from host to host performing some task. Some of the hosts are friendly, while others are hostile. The mobility of these agents introduces serious security risks, as agents need to move from machine to machine while preserving their state information from hosts, some of which, as mentioned, may have hostile intentions. In addition the network situation can be highly volatile, so that hosts that are expected to be friendly might have been compromised.

Network security techniques have traditionally been centralized in approach and server-oriented, which makes it difficult to apply them to agent security problems such as the one described above. This has led to the development of distributed approaches to handle network attacks in mobile agent communities, where a mobile agent may interact with, and ask for help from, other mobile agents in its community. A common security mechanism for mobile agent security is the Buddy Model [5][6], where each agent has two or more buddy agents, who look after each others well-being.

However, the Buddy model and other similar models have a weakness in that they are essentially reactive. They do not try to prevent an agent from visiting a malicious host. Instead the buddy agents only perform the task of rescue when they learn that a buddy is in trouble. However, by that time the agent may be significantly damaged and agent repair can be very expensive. Also the agent might be carrying valuable information that might be irretrievably lost or compromised as a result of an attack.

All these facts make the idea of a proactive approach, in which agents actively co-operate and share information to avoid malicious hosts, thereby minimizing the risk of such attacks, attractive. In this paper, we explore the idea of reputation-based trust as an approach towards providing soft security for a mobile agent community on a network. We focus on situations where the network state is constantly evolving, that is, a friendly host may become hostile or inaccessible with time, or a host that has turned hostile or broken down may

be 'up' again. We compare two algorithms, based on different online learning frameworks [23] that an agent in a community can use to model its situation, to proactively recognize and avoid malicious hosts. We find that these algorithms perform very differently in static and dynamic scenarios.

II. Related Work

Jansen and Karygiannis [7] divide security threats in mobile agent systems into four categories: a) agent to platform/host, b) host to agent, c) agent to agent, and d) other (another host, or outside entity) to host. Out of these categories *a* and *d* are amenable to the application of traditional security techniques. In comparison, categories *b* and *c* present very different challenges, as they are particular to the mobile agent framework. In this paper, we focus on category *b*, that is the threat a hostile host may present to an agent.

The major challenge to security in a mobile agent system is the decentralized nature of the agent model, which makes a centralized approach to security problematic. Wedde and Lischka [8] discuss some of the problems faced with centralized models to agent security. One common problem is that it is often not possible to have all agents monitored from a single location. Even in cases where this may be possible, the location may become overloaded and function as a bottleneck in the system. Centralized systems also tend to enforce homogeneity in terms of the security framework they apply to the system, while agents in a system can be highly varied and have very different requirements. Also, the centralized location provides a single point of attack for malicious entities.

The buddy model of security, first introduced by Page et al [5][6], has been developed to counter some of these problems. The key idea of the buddy model is the decentralization of the security framework. In this model, agents in a community protect their neighbors, which are known as their *buddies*. Every agent in this model has two or more buddy agents. All agents are constructed identically, so that no agent is distinguishable from another. In this way, it is hard for an attacker to find a single target to aim at. Buddies keep track of each other by periodically issuing tokens. A buddy is assumed to be in good health if the token is received within a certain time frame. If the token from an agent is not received in time, its buddies will assume that it is unable to communicate. The other agents will be informed by a global broadcast and the agents will come together to perform a 'rescue' of the compromised agent.

While the buddy model is well-tailored to mobile agent communities, it has a serious weakness, in that it is entirely reactive. It does not attempt to prevent an agent from being compromised, but only attempts to 'rescue' agents that have already been compromised. This can prove to be very expensive, especially if an agent is carrying valuable information, which may be lost/compromised before a 'rescue' has been affected. Trust-based 'soft' security, where agents maintain and share information about the nature of hosts, and exchange this information to avoid visiting a malicious host, can be extremely useful in such situations.

Trust-based approaches can generally be divided into two approaches: a) experience-based trust modeling, and b) reputation-based trust modeling. In experience-based trust modeling [9], agents decide whether or not to trust hosts based on their past experience. In reputation-based modeling [10][11][12], the agent tries to estimate the trust it can place in a particular host based on trust information, or *reputations*, the other agents provide.

We focus on reputation-based models applied to agents over a constantly changing environment.

III. Paper Contributions

Reputation-based models have generally used reinforcement learning approaches such as q-learning [10][14], or online algorithms based on a mixture-of-experts framework, such as the weighted majority algorithm [11][12]. These methods rely on certain assumptions about the behavior of other entities in a system. The two essential assumptions are: hosts that have been trustworthy in the past are more likely to be trustworthy, and b) agents that have provided correct reputation estimates in the past will continue to provide correct estimates. These assumptions hold true in different setups to different degrees.

For example, consider our setup where agents are moving around a network of hosts. In such a situation agents might share information with each other, and also remember their past experience with a host, to make decisions on whether to visit or avoid a host [13]. However, in a highly dynamic network scenario where hosts are constantly being compromised/crashing and then being brought up, assumption a) above may not hold. Similarly, assumption b) may not hold in a large and heterogeneous network where some agents may have excellent information about parts of the network they have visited recently, but outdated information about areas of the network they visited long ago. Thus, even if we assume that all agents are altruistic and honest, an agent may still provide false information, because in a dynamic scenario it may not be aware that its information is outdated. Similarly an agent we have distrusted in the past may suddenly become the best source of correct information if we move to a region of the network it is an expert of. All these factors may create problems with a reputation-based approach.

In this paper, we assume that all agents are altruistic and honest in providing reputations. This often holds true in a mobile agent community. This is because they often are identical to each other, and have the same origin. Instead, we focus on another dimension of reputation estimates: *timeliness* of the information provided. In highly dynamic scenarios, even perfectly altruistic agents may unintentionally provide misleading information as timeliness of information can deteriorate very fast. So reputation models, in order to be successful, need to evolve very quickly in such scenarios.

We investigate various adaptive as well as static reputation-based 'soft' security algorithms in a mobile agent community. These algorithms follow the framework of most reputation-based security approaches. We give the outline below.

1) Given a network of hosts, over which a community of agents is operating, an agent expresses an intent to visit a host.

2) It asks other agents in the network (who we also refer to as *buddies*) about the nature of the host (friendly/hostile).
3) Each of the agents queried sends back its estimate to the querying agent.
4) The agent makes a decision whether or not to visit the host based on the feedback from the agents.

There is often a fifth step in this process where, if an agent decides to visit a host, it re-evaluates the estimate it has of the quality of information each of the other agents/buddies usually provides, based on its experience with the host. These updated estimates will be used the next time to execute step 4. Hence, in our framework, each agent maintains two types of reputation values: a) *agent reputation*, an estimate of the trust it places in the quality of reputation information other agents provide, b) *host reputation*, the degree of trust it has that a host is friendly, based on its past experience with the host (this is the information the agent provides as reputation, when queried by other agents).

The host reputation for an agent is a value between 0 and 1. It is evaluated as the fraction of positive experiences of the agent of the total, with a given host so far. The agent reputations are also restricted between 0 and 1. As we focus on how these values may be learned, their evaluation is more complicated.

In this paper, we focus on two online learning algorithms for the task of learning agent reputations. The two algorithms we focus on are the *weighted majority algorithm* [16][18][23], and *Winnow* [20][22][23]. A version of the weighted majority algorithm has been used by Yu *et al* [11][12] for the task of learning agent reputations in a multi-agent system. To the best of our knowledge, Winnow has not previously been used in the agent framework.

In the following section we discuss the mixture-of-experts and the learning-from-examples frameworks for online learning, and also the weighted majority algorithm and Winnow.

IV. The Predicting-from-Expert-Advice Framework

The traditional model for supervised machine learning [15] has been the offline/batch learning model. The batch learning model consists of two phases: the training and the testing phase. The training period is a 'grace' period, where the algorithm may focus on learning the function mapping data to correct labels without having to bother about its performance. However, there are common real-world situations where an algorithm may not be afforded a grace period, but is required to be competitive from the start. Also, it is often useful to make use of the experience an algorithm gains as it gets exposed to more input.

Such situations have led to an increasing interest in online algorithms in machine learning. In the online learning [23] framework, there is no training phase. The algorithm is presented the data one input at a time, and the algorithm is required to label the input. It is then told the correct label for the data point, and can modify itself based on this feedback.

A common problem analyzed in the online learning framework is the mixture-of-experts or predicting-from-expert-advice problem [17][19][23]. In the predicting-from-experts problem, the algorithm has access to a number of experts, who it can query for their prediction for the label on a particular data point. It can then combine the information provided by the experts, based on how strongly it trusts the judgment of each expert, to provide its prediction of the label. After the algorithm has made its prediction, it is allowed to see the correct label on the data point. It can then update its estimate of the experts based on which of them were correct and which were not.

The prediction-from-experts framework can be applied to the trust problem in our mobile network framework. At each step, a mobile agent that wants to move to a particular host asks other agents (who function as experts for the agent) for feedback on whether the host is friendly or hostile. Initially each agent trusts all other agents equally. The weights for the agents are adjusted as time goes by.

One important feature of the mixture-of-experts framework is that no statistical assumptions are made about the nature of the data, or the quality or independence of the experts. As a result, the performance of an algorithm is often measured not in absolute terms, but in comparison to the performance of the best expert the algorithm has access to.

A. The Randomized Weighted Majority Algorithm

A simple but highly effective algorithm, known as the weighted majority algorithm, was suggested for the mixture-of-experts problem, by Littlestone and Warmuth [18] and was used by Yu *et al* [11][12] for the task of reputation estimation in agent and peer-to-peer networks. In this paper we use another version of the weighted majority algorithm, suggested by Kivinen and Warmuth [16]. Though in our experiments we found no significant difference in the performance of the two algorithms, we use the latter algorithm as the expression for its worst case performace is easier to interpret and draw conclusions from.

In the interest of completeness, we give the algorithm below:

1) Set the initial weights $w_{1,1}, w_{1,2}, \ldots, w_{1,n}$, of all the experts to 1/n, where n is the number of experts. Set c to some positive value.

2) For t = 1. . .k-1, where k is the number of trials

   a) Get a set of predictions $x_{t,1}, x_{t,2}, \ldots, x_{t,n}$ by the experts.
   b) Output the prediction $\hat{y_t} = \Sigma_{i=1}^{n} w_{t,i} \cdot x_{t,i}$.
   c) When the correct answer y_t is received, update the weights for the next iteration as follows:

   $$w_{t+1,i} = w_{t,i} \cdot exp(-L(y_t, x_{t,i})/c)/Z_t$$

   where

   $$Z_t = \Sigma_{i=1}^{n} w_{t,i} \cdot exp(-L(y_t, x_{t,i})/c)$$

Here L is a loss function, and represents by how much the prediction made by an expert was off the correct answer. Some common functions that can be used to measure the loss are the square loss function, the relative entropy loss function [16]. We use the square loss function, defined as $(y - \hat{y})^2$. Hence, the total loss of the weighted majority algorithm is given by $\Sigma_{t=1}^{k}(y_t - \hat{y}_t)^2$ over k trials.

Notice that the prediction $\hat{y}_t$ of the algorithm is not binary but a real value between 0 and 1. We treat the estimate as the estimated probability that the host the agent plans to visit is benign. Hence, the agent visits the host queried for with probability $\hat{y}_t$. For this reason, we refer to this algorithm as the *Randomized Weighted Majority* algorithm in this paper.

There is an elegant proof [16] which shows that, on any sequence of trials, for the square loss function, the total loss $L_{RW}(S)$ of the Randomized Weighted Majority algorithm over a sequence of trials S satisfies:

$$L_{RW}(S) \leq L_{E_i}(S) + 2 \ln n$$

where n is the number of experts, and $L_{E_i}(S)$ is the square loss of any of the n experts. In other words, the loss can be seen as upper bound by the performance of the best expert. Also, it grows only logarithmically as the number of experts grows. This suggests that when assigning experts to a particular agent, assigning a large number of poor experts to an agent will not greatly impact the results.

B. Randomized Weighted Majority: Discussion

As the expression for performance bound indicates, the performance of the randomized weighted majority algorithm is in terms of the quality of the best expert available to it. Essentially the algorithm can be seen as trying to identify the best expert as quickly as possible, based on its past experience. So, the algorithm can be expected to provide better performance when there are a few experts that perform extremely well consistently, than when there are a large number of experts who are right, say half the time.

We now discuss what this means in terms of a dynamic network setup. Consider a network with a hosts, and agents that travel between the hosts, performing some task. Suppose the number of hosts is much larger than the number of agents, and the network situation is evolving constantly. In this case, it is highly unlikely that any single agent will consistently have the correct information about the status of the network. An agent may be expected to know the status only of the hosts it has visited recently. If the order in which it visits the hosts is random, past predictions shall contain little information about the credibility of future reputations provided by an agent. If an agent gave the correct prediction for host *h*, there is little reason to believe that it will be able to predict the status of another host *h'* correctly. This would be especially true if the network is large, or constant changing so that information collected even in the slightly remote past is no longer useful.

However, fortunately, agent access of hosts is seldom random. There is likely to be what we term *locality* or *specialization* of some sort. In the simplest case, an agent that visits a host will usually have to pass through the hosts adjacent to it. However, locality need not only be physical: often it is logical. If certain hosts provide a certain service, agents that require that service will visit those hosts more often than other hosts. Hence, an agent can be highly successful in identifying safe hosts if it can quickly identify agents that have interests and an itinerary similar to its own. However, as the interests of agents change with time, it should also be able to identify when its interests are beginning to diverge with a buddy agent.

However, in a mobile agent community where agents do not have distinct *specializations*, which we may call *domains*, a correct prediction by an agent in one instance says nothing about the validity of its next prediction, and a reputation-based approach is unlikely to work. Over a large or constantly changing network where no such domains exist, it is likely that an adaptive algorithm will perform no better than some simple algorithm that merely averages the results.

V. Learning from Examples

In the online learning-from-examples framework, at each step, the algorithm is provided with an 'example' data point. This example data point is drawn from a sample space *X* with n attributes, usually $\{0,1\}^n$. The algorithm is expected to predict a label for the example. After its prediction, the algorithm is shown the correct label from the example. The assumption is that the label is a function of the attributes of the example. The algorithm *Winnow* assumes that the final label is a disjunction of the attributes.

A. Winnow

Winnow, first introduced by Littlestone [20], is an algorithm for learning monotone disjunctions of the form

$$h = x_1 \vee x_2 \vee \ldots x_n$$

The algorithm is presented at each step with the values of x_i and is asked to make a prediction on the value of h. It is then presented with the correct value of h. In our mobile agent model, each of the buddies of the mobile agent under consideration can be treated as a literal x_i in the above expression. The assumption is that a certain subset of the buddies is trustworthy, and the agent can assume that a given host is friendly, if any one of the trusted subset of buddies says that the host is friendly. Or one can also assume that a host is friendly by default, unless on the trusted agents says that the host is unfriendly. It is clear that the latter approach is far less conservative that the former. In our simulation, we go with case 1, where an agent assumes a host to be hostile, unless it is recommended by other agents.

Note that if it is required to use a conjunction instead of a disjunction, this can be done simply by converting a conjunction to a disjunction using De Morgan's laws. So if an agent wants to arrive at a set of experts, so that it moves to a host only when *all* of them agree that it is safe, Winnow can be used for the purpose.

We give the Winnow [20] algorithm below:

1) Initialize the weights $w_1, w_2, \ldots, w_n$, for all the experts to 1.
2) Given an example $x_1, x_2, , x_n$, output $h' = 1$ if $w_1x_1 + w_2x_2 + \ldots + w_nx_n \geq n$, and output $h' = 0$ otherwise.
3) Receive the correct value of h.
4) If the algorithm made a mistake:
   a) If the algorithm predicts negative on a positive example, then for each $x_i = 1$, double the value of w_i.
   b) If the algorithm predicts positive on a negative example, then for each $x_i = 1$, cut the value of w_i in half.
5) Go to step 2.

It can be shown [19][23] that Winnow makes at most

$$2 + 3r(1 + \log n)$$

mistakes, where the target concept is a disjunction of r variables.

In the case of a constantly evolving network, it is very unlikely that a consistent target disjunction will exist. In this case, we are trying to predict what is known as a shifting/drifting disjunction [22][23], or a disjunction that changes with time. In a drifting disjunction setup, variables are added to and removed from the target expression from time to time. There is a surprisingly simple variation of Winnow in that case, which gives $(\log n)$ mistakes, where n is the total number of variables that are added to or removed from the original disjunction during the course of the trials.

In the version of Winnow for drifting disjunctions, we never allow the weight of any variable(w_i) to decrease below $\frac{1}{2}$. The intuition behind this simple change is that the weight of any expert never decreases to such a small value, that it is not able to make a comeback when the situation changes and it starts predicting correctly. In the original version, if an expert made too many mistakes, its value could get so close to zero, that it would not be able to recover even when it starts predicting correctly. The 'drifting disjunction' version of Winnow is, thus, very well-suited to a dynamic framework.

VI. Simulation Setup

As part of our experiments, we try to see how well the randomized weighted majority and the Winnow algorithm perform over a network that is stable, and over networks with increasing levels of instability. We test the following hypotheses:

1) Online learning algorithms will be able successfully help agents to avoid malicious hosts even over a large network if:
   a) the network is stable (host statuses do not change).
   b) there exists some kind of locality property, that is, there is some kind of relation between hosts, so that agents that have information about certain hosts, are likely to have information about certain other hosts.
2) In the absence of the locality property, these algorithms will not do better than an unweighted averaging approach. However, if such relationships exist, the algorithms can discover them reasonably quickly, even when they have no advance information about such relationships.
3) Online learning algorithms will perform reasonably well, and better than other simpler approaches, even in an increasingly unstable network situation where hosts constantly change status (friendly to hostile, and vice versa).

To test our hypotheses, we use four kinds of test agents:

- Agent A_{wm}: uses the randomized weighted majority algorithm to maintain reputation information for other hosts (we set β to 0.25).
- Agent A_{winnow}: uses the Winnow algorithm, modified for drifting disjunctions, to maintain reputation information for other agents.
- Agent A_{mean}:uses the unweighted mean of the feedback from all users to make its decision.
- Agent A_{exp}: remembers its past experience with each host, and uses this experience to make its decision.

The simulation procedure is given below:

1) As part of the simulation we create a set of 2048 hosts. Out of these hosts, around 50% are randomly labeled as hostile at the beginning of the simulation.
2) A set of 128 agents is also created, and the agents are randomly assigned to hosts.
3) Each agent stores a reputation weight for all the other agents in the community. It also stores a weight for all the hosts in the system. The agent reputation weights are initially all set to 1. The host weights are initialized randomly to a value between 0 and 1.
4) Each host is randomly assigned to a domain. A domain is simply a list of hosts. We initially create 10 domains in the network. Each agent is also assigned a domain.
5) At each iteration, an agent randomly picks a host from the ones assigned to its domain. This ensures that the movement of the agents around the hosts is non-random, and there is a certain locality to the movement of agents, that is, agents that visit a certain host, are also likely to visit others.
6) The agent queries all the other agents about the status of the host (friendly/hostile). On receiving a binary feedback from all the agents about the host, the agent decides whether to visit the host or not.
7) If the agent decides not to visit the host, the iteration ends. If the agent visits the host, it scores +10 points if the host is friendly, and -10 if the host is hostile.
8) Also, during each iteration:
   a) With probability γ, the domain of an agent is randomly changed.
   b) With probability θ, the status of a host is randomly changed.

The variables γ and θ represent the degree of instability in the network. In each iteration, with probability γ, an agent leaves its current domain and moves to another. Every time an agent leaves its domain, it has to contend with a new set

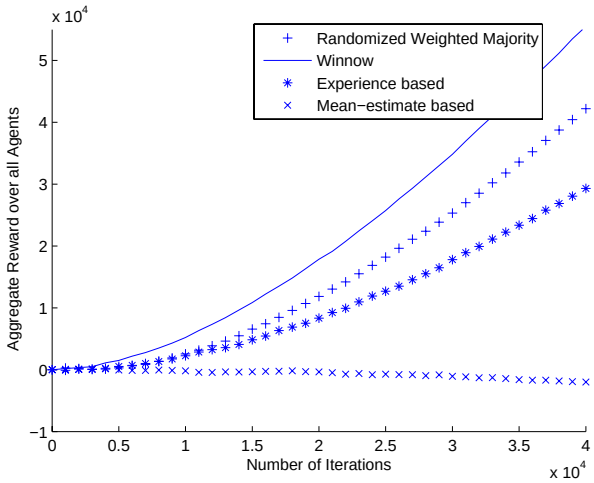

Fig. 1. Algorithm performance over completely stable network ($\gamma = 0, \theta = 0$)

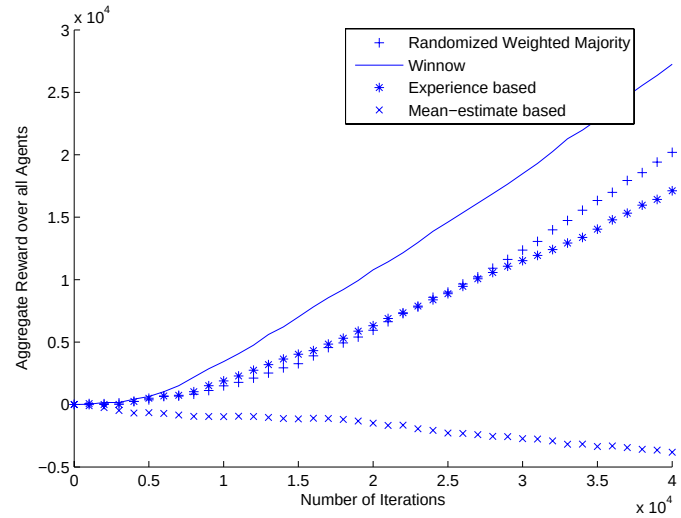

Fig. 2. Algorithm performance over slightly unstable stable network ($\gamma = 0.01, \theta = 0.01$)

of hosts. Also, the agents that provided accurate reputation information in the previous domain no longer provide accurate information. The agent now quickly needs to realize that the network situation has changed, and needs to adapt accordingly.

The current status of a host picked at random, changes (from friendly to hostile or vice versa) at each iteration, with probability θ. The higher the value of θ, the more often hosts change their status in the network, and the more unstable the network is.

We also set up another experiment where no domains are assigned to agents to hosts, and agents randomly decide to visit any host in the setup, in each iteration. This is our control experiment, which tests our hypothesis that online learning algorithms can only be effective if there is some kind of locality/domains existing over the network.

VII. Experimental Results

We ran simulations for the four agents described above, for 40,000 iterations. The graphs show results averaged over 10 test runs.

Figure 1 shows the performance of all four agents in a completely stable network, where 50% of the hosts are hostile, but hosts never change their state. As can be seen from the graph, Winnow performs better than the other algorithms. The experience based agent also performs reasonably well. The mean-estimate based agent, however, barely breaks even. The reason for this is, over a large network, most of the other agents will have incorrect information. Hence any agent that weights the information provided by other agents equally, will end up drowning the correct information under the noise.

Figure 2 shows the performance of the agents over a moderately unstable network ($\gamma = 0.01, \theta = 0.01$). The values of γ means that an average agent changes its domain around 4 times during the course of the simulation. Also around 400 hosts (approximately 20%) change their status during the course of the simulation. As can be seen from the results, the performance of both A_{wm} and A_{winnow} comes down by around 50%, in comparison to the completely stable network

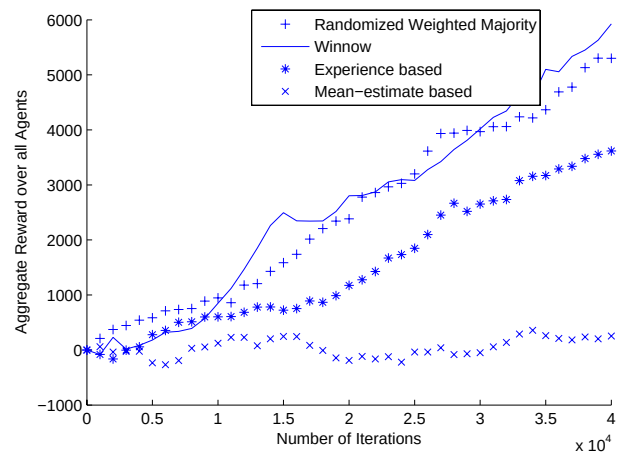

Fig. 3. Algorithm performance over highly unstable stable network ($\gamma = 0.1, \theta = 0.1$)

setup (Figure 1). The reward attained by A_{exp} goes down by around 67%. However, A_{winnow} still performs better than the other agents.

Figure 3 shows the performance for a highly unstable network ($\gamma = 0.1, \theta = 0.1$). In this case, an agent changes its domain around 40 times, and the status of each host changes around twice, during the course of the simulation. As can be seen from the graph, A_{winnow} still performs better than other agents, even though the reward accumulated now is only 10% of the reward accumulated in the completely stable case.

As can be seen from the graphs, the performance of reputation-based online learning algorithms fall drastically as the instability in the environment increases. However, they are still able to do reasonably well. In comparison, given that 50% of the hosts are hostile at any given time, a completely naive approach can be expected to accrue zero reward. Clearly these algorithms do much better than that.

Figure 4 gives the results where the agents move randomly

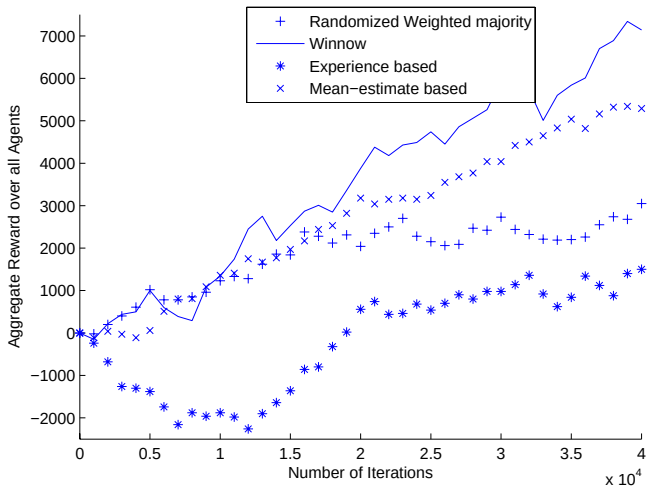

Fig. 4. Algorithm performance over stable network with no locality information

along the network. This may be considered to be the noisiest case, where there is no locality information for the algorithms to exploit. In this case, A_{mean} actually does better than A_{wm} as well as A_{exp}. However A_{winnow} still does better than the rest.

As can be seen from the graphs, A_{winnow} consistently does better than all other algorithms. The reason for this is that the other agents, particularly A_{wm} cannot 'unlearn', that is, once they start placing a high value on the reputation information provided by a particular source, when the source becomes irrelevant or deteriorates in quality, they take a long time to realize this. A_{winnow}, on the other hand, is able to adapt much faster to changes in its environment.

Also, as can be seen from Figure 4, when there is no domain/locality information in agent movement, this leaves no information for the adaptive algorithms to exploit. Winnow is actually able to realize this is the case, and thus manages to perform reasonable well. Randomized weighted majority, on the other hand, performs even worse than a simple averaging based approach.

VIII. Conclusions and Future Work

In the past, research in reputation modeling for trust has generally focused on environments that are stable or evolve very slowly. In this research, we tried to analyze how well traditional approaches such as the weighted majority algorithm are suited to a fast evolving scenario. We also experimented with Winnow, another algorithm that uses the mixture-of-experts framework. We found that Winnow consistently performed better than the weighted majority algorithm, and was much more robust to environment instability. However, more complex algorithms exist that have been developed specifically to be capable of handling highly dynamic situations. Some of these attempt to modify the weighted majority algorithm to make it more responsive to change [21], others make modifications to Winnow to allow for more parametric flexibility [22]. We plan to investigate these algorithms in the future.

We also found that it is important that certain conditions hold in the network/environment for an adaptive reputation-based algorithm to be useful, compared to a simple mean-based approach. We defined this condition as *locality* or *specialization*. We found that if such conditions existed in the network, online learning algorithms are capable of discovering and exploiting them to great advantage. However, if these condition do not exist, we found that a weighted approach actually does worse than a simple mean based approach. We also found that adding even a small amount of instability to the environment caused the algorithms' performance to fall drastically. Though for our experiments, we modeled a mobile agent community operating over a network, our conclusions can be seen to be generally applicable.

References

[1] S. Franklin and A. Graesser, Is it an Agent, or just a Program?: A Taxonomy for Autonomous Agents, *Third International Workshop on Agent Theories, Architectures and Languages*, Springer-Verlag, 1996.

[2] D. Lange and M. Oshima, Seven Good Reasons for Mobile Agents, *Communications of the ACM*, Volume 42 , Issue 3, pp. 88-89, 1999.

[3] N. R. Jennings and M. Wooldridge, Applications Of Intelligent Agents, *Agent technology: foundations, applications, and markets*, pp. 3-28 Springer-Verlag NY(1998).

[4] C. Badica, M. Ganzha, M. Paprzycki, Mobile Agents in a Multi-agent E-Commerce System, *Proceedings of the Seventh International Symposium on Symbolic and Numeric Algorithms for Scientific Computing*, 2005.

[5] J. Page, A. Zaslavsky, M. Indrawan, A Buddy Model of Security for Mobile Agent Communities Operating in Pervasive Scenarios, *Proceedings of the second workshop on Australasian information security, Data Mining and Web Intelligence, and Software Internationalization*, Vol. 32, 2004.

[6] J. Page, A. Zaslavsky, M. Indrawan, Countering Security Vulnerabilities using a Shared Security Buddy Model Schema in Mobile Agent Communities, *Proceedings of the 1st International Workshop on Safety and Security in Multi- Agent Systems (SASEMAS2004)*, pp. 85-101,New York (USA), July 2004.

[7] W. Jansen and T. Karygiannis, Mobile Agent Security.*NIST Special Publication 800-19*. National Institute of Standards and Technology, 2000.

[8] H.F. Wedde and M. Lischka, Modular Authorization, *Proceedings of The 6th ACM Symposium on Access Control Models and Technologies (SACMAT)*, Chantilly, Virginia, USA May 3-4, pp. 97-105.

[9] B. R. Banerjee, R. Mukherjee, S. Sen, Learning Mutual Trust, *Working Notes of AGENTS-00 Workshop on Deception, Fraud and Trust in Agent Societies*, 2000.

[10] K. Fullam and K. S. Barber, Learning trust strategies in reputation exchange networks, *Fifth international joint conference on Autonomous agents and multiagent systems*, 2006.

[11] B. Yu, and M.P. Singh, An Evidential Model of Distributed Reputation Management, *Proceedings of the first international joint conference on Autonomous agents and multiagent systems*, Bologna, Italy, pp. 294-301, 2002.

[12] B. Yu, M.P. Singh, K. Sycara, Developing Trust in Large-Scale Peer-to-Peer Systems, *Proceedings of First IEEE Symposium on Multi-Agent Security and Survivability (MASS)*, August 2004, pages 1-10.

[13] D. DeAngelis, K. Fullam, K.S. Barber, Effects of Communication Disruption in Mobile Agent Trust Assessments for Distributed Security, *Proceedings of The Workshop on Trust in Agent Societies at The Fourth International Joint Conference on Autonomous Agents and Multiagent Systems* (AAMAS-2005), Utrecht, July 25-29, pp. 27-37, 2005.

[14] R. Sutton and A.G. Barto, *Reinforcement Learning: An Introduction*, MIT Press, Cambridge, MA, 1998.

[15] T. Mitchell, *Machine Learning*, McGraw Hill, 1997.

[16] J. Kivinen and M. Warmuth , Averaging Expert Predictions, *Computational Learning Theory: 4th European Conference (EuroCOLT 99)*, pages 153-167. Springer, Berlin, 1999.

[17] Y. Freund, R. Schapire, Y. Singer, M. Warmuth, Using and combining predictors that specialize, *Proceedings of the Twenty-Ninth Annual ACM Symposium on the Theory of Computing*, pages 334–343, 1997.

[18] N. Littlestone and M. Warmuth, The weighted majority algorithm, *Information and Computation*, 108:212-261, 1994.

[19] N. Cesa-Bianchi, Y. Freund, D. Haussler, D. Helmbold, R. Schapire, M. Warmuth, How to Use Expert Advice, *Journal of the ACM*, Vol. 44, No. 3, pp 427-485, 1997.

[20] N. Littlestone, Learning Quickly When Irrelevant Attributes Abound: A New Linear-Threshold Algorithm, *Journal of Machine Learning*, Volume 2(4), April 1988.

[21] M. Herbster and M. Warmuth, Tracking the Best Expert, *Journal of Machine Learning*, Volume 32(2), pp: 151 - 178, 1998.

[22] P. Auer and M. Warmuth, Tracking the Best Disjunction, *Journal of Machine Learning* Vol. 32(2), pp. 127-150, August 1998.

[23] A. Blum, On-line Algorithms in Machine Learning, *Lecture Notes In Computer Science*, Vol. 1442.

On the convenience of using similarity between tasks in a trust and reputation model for contracting web services

Alberto Caballero*
Instituto Superior Politécnico
Jose Antonio Echeverría
La Habana, Cuba
albe_cu@yahoo.com

Juan A. Botía
Universidad de Murcia
Campus Espinardo
Murcia, España
juanbot@um.es

Antonio F. Gómez-Skarmeta
Universidad de Murcia
Campus Espinardo
Murcia, España
skarmeta@um.es

ABSTRACT

This paper analyzes the behavior of a trust and reputation model for agents in P2P environments, taking as starting point the proposals made in previous works. We present partial results of the refinement process of our model in order to identify the suitable structure, functions and parameters. In our model, where each agent can be either consumer or provider of resources, the aim is to propose the best response for the specified requirement, using trust and reputation information. WSMO is the base for definition of requirements of consumers and responses of providers, giving a common representation to facilitate the adaptation of the model for any application domain. This paper talks about the principal features of structure and functionality of the model, showing the changes introduced since the last write up of the model. Also, it defines and proves some alternatives of definition for two functions: quality of response of an agent given the request, and estimation of trust and reputation values. It gives experimental information that makes possible to analyze a number of ways to obtain the quality of the response given the requirements. Also, it shows results related with the increase of the satisfaction of the recommended response when it uses a right way to estimate trust and reputation values using stored information about similar tasks.

Categories and Subject Descriptors

I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence—*Multiagent systems, Intelligent agents*

*Supported by the Programme Alban, the European Union Programme of High Level Scholarships for Latin America, scholarship No.E05D049799CU and also by the Spanish Ministry of Education and Science by the Research Project TIN-2005-08501-C03-02.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
AAMAS '07 Honolulu, Hawaii
Copyright 200X ACM X-XXXXX-XX-X/XX/XX ...$5.00.

General Terms

Models, Reliability

Keywords

Trust, Reputation, Web Services, WSMO

1. INTRODUCTION

Trust and reputation models can be very useful in P2P environments, where peers interact in a decentralized manner trying to obtain the best system solution for a given user problem. Peers can be agents that represent providers or consumers of resources, interacting without any intermediate node [10, 18].

Each agent may expose very different behaviors. Generally, consumers would want to contract only providers with the best behaviors. A trust and reputation model, where each agent manages his own updated model about the behavior of the rest of agents in the system, can be used to bring out information to separate good-performance nodes and bad-performance ones in order to solve a given problem.

In previous works [2, 1], we consider a P2P environment where agents must be capable to contract tasks to agents and try to achieve the best performance. For that, we propose a model wich delivers mechanisms to support adaptive negotiations between agents. The shown model tries to decide which are the agents to negotiate with, based on the calculation of confidence values associated in every moment with the specification of the task that it is necessary to contract. These values are obtained from the agent experience and the information interchanged between agents (i.e. reputation), for the interaction task and any similar.

The information referring experience of each agent must include the satisfaction that this agent obtained from others. Generally, the way to measure the satisfaction may be very different according to the application domain and representations for tasks and responses.

In our trust and reputation model, WSMO [5, 20] is used to represent tasks as a Web service request and the response as a Web service description that satisfied it. Also, others domain-dependent features of the model, such as satisfaction of the task (service request) given the response (service description), and similarity between two tasks (when the trust values for unknown tasks are obtained from trust values for similar tasks), are described by means of WSMO.

This model is in a refinement and adjusting process, in or-

der to find the most suitable definition for some of its functions and knowledge representations. The main objective of this paper is to propose and experimentally compare several alternatives for two functions of the model: satisfaction of the responses obtained after the execution of the contracted task; and the way to estimate trust and reputation values for unknown tasks from knowledge about similar tasks.

The rest of the paper is organized as follows: section 2 introduces the main characteristics of the model, its structure and general functionality (subsection 2.1), the approach used to estimate trust and reputation from similar tasks (subsection 2.2), and the definitions of domain-dependent functions of the model (subsection 2.3). Section 3 compares experimental results to select a suitable way to obtain the satisfaction of solution given a task, and the way to estimate trust and reputation for unknown tasks using similarity between tasks. In the final section, we draw conclusions and open issues.

2. TRUST AND REPUTATION MODEL

In previous works we define a trust an reputation model between agents to propose a suitable response for a consumer requirement [2, 1].

Following the classification given by Ramchurn *et al.* [14], this is a learning based and evolutionary trust model at individual-level trust, where an agent has some beliefs about the honesty or reciprocative nature of its interaction partners. This model considers trust as an emergent property of direct interactions between agents and assume that agents interact many times. Like others, trust is considered as a social phenomenon based on multiple interactions between two parties [12, 15, 16, 17, 19].

Our model is characterized by the following three assumptions: (1) it is possible to calculate trust and reputation values to indicate who trusts who, (2) these values are based on the experience of the system taken from past interactions, among others; and finally that (3) it is possible to refine this value using new acquired knowledge added to the experience of the agent.

Contrary to other models [3, 15, 13, 22, 18, 11, 9, 7], where trust and reputation values are obtained as global values only associated to a peer, our model associates trust and reputation to the specification of the task that agents need to delegate. Griffiths is an example that proposes a model to manage trust between agents with respect to a particular task [8].

Trust and reputation values are associated to the task the agent needs to solve. For example, if a given agent needs to obtain the confidence in other agent, it reviews only the data related with the task, it considers only the experiences related with the specification of the task. But, it is possible that the agent does not have any information about the previous behavior of its partner performing the specified task but he knows instead the previous behavior performing similar tasks. In this cases, our model offers the way to approximate trust and reputation values for the specified task using available trust and reputation information about similar tasks. Here, based on experimental results, we proposes some ways to estimate trust and reputation from similar tasks. This is one contribution of this work (please, see sections 2.2 and 3.2 for details on this particular issue).

Our model uses WSMO, a W3C standard proposal, as the base for definition of knowledge structures needed by model. Service requests can be represented by tasks using the WSMO concept of "Goal". In the other hand, the response (describing the Web service that satisfies the task) can be represented using the concept of "Web Service" given by this standard [20, 4].

In previous works we identify some interesting non - functional properties of Web services and goals to manage trust, quality, costs, etc. in order to establish measures of similarity criteria between goals and measures of goal satisfaction given Web services descriptions [2, 1].

At present, the model is in process of refinement. Following, we present the main features of the model and functions which evolved since the last write up [1].

2.1 Model structure

Our model is composed by a set of information bases that each agent stores about the behavior of the others and a set of functions to operate with these bases. Following a distributed approach imposed by P2P environments, each agent manages its own bases of experiences. Functions produces values to guide the interactions between agents, based on trust and reputation concepts.

Basically, the model is structured and operates following the schema given in the figure 1.

There are two bases of experiences to obtain trust and reputation values for a given task: base of experiences for trust (IET) and for reputation (IER). These values are produced by means of the right combination of some functions. First, by introspection of the bases of experiences, the model calculates direct trust (DT), reputation (R) and reliability of DT ($DTRL$), combined to produce a unique value of trust, using the function (T). This value, aggregated form direct trust and reputation, is used to select the partners in the interaction, to ask about the solution or about others.

If the bases of experiences do not have information for a given task, the model obtain the values of trust (by means of function DT) and reputation (by means of function R) for a similar task and combines these values with the similarity degree between two tasks, given by function D. For that, the model uses functions IT and IR to select the partners in the interaction for both purposes.

Initiator agent interacts with the recommended agent and solves his task with the response offered by partner. In this moment, the initiator agent produces information related with the interaction that he may use to update the bases of experiences for trust and reputation. For that, our model gives two functions: fulfillment of the promised satisfaction (P) and quality of the response (Q).

Following, we detail the structure and functionality of the model presented briefly above. Other functions are described in the rest of the paper or in the previous related works [2, 1].

The base of experience for trust IET (Initiator's Experience of Trust) must be represented in a tuple, where initiator agent stores information about others when they are contracted for to execute the task:

$$IET_i = \{(a_j, s_k, et_{i,j,k,l}) | a_j \in A, s_k \in S, et_{i,j,k,l} \in [0,1]\}$$

where IET_i is the trust experience of agent a_i at time t, A is the set of agents in the system, S is the set of possible specifications of tasks that the agent needs to contract, $et_{i,j,k,l}$ is the satisfaction degree of agent a_i when agent a_j offers a solution to the task s_k for the l-th time.

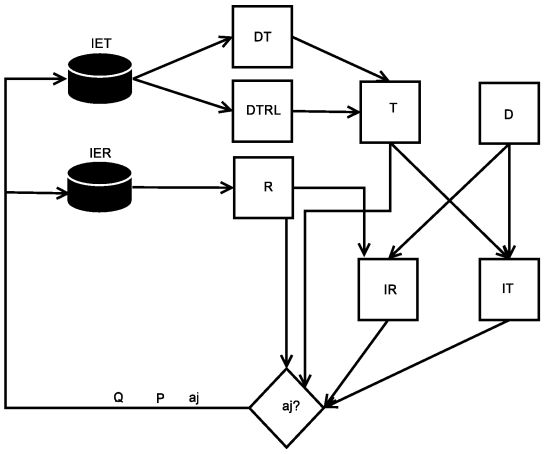

Figure 1: Relationship between different parts of the model

Also, the initiator agent must store data about the reliability of other agents when they offer reputation information IER (Initiator's Experience of Reputation):

$$IER_i = \{(a_j, s_k, er_{i,j,k}) | a_j \in A, s_k \in S, er_{i,j,k} \in [0,1]\}$$

where $er_{i,j,k}$ is the satisfaction degree of agent a_i when a_j offers reputation values about other agents when they performed the task s_k.

These bases of experiences are updated when an interaction is finalized. In order to update the bases of experiences the agent a_i evaluates the interaction (using a quality function Q described in section 2.3.1), taking into account the solution w_j as response of the task s_k.

The information about each particular interaction, that agent a_i carried out for a given task s_k, may be grouped in the set:

$$I(a_i, s_k) = \{(a_j, w_j) | a_j \in C(a_i, s_k),\ w_j \in W\},$$

where w_j is the response given in this interaction by agent a_j; W is the set of all possible responses, and $C(a_i, s_k)$ is the set of the most reliable agents to give solutions to task s_k according to the experience of agent a_i. This is a subset of the set of neighbors. There are some strategies to populate $C(a_i, s_k)$, one of them is proposed in previous works [2, 1]. Here, to simplify the study of the model, we consider the simple case where $C(a_i, s_k)$ is composed by all neighbors. The behavior of performance of the system in front of different strategies to select the neighbors is not the aim of this paper; it is considered in future works.

To update the base of experiences IET_i, for each agent a_j, that gives the solution w_j to the task s_k, the agent has a new experience:

$$edt_{i,j,k} = (a_i, a_j, s_k, et_{i,j,k})$$

with

$$(a_j, w_j) \in I(a_i, s_k)$$

where the trust value $et_{i,j,k}$ is a measure obtained from the real quality of the solution (Q) and the fulfillment (P) of the promised satisfaction ($ec_{i,j,k}$):

$$et_{i,j,k} = Q(w_j, s_k) \cdot P(ec_{i,j,k}, Q(w_j, s_k))$$

This way, the proposed model avoids that an agent a_j, with a low promised satisfaction $ec_{i,j,k}$ and a medium-quality solution for task s_k, may obtain a high satisfaction degree $et_{i,j,k}$. The satisfaction degree must be the combination of real quality of the solution (Q) and the fulfillment of the promised quality (P). The definitions of functions P and Q are given in section 2.3.1.

Here, the model takes into account that is possible to add the new experience $edt_{i,j,k}$, without having to analyze how many experiences are in the base IET_i.

The base of experiences for reputation IER_i has an unique value of reputation $er_{i,j,k}$ to indicate, according to experience of agent a_i, the reliability of agent a_j to give reputation information about other agents performing task s_k. For that, the model takes into account the variation (produced during the interaction) on trust about agents recommended by a_j. Agents recommended to a_i by a_j for the task s_k is denoted by $M_j(a_i, s_k)$ (for more information, please, see [2]).

For each recommended agent $a_r \in M_j(a_i, s_k)$, the model obtains the trust value that agent a_i had about a_r (denoted by $f^{(0)}_{i,r,k}$) at the beginning of the interaction and the new value (denoted by $f_{i,r,k}$) at the end. The trust value to give reputation information at the begining of the interaction $er^{(0)}_{i,j,k}$ is modified combining the mean of all differences between final and previous trust values for each agent a_r, about agent a_j:

$$er_{i,j,k} = \begin{cases} er^{(0)}_{i,j,k} + (1 - er^{(0)}_{i,j,k}) \cdot \Delta & : \quad \Delta \geq 0 \\ er^{(0)}_{i,j,k} \cdot (1 + \Delta) & : \quad \Delta < 0 \end{cases}$$

where Δ is the mean of all differences between final and previous trust values:

$$\Delta = \frac{\sum_{a_r \in M_j(a_i, s_k)} f_{i,r,k} - f^{(0)}_{i,r,k}}{|M_j(a_i, s_k)|}$$

This function exposes a suitable behaviour in updating the value of confidence on an agent given information about the performance of others.

The value of the reputation $er_{i,j,k}$ will be better than $er^{(0)}_{i,j,k}$ when the trust on recommended agents from $M_j(a_i, s_k)$ is improved during the interaction. In other words, the confidence of an agent as recommender of others is increased or decreased in the way of the variation of trust values on recommended agents. The confidence of the agent, giving reputation information, is increased when the mean of the differences of trust values is positive. If the trust of recommended agents is decreased, the confidence of the agent as recommender is decreased too.

Obtaining trust and reputation As we present in [2, 1], the global value of trust $f_{i,j,k}$ is obtained from the bases of experiences like in REGRET [15], using this function:

$$f_{i,j,k} \equiv T(a_i, a_j, s_k) = \\ DTRL(a_i, a_j, s_k, IET_i)\ DT(a_i, a_j, s_k, IET_i) + \\ (1 - DTRL(a_i, a_j, s_k, IET_i)) R(a_i, a_j, s_k, CR_{sup}(a_i, s_k))$$

where $DT(a_i, a_j, s_k, IET_i)$ represents the direct trust value that agent a_i assigns to agent a_j for task s_k according to his own experience in IET_i, $DTRL(a_i, a_j, s_k, IET_i)$ is the reliability of this measure, according to the multiplicity of

the information and the deviation around the mean, and $R(a_i, a_j, s_k, CR_{sup}(a_i, s_k))$ is the reputation value that agent a_i assigns to agent a_j for task s_k according to the experiences of the most reliable agents to give reputation ($CR_{sup}(a_i, s_k)$) taking into account the information stored in his own base IER_i. It is obtained using the equation proposed in [2], taking into account some trust and reputation models, given by Golbeck and Hedler [7, 6], Zacharia [23] and Schillo [16].

2.2 Obtaining trust and reputation from similar tasks

It is possible that an agent does not have information about performance of other agents for a given task. In this case, we propose to approximate the trust and reputation values using a similar task whose accomplishment has been previously done by known agents and requested by a_i. The model obtains this approximation using a similarity degree between the most similar well-known task and the unknown one.

Our model defines the function D to obtain the similarity degree between two tasks from the comparison of the tasks attributes. The definition of this domain-dependent function is given in section 2.3.2.

This way, combining the trust or reputation in the most similar task (s_p) with the similarity degree between the two tasks $D(s_k, s_p)$, we define indirect trust (IT) and indirect reputation (IR) functions to approximate direct trust or reputation values, respectively:

$$IT(a_i, a_j, s_k, s_p, IET_i)) = DT(a_i, a_j, s_p, IET_i) \oplus D(s_k, s_p),$$

and

$$IR(a_i, a_j, s_k, s_p, CR_{sup}(a_i, s_k)) = R(a_i, a_j, s_k, CR_{sup}(a_i, s_k)) \oplus D(s_k, s_p)$$

There are several ways to combine trust and reputation values with similarity. To simplify the notation, in order to define several alternatives to the operator $\oplus$, we use:

- IT to refer to $IT(a_i, a_j, s_k, s_p, IET_i))$,
- IR to refer to $IR(a_i, a_j, s_k, s_p, CR_{sup}(a_i, s_k))$,
- DT to refer to $DT(a_i, a_j, s_p, IET_i)$,
- R to refer to $R(a_i, a_j, s_k, CR_{sup}(a_i, s_k))$, and
- D to $D(s_k, s_p)$

We may consider the following alternatives for the operator $\oplus$ to combine trust and reputation with similarity:

IT_a: $IT = DT \cdot D$
$IR = R \cdot D$

IT_b: $IT = \sin\left(\frac{\pi}{2} DT \cdot D\right)$
$IR = \sin\left(\frac{\pi}{2} R \cdot D\right)$

IT_c: $IT = DT \cdot \sin\left(\frac{\pi}{2} D\right)$
$IR = R \cdot \sin\left(\frac{\pi}{2} D\right)$

IT_d: $IT = \sin\left(\frac{\pi}{2} DT\right) \cdot \sin\left(\frac{\pi}{2} D\right)$
$IR = \sin\left(\frac{\pi}{2} R\right) \cdot \sin\left(\frac{\pi}{2} D\right)$

IT_e: $IT = \frac{DT+D}{2}$
$IR = \frac{R+D}{2}$

IT_f: does not consider any estimation

The previous alternatives are considered in order to compare its performance because they seem the most trivial options according to the considerations and characteristics of our model.

The alternatives IT_a and IT_e combines trust/reputation with similarity following the multiplication and the arithmetic mean of these values, respectively. In spite of bearing them in mind, we think that these variants do not improve the satisfaction that produces the alternative IT_f (does not consider any estimation). IT_a produces values very much minor that the values to combine. For very high values of trust/reputation and similarity, it can obtain average values of estimation. For that, we think that is necessary a previous transformation of the values to combine, according to the intentions of our model. Alternative IT_e does not seem to be the suitable one either; does not have sense to talk about the middle point between two values that represent different magnitudes.

The model must consider the estimation of the trust / reputation when there is a high degree of similarity between the tasks. The rest of the alternatives that we propose incorporate the function sin to amplify the value of the similarity, the trust / reputation, or their combination. The value of estimation must be as similar as possible to the original value for those cases where the similarity is very high. The most suitable variant seems to be IT_c.

Section 3.2 shows some experimental results in order to compare these alternatives.

2.3 Domain-dependent functions: Quality and Similarity

There are two functions in our model whose definition depends on the representations of tasks and responses: quality and similarity.

Each task request s_k and response w_j are described by the set of non-functional WSMO properties (please, see section 2 and paper [2]), and others added by the user, depending on the application domain.

For each property of request or response, the model must define a normalization function to make independent the domain of the real world values from model-managed values. The model uses values in the range [0,1] to represent the convenience of the property, independent of the original property domain.

Both non-functional properties defined by WSMO as application domain properties, are transformed using normalization functions. The model manages values in the range [0,1], where a value near to 0 indicates a non-desired value in the original property, and values near to 1 indicate high-desired values in the original properties.

Following, we show the approach we use to define these functions using the WSMO concepts to represent tasks and responses.

2.3.1 *Quality of the solution given a task*

Our model uses two functions to evaluate the satisfaction of the initiator agent through the fulfillment of the promised satisfaction degree and the quality of the solution according to the task.

The fulfillment of the promised satisfaction indicates to what extent, the responder agent fulfills the promised quality ($ec_{i,j,k}$). Basically, the value of this function results from a comparison between the agreement quality value $ec_{i,j,k}$ and the real quality of the given solution, denoted by $Q(w_j, s_k)$. To determine the fulfillment of the satisfaction agreement, we may define a function P:

$$P(ec_{i,j,k}, Q(w_j, s_k)) = \begin{cases} 1 & : \quad Q(w_j, s_k) \geq ec_{i,j,k} \\ 1 - (ec_{i,j,k} - Q(w_j, s_k)) & : \quad Q(w_j, s_k) < ec_{i,j,k} \end{cases}$$

The value of the function $P(ec_{i,j,k}, Q(w_j, s_k))$ represents the fulfillment degree of promised quality ($ec_{i,j,k}$), comparing this value with the quality of the solution w_j for task s_k. If the real satisfaction degree overcomes the promised value, the function returns 1, otherwise it is an indicator of the difference between promised and real values.

The quality of the solution, denoted by $Q(w_j, s_k)$, indicates how much the response w_j satisfies the requirements specified in the task s_k. Calculation of this value is based on the comparison of both concepts.

According to the convenience of the value of each property, and following some ideas given by WSMO [21, 20] we can to define a set of the most relevance attributes of a given concept. For each task (s_k) or response (w_j), we can split the good-value attributes into the sets R_g and R_w, respectively.

If R_u is the set of properties used to defined a task or response, then $R_g \subset R_u$ and $R_w \subset R_u$ consist of the most prominent attributes for each concept, according to the value of each attribute. To construct these sets, we consider that the attribute b_i of s_k (we denote with $s_k.b_i$) is a good-value attribute and hence $b_i \in R_g$ if $s_k.b_i \geq \lambda_i$ (λ_i is a domain-dependent threshold value). In the same way, an attribute b_i of w_j is a good-value and $b_i \in R_w$ if $w_j.b_i \geq \lambda_i$.

Using these sets we can define three alternatives to obtain the quality of the response w_j to satisfy the task s_k:

Q_a: following the WSMO Web service discovery process

$$Q(w_j, s_k) = \begin{cases} 1 & : \quad R_g = R_w & Match \\ 0.75 & : \quad R_g \subseteq R_w & Match \\ 0.5 & : \quad R_g \supseteq R_w & PartialMatch \\ 0.5 & : \quad R_g \cap R_w \neq \emptyset & PartialMatch \\ 0 & : \quad R_g \cap R_w = \emptyset & NoMatch \end{cases}$$

According to this definition of the satisfaction degree function, maximum satisfaction degree is obtained when all important (good-value) attributes desired in goal s_k are important (good-value) attributes in Web services w_j. Contrary, the worst satisfaction is obtained when no prominent attributes of goal s_k are satisfied by important attributes of Web service w_j. Also, the satisfaction function considers others intermediate cases.

In this alternative, discovery process based on simple semantic descriptions of services, given by WSMO, acts as a function that indicates the matching degree of the Web service (response w_j) and the desired goal (task s_k), taking into account the matching between the most important attributes of these concepts.

Q_b: considering how many task attributes are satisfied by response

$$Q(w_j, s_k) = \sin\left(\frac{\pi}{2} \cdot \frac{|Q'_{w_j,s_k}|}{|R_u|}\right)$$

where R_u is the set of all properties of tasks and responses, $Q'_{w_j,s_k} \subset R_u$ is the set of these properties such that its values in the task s_k are less restrictive than the values in the response w_j:

$$Q'_{w_j,s_k} = \{b_i | b_i \in R_u, v_{s_k}(b_i) \leq v_{w_j}(b_i)\}$$

Satisfaction degree function is like a ratio between the satisfied s_k attributes and the total number of attributes R_u of any task or response. The maximum satisfaction degree is obtained when all (not only good-value) attributes desired in goal s_k are satisfied by Web services w_j. Contrary, the worst satisfaction is obtained when any attribute of goal s_k is satisfied by attributes of Web service w_j.

Q_c : considering how many good-value task attributes are satisfied by response

$$Q(w_j, s_k) = \sin\left(\frac{\pi}{2} \cdot \frac{|Q''_{w_j,s_k}|}{|R_g|}\right)$$

where R_g is the set of good-value attributes of task s_k, $Q''_{w_j,s_k} \subset R_g$ is the set of these properties such that its values in the task s_k are less restrictive than the values in the response w_j, if the property is consider a good-value for the response w_j:

$$Q''_{w_j,s_k} = \{b_i | b_i \in R_g, b_i \in R_w, v_{s_k}(b_i) \leq v_{w_j}(b_i)\}$$

This alternative is similar to previous one. But, in this case, satisfaction is like a ratio between the satisfied s_k important attributes and the number of important attributes R_g of the task s_k. The maximum satisfaction degree is obtained when all good-value attributes desired in goal s_k are satisfied by Web services w_j. Contrary, the worst satisfaction is obtained when any good-value attribute of goal s_k is satisfied by attributes of Web service w_j. This is an intermediate case between the two previous alternatives Q_a and Q_b.

A priori, we expect that Q_a does not produce the better values of satisfaction because it does consider only a few range of values of satisfaction (0, 0.5, 0.75, 1). The result values depend on the type of matching between task and responses, following the criteria of the Web service discovery process given by WSMO. It considers the satisfaction of groups of important attributes as a whole, but not the satisfaction of attributes independently.

In front of limitations of approach given by Q_a, we proposes an alternative where the satisfaction produces a wider spectrum of values. Q_b considers the amount of task attributes that are satisfied by response, with out taking into account the relevance of the attributes. This alternative should offer better values of satisfaction, because it is capable to represent in a better way the great amount of cases that occur.

Other hand, Q_b is an intermediate case between Q_a and Q_b that takes into account the satisfaction of each attribute (like Q_b) but only of good-values attributes (like Q_a).

Given the limitations of the Q_a approach (following the Web service discovery process in WSMO), we suppose that the best alternatives to obtain the quality of the response should be Q_b or Q_c. In these cases, the match between task and response is given by the satisfaction of each attribute

of the task, and not by the relationship between groups of attributes (like Q_a).

Section 3.1 shows an experimental comparative analysis between these three alternatives.

2.3.2 Similarity between tasks (D)

The similarity between two tasks s_k and s_p is obtained from the comparison of the task attributes.

$$D(s_k, s_p) = 1 - \frac{1}{n} \cdot \sum_{i=1}^{n} |s_{k_i} - s_{p_i}|$$

where n is the number of task attributes, s_{k_i} is the i-th attribute of task s_k, and s_{p_i} is the i-th attribute of task s_p.

The similarity is defined as a the mean of the absolute difference between the n attributes of each task. The values produced by this function are real values between 0 and 1: 0 indicates the lowest similarity (entirely different tasks) and 1 the highest similarity (equal tasks).

The model proposes the definition of these two functions (quality and similarity) using a simple method based on the representation of domain concepts by means of WSMO. However, it may incorporate other properties according to the application domain, given the simplicity of these two domain-dependent functions [2].

3. EXPERIMENTS

We carried out some experiments in order to designate the best alternative for the functions of our model. Following, we present most relevant results related with the way to obtain the quality of the response given a task, and the way to approximate trust and reputation using knowledge related with similar tasks.

These experiments took place over the same generals conditions, using our own implementation in Java. We created 10 instances of agents, populating a community of agents with different responses. There were agents of several qualities, depending on the quality of its responses. Agents interacted during t=450 time steps. In each time step, one of them was selected to act as initiator in a negotiation with the rest. The WSMO goal of the initiator in the negotiation was a task selected from a predefined set of 10 different tasks. This task represented the requirement of the selected agent in the given time step.

In all time steps, initiator agent identified the entire community of agents as reliable agents because this experiment did not analyze the influence of any pre-selection of agents at the beginning of the interaction depending on trust.

Also, in order to obtain trust value from the bases of experiences we used the value $\delta = 0.8$ to the time modulating parameter to give higher importance to experiences closer to the current time step (please, see [2]).

The experiments were repeated 20 times and results averaged after that.

3.1 Alternatives to measure the quality

The experiment compares the evolution of the satisfaction for the solution recommended by the model, for three different alternatives to obtain the satisfaction degree. The recommended solution is the response offered by the most reliable agent.

The three alternative to obtain the quality (Q_a,Q_b,Q_c), compared in this experiment, are described in section 2.3.1.

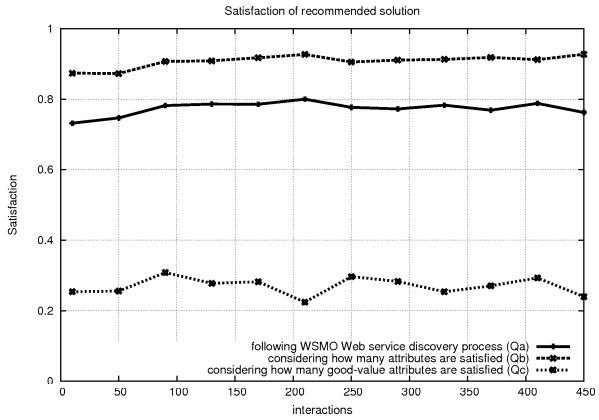

Figure 2: Comparison of satisfaction degree of recommended solutions, for each alternative to obtain the quality of the response given the task

In this case, we do not consider similarity between tasks, it means that the models does not estimate trust and reputation values for unknown tasks.

Figure 2 shows the satisfaction value of the responses given by the most reliable agent for each alternative. In each considered alternative, the satisfaction of recommended solution is stabilizes after a small number of experiences. Best satisfaction values are obtained using Q_b, where the quality measure considers the amount of task attributes that are satisfied by response. This alternative improves the satisfaction in the solution offered by the most reliable agents. It is the most suitable situation because the trust in the recommended agent is determined by the satisfaction degree stored in the bases of experiences (please, see section 2). We use this alternative in the next experiments in order to adjust the model.

3.2 Alternatives to estimate trust and reputation from unknown tasks

When agents do not have any experience about the performance of other agents for a given task, they need to estimate trust and reputation values. In this way, the model proposes to uses trust and reputation values associated with other task. For that, it gives a similarity function (please, see section 2.3.2) to determine which is the most similar task, in order to use it to approximate these values. Besides, the model needs to give the way to obtain the estimated values from similarity degree and the values for the most similar task. Section 2.2 gives some alternatives. This experiment compares them in order to determine the best way to estimate trust and reputation from a similar task.

Quality of response is obtained using the best alternative proposed in the previous experiment (given by Q_b), that considers the amount of task attributes that are satisfied by response. (Please, see sections 2.3.1, 3.1).

Figure 3 shows the satisfaction value of the responses given by the most reliable agent for each alternative. The best results are obtained using the alternative IT_c. When trust and reputation are estimated using equations

$$IT(a_i, a_j, s_k, s_p, IET_i)) = DT(a_i, a_j, s_p, IET_i) \cdot \sin\left(\frac{\pi}{2} D(s_k, s_p)\right)$$

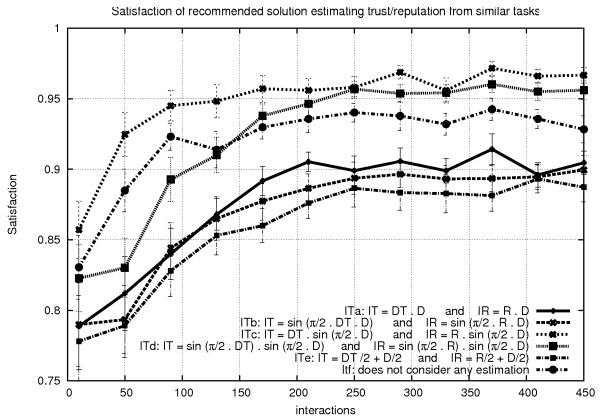

Figure 3: Comparison of satisfaction degree of recommended solutions, for several alternatives to combine trust / reputation and similarity between task in order to estimate trust/reputation for unknown task (please, see the definition of the alternatives in section 2.2).

and

$$IR(a_i, a_j, s_k, s_p, CR_{sup}(a_i, s_k)) = R(a_i, a_j, s_k, CR_{sup}(a_i, s_k)) \sin\left(\frac{\pi}{2} D(s_k, s_p)\right),$$

then, the satisfaction of the recommended solution, offered by the most reliable agent, is greater than other alternatives. It is greater than values obtained without consider any approximation using similar task. Hence, we demonstrate that using similarity between tasks improves the selection process.

3.3 Evolution of trust

The main objective of the trust and reputation model is to guide the response selection process. The model must select the response offered by high-reliable agent (with highest trust value for the task), but not in the response by it self because it is unknown.

It is important to know how trust values change when the model estimates trust and reputation from similar tasks. We analyze the evolution of trust values when the model uses these alternatives to combine trust and reputation with similarity. In this case, we study the evolution of trust of three types of agents. We classify the agents in accordance with the quality of the offered responses. We define three qualities of agents:

- $|R_w|$ near to $|R_u| \rightarrow$ HIGH
- $|R_w|$ near to $\frac{1}{2}|R_u| \rightarrow$ MEDIUM
- $|R_w|$ near to $0 \rightarrow$ LOW

The figure 4 shows trust evolution for three types of quality of the response that agents give, using the alternatives to combine trust/reputation and similarity between task in order to estimate trust/reputation for unknown task.

All the alternatives behave of the same way: trust in high-quality agents is increased; trust in medium-quality agents do not shows a significative variation; and trust in low-quality agents is lightly decreased. This indicates that the

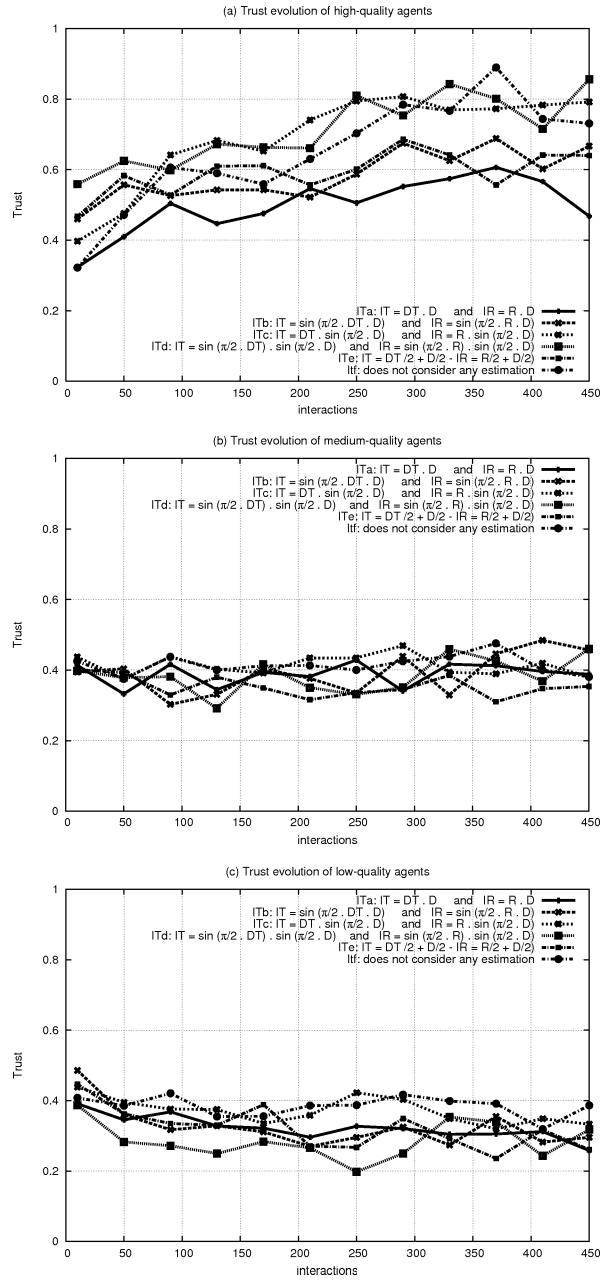

Figure 4: Trust evolution of different types of agents (a: high-quality, b: medium-quality, c: low-quality), using several alternatives to combine trust/reputation and similarity between task in order to estimate trust/reputation for unknown task (please, see the definition of the alternatives in section 2.2).

estimation of trust and reputation values does not affect the evolution of trust values for any type of agent. The variation of trust values associated to a given agent is determined by the quality of the solution that it offers, but not in the way that model uses to estimate this value for unknown tasks.

Also, figure 4 evidence the capability of the model to represent the quality of agents using trust values. It increases trust in agents with response of appreciated quality, to keep the values in medium-quality agents and to penalize the low-quality agents. In other words, model has a differenciated performance in front of different types of agents (high-, medium-, and low-quality).

In high-quality agent figure (figure 4.a), trust evolves from a default value (given by model at the beginning of simulation) to a value near to 1. It means that this type of agent offers good responses and, consequently, agents increase their associated trust value.

Figure 4.b shows how that model keeps the trust around an average value for agents with a medium-quality response. Trust in low-quality agents (4.c) behaves in a similar way, but with a little decrease. The model penalizes the bad solutions decreasing the trust value for agent that gives it.

Concluding, this experiment evidences that satisfaction in the response is increased when the model estimates trust and reputation for unknown tasks in a suitable way. The combination of trust and reputation with similarity between task using the alternative IT_c (please, see section 2.2) enhances the satisfaction of recommended response. This alternative produces greater satisfaction values than the alternative when system does not use any approximation of trust and reputation. Also, this way to estimate trust and reputation has a desired differentiated behavior according to the quality of the response given by each agent. Agents with a high-quality response increases its trust.

4. CONCLUSIONS AND FUTURE WORK

This paper analyzes the behavior of a model used to manage trust and reputation. In this model, agents should be able to contract the Web service of best behavior based on trust information, taking WSMO as conceptual framework in a P2P environment.

Trust and reputation information can be different in dependence of the specified task or requirement. Nevertheless, if the model ignores the behavior of the service for a given task, the values of trust can be approximated using the similarity degree between this and a well known task.

For the description of the services and their requests, the model uses some concepts given by WSMO: Web Service and Goal. This way, it facilitates the definition of some characteristics and functions that are dependent of the application domain, such as the satisfaction of a task given the answer or the similarity between two tasks.

The paper proposes some alternatives to obtain the satisfaction of a task given the answer. Shows some experimental results in order to compare these alternatives and decides which is the most suitable way to define the quality function. Quality measure obtained from the amount of task attributes that are satisfied by response shows better behavior than the measure obtained using WSMO Web service discovery process.

Also, it gives some ways to combine trust and reputation with similarity, when it needs to estimate these values for unknown tasks. It shows some experimental results comparing several alternatives and discusses which is the most appropriated.

Our proposed model increases the satisfaction of the recommended response when it uses a suitable way to combine trust and reputation with similarity, with respect to cases in which it does not use any estimation. The proposed way is capable to show different behaviors according to the quality of the response offered by agents. Nevertheless, we need to do some experiments where the quality of the agent response changes during the experiment, in order to prove the adaptability of the model.

We intend to define and prove some alternatives for other functions of the system (i.e. array of agents for asking them about trust and reputation, bases of experience updating process, etc.), in order to identify the parameters that affect the system performance and their high-recommended values.

5. REFERENCES

[1] A. Caballero, J. Botia, and A. Skarmeta. A new model for trust and reputation management with an ontology based approach for similarity between tasks. In F. et al., editor, *Multiagent System Technologies*, 4196. Springer, September 2006.

[2] A. Caballero, J. Botia, and A. Skarmeta. Trust and reputation model based on wsmo. In R. F. et. al, editor, *Proc. of Trust Workshop at AAMAS*, pages 9–18, May 2006.

[3] P. Dasgupta. Trust as a commodity. *Trust: Making and Breaking Cooperative Relations, D. Gambetta (ed.)*, pages 49–72, 1998.

[4] J. de Bruijn, H. Lausen, R. Krummenacher, A. Polleres, L. Predoiu, M. Kifer, and D. Fensel. *D16.1v0.2 The Web Service Modeling Language WSML. WSML Final Draft 20 March 2005.* W3C, http://www.wsmo.org/TR/d16/d16.1/v0.2, 2005.

[5] J. Dominique, D. Fensel, and D. Roman. Semantic Web Services with the Web Service Modeling Ontology (WSMO). *Agentlinks News*, (19):7–9, November 2005.

[6] J. Golbeck and J. Hendler. Accuracy of metrics for inferring trust and reputation in semantic web-based social networks. In *Proc. of EKAW'04*, 2004.

[7] J. Golbeck and J. Hendler. Filmtrust: Movie recommendations using trust in web-based social networks. In *Proc. of the IEEE Consumer Communications and Networking Conference*, January 2005.

[8] N. Griffiths. Task delegation using experience-based multi-dimensional trust. In *Proc. of Trust Workshop at AAMAS*, 2005.

[9] S. Marti. *Trust and Reputation in Peer-to-Peer Networks.* PhD thesis, Stanford University, 2005.

[10] D. Milojicic, V. Kalogeraki, and R. L. R. Peer-to-peer computing. Tech report: Hpl-2002-57, Hewlett Packard, 2002.

[11] M. Montaner, B. Lopez, and J. L. de la Rosa. Developing trust in recommender agents. In C. Castelfranchi and L. Johnson, editors, *Proc. of the First International Joint Conference on Autonomous Agents and Multi-Agent Systems*, 2002.

[12] L. Mui, M. Mohtashemi, and A. Halberstadt. A computational model of trust and reputation for e-business. In *Proc. of 35th Hawaii International*

Conference on System Science. IEEE Computer society, 2002.

[13] M. Prietula. Advice, trust, and gossip among artificial agents. *Dynamics of Organizations, A. Lomi and E. R. Larson (eds.)*, pages 141–177, 2000.

[14] S. Ramchurn, D. Huynh, and N. Jennings. Trust in multi-agent systems. *Knowledge Engineering Review*, 1(19):1–25, 2004.

[15] J. Sabater and C. Sierra. Regret: a reputation model for gregarious societies. In C. Castelfranchi and L. Johnson, editors, *Proc. of the First International Joint Conference on Autonomous Agents and Multi-Agent Systems*, pages 475–482, 2002.

[16] M. Schillo, P. Funk, and M. Rovatsos. 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, 2000.

[17] S. Sen and N. Sajja. Robustness of reputation-based trust: Boolean case. In C. Castelfranchi and L. Johnson, editors, *Proc. of the First International Joint Conference on Autonomous Agents and Multi-Agent Systems*, pages 288–293, 2002.

[18] Y. Wang and J. Vassileva. Trust and reputation model in peer-to-peer networks. In *Proc. of IEEE Conference on P2P Computing*. Linkoeping, Sweden, September 2003.

[19] M. Witkowski, A. Artikis, and J. Pitt. Experiments in building experiential trust in a society of objective-trust based agents. *Trust in Cyber-societies, R. Falcone, M. Singh, and Yao-Hua Tan (eds.)*, pages 111–132, 2001.

[20] WSMO Team. *Web Service Modeling Ontology (WSMO)*. W3C, http://www.w3.org/Submission/WSMO/, 2005.

[21] WSMO Team. *WSMO Web Service Discovery. WSML Working Draft*. W3C, http://www.wsmo.org/2004/d5/d5.1/v0.1/, 2005.

[22] T. Yamagishi, K. Cook, and M. Watabe. Uncertainty, trust, and commitment formation in the United States and Japan. *American Journal of Sociology*, 104(1):165–194, 1998.

[23] G. Zacharia. *Collaborative reputation mechanisms for communities*. PhD thesis, Massachusetts Institute of Technology, 1999.

Cognitive Model of Trust as Relational Capital

Cristiano Castelfranchi
ISTC-CNR
Via San Martino della Battaglia 44
00185, Rome, Italy
+390644595283
cristiano.castelfranchi@istc.cnr.it

Rino Falcone
ISTC-CNR
Via San Martino della Battaglia 44
00185, Rome, Italy
+390644595253
rino.falcone@istc.cnr.it

Francesca Marzo
ISTC-CNR
Via San Martino della Battaglia 44
00185, Rome, Italy
+390644595310
francesca.marzo@istc.cnr.it

ABSTRACT
This paper addresses the way in which trust can be considered as relational capital for agents that are trusted in a social network. The study is an elaboration of a previous work where the issue was presented. The basic idea is that trust can be viewed as an instrument both for an agent selecting the right partners in order to achieve its own goals (the point of view of the trustor), and for an agent of being selected from other potential partners (the point of view of the trustee) in order to establish with them a cooperation/collaboration and to take advantage from the accumulated trust. In order to analyze trust as the agents' relational capital, we started from the classical dependence network (in which needs, goals, abilities and resources are distributed among the agents) with potential partners, we introduced the analysis of what it means for an agent to be trusted and how this condition could be strategically used from it for achieving its own goals, that is, why it represents a form of power. Together with this analysis, the paper present an insight into the concept, by studying its dynamical characteristic in the attempt to understand what are the features that make trust a capital for individuals, in which sense "trust" is a capital and how this capital is built, managed and saved as well as how this capital is the result of the others' beliefs and goals. Furthermore, the difference between relational capital and social capital is addressed: we argue about the fact that individual trust capital (relational capital) and collective trust capital not only should be disentangled, but their relations are quite complicated and even conflicting.

Categories and Subject Descriptors
J.4 Social and behavioral sciences

General Terms
Human Factors, Theory.

Keywords
Trust, Dependence networks, Social Capital, Relational Capital.

1. INTRODUCTION
In multi-agent systems trust is a growing field of analysis and research and ways to calculate it have already been introduced to enhance studies on commercial partnership, strategic choice, and on coalition formation.

In order to advance such studies it is strategically important to know not only who is trusted by who and how much, but also to understand how being trusted can be used by the trustee.

Therefore, following a previous work on trust from the trustee point of view [23], in this paper we go deeper into the latter point, by showing how trust can be considered a strategic resource for agents that are trusted, by refining our model of 'trust as a capital' for individuals and by proposing an analysis of how the dynamics of trust can be taken into account for the trustee to perform strategic action.

The basic thesis from which our study move is that to be trusted, on one hand, increases the chance to be requested or accepted as a partner for exchange or cooperation and, on the other hand, improves the 'price', the contract that the agent can obtain[1]. More in details, the claims we are going to investigate in the paper are:

- it is possible *to measure the capital constituted by received trust*, by understanding its relational nature – it depends on the position in a network of relationships – and by investigating the affection it has on the '*negotiation power'* of Ag_i *(the trustee)* that cannot simply be derived from the "dependence bilateral relationships";

- to be trusted usually is an advantage, an *asset for the trustee* (agent Ag_i) which is related to the advantage of trust for the community but not in an unproblematic way: there can be a conflict between the two notion of relational capital (considered as individual capital) and social capital (considered as an asset for the community);

- received trust is a *context dependent and dynamic* capital and *has different sources*; it can be invested and requires decisions and costs to be cumulated.

[1] This point does not necessary imply and does not mean that a deceiving trustee would have surely an individual advantage. This misinterpretation is a typical point of view coming from domains like commerce and exchange in which trust is considered just as an instrument for solving questions like Prisoner Dilemma problems. In fact, we are interested to model trust in more general domains and contexts: for example, in strict cooperation in which a deceiving trustee jeopardizes its own interests and goals.

2. Trust and Relational Capital

Although there is a big interest in literature about 'social capital' and its powerful effects on the wellbeing of *both societies and individuals*, often it is not clear what is the object under analysis. How is it possible to say that "trust" is a capital? How is this capital built, managed and saved? The aim of our work is to answer these questions and to analytically study the cognitive dynamics of this object, with a particular focus on how they depend on beliefs and goals.

One of the basic issues that need to be studied is how it can happen that networks of relations can be built, which ways they develop, and how they can both influence individual behaviors and be considered as an individual capital (21).

While by social a lot of scholars mean 'collective', referring to some richness, advantage for a certain society, and to something that favors cooperation, we assume here an individualistic perspective, considering the advantages of the trusted agent, not the advantages for the community, and distinguishing between 'relational capital' (8) and the more ambiguous and extended notion of 'social capital'. Although the idea of a clear distinction of the two levels is not completely new in literature, usually relational capital is addressed in relation with meta-cognitive aspects of human capital (22) rather than being studied through an analysis of its own cognitive mechanisms.

In economic literature the term "capital" refers to a commodity itself used in the production of other goods and services: it is, then, seen as a human-made input created to permit increased production in the future. The adjective "social" is instead used to claim that a particular capital not only exists in social relationships but also consists in some kind of relationships between economical subjects. It is clear that for the capital goods metaphor to be useful, the transformative ability of social relationships to become a capital must be taken seriously. This means that *we need to find out what is the competitive advantage not simply of being part of a network, but more precisely of being trusted in that network.*

In particular, what does imply to be trusted for the trustee? As we said, the intuitive answer could be that both the *probability to be chosen* for exchange or for partnership will grow and the *negotiation power* of that agent will increase. In order to account for both these implications (that are not completely independent), it is necessary to rethink the whole theory of negotiation power based on dependence (9,10,11,12), including trust.

What we need, then, is a comprehensive theory of trust from the point of view of the trusted agent, in order to find out the elements that, once added to the theory of dependence, can explain the *individual social power in a network*, on one hand, and, the *social capital meant as a capital for the society,* in a second phase. Once a quantitative notion of the value of a given agent is formulated calculating on *how much the agent is valued by other agents in a given market for a given task*, we can say that this trust-dependent value is a real capital. It consists of all the relationships that are possible for the agent in a given market and, together with the possible relationships in other markets, it is the so-called *relational capital* of that agent. It differs from simple relationships in given networks, which are a bigger set, since it only consists of those relationships the agent has with those who not only need it but have a good attitude toward it and, therefore, who are willing to have it as a partner.

How much the agent is appreciated and requested? How many potential partners depends on Ag_i and would search for Ag_i as partner? How many partners would be at disposal for Ag_i's proposals of partnership, and what "negotiation power" would Ag_i have with them?

These relationships form a capital because (as any other capital) it is the result of investments and it is costly cumulated in order to be invested and utilized. In a certain sense it represents a strategic tool to be competitive, and, as it happens with other capitals such as the financial one, it is sometimes even more important than the good which is sold (being it either a service or a material good). For example when Ag_i decides of not keeping a promise to Ag_j, it knows that Ag_j's trust in Ag_i will decrease: is this convenient for future relationships with Ag_j? Will Ag_i need counting on Ag_j in future? Or, is this move convenient for reputation and other relationships?

For all these raising questions it is very important to study how it is possible for the agent to cumulate this capital without deteriorating or waste it: since the relational capital can make the agent win the competition even when the good it offers is not the best compared with substitutive goods offered in the market, it should be shown quantitatively what this means and what kind of dynamical relationships exist between quality of offered good and relational capital. In other, and more general, terms the relational and reputational capital of an agent is more valued than its immediate reward.

3. Measuring Relational Capital

In order to study all the important aspects of relational capital anticipated above, it is necessary to understand how it can be measured. To this aim, we must first introduce the *notion of dependence* and the related concept of *negotiation power*.

Indeed, the theory of trust and the theory of dependence are not independent from each other, not only because – as we modeled (1, 2), before deciding to actively trust somebody, to rely on it (Ag_i), one (Ag_j) has to be dependent on Ag_i: Ag_j needs an action or a resource of Ag_i (at least Ag_j has to believe so), but also because *objective* dependence relationships (9) that are the basis of adaptive social interactions, are not enough for predicting them. *Subjective* dependence is necessary (that is, the dependence relationships that the agents know or at least believe), but still not sufficient. Therefore, we need to consider two beliefs: (i) the belief of being dependent, of needing the other, (ii) the belief of the trustworthiness of the other, of the possibility of counting upon it.

Indeed, the theory of dependence includes two types of dependences:

(1) the *objective dependence*, which says who needs whom for what in a given society (although perhaps ignoring this). This dependence has already the power of establishing certain asymmetric relationships in a potential market, and it determines the actual success or failure of the reliance and transaction;

(2) the *subjective (believed) dependence*, which says who is believed to be needed by who. This dependence is what

determines relationships in a real market and settles on the negotiation power; but it might be illusory and wrong, and one might rely upon unable agents, while even being autonomously able to do as needed.

More Formally, let $Agt=\{Ag_1,..,Ag_n\}$ a set of *agents*; we can associate to each agent $Ag_i \in Agt$:

- a set of *goals* $G_i=\{g_{i_1},..,g_{i_q}\}$;

- a set of *actions* $Az_i=\{\alpha_{i_1},.., \alpha_{i_z}\}$; these are the elementary actions that Ag_i is able to perform;

- a set of plans $\Pi=\{p_{i_1},..,p_{i_s}\}$; the Ag_i's plan library: the set of rules/prescriptions for aggregating the actions; and

- a set of *resources* $R_i=\{r_{i_1},..,r_{i_m}\}$.

The achievement/maintenance of each goal needs actions/plans/resources. Then, we can define the *dependence relationship* between two agents (Ag_j and Ag_i) with respect a goal g_{jk}, as: *Obj-Dependence* (Ag_j, Ag_i, g_jk) and say that:

An agent Ag_j has an Objective Dependence Relationship with agent Ag_i with respect to a goal g_{jk} if for achieving g_{jk} are necessary actions, plans and/or resources that are owned by Ag_i and not owned (or not available) by Ag_j.

More in general, *Ag_j has an Objective Dependence Relationship with Ag_i if for achieving at least one of its goals $g_{jk} \in G_j$, are necessary actions, plans and/or resources that are owned by Ag_i and not owned (or not available) by Ag_j.*

As in (11) we can introduce the *unilateral*, *reciprocal*, *mutual* and *indirect* dependence (see Figure1). In very short and simplified terms, we can say that the difference between reciprocal and mutual is that the former is on different goals

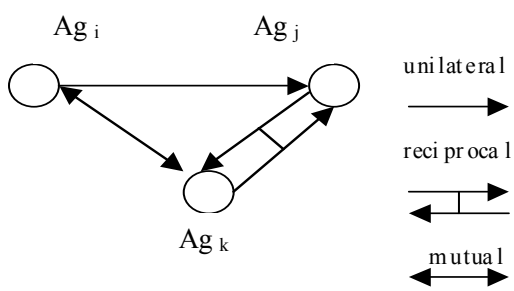

Figure 1

while the latter is on the same goal.

3.1 Dependence and Negotiation Power

The important relationship is the network of dependence *believed by each agent*. In other words, we cannot only *associate* to each agent a set of goals, actions, plans and resources, but we have to evaluate these sets as believed by each agent (the subjective point of view), also considering that they would be partial, different from each of others, sometime wrong, and so on. In more practical terms, each agent will have a different (subjective) representation of the dependence network as exemplified in Figure1. So, we introduce the Bel_kG_z that means the Goal set of Ag_z believed by Ag_k. The same for Bel_kAz_z, $Bel_k\Pi_z$, and Bel_kR_z. That is to say that the dependence relationships should be re-modulated on the basis of the agent subjective interpretation. We introduce the *Subj-Dependence*(Ag_j, Ag_i, g_{jk}) that represents the Ag_j's point of view with respect to its dependence relationships. In a first approximation each agent should correctly believe the sets it has, while it could mismatch the sets of other agents. We define *Dependence-Network(Agt,t)* the set of dependence relationships (both subjective and objective) among the agents included in *Agt* set at the time *t*. Each agent $Ag_j \in Agt$ must have at least one dependence relation with another agent in *Agt*.

Given a *Dependence-Network(Agt,t)*, we define *Objective Potential for Negotiation* of $Ag_j \in Agt$ about a goal of own g_{jk} - and call it $OPN(Ag_j, g_{jk})$- the following function.

$$OPN(Ag_j, g_{jk}) = f\left(\sum_{i=1}^{n} \frac{1}{1+p_{ki}}\right)$$

Where: f is in general a function that preserves monotonicity (we will omit this kind of functions in the next formulas); n represents the number of agents in *Agt* set that have a dependence relation with Ag_j with respect to g_{jk} (this dependence relation should be either reciprocal or mutual: in other words, there should also be an action, plan, or resource owned by Ag_j that is necessary for some goal of Ag_i); p_{ki} is the number of agents in Agt that are competitors with the Ag_j on the same actions/plans/resources owened by Ag_i (useful for g_{jk}) in a not compatible way (Ag_i is not able to satisfy at the same time all the agents).

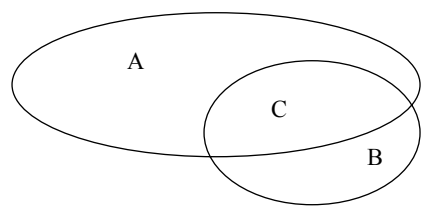

Figure 2

In Figure 2 we show the objective dependence of Ag_j: A represents the set of agents who depend from Ag_j for something (actions, plans, resources), B represents the set of agents from which Ag_j depends for achieving an own specific goal g_{jk}. The intersection between A and B (part C) is the set of agents with whom Ag_j could potentially negotiate for achieving support for g_{jk}. The greater the overlap the greater the negotiation power of Ag_j in that context. In other and more simple words, the more the agents being at the same time depending and depended upon Ag_j, the greater the negotiation power of Ag_j. However, the negotiation power of Ag_j also depends on the possible alternatives that its potential partners have: the few alternatives to Ag_j they have, the greater its negotiation power (see below).

We can define the Subjective Potential for Negotiation of $Ag_j \in Agt$ about an its own goal g_{jk} -and call it $SPN(Ag_j, g_{jk})$- the following function:

$$SPN(Ag_j, g_{jk}) = \sum_{i=1}^{n} \frac{1}{1+p_{ki}}$$

Where we have the same meanings as for the previous formula but now we make reference to the believed (by Ag_j) dependence relations (not necessarily true in the world): in particular both n (the number of direct dependences) and p (the indirect, competitive dependences) are believed.

Analogously, we can interpret Figure 2 as the set of believed relationships (by Ag_j) among the agents. In this case we have the subjective point of view. It is also possible to introduce a modulation factor (m) that takes into account the special kind of dependence: reciprocal (x=r), mutual (x=mu):

$$SPN(Ag_j, g_{jk}) = \sum_{i=1}^{n} \frac{m_x}{1+p_{ki}}$$

with $0<m_x<1$

Usually, we can say that $m_m \geq m_r$.

More in general, we can say that the Subjective Potential for Negotiation of Ag_j□Agt about the whole set of its own goals (G_j) in the Dependence-Network(Agt,t) is:

$$SPN(Ag_j, G_j) = \frac{1}{s} \sum_{k=1}^{s} \sum_{i=1}^{ns} \frac{m_x}{1+p_{ki}}$$

Where s is the number of goals of Ag_j, and *ns* is the number of other agents in the set *Agt*, that have a dependence relation with Ag_j with respect to the goal g_{jk}. p_{ki} is the number of agents in *Agt* that are competitors with the Ag_i on the same actions/plans/resources (useful for g_{jk}) in a not compatible way. In words, the global subjective potential for negotiation of an agent in a dependence network with respect of all its own goals is the sum of the believed terms above showed[2].

3.2 The Trust Role in Dependence Network

Before taking into account the trustee's point of view we would like to introduce into the dependence network also the trust relationships. In fact, *although it is important to consider dependence relationship between agents in a society, there will be not exchange in the market if there is not trust to enforce these connections*.

Considering the analogy with the Figure2, we will have now a

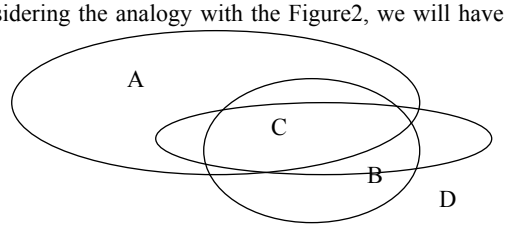

Figure 3

representation as given in Figure3 (where D includes the set of agents that *Agj* considers trustworthy for achieving g_{jk}).

We have now a new subset (showed outlined in Figure4) containing the potential agents for negotiation. The analysis of the part E, F and G will result in: part E includes agents who depend from Ag_j, who are trusted but on different tasks; part F includes agents not depending from Ag_j and trusted on different tasks; part G includes agents trusted for the achievement of the goal g_{jk} but not depending from Ag_j.

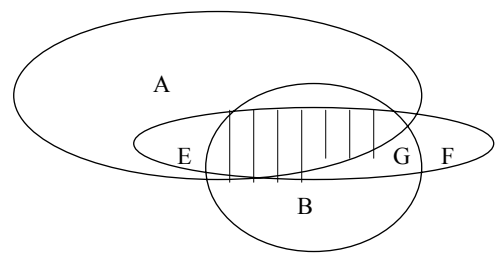

Figure 4

Not only the decision to trust presupposes a belief of being dependent, but notice that a dependence belief (*BelDep*) implies on the other side a piece of Trust (as modelled in (2, 3)). In fact *to believe to be dependent means*:

- (*BelDep-1*) to believe not to be able to perform action α and to achieve goal *g*, and
- (*BelDep-2*) to believe that Ag_i is *able* and in condition to achieve *g*, to perform α.

Notice that (*BelDep-2*) is precisely one component of Trust in our analysis: the *positive evaluation* of Ag_i as competent, able, skilled, and so on. However, the other fundamental component of trust as evaluation is lacking: reliability, trustworthiness: Ag_i really intends to do, is persistent, is loyal, is benevolent, etc. Thus he will really do what Ag_j needs.

Given the basic role played by "believed networks of dependence", established by a believed relationship of dependence based on a belief of dependence, and given that this latter is one of the basic ingredient of trust as a mental object, we can claim that this overlap between theories is the crucial issue and our aim is namely to study it deeply. So introducing also in the *Subjective Potential for Negotiation* (of $Ag_j \in Agt$ about an its own goal g_{jk}) the basic beliefs about trust (1,2) we have:

$$SPN(Ag_j, g_{jk}) = \sum_{i=1}^{n} \frac{Bel_j(DoA_j * DoW_i)}{1+p_{ki}}$$

Where: $m_m=m_r=1$; $Do(Bel_j(A_i))$ is the degree of (believed by Ag_j) ability (with respect of the goal g_k) of the agent Ag_i; $Do(Bel_j(W_i))$ is the degree of (believed by *Agj*) willingness

---

[2] An interesting problem is that an agent could be a competitor towards itself for achieving its own goals; for example:

1) Ag_j needs action τ_r both for g_s and g_t and there is only an agent in *Agt* that has τ_r but is unable to provide two times the action τ_r.

2) Ag_j needs action τ_r for g_r and τ_s for g_s and for both the actions τ_r and τ_s it depends only from Ag_i that can provide only an action.to believe not to be able to perform action α and to achieve goal *g*.

(with respect the goal g_k) of the agent Ag_i. We do not consider here the possible relations between the values of $Do(Bel_j(A_i))$ and $Do(Bel_j(W_i))$ with the p_{ki} variable.

$1 \geq Do(Bel_j(A_i)), Do(Bel_j(W_i)) \geq 0$.

Let us, now, explicitly recall what are the cognitive ingredients of trust and *reformulate them from the point of view of the trusted agent*. In order to do this, it is necessary to limit the set of trusted entities. It has in fact been argued that trust is a mental attitude, a decision and a behavior that only a cognitive agent endowed with both goals and beliefs can have, make and perform. But it has been also underlined, that the entity that is trusted is not necessarily a cognitive agent. When a cognitive agent trusts another cognitive agent, we talk about social trust. We consider that the set of actions, plans and resources owned/available by an agent can be useful for achieving a set of tasks ($\tau_1, \ldots, \tau_r$).

We take now the *point of view of the trustee agent in the dependence network*: therefore *we present a cognitive theory of trust as a capital*, which is, in our view, a good starting point to include this concept in the issue of negotiation power. *That is to say that if somebody is potentially strongly needed by other agents, but it is not trusted, its negotiation power does not improve*.

We call the *Subjective Trust Capital* of $Ag_i \in Agt$ about a specific task τ_k, the function:

$$STC(Ag_i, \tau_k) = \sum_{j=1}^{n} Bel_i(Bel_j DoA_i * Bel_j DoW_i)$$

Where n is the number of agents that need the task τ_k.

$Ag_j, Ag_i \in Agt$.

$Do(Bel_{ji}(A_{ik}))$ means the A_j's degree of belief (believed by A_i) with respect the A_i's ability about the task τ_k.

$Do(Bel_{ji}(W_{ik}))$ means the A_j's degree of belief (believed by A_i) with respect the A_i's willingness about the task τ_k.

In words, the cumulated trust capital of an agent Ag_i with respect of a specific task τ_k, is the sum (on all the agents needing that specific task in the network dependence) of the corresponding abilities and willingness believed by each dependent agent. The subjectivity consists in the fact that both the network dependence and the believed abilities and willingness are believed by (are the point of view of) the agent Ag_i.

We call *Degree of Trust* of the Agent Ag_j on the agent Ag_i about the task τ_k (DoT($Ag_j\ Ag_i\ \tau_k$)):

$$DoT(Ag_j Ag_i \tau_k) = Bel_j DoA_i * Bel_j DoW_i$$

Analogously, we can also call the self-trust of the agent Ag_i about the task τ_k we can write:

$$ST(Ag_i, \tau_k) = Bel_i(DoA_i * DoW_i)$$

From the comparison between $STC(Ag_i, \tau_k)$, $DoT(Ag_j\ Ag_i\ \tau_k)$ and $ST(Ag_i, \tau_k)$ a set of interesting actions and decisions are taken from the agents (we will see in the next paragraph).

Starting from the Trust Capital we would like evaluate the usable part of this trust capital. In this sense, we introduce the *Subjective Usable Trust Capital* of $Ag_i \in Agt$ about an its own task τ_k as:

$$SUTC(Ag_i, \tau_k) = \sum_{j=1}^{n} \frac{Bel_i(Bel_j DoA_i * Bel_j DoW_i)}{1 + p_{kj}}$$

where p_{kj} is (following the Ag_i's belief about the beliefs of Ag_j) the number of other agents in the dependence network that can achieve the same task with a trust value comparable with the one of Ag_i. We have two *comparable trust values* when the difference between them is in a range under a given threshold that could be considered meaningless with respect to the achievement of the task.

4. Trust as an asset: relational capital vs social capital

The model presented in the previous section makes it clear that trust is an advantage for the trustee. Let us now address the problem of what relationship is between this and the advantage for the community.

We would like to underline that there is no advantage to use social capital at individual level because the two interpretation of it (social and relational) are not only ambiguous but also contradictory. Social capital at individual level (relational capital) could be in conflict with the collective capital: for example, for an individual is better to monopolize trust, while for the community it is better to distribute it among the several individuals. Furthermore, the individual Ag_i could use its capital of trust, for *anti-social* purposes.

Therefore it is important to understand this difference and try to model the interaction between the two forms of capital. For this purpose it is useful to clarify some concepts which are also mixed up in literature. Let us start with the concept of "generalized trust".

In the common use, "generalized trust" is used both for indicate what characterize the point of view of an individual trustor towards a given community of agents, and to refer to the cognitive and relational notion which relate to a "diffused trust" in that community.

"*Generalized trust* of Ag_j towards a set (community) *Agt*" means that Ag_j has a trust attitude towards everybody in *Agt*, without discriminating among *Agt* members, which would mean to allocate trust only in some of them and not in the others.

$$\forall Ag_i : Ag_i \in Agt$$

$$Bel_j DoA_i * Bel_j DoW_i$$

Of course, this notion, which is logical and non quantitative, can be transformed in a quantitative notion and fuzzy formula: for instance, it is possible to quantify by saying that Ag_j trusts *more or less* the totality or the (great) *majority* of Ag_i), or that Ag_j trust more people than what he distrusts. Anyway, this notion implies an individual point of view and does not need any "diffused trust" in C. It is just a unilateral piece of a trust network.

Of course, multilateral networks like this can create a correct case of *diffuse trust*, where "everybody trusts everybody in

Agt"[3].

$$\forall Ag_i, Ag_j : Ag_i, Ag_j \in Agt$$

$$Bel_j DoA_i * Bel_j DoW_i$$

and

$$Bel_i DoA_j * Bel_i DoW_j$$

There can be also the case of an "over-generalized trust", where Ag_j trusts not only the members of the community Ag_t, but also of a larger set – the one which include Ag_t – which can be the broad community of communities. In this case it is possible to say that Ag_j trust everybody.

This distinction between unilateral and multilateral generalized trust, is very important for our view of the trustee perspective and, in particular, for the distinction between the relational capital of each single agent and the social capital of the community. In fact, although the two phenomena are related to each other, their distinction becomes even clearer if we consider that there might be a conflict between them: the selfish interest of the trustee is to monopolize trust, to be more trusted than the others in the community, since this will increase her negotiation power.

Let Ag_i be the trustee, she would tend to maximize the number of agents that trust her in the community. What she will tend to can be expressed, therefore, by the transformation of the notion of generalized trust from the trustee point of view:

$$\forall Ag_j : Ag_j \in Agt$$

$$Bel_j DoA_i * Bel_j DoW_i$$

At the same time, by tending to minimize the number of other agents in the community that are trusted, she would be interested in not having the diffuse trust in that community.

In other words, it is convenient for Ag_i to be trusted by a lot of agents but it is not convenient for Ag_i that also the others have a lot of agents they are trusted by and, then, that there is a diffuse trust in the community. The combination of this two elements – Ag_i can be trusted by everybody; they trust only Ag_i – provides Ag_i the maximum possible negotiation power in the community. So, not always and not necessarily a diffuse trust, a trust atmosphere is a good condition for all the agents in the community.

There are possible advantages and possible disadvantages that must be investigated deeply, as well as other issues related to social capital and the notion of diffused trust. In fact, although we are not going to address them here, it is important to remark that what is also interesting to model in the diffused trust, is not just the "generalization" from a specific agent to a set/class/group. While generalization can maintain the specific dimension of the task[4], by talking of diffused trust an abstraction is made from the task: the task becomes *unspecified*. In this terms, the relevant aspect of trust becomes the "reliability", the fact that everybody will be good-willing or better adoptive (usually meant on a "moral" base), together with other basic constituents of trust – to feel "safe", without hostility, without danger from other agents' side. We remind the analysis of these very interesting issues to future work.

[3] Note that while diffuse trust implies several generalized trust, it is not true the other way around.

[4] One might for example generalize that not only Ag_j but all the group or class she is member of is trustworthy as for action *A*; which means that all the agents are both_competent/skilled and reliable as for doing *A*, and individual instantiate this trust case by case. This is the kind of trust we have towards nurses, policemen, etc.

5. Dynamics of Relational Capital

So far, we have seen in what sense received trust is a capital: we have addressed the problems of how relational capital can be measured in a network, what is the advantage for individual to cumulate it and how this advantage is different and sometimes in conflict with the advantage for the community). What we need to study, now, is the dynamical aspect of it: namely, we are going to present both how this capital can increase, decrease and be transferred, and what kind of strategies can be performed by agents that want to manipulate it.

Let us stress the point that what has not been considered enough in organization theory is the fact that the *relational capital* is peculiar in its being crucially based on beliefs: again, what makes relationships become a capital is not simply the structure of the networks (who "sees" whom and how clearly) but the *levels of trust which characterizes the links in the networks* (who trusts whom and how much). Since trust is based on beliefs – including, as we said, also the believed dependence (who needs whom) – it should be clear that *relational capital is a form of capital, which can be manipulated by manipulating beliefs*.

Thanks to a structural theory of what kind of beliefs are involved it is possible not only to answer some very important questions about agents' power in network but also to understand the dynamical aspects of relational capital. In addition, it is possible to study what a difference between trustee's beliefs and others' expectations on her implies, in terms of both reactive and strategic actions performed by the trustee.

5.1 Increasing, Decreasing, and Transferring

For what concerns the dynamic aspects of this kind of capital, it is possible to make hypotheses on how it can increase or how it can be wasted, depending on how each of the basic beliefs involved in trust are manipulated.

First, let us consider what kind of strategies can be performed to enforce the other's dependence beliefs and in particular his beliefs about agent's competence.

i) Ag_i can make the other agent dependent on him by making the other lacking some resource or skill (or at least inducing the other to *believe* so).

ii) Ag_i can make the other agent dependent on him by activating or inducing in it a given goal (need, desire) on which the other is not autonomous (13) (or it believes so).

iii) Since dependence beliefs are strictly related with the possibility of the others to see the agent in the network and to know her ability in performing useful tasks, the goal of the agent who wants to improve her own relational capital will be to *signaling* her presence and her skills (14,15,16). While to show her presence she might have to shift her position (either

physically or figuratively like, for instance, changing her field), to communicate her skills she might have to hold and show something that can be used as a signal (such as certificate, social status etc.). This implies, in her plan of actions, several and necessary sub-goals to make a signal. These sub-goals are costly to be reached and the cost the agent has to pay to reach them can be taken as the evidence for the signals to be credible (of course without considering cheating in building signals). It is important to underline that using these signals often implies the participation of a third subject in the process of building trust as a capital: a third part which must be trusted (2). We would say the more the third part is trusted in the society, the more expensive will be for the agent to acquire signals to show, and the more these signals will work in increasing the agent's relational capital. We will see later how this is related with the process of transferring trust from an agent to another (building reputation).

Obviously also Ag_i's *previous performances* are 'signals' of trustworthiness. And this information is also provided by the circulating *reputation* of Ag_i (17, 18). In formal terms, we can say that Ag_i has to work for increasing:

$Do(Bel_j(Ag_i))$ and consequently $Do(Bel_{ji}(Ag_i))$.

iv) Alternatively, Ag_i could work for reducing the believed (by Ag_j) value of ability of each of the possible competitors of Ag_i (in number of p_{kj}) on that specific task τ_k.

Let us now consider how willingness beliefs can be manipulated. In order to do so, consider the particular strategy performed to gain the other's good attitude through gifts (19). It is true that the expected reaction will be of reciprocation, but this is not enough. While giving a gift the agent knows that the other will be more inclined to reciprocate, but she also knows that her action can be interpreted as a sign of the good willingness she has: since she has given something without being asked, the other is driven to believe that the agent will not cheat on him. Then, the real strategy can be played on trust, sometimes totally and sometimes only partially – this will basically depend on specific roles of agents involved. Again in formal terms, we can say that Ag_i has to work for increasing:

$Do(Bel_j(W_i))$ and as a consequence $Do(Bel_{ji}(W_i))$. Alternatively, it could work for reducing the believed (by Ag_j) value of willingness of each of the possible competitors of Ag_i (in number of p_{kj}) on that specific task τ_k.

An important consideration we have to do is that a dependence network is mainly based on the set of actions, plans and resources owned by the agents and necessary for achieving the agents' goals (we considered a set of tasks each agent is able to achieve). The interesting thing is that the dependence network is modified by the dynamics of the agents' goals, from their variations, from the emergency of new ones, from the disappearance of old ones, from the increasing request of a subset of them, and so on (20). On this basis changing the role of each agent in the dependence network, it changes in fact the trust capital of the agents.

Relational capital can also circulated inside a given society. If somebody has a good reputation and is trusted by somebody else, she can be sure this reputation will pass and transfer to other actors – and this is always considered in marketing strategies of making voice circulate. What is not clear yet is how these phenomena work. But when trust on an agent circulates, it is strategically important for the agent to know very well how this happens and in which ways (not only figurate) trust begin to expand and keep on doing it. In fact, not all the ways are the same: it is possible that being trusted by a particular agent can mean that she just has one more agent in her relational capital, but gaining the trust of another agent can be very useful to her and exponentially increase her capital thanks to the strategic role or position of this other agent. That said, it should be clear the importance of understanding if and how much an agent is able to manage this potentiality of her capital. Basically, here also, the role of agents involved play a crucial part: for this reason it is necessary for agent to know the multiplicative factor represented by the recognized and trusted evaluator in the society. It is not necessarily true, in fact, that when somebody trusts somebody else and this latter trusts a third one, the first one will trust the third one: the crucial question is "which role the first recognizes to the second". If the second one is trusted as an evaluator by the first one, than she can trust the third one for specific goals. Usually how well these transitive process work depends on what kind of broadcasting and how many links the valuator has as well as on how much she is trusted in each of those links, so, basically, it recursively depends on the valuator's relational capital.

5.2 Strategic Behavior of the Trustee

Until now we just have considered trust as something quantitatively changeable, but we did not talk about subjective difference in the way trust is perceived by the two parts of the relationship. Nevertheless, to be realistic, we must take into account the fact that *there is often a difference between how the others actually trust an agent and what that agent believes about that*; and also between this (believed trust of others on her) and the level of trustworthiness that agent perceives in herself. Since being able is not necessarily the cause of trust: it can be the case of a diffuse atmosphere that makes the others to trust the agent although the agent has not all the characteristics to be trusted. In fact, these subjective aspects of trust are fundamental in the process of managing this capital, since it can be possible that there is the capital but the agent does not know to have it. Can be possible to use the relational capital even if who uses it is not aware of having it?

At the base of the possible discrepancy in subjective valuation of trustworthiness there is the perception of how much an agent feels herself trustworthy in a given task and the valuation that agent makes about how much the others trust her for that task. In addition, this perception can change and become closer to the objective level while the task is performed: the agent can either find out of being more or less trustworthy than what he believed, or realize that the others' perception was wrong (either positively or negatively). All this factors must be taken into account and studied together with the different components of trust, in order to build hypotheses on strategic actions that the agent will perform to cope with her relational capital.

Then, we must consider what can be implied by these *discrepancies* in terms of strategic actions: how they can be individuated and valued? How the trusted agent will react when aware of them? She can either try to acquire competences in order to reduce the gap between others' valuation and her own one, or exploiting the existence of this discrepancy, taking

advantage economically of the reputation over her capability and counting on the others' scarce ability of monitoring and testing her real skills.

6. Conclusions

As we said, individual trust capital (relational capital) and collective trust capital not only should be disentangled, but their relations are quite complicated and even conflicting. In fact, since the individual is in competition with the other individuals, he has a better position when trust is not uniformly distributed (everybody trusts everybody), but when he enjoys some form of concentration of trust (an oligopoly position in the trust network); on the contrary the collective social capital could do better with a generalized trust among the members of the collectivity.

7. ACKNOWLEDGMENTS

This paper has been founded by the European Project ***MindRACES*** (*from Reactive to Anticipatory Cognitive Embodied Systems*): Contract Number: FP6-511931.

8. REFERENCES

[1] Castelfranchi C., Falcone R., Principles of trust for MAS: cognitive anatomy, social importance, and quantification, *Proceedings of the International Conference of Multi-Agent Systems (ICMAS'98*), pp. 72-79, Paris, July, 1998.

[2] Falcone R., Castelfranchi C., (2001). Social Trust: A Cognitive Approach, in *Trust and Deception in Virtual Societies* by Castelfranchi C. and Yao-Hua Tan (eds), Kluwer Academic Publishers, pp. 55-90.

[3] Falcone R., Castelfranchi C. (2001), The socio-cognitive dynamics of trust: does trust create trust? In *Trust in Cyber-societies: Integrating the Human and Artificial Perspectives* R. Falcone, M. Singh, and Y. Tan (Eds.), LNAI 2246 Springer. pp. 55- 72.

[4] Bourdieu, P. 1983: Forms of capital. In: Richards, J. C. ed. Handbook of theory and research for the sociology of education, New York, Greenwood Press.

[5] Coleman, J. C. 1988: Social capital in the creation of human capital. American Journal of Sociology 94: S95-S120.

[6] Putnam, R. D. 1993: Making democracy work. Civic traditions in modern Italy. Princeton NJ, Princeton University Press.

[7] Putnam, R. D. 2000: Bowling alone. The collapse and revival of American community. New York, Simon and Schuster.

[8] Granovetter, M. (1973). The strength of weak ties. American Journal of Sociology, 78, 1360-1380.

[9] Castelfranchi C., and Conte R., The Dynamics of Dependence Networks and Power Relations in Open Multi-Agent Systems. In Proc. COOP'96 – Second International Conference on the Design of Cooperative Systems, Juan-les-Pins, France, June, 12-14. INRIA Sophia-Antipolis, 1996. P.125-137.

[10] Sichman, J, R. Conte, C. Castelfranchi, Y. Demazeau. A social reasoning mechanism based on dependence networks. In *Proceedings of the 11th ECAI*, 1994.

[11] Castelfranchi, C., Miceli, M. e Cesta, A., Dependence relations among autonomous agents. In E. Werner, Y. Demazeau (Eds), *Decentralized A. I.* - 3 , pp. 215-227, North Holland, Amsterdam, 1992.

[12] Conte, R. e Castelfranchi, C. (1996) Simulating multi-agent interdependencies. A two-way approach to the micro-macro link. In U. Mueller & K. Troitzsch (eds) *Microsimulation and the social science*. Berlin, Springer Verlag, Lecture Notes in Economics.

[13] Castelfranchi, C. Falcone R. (2003), From Automaticity to Autonomy: The Frontier of Artificial Agents, in Hexmoor H, Castelfranchi, C., and Falcone R. (Eds), Agent Autonomy, Kluwer Publisher, pp.103-136.

[14] Schelling, T., *The Strategy of Conflict*. Cambridge, Harvard University Press, 1960.

[15] Spece, M. 1973 Job market signaling. *Quarterly Journal of Economics, 87*, 296- 332.

[16] R. Bliege Bird & E. Alden Smith "Signaling Theory, Strategic Interaction, and Symbolic Capital", *Current Antropology*, vol. 46, n.2. April 2005.

[17] R. Conte and M. Paolucci, Reputation in Artificial Societies. Social Beliefs for Social Order. Kluwer 2002.

[18] A. Jøsang and R. Ismail. *The Beta Reputation System*. In the proceedings of the 15th Bled Conference on Electronic Commerce, Bled, Slovenia, 17-19 June 2002.

[19] Cialdini, R. B. 1990: Influence et manipulation, Paris, First.

[20] Pollack, M., Plans as complex mental attitudes in Cohen, P.R., Morgan, J. and Pollack, M.E. (eds), *Intentions in Communication*, MIT press, USA, pp. 77-103, 1990.

[21] Nan Lin 2001: Building a Network Theory of Social Capital, In Nan Lin, K. Cook, and R. Burt (Eds), *Social Capital: Theory and Research*, Somerset NJ, Aldine Transaction.

[22] Glaeser, E.L., Laibson D., Scheinkman J.A., and Soutter C.L., 2000: Measuring Trust, *The Quarterly Journal of Economics*, 115: 811-46.

[23] Castelfranchi C., Falcone R. and Marzo F. (2006), Being Trusted in a Social Network: Trust as Relational Capital, *Lecture Notes Computer Science*, 3986: 19-32.

Applying Trust to a Secure Message Passing Domain

David DeAngelis and K. Suzanne Barber

The University of Texas at Austin
Laboratory for Intelligent Processes and Systems
1 University Station C5000, Austin, TX 78712-0240
{dave, barber}@lips.utexas.edu

Abstract

Technologies developed in the trust research community can be applied to solve real problems. These technologies include multi-dimensional trust and trust model sharing. This paper presents a case study that was performed to demonstrate the application and performance of these trust-based tools. The case study problem domain presents a constraint satisfaction and optimization problem, and we show that multi-dimensional trust can be used to accurately model domain phenomena, maximizing goal achievement.

Introduction

This work presents a case study in which two trust based technologies are applied to a secure message passing domain. These technologies leverage the inherent advantages of agents, including agents' ability to construct and share trust-based models of their operating environment, to solve a problem while maximizing social welfare.

According to Marsh, there are two types of trust: general trust and situational trust (Marsh 1994). General trust involves evaluating a partner agent's trustworthiness according to its performance during previous interactions. Situational trust is an evaluation based on an agent's behavior in similar situations. Both general trust and situational trust are needed for developing a secure system. General trust is most useful when a system has a small, fixed number of users and a security failure is not catastrophic, since previous interactions, both positive and negative, must be encountered for a general trust based model to be effective. Situational trust schemes have the added benefit of being able to prevent security failures by recognizing similar scenarios or situations. These situations may include any number of direct pertinent information about the interacting agents, or even simple contextual information. The notion of soft security implies that agents incorporate local trust models to identify malicious operating environments, or hosts (Barber and Kim 2002). Mobile agents must move among these hosts whose trustworthiness cannot be verified. Degree of trustworthiness serves as a decision criterion for whether a mobile agent should relocate to the host whose trust is being modeled. A trust model can be built by comparing the number of times a host performs innocently versus maliciously. For example, trust has been assessed by measuring the number of positive and negative "experiences" an agent undergoes with the host (Jonker and Treur 1999).

Applied Trust Technologies

Two agent technologies, multi-dimensional trust and trust model sharing, are discussed in previous work by DeAngelis, Barber, Fullam and Gujral. The application and performance of these technologies is demonstrated in a case study based on a more robust communication domain.

In a multi-agent system, a goal driven agent (human or software) is motivated to trust others when its resources are too limited to permit goal achievement in isolation. These goals may have multiple requirements (quality, completion timeliness, costs) influencing the reward received from goal achievement. To maximize the reward it receives from achieving a goal, an agent must consider the trustworthiness of potential partners relative to multiple dimensions accounting for multiple goal requirements (which determine rewards) and multiple partner constraints (which estimate partner behavior). Multi-dimensional trust endows agents with the ability to assert how much they should trust multiple facets of a potential partner's behavior – the availability of the partner to deliver quality and on time solutions within cost – in the context of multiple goal requirements. The partner selection algorithm introduced in (Gujral et al. 2006) allows an agent to use multiple dimensions (goal requirements and estimated partner behavior) to estimate future goal achievement.

The concept of multi-dimensional trust is a flexible tool that can be used in any situation exhibiting goal driven behavior where the goal has multiple components. The experimental results presented in (Gujral et al. 2006) show significantly higher goal achievement when using the complete partner selection strategy over a single dimensional strategy in cases with multidimensional rewards. Another benefit is that using multi-dimensional trust models allows for the uninterrupted pursuit of rapidly changing goals. If a new goal is introduced to an agent system but the goal can be decomposed into the same basic elements as a previous goal, no new trust models need to be learned.

The value of modeling host behavior using mobile agents has been shown for intrusion detection scenarios (DeAngelis, Fullam, and Barber 2005). A group of agents

interact with a collection of hosts, and the agents model each host based on reports of malicious host behavior from other agents. Instead of monitoring the steady state system performance once models are established, the focus is on the transient behavior during model building because effective models must be constructed quickly to minimize the damage caused by malicious hosts. Using this transient analysis, system designers can manage network connectivity based on initial connectivity and minimum connectivity thresholds. A system of agents that is fully connected can share interaction experience to identify and avoid a malicious intruder. Degradation of the connectivity network has little impact on the algorithm effectiveness, and only a fraction of the total connections are needed to quickly converge to accurate models of hosts. This security scheme scales well; when more agents participate, more interaction experience is shared amongst the group, allowing the models to evolve more quickly. The extra communication overhead required is also not typically a limiting factor. Only 20% of the total number of communication links need to be present to achieve nearly 75% of the performance of a fully connected system. The only significant overhead lies in the computation of each trust model.

Domain Problem

The secure message passing problem domain is inspired by military and commercial communications systems. In this problem, a fixed number of decision makers must send messages according to a varied set of criteria. These messages have certain properties; the utility of sending a message depends on the characteristics of the message transmission. This is a highly parallel and dynamic resource constrained domain.

Multiple decision makers are assigned messages to be sent. The decision makers must determine the best method for transmitting the message. One simulation cycle is complete when each decision making agent sends exactly one message, successfully or not. There are a fixed number and type of modes of transmission, and each message has properties making it well suited for only a subset of the available modes. One such property is a *security* requirement for each message. If it is deemed that a sensitive message was sent in an insecure manner, a severe penalty is incurred. Other properties include the *timeliness* and *quality* requirements of a message, meaning the reward, or utility, associated with a successful transmission has a decay factor based on the amount of time taken. Adding uncertainty to the domain is the concept of availability of the transmission modes. The modes are a valuable resource with associated costs for utilization, and they may be unavailable depending on the state of the decision maker. The decision maker is always in one of four operating states, and for each state, the set of available transmission modes is a subset of all the modes in the system. The message *type* also affects the set of available modes. The ideas of message *type* and decision maker state form hard constraints that must be satisfied for a successful transmission. The scenario is designed to minimize the occurrence of a combination of decision maker state and message *type* that yields zero available modes. Therefore, it is nearly always possible to satisfy the hard constraints, though the entire transmission may not always yield positive rewards.

Trust is applied during the decision algorithm for choosing which transmission mode to utilize. The behavior and availability of the transmission modes are modeled internally by each decision maker, and these models evolve as the decision making agent gains more experience. This experience can be shared among other decision making agents to allow for faster convergence to an ideal behavioral model. This sharing technique is very similar to the communication of trust models presented in (DeAngelis, Fullam, and Barber 2005). These many facets of the transmission modes' behavior are combined into a multi-dimensional trust assessment for evaluating the suitability for transmitting a particular message. Like the reward functions in (Gujral et al. 2006), the goal (the reward associated with a successful message transmission) is multi-dimensional, even though it is based on the properties of the message and is not a predetermined fixed function. Because the messages are all unique, this simulation demonstrates that multi-dimensional trust has the flexibility to accommodate rapidly changing reward functions with no penalty. This dynamic message passing domain shows an appropriate application of agents and demonstrates the effectiveness of trust based techniques on a more complex problem.

Experiment Parameters

All simulations were run with ten simultaneous decision making agents unless specified (Figure 4) and the same pool of six transmission modes. These modes are secure satellite, media satellite, data cable, secure telephone, unsecured telephone, and traditional mail service. The values chosen do not represent any one real system, but they are intended to be reasonable approximations to the choices a real decision maker may have available.

Message Generation

Messages are generated according to the properties *type, security, size, quality, and deadline* as described in Table 1. Currently each property is independent, meaning a message of *type* video is not more likely to have a large *size* than a plain text message. The message type is one of four values with the probabilities shown, and is used to generate mode availability constraints. 15% of the messages generated require secure transmission. The decision making agent will incur a severe penalty if the security requirements are not met. This penalty is normally distributed with a mean of 2000 reward units and a standard deviation of 500 units. A message is designated as having short, medium, or long length and the size in MB is generated according to the posted normal distributions. The

Table 1 Message generation

Message Type	Security	Size (MB)	Quality	Deadline
35% Text 30% Web 25% Audio 10% Video	85% unsecured 15% secure Penalty ~N(2000,500)	40% Short ~N(3,3) 40% Medium ~N(50,25) 20% Long ~N(1000,200)	~N(50,30)	33% $\frac{1}{t}$ 33% $\frac{1}{t^{1/5}}$ 34% $\frac{1}{t^{1/20}}$

quality requirement is a minimum threshold of fidelity that must be maintained during transmission to receive any reward. The deadline of a message is a decay function. When a mode transmits a message, it replies with the amount of time t in domain time units (typically a fraction of one cycle) necessary for sending the message. The deadline field is a multiplier on the reward payoff that depends on the value of t. All values utilizing a normal distribution are bounded by a value of zero.

Table 2 Transmission mode capabilities

	Security	Quality	Cost (size)	Throughput
Secure Satellite	99%	~N(0.99,10)	100 /MB	1000 MB/Cycle
Media Satellite	85%	~N(0.9,10)	60 /MB	2000 MB/Cycle
Data Cable	75%	~N(0.9,10)	30 /MB	300 MB/Cycle
Secure Telephone	95%	~N(0.9,10)	65 /MB	30 MB/Cycle
Unsecured Telephone	65%	~N(0.8,10)	35 /MB	30 MB/Cycle
Mail	5%	~N(0.7,10)	10 /MB	5 MB/Cycle

Transmission Mode Properties

The six transmission modes are assigned behavior traits for *security*, *quality*, *cost,* and *throughput* as shown in Table 2. *Security* is the percentage of message transmissions that will be sent in a secure manner. A higher security value means the mode is less likely to compromise the secrecy, authenticity, and integrity of a message. *Quality* is the fidelity at which a message is sent. This is represented as a normal distribution (bounded by 0). Both *security* and *quality* are unknown to the decision making agent. These two mode behavior traits create a portion of the decision makers' multi-dimensional trust models of the modes. *Cost* and *throughput* are public knowledge; therefore each decision maker is aware of these values. *Cost* is a function of message *size*. *Cost* is the price in reward units for sending a message per MB. This includes all infrastructure and transmission costs, but not any applicable security penalty. Throughput is the speed at which a message is transmitted, measured in MB per simulation cycle. Each mode can transmit only 1 message at a time, and messages are queued according to the rank of the decision provider.

Transmission Mode Availability

Table 3 shows the hard constraints imposed by the state of the message sender and the message *type*. Each decision maker is always in one of four states with the probabilities shown. Most often, a decision maker is in the *full access* state, meaning all modes are available. Other states including *Office*, *Field*, and *Travel* are more restrictive. The message *type* also affects the availability of a mode. It can be argued that all message types are simply data that can be stored in a digital form and transmitted using all modes, but this simulation focuses on the most common usage of each type of transmission mode.

Table 3 Mode availability constraints

	Message Sending States				Message Type			
	Full 75%	Office 15%	Field 8%	Travel 2%	Text 35%	Web 30%	Audio 25%	Video 10%
Secure Satellite	+		+		+	+	+	+
Media Satellite	+				+	+	+	+
Data Cable	+	+			+	+	+	+
Secure Telephone	+	+	+				+	
Unsecured Telephone	+			+			+	
Mail	+	+		+	+			

Transmission mode selection algorithm

The decision maker's goal is to choose the best available transmission mode in its current environment. This environment includes the current message, the state of the decision maker, and the state of the modes. Throughout the entire simulation, each mode maintains a queue of messages to send. The size of this queue depends on the *throughput* of the mode and the number of times the mode is chosen. Three algorithms are presented for choosing the best transmission mode. The first algorithm, called *Reward Modeling*, uses multi-dimensional trust and the communication of trust models. Two other algorithms are provided for comparison. The *Cost Minimizing* algorithm always chooses the mode with the lowest transaction cost for the given message once all hard constraints (Table 3) are satisfied. The *Random* algorithm chooses a random transmission mode once all hard constraints are satisfied. Figure 1 is a flow chart describing the simulation and decision process for one decision making agent.

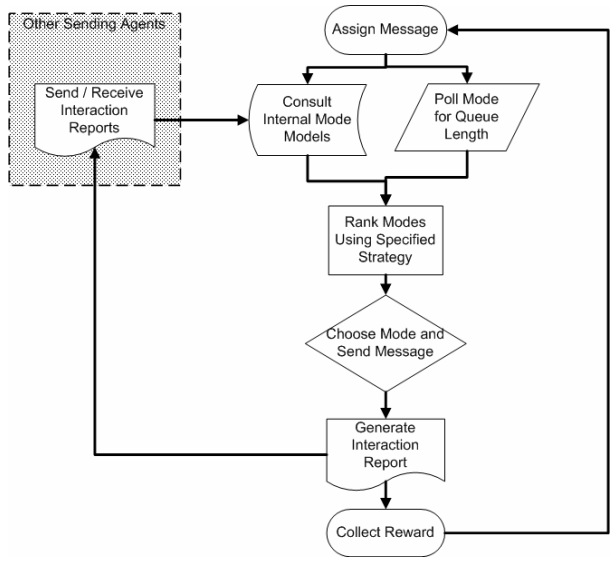

Figure 1 Mode selection algorithm

The simulator assigns a message, and the decision making agent then gathers all available resources by polling each mode for queue length and consulting internal models of mode behavior. Then all modes that meet the hard constraints are ranked according to one of three methods. These methods are *Reward Modeling*, *Cost Minimizing*, and *Random*. Once the modes are ranked, the top ranked mode is chosen and the message is sent. The mode generates an interaction report that outlines all properties of the transmission including security, quality, time, and the identification of the mode. Rewards are then collected by the decision making agent. This interaction report is then used in the *Reward Modeling* algorithm, and the process repeats.

Reward Modeling Decision Algorithm

The *Reward Modeling* decision algorithm ranks all available modes according to the expected reward for a successful transmission. The expected reward is calculated from the models of mode behavior and the simulator reward equations.

$$P_{\overline{Q}} = -C_T$$
$$P_{Q\overline{S}} = 100 \times T \times S \times -C_T - Z$$
$$P'_S = 100 \times T \times S \times -C_T$$
$$P' = 50 \times T \times S \times -C_T$$

Figure 2 Simulator reward equations

The reward when the quality is not met, P_Q, is simply the negative transaction cost, $-C_T$. No security penalty is assumed in this case because the fidelity was too poor for successful message transmission. The reward when the quality is met, but the message security is not met, P_{QS}, includes the timeliness factor T (from Deadline, Table 1), message size *S*, and security penalty *Z*. When the message must be secure, and both the security and quality of the transmission are met, the reward is P'_S. This reward has a factor of 100 because successfully transmitting a secure message yields higher rewards. The final reward function, P', is applicable when the message does not need to be secure, and the quality is met.

To maximize the expected reward, each decision making agent maintains a model of every mode. These models are used to provide estimates of the transmission security and quality. Mode *throughput* and *cost* are fixed numbers that are known to each decision maker. *Security* and q*uality* are established as probability distributions. These distributions are kept secret from the decision makers. Internal models of these values are established by averaging previous interaction experience. These models are initialized optimistically to encourage exploration among the modes. As shown in Figure 1, another source of interaction experience is other decision makers. Using the scheme of fully connected modeling, agents send their interaction reports to other agents, effectively sharing their experience. This allows the models of mode security and quality to converge to the true values more quickly.

Experimental Results

The first experiment compares the different mode selection strategies. Average cumulative reward (*ACR*) is an accurate measure of algorithm performance because it shows the cumulative sum of the average number of points earned per decision maker in each simulation cycle. This number is analogous to a total system bank balance, scaled by the number of decision makers in the system. This experiment was run for 100 simulation cycles using ten decision makers, and 100 of these games were then averaged. It is clear from Figure 3 that the *Reward Modeling* algorithm significantly outperforms the others. Also interesting is that the *Random* strategy outperforms the *Cost Minimizing* strategy. The *Random* strategy is provided as a base line assessment, and *Cost Minimizing* is chosen to simulate the behavior of hurried human operator by minimizing the expected cost while meeting the hard constraints. It performs poorly in this case because the mode properties initialization (Table 2). The least costly mode of transmission is *mail*, but *mail* has many undesirable properties such as a lack of security, low transmission quality, and slow throughput. The reward points saved by choosing *mail* most often is not enough to offset the missed opportunity cost provided by the other transmission modes.

To evaluate the effectiveness of sharing trust models, we investigated the average reward earned per cycle for two different strategies: agents sharing trust models and each agent relying solely on its own models. The average reward earned per cycle is essentially a first derivative of the *ACR*. Our experiment shows that sharing interaction reports does not significantly affect the overall system performance when using the *Reward Modeling* algorithm. The models of mode quality and security quickly converged to the true behavior of the system, and that there was no learning benefit gained from more experience. As shown in (DeAngelis, Fullam, and Barber 2005), sharing

trust models can sometimes greatly benefit all those involved. Here, sharing trust models makes no significant difference. This is because the mode models are initialized with conservative values close to the true behavior of the system, and the response is under-damped. The effects of a single security or quality failure are not strong enough to prevent future interactions with each mode.

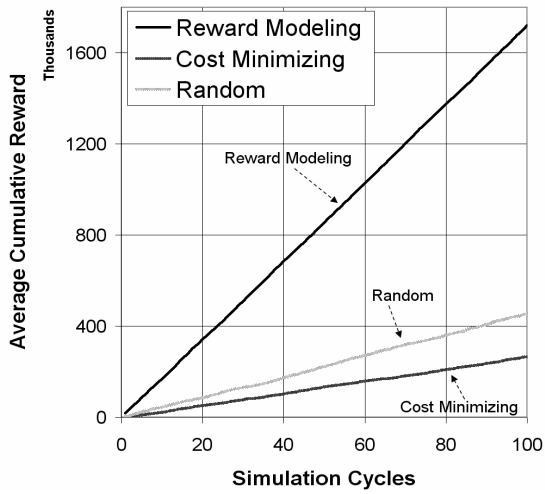

Figure 3 Mode selection algorithm performance

The scalability of the transmission mode selection algorithm is shown in Figure 4. This experiment varies the number of decision makers in the system, and therefore the number of messages sent, while the number and type of transmission modes remains the same. The vertical axis in Figure 4 is the total average cumulative reward (*ACR*) at exactly fifty simulation cycles. This indicates the scalability of each algorithm as the system grows more crowded, yet the resources are fixed. An operating point of fifty cycles was chosen because it is a reasonable assumption that the mode models are fixed near their final values after fifty cycles.

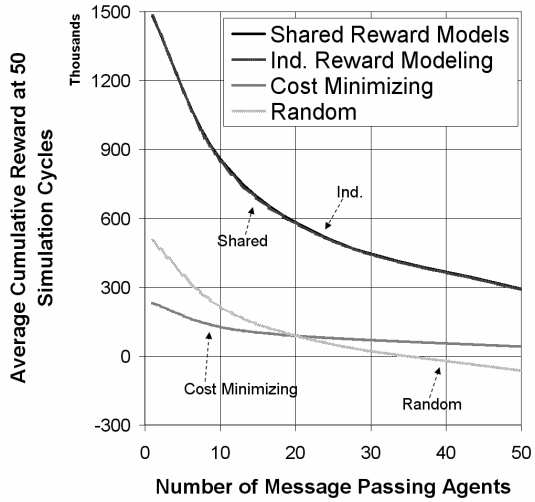

Figure 4 Algorithm scalability

The *Reward Modeling* algorithm again performed the strongest, and sharing interaction reports was insignificant. This is expected because the mode models converge very quickly. Especially interesting is the behavior of the *Random* and *Cost Minimizing* strategies. As explained above, *Random* outperforms *Cost Minimizing* for smaller systems (<20 agents). As the system grows, more collisions are incurred using the *Random* strategy. When a single mode is chosen many times in a single cycle, it develops a long queue of messages. This causes a longer delay in transmission, and a smaller time factor *T* as a coefficient of the reward. Because the *Cost Minimization* strategy queries modes for an estimated cost, collisions are minimized. The monotonic downward trend for each strategy indicates that collisions do happen and smaller rewards are the result. Experimental evidence supports that the *Reward Modeling* strategy outperforms the others even as the number of agents grows much larger.

Conclusions

This case study drawn from the secure message passing domain applies model sharing and multi-dimensional trust techniques. The intent is not merely to show that considering more information leads to higher performance, but to demonstrate how trust-based techniques can be applied and evaluated in a more robust artificial domain. Due to quick model convergence, the sharing mode interaction experience has little effect. Building interaction experience by sharing rewards models does not significantly affect the average number of points earned by a decision maker in each cycle. Therefore, impaired communication ability among the message sending agents would have no detrimental effect on overall system performance. Using multi-dimensional models, however, has a substantial benefit. The concept of multi-dimensional trust is used to accurately model message transmission mode behavior. When the message sending agent uses these models to choose the optimum transmission mode, the reward earned is quadrupled compared to random selection among valid modes.

References

Barber, K.S.; and Kim, J. 2002. Soft Security: Isolating Unreliable Agents. In *The Workshop on Deception, Fraud, and Trust in Agent Societies* at Autonomous Agents, 8-17.

DeAngelis, D.; Fullam, K.; and Barber, K.S. 2005. Effects of Communication Disruption in Mobile Agent Trust Assessments for Distributed Security. In *The Workshop on Trust in Agent Societies* at *The Fourth International Joint Conference on Autonomous Agents and Multiagent Systems*, 27-37.

Gujral, N.; DeAngelis, D.; Fullam, K.; and Barber, K.S. 2006. Modeling Multi-Dimensional Trust. In *The Workshop on Trust in Agent Societies* at *The Fifth International Joint Conference on Autonomous Agents and Multiagent Systems*, 35-41.

Jonker, C.M.; and Treur, J. 1999. Formal Analysis of Models for the Dynamics of Trust Based on Experiences. In *The 9th European Work-shop on Modeling Autonomous Agents in a Multi-Agent World*: Multi-Agent System Engineering, 221-231.

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

Beyond Trust. Fuzzy Contextual Corrective Filters for Reliability Assessment in MAS. Application to the ART Testbed

Esteve del Acebo and **Nicolás Hormazábal** and **Peplluís de la Rosa**
Agents Research Lab
Institut d'Informàtica i Aplicacions
Universitat de Girona. Spain

Abstract

Trust modelling is widely recognized as an aspect of essential importance in the construction of agents and multi agent systems (MAS). As a consequence, several trust formalisms have been developed over the last years. All of them have, in our opinion a limitation: they can determine the trustworthiness or untrustworthiness of the assertions expressed by a given agent, but they don't supply mechanisms for correcting this information in order to extract some utility from it. In order to overcome this limitation, we introduce the concept of reliability as a generalization of trust, and present Fuzzy Contextual Corrective Filters (FCCF) as reliability modeling methods loosely based on system identification and signal processing techniques. Finally we illustrate their applicability to the appraisal variance estimation problem in the Agent Reputation and Trust (ART) testbed.

Introduction

Trust is one of the main concepts upon which human and animal societies are built. It is evident, therefore, the importance of its formalization for the construction of artificial or electronic societies, which so vast amount of interest have caused not only in the Artificial Intelligence and Computer Science communities, but also in such different ones as Sociology, Economics and Biology. Quoting (Sabater 2002):

> Artificial Intelligence is quickly moving from the paradigm of an isolated and non-situated intelligence to the paradigm of situated, social and collective intelligence. The new paradigm of the so called intelligent or adaptive agents and Multi-Agent Systems (MAS) together with the spectacular emergence of the information society technologies (specially reflected by the popularization of electronic commerce) are responsible for the increasing interest on trust and reputation mechanisms applied to electronic societies.

Over the last years, several attempts to make such formalization have been carried out from diverse points of view (recommender systems, social networks, electronic commerce...) All of them suffer, in our opinion, from a quite serious limitation. While they can provide a number, category or even fuzzy statement measuring the trustworthiness of a given agent or, more precisely, the trustworthiness of the information provided by a given agent, they fail in the sense that they don't supply any filtering or correcting method in order to make the provided information useful, even if wrong. The main point of this paper is: In some cases, false information transmitted by an agent can be useful if conveniently filtered.

The aim of this paper is threefold. In the first place, we want to make evident the importance of such filtering mechanisms in order for an agent to improve its performance in a multiagent environment, introducing the concept of reliability as an extension or generalization of trust. Secondly, we present fuzzy contextual corrective filters (FCCF) as a convenient and straightforward way, loosely inspired in systems identification and signal processing techniques, to implement those filtering/correcting mechanisms. Finally, we demonstrate the usefulness of FCCF by applying it to the appraisal variance estimation problem in the Agent Reputation and Trust (ART) testbed domain.

The Need for Trust Formalization in MAS

An essential characteristic of MAS is the existence of an information interchange between the individual agents forming the system. In the case of collaborative MAS, the aim of this communication is the improvement of the global performance of the system. Therefore agents, in general, do not lie each other consciously. In the case of competitive environments, however, individual agents are selfish, in the sense that its behavior is addressed to maximize some kind of individual utility function, even if that means a prejudice for the individual interests of the other agents or the diminution of the overall performance of the system. Communicative acts in competitive MAS are therefore addressed to obtain individual benefit and it is more suitable (because it can be profitable) the conscious communication of false information.

Both in collaborative and in competitive MAS, however, an emitter agent can communicate false information to a recipient agent because of several reasons. The

main ones being:

1. The emitter agent is, simply, wrong. He is honest, in the sense that he believes he is communicating a true statement, but the transmitted information is false.
2. Emitter and recipient agents do not use the same language. The message encloses a true statement, as understood for the emitter agent, but has a different and false meaning for the recipient agent. That's why ontologies are used, just to try to assure that all the agents in a domain speak the same language
3. A transmission error occurred. The emitted and received messages are different.
4. The emitter agent consciously transmits a false information to the recipient agent. The aim of such behavior can be supposed to be the obtaining of some benefit from the prejudicing of the recipient agent. That is the typical behavior we can expect in competitive environments

Whatever could be the reason behind the transmission of false information, individual agents need some kind of mechanism that allow them to deal with it. Agents can't afford (specially in competitive environments) to believe everything the other agents tell to them. A car vendor agent who commits itself to deliver a car "soon" and who says that the car is "fast" can be honest even if the car lasts a year to arrive and it can not run faster than 100 kilometers per hour. Perhaps he really believed what he was saying, perhaps the words "soon" and "fast" have a different meaning in the car vending language or even, perhaps, he said "late" and "slow" but somehow the sounds changed in their way from their mouth to our hears. More probably, however, he is deliberately lying to take profit from us. In either case, we need to learn from our experience in order to know what can be expected from him in further deals. Here is where trust and reputation modeling methods come in as an important field of study inside the theory of MAS.

Beyond Trust. Reliability

It is not inside the scope of this document to give a detailed account of the several trust and reputation formalizations that have been proposed along the last years, so we refer the interested reader to (Sabater 2002) for a survey of them. Nevertheless, a point seems to have been so far overseen, to our knowledge, by these trust formalisms. *It is not necessary to trust an agent (in the sense of believing it is saying the truth) in order to get some utility from the information provided by it.* This information can be useful even if it is false, provided we had some method to correct it.

Let's put an example: a watch agent that goes two and a half hours in advance will never tell you the right time, so you will do good not trusting it. Does it implies that you can't get any utility from it?. No. On the contrary, you can completely rely on it. Its regularity makes possible to correct the information it provides and get the exact time. We can say much the same thing about our car vendor agent. Better for us do not believe everything he could tell us, of course, but even if we don't trust him, we can yet extract some probably useful information from his offers, perhaps in the form of upper or lower bounds. Moreover, with the time, if we deal with him often enough, we can arrive to learn its language, that is, to capture regularities in it which can allow us to, for example, reject at once a car if he says of it to be "not very old".

The key concept in order to be able to correct messages coming from other agents is reliability[1]. If an agent tends to communicate similar information under similar circumstances, a moment will arrive when we will be able to extrapolate the circumstances, more or less correctly, from the received messages. On the contrary, if an agent emits just random messages it will be very difficult, if not impossible, to obtain from them any utility at all. Going back to the causes because of which agents can communicate false information, reliability can be expected mainly in cases 1, 2 and, perhaps to a minor extent, 3.

The corrective mechanisms, which we will call filters, can have very different structures. The filter for a watch agent that goes two hours and a half in retard could be as simple as adding 150 minutes to the time he says it is. On the other hand, we will need a much more sophisticated filter when dealing with the car vendor agent, maybe some kind of expert system. In the following section we will show how simple filters, based on fuzzy systems, can be constructed and how they can be learned and used to improve the performance of individual agents in their environment.

Fuzzy Contextual Corrective Filters

Think about the following problem: An agent A interacts with several other agents in a multi-agent environment requesting from them some kind of information, which they supply (this information can be false because of any of the reasons exposed in section 2). Suppose also that the correct answers to A's requests are made available to A by the environment in a posterior time instant, in such a way that A is able to know which agents told the truth and which agents lied, and how much. Our point is: for A to be able to perform well in this kind of environment it has to maintain a set of filters (one of them for each agent it interacts with) which allows it to correct the information received from the other agents, as well as to assess the possible utility of the corrected information. These filters must be dynamic, in the sense that they must evolve and adapt to changes in the environment and in the behavior of the other agents. So, (see figure 1) filters act as a translative layer that eases the process of interpretation of the messages sent by other agents. They can also, on the other hand, help the agent to translate the informa-

[1] From reliable, in the sense of "giving the same result in successive trials". (Merriam-Webster 2007)

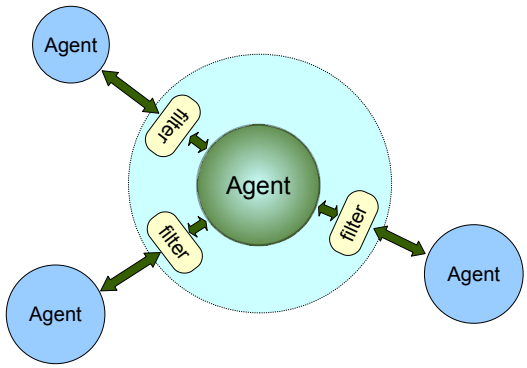

Figure 1: The set of filters of an agent act as a translative layer.

tion it wants to transmit to the language spoken by the other agents, increasing therefore the probability of being correctly understood.

It is also very important for the agent that owns the filter to have some kind of measure of the correctness of the filtered information, that is, the degree to which it can be expected to reflect the reality. We will call this value *reliability* and the filter will compute it from the observed regularities in the behavior of the filtered agent in past interactions.

Figure 2 shows the suggested structure for the construction of these filters, which we call fuzzy corrective contextual filters (FCCF) (del Acebo & de la Rosa 2002). A FCCF F has two parts, the corrective module and the reliability calculation module. The corrective module is a special case of a Mamdani fuzzy inference system [2] where the fuzzy rules have the form:

If A_1 is S_1 and ... and A_n is S_n and V is L_1 then W is L_2

where:

- $S_1, S_2 \ldots S_n$ are linguistic labels, defined by fuzzy sets on universes of discourse $X_1, X_2 \ldots X_n$, respectively.
- $A_1, A_2 \ldots A_n$ are are fuzzy variables taking values over the fuzzy power sets of $X_1, X_2 \ldots X_n$, respectively.
- L_1 and L_2 are linguistic labels defined by fuzzy sets over the universes of discourse U_1 and U_2, respectively. U_1 and U_2 can be, and usually are, the same set.
- V and W are are fuzzy variables taking values over the fuzzy power sets of U_1 and U_2, respectively.

We will call $A_1, A_2 \ldots A_n$ the *context variables*, V the *main variable* and W the *filtered variable*. We can see

[2] A explanation of fuzzy sets, fuzzy logic and fuzzy inference systems' theories is beyond the scope of this paper. We refer the interested reader to (Jang, Sun, & Mizutani 1997) where several excellent introductory chapters can be found.

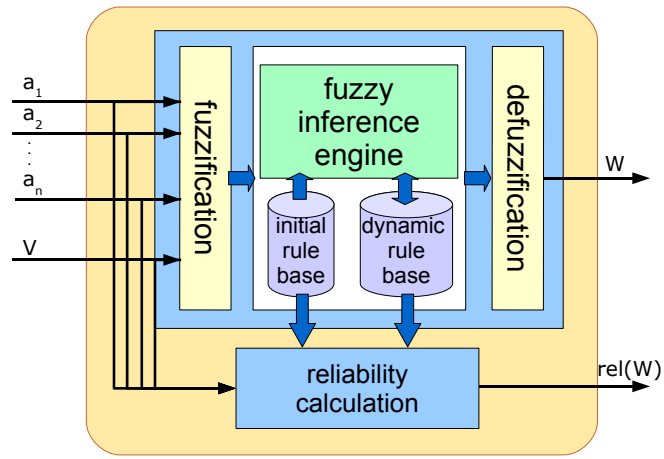

Figure 2: Structure of a fuzzy contextual corrective filter

the operation of the corrective module as a transformation of fuzzy sets over a certain universe U_1 to fuzzy sets over the universe U_2, in a way that depends on the values of the context variables as well as on the value of the main variable. The corrective module of a FCCF, then, filter the values (fuzzy sets) of the main variable to obtain new values (fuzzy sets over the same universe or another one) which are expected to be more suitable for some purpose. As is the case with general Mamdami fuzzy systems, it is possible to use FCCF on crisp input values to produce crisp filtered values by using appropriate fuzzification and defuzzification procedures.

The rule base of the corrective module has two components, the static and dynamic rule bases. The static rule base is fixed (and possibly the same) for every agent. It expresses the *a priori* assumptions about the behavior of the other agents in the environment and serve as a departing point in the interpretation of other agents's assertions. It can be as simple as identity or can, for instance, incorporate some common sense knowledge about the behavior which can be expected from certain kinds of agents. The dynamic rule base is built upon the information extracted (in the form of fuzzy rules) from the interactions between the agent which owns the filter and the filtered agents. It is dynamic in the sense that it evolves with time and can adapt itself to changes in the environment and in the behavior of the filtered agents. The construction of the dynamic rule base can be viewed as a system identification task where the behavior of the filtered agent has to be modeled from a set of examples, the results of past interactions between the modeling and the modeled agents. As a system identification problem, several modeling methods can be used, ranging from those based on a neuro-fuzzy, backpropagation-based approach (Jang's ANFIS (Jang, Sun, & Mizutani 1997) would be a good example of this) to those based on

lookup tables (Wang 1994) or, even, genetic algorithms (Cordon & Herrera 1995).

The function of the second part of the FCCF, the reliability calculation module, consists in computing the reliability of the filtered value obtained by the corrective module. Reliability will be a function of the input and context variables and will depend upon the number of prior similar interactions between filtering and filtered agents as well as upon the regularities observed during that interactions.

The ART Testbed

The Agent Reputation and Trust (ART) testbed (Fullam *et al.* 2005) is a framework, based on the art appraisal domain, for experimentation and comparison of trust modeling techniques. Agents function as painting appraisers with varying levels of expertise in different artistic eras. Clients request appraisals for paintings from different eras; if an appraising agent does not have the expertise to complete the appraisal, it can request opinions from other appraiser agents. Appraisers receive more clients, and thus more profit, for producing more accurate appraisals.

Let's focus in the opinion requesting part: when an agent A does not have expertise enough to guarantee a good appraisal for a given painting, it can buy the opinion of other, more expert, agents. The process is the following: first, agent A asks all or part of the other agents to provide a value stating their confidence in the accuracy of their appraisal of the painting. Then, A decides, upon the received confidence values, which agents to trust, that is, which opinions to purchase.

This is the main point where the communication of false or misleading information can happen in the ART testbed. An agent can declare a great confidence in its appraisal just to fool the requesting agent into purchasing it, and then produce a very bad appraisal. This will result in a big error in the requesting agent's appraisal and, consequently, a big loss in its the client share. On the other hand, the requesting agent has no way to know what the confidence value provided by an agent means. It is a value over an arbitrary range that has to be interpreted. It is perfectly possible for a given confidence value to mean completely different confidence levels for different agents.

Using FCCF for Appraisal Variance Estimation in the ART testbed

One of the main problems we faced in the construction of our ART testbed agent was the following: given two appraisals with variances α_1^2 and α_2^2, we knew the way to combine then in order to obtain the appraisal with the minimal expected relative error. We needed, therefore, a way to guess appraisals' variances departing from the confidence values supplied.

We solved the problem providing our agent with a set of FCCF, one for each agent other than itself in the environment. The structure of the filters is very simple. They have, as input variable, the confidence value stated by the seller agent, and, as context variable, the era to which the painting belongs. The filtered variable is the square of the relative error of the appraisal. We will see how the filtering module will produce, as output, a correction of the confidence value in the form of the expected variance of the seller agent's appraisal.

Rules in the initial rule base are predefined by design and serve the purpose of providing a sensible starting point to the interpretation process. Rules in the dynamic rule base, on the other hand, are continuously obtained from interactions between our agent and the filtered agents. Each of the rules in the rule bases, however, has the same form:

$$R_i : \text{If } era = E_i \text{ and } conf = C_i \text{ then } qError = Q_i$$

where E_i [3] and Q_i are singleton fuzzy sets over the sets of the eras and the positive reals, respectively and C_i is a fuzzy real number. So, for instance, if we purchase an appraisal for a cubist painting for which the seller agent declares to have a confidence 0.5, and the provided appraised value is 20000 but the real price of the painting turns out to be 25000 (giving a relative error of 0.2), we will add to our dynamic rule base the following rule: If $era = cubism$ and $conf = 0.5$ then $qError = 0.04$

We will have, then, a possibly large number of fuzzy rules in this form. Now suppose that we want to consider the possibility of purchasing an appraisal for a painting of a given era e from an agent which states that it has a confidence c in its appraisal. How to estimate the variance of the appraised value?. We know that the variance is defined as the expectation of the quadratic error, so, in principle it would be enough to gather all the interactions in which the agent has stated the very same confidence in its appraisal of a painting of the same era and estimate the variance from these data as the mean of the quadratic errors made. Unfortunately, confidence values will be, in general, scattered along a big range of values, so we can hardly expect to have enough of them to make the estimation accurate. We can, nevertheless, estimate the variance computing the output of the fuzzy system in the following way:

$$\sigma_{e,c}^2 = \frac{\sum_i \pi_i(e) \cdot \mu_i(c) \cdot TimeStep_i^K \cdot Q_i}{\sum_i \pi_i(e)\mu_i(c) \cdot TimeStep_i^K} \qquad (1)$$

where $\pi_i(e)$ will take the values 1 or 0 depending on whether the era of the painting corresponding to fuzzy rule R_i was e or not, Q_i is the quadratic error corresponding to fuzzy rule R_i, $TimeStep_i$ is the iteration in which the interaction corresponding to R_i happened (we can, then, use the parameter K to vary the relative influence of the rules in the computed result, giving

[3] In the ART testbed, paintings can only belong to one era, and they belong to it completely. It is possible, however, to imagine instances of the problem where paintings could belong, to a certain degree, to different eras. Our method is general enough to cope with this.

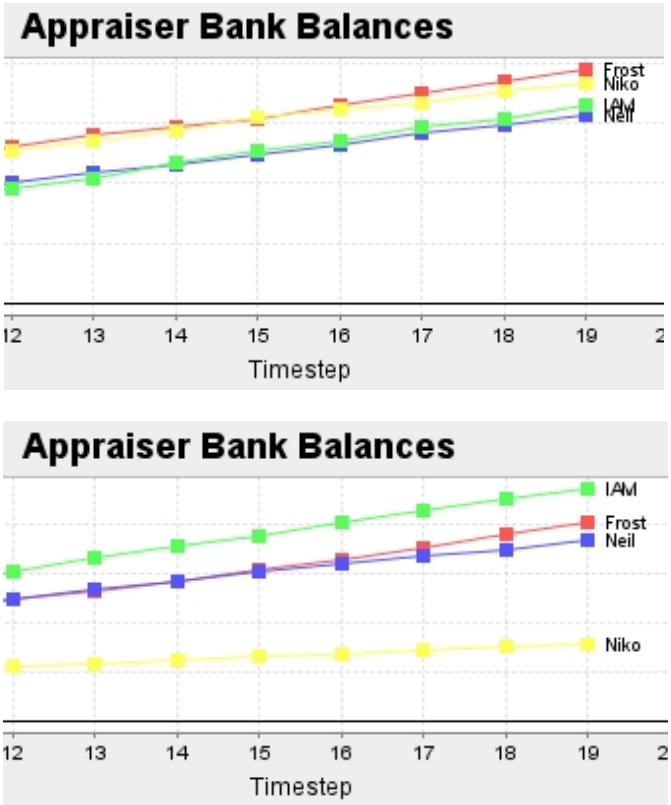

Figure 3: Nico *versus* IAM, Neil and Frost. *Bottom*: Results without filtering. *Top*: Results using FCCF.

more or less importance to more recent interactions) and $\mu_i(c)$ is the degree to which the value c belongs to the fuzzy number C_i, which we compute as:

$$\mu_i(c) = exp\left(-\frac{(c - C_i^*)^2}{AMP^2}\right) \quad (2)$$

where C_i^* is the central value of the fuzzy number C_i, the width of which can be controlled by the parameter AMP.

The implementation of the reliability calculation module for our problem is based in previous work by the authors (del Acebo *et al.* 1998). It mainly takes into account the completeness of the rule bases (roughly speaking, the number of rules that fire in the calculation of the variance).

Results and Future Work

We have tested our agent, Niko, against the winner agents in the last edition of the ART contest. A representative set of results is shown in figure 3. In the lower part of the figure we can see the performance of our agent when no FCCF are used. Our agent believes literally each confidence value received and, as was to be expected, the performance is quite poor. In the upper part of the figure we see the result of the competition against the same agents when our agent uses fully its filtering capabilities. As can be seen, in lack of further refinements, FCCF situates our agent at quite the same level as the best of them.

Our future work concerning the ART testbed will focus on two main areas. First, we want to find a method to decide what appraisals to buy, given their expected variances, in order to maximize the expectation of earnings. This could also be extended to the determination of the optimal quantity of money to be expended in the generation of our agent's appraisals. Secondly, we want to extend the application of FCCF to the filtering of reputation values provided by the agents. The good results obtained so far make us very optimistic, given the vast room for improvement that we have yet.

Acknowledgements

This work was supported in part by the Grant TIN2006-15111/Estudio sobre Arquitecturas de Cooperación from the Spanish government and by the EU project Num.34744 ONE: Open Negotiation Environment, FP6-2005-IST-5, ICT-for networked Businesses.

References

Cordon, O., and Herrera, F. 1995. A general study on genetic fuzzy systems. In *Genetic Algorithms in Engineering and Computer Science*. John Wiley & Sons.

del Acebo, E., and de la Rosa, J. L. 2002. A fuzzy system based approach to social modeling in multi-agent systems. In *The First International Joint Conference on Autonomous Agents & Multiagent Systems, AAMAS 2002, July 15-19, Bologna, Italy*, 463–464.

del Acebo, E.; Oller, A.; de la Rosa, J. L.; and Ligeza, A. 1998. Statistic criteria for fuzzy systems qualitiy evaluation. In Pobil, A. P. D.; Mira, J.; and Ali, M., eds., *IEA/AIE (Vol. 2)*, volume 1416 of *Lecture Notes in Computer Science*, 877–887. Springer.

Fullam, K.; Klos, T.; Muller, G.; Sabater, J.; Schlosser, A.; Topol, Z.; Barber, K. S.; Rosenschein, J.; Vercouter, L.; and Voss, M. 2005. A Specification of the Agent Reputation and Trust (ART) Testbed: Experimentation and Competition for Trust in Agent Societies. In *The Fourth International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS-2005)*, 512–518. ACM Press.

Jang, J.; Sun, C.; and Mizutani, E. 1997. *Neuro-Fuzzy and Soft Computing*. MATLAB Curriculum Series. Prentice-Hall, first edition.

Merriam-Webster. 2007. Merriam-webster online dictionary. http://www.m-w.com/.

Sabater, J. 2002. *Trust and Reputation for Agent Societies*. Monografies de l'Institut d'Investigació en Intel·ligència Artificial (20). Consell Superior d'Investigacions Científiques.

Wang, L.-X. 1994. *Adaptive Fuzzy Systems and Control. Design and Stability Analysis*. Prentice-Hall.

Expectations Enhanced Trust Model

Adel ElMessiry*
North Carolina State University
Department of Electrical and Computer Engineering
Raleigh, NC, USA
ammessir@ncsu.edu

ABSTRACT

With the emergence of the web of services and the semantic web, the need to find different methods of web service selection is gaining importance. Many efforts are focusing on the aspects of building the trust networks on the semantic web as well as agent-based models to utilize the trust in selecting the appropriate web service from a group of available services.The main direction of the research is to provide a model by which trust can be computed for several service providers that are available to accomplish the same task. In this work, we explore the concept of *'User Expectations'*. The trust is based on the rating assigned by the user. The rating is based on user satisfaction, which is based on the fulfilment of the user expectations. Without taking into consideration the original user expectations, the assigned ratings are not objective, but rather subjective. Thus, we propose a semantic web based model for capturing the user expectations to enhance the trust rating based on the requesting user's own expectations. We evaluate our approach via simulations on a simplified but realistic service instances and past and current consumers each with their own expectations. Our results show that using these considerations for user expectations, service agents are able to determine over time the best service selection for a consumer.

Categories and Subject Descriptors

H.4 [**Trust**]: Trust propagation

General Terms

Trust, Trust propagation, Semantic web

Keywords

Trust, Trust propagation, Semantic web

*A Doctoral Student.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
Trust Workshop at AAMAS'07 May 14–18 2007,Honolulu, Hawai'i, USA.
Copyright 2007 IFAAMAS .

1. INTRODUCTION

The rapid growth of web based transactions utilizing trust (as in eBay) has emphasized research into trust and trust propagation.The approaches to the study of trust are various and different from each other [3]. One approach is to propagate the representation of trust via a probability certainty distribution [14]. Others propose an atomic propagation [5] model, yet another approach is to represent trust as a vector [10]. The common between all the approaches is the focus on the Trust Rating without considering the other factors that may affect or improve the interpretation of that rating. Singh and Maximilien [13] introduce a trust model that is centered on a shared conceptualization for QoS (ontology) and a QoS preference model that considers consumer's tradeoffs among qualities as well as relationships between qualities. Lamsal [7] examine several approaches to define trust. Bernard Barber [8] defines trust as:

1. Expectation of the persistence and fulfilment of the natural and moral social orders.
2. Expectation of technically competent role performance from those we interact within social relationships and systems.
3. Expectation that partners in an interaction will carry out their fiduciary obligations and responsibilities, that is, their duties in certain situations to place others interests before their own.

The fulfillment of a consumer's expectation is essential to a consumer's satisfaction, and may indirectly affect the repurchase intention. Kim et al [6]. Different satisfaction models and theories have been developed in order to define and explain the phenomenon related to consumer satisfaction. The dominant conceptual model in the consumer satisfaction area is the Expectation-Confirmation Theory[1, 6]. We expand on this approach to include the user expectations in accounting for trust. User expectations are represented semantically .

2. TRUST AND REPUTATION

Whether it is a purchase transaction conducted in eBay, or an information transaction where a user obtains information from a web based resource, the user must have a certain level of trust that the transaction will be conducted correctly. The trustworthiness of one service provider over the others may lead the user to favor that provider. Mechanisms that support finding trust estimations based on a community are

called reputation systems [4]. There are two kinds of reputation systems: centralized and distributed. One successful example is eBay, where after each completed transaction, the user can leave a rating. The ratings of the provider are then averaged to provide an overall rating. The better the vendor's reputation, the more trustworthy becomes the vendor. The fundamental deficiency of this approach lies in its biases, that we can simply combine the different user ratings. That assumes that the different users have the same frame of reference. While this is true in case of a huge number of user ratings, where the normal distribution would hold true, is not valid for more practical and smaller set of ratings. Many efforts address the issue of malicious ratings such as a vendor employing others to provide false high ratings, or to provide false low ratings of another vendor in order to distort the other vendor's reputation. Other efforts concentrated on the propagation of trust through several second-hand sources. Few have explored what other factors can affect the rating of a user and thus affect the computation of the trust. One such effort is the work of [13], they introduced the use of a trust model that is centered on a shared conceptualization for QoS (ontology) and a QoS preference model that considers consumer's tradeoffs among qualities as well as relationships between the qualities.

3. USER EXPECTATIONS AND SATISFACTION

Merriam-Webster Online [9] offers the following definitions:

Expectation The act or state of expecting : ANTICIPATION in expectation of what would happen.

Satisfaction Fulfillment of a need or want.

While Cambridge offers the following definitions [2]:

Expectation noun 1 [C usually plural] when you expect good things to happen in the future.

Satisfaction noun 1 [C or U] a pleasant feeling which you get when you receive something you wanted, or when you have done something you wanted to do.

To understand the relation between expectations, satisfaction and reputation, we consider a three phase model as:

Pre Transaction Phase In this phase, the customer needs to fulfill a list of requirements, which the customer is seeking a vendor for, and has a list of expectations.

Transaction In the transaction phase, the customer will select the vendor that can fulfill the requirements.

Post Transaction After the transaction has completed, the customer is comparing the expectations that the customer had pre-transaction with the actual transaction. Those expectations can be either met or not met. Meeting the customer's expectations results in more satisfaction. We use the same logic to deduce that the customer will be more willing to give the vendor a higher rating. The opposite will incline the customer to give the vendor lower ratings.

This model is illustrated in Figure 1.

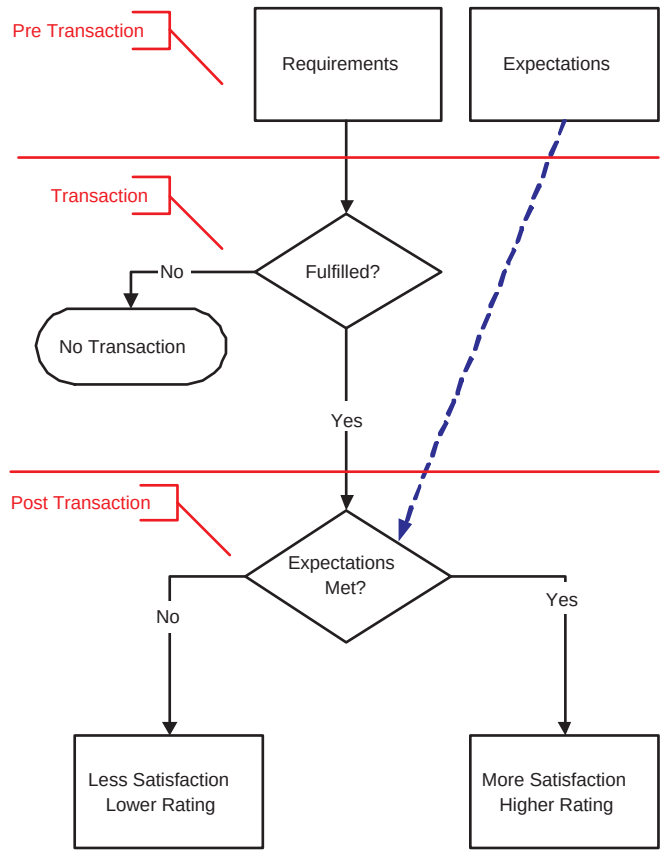

Figure 1: Transaction model

We can see that the relation between satisfaction and expectations is very strong. The more the user expectations are met, the more satisfaction is obtained. User expectations are different from other functional aspects of the service and may not be directly related to the actual service. An example would be package ordered online and shipped ground five to seven business days. We can note the following:

Not directly related to the service Although the shipping method is ground which takes five to seven business days, user expectations is that the package would arrive in 5 days that may include the weekend. The user could expect the package to be wrapped not boxed.

User relative experience The user may have been used to ordering from vendors that are nearer geographical proximity, thus, the ground shipping actually arrives in three days.

Current user context The user may have deadlines that depend on the ordered package and thus, expects - hopes- for faster delivery.

It is logical that users with shared expectations would share similar degree of satisfaction, in contrast, those user's rating of the rendered service would be more relevant. The main challenge is how to express user expectations? Further more, how can machines understand and compare those expectations? We need to ensure that user expectations can be dynamically defined based on the context of usage, but at the

same time we need to be able of relating those definitions. The answer to represent expectations semantically.

4. USER EXPECTATIONS PROFILE

In order to simplify the representation of user expectations, we suggest that user expectations can be represented by several pairs of elements and associated expectation levels. The element and unit of measurement are also be semantically expressed. The benefits of such a representation are that different measurement units and scales can be converted to each other and thus compared. We can define a *User Expectation pair* as:

$$UE(a, l) \tag{1}$$

where

$\boldsymbol{a}$ is a user expectations attribute

$\boldsymbol{l}$ is the level of expectation for that attribute

We can construct a *User Expectations Profile* by collecting user expectation pairs for all relevant attributes.

$$UEF = (UE_1, UE_2, .., UE_n) \tag{2}$$

4.1 Comparing User Expectations

Without the ability to express the meaning of the data, comparing two user expectations would not be possible. Two user expectations profiles can be compared semantically. We expect to have one of the following cases:

Explicit Match User expectation pairs with same definition in each profile, the two levels can be compared. the two user expectations profile match the attribute. Thus, the two levels can be compared.

Implicit Match Expectation pairs in the two user expectations profiles are related through one or more other definitions. Thus, the levels can be compared after resolving the relation between the pairs.

Mismatch No attribute in the two user expectations profiles can not be explicitly or implicitly matched. In this case, the missing pair is assumed to be neutral.

Now, let's assume we have two users, x and y, each with its expectation profile. To compare y's profile to x's profile, we perform the following steps:

This process will result in a new relative user expectation:

$$UE_{xy}(a) = 1 - |(UE_x(a) - UE_y(a))| \tag{3}$$

Thus, the more similar is the expectations of the two users the closer the relative expectation approaches. Consider that user x has n attributes, while user y has m attributes, we can apply the same to all of the user expectations to get the relative user expectations profile:

$$UEF_{xy} = Compare(UEF_x, UEF_y) \tag{4}$$

Where:

$$Compare(UEF_x, UEF_y) = (UE_{xy}(a_1), .., UE_{xy}(a_n)) \tag{5}$$

Note that:

$$UEF_{xy} \neq UEF_{yx} \tag{6}$$

As the items ignored in UEF_y when compared to UEF_x will not be ignored when comparing UEF_x to UEF_y.

```
foreach expectation pair do
    compare the pair;
    if the pair match then
        compute the inverse of the difference between
        pair levels
    else
        if pair in UEF_x, but not in UEF_y then
            assume the user is off by a percent, say 25%
        else
            if item in UEF_y but not in UEF_x then
                igonre
            end
        end
    end
end
```

Algorithm 1: Comparing Expectation Pair

4.2 User Expectation Enhanced Ratings

Assume we have two users, x *and* y, user y has a rating of a service p as:

$$R_y(p) \tag{7}$$

Using the presented approach, we can obtain UEF_xy. To compute a numeric value, we average all the pairs in UEF_xy:

$$UEF_{xy} = \frac{\sum_{i=1}^{n} UE_{xy}(a_i)}{n} \tag{8}$$

Thus, the user expectations enhanced rating for user y can be defined as:

$$R_{yx}(p) = \alpha UEF_{xy} R_y(p) \tag{9}$$

where: α: Expectations Enhancement Coefficient.

5. EVALUATION

In order to evaluate this approach, we need to create a well defined case demonstrating how to account for the impact of user expectations and examine the result with and without using this approach.One of the most widely studied examples of using trust on the internet is eBay. eBay implements a simple rating system, for each transaction both parties involved can leave a feed back and a rating. The rating can be one of three values:

+1 Point For each positive comment.

0 Points For each neutral comment.

–1 Point For each negative comment.

The Positive Feedback Percentage is then calculated by dividing the positive feedback by the total feedback. This simple model is greatly enhanced by applying the results of this research.

5.1 Hypothesis

The user's perception of trust is acquired via the provider's reputation, which depends not only on the objective behavior between the seller and the buyer contributing to the provider's reputation, but also on how the behavior relates to the buyer's expectations. The user's perception of trust can be enhanced by accounting for each buyer's expectations relatively to the user's own expectations when assessing their contribution to the seller's reputation.

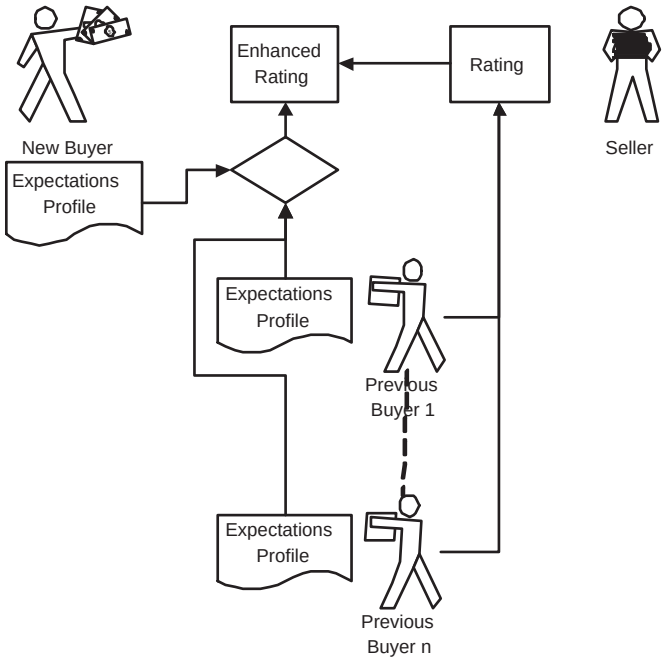

Figure 2: Expectations Enhanced Rating Model

The expectations enhanced model is illustrated in Figure 2. We will need to answer the following:

Previous buyer's rating Data set directly obtained from eBay

Previous Buyer's expectations Deduced from the buyer's text feedback [12]. Since judging expectations is subjective, we need to focus on the negative feedbacks [12, 11]. Find the key complaint and assume that the buyer expected better.

User's expectations Assumed to see the effects. We create a profile based on the expectations that were found. Showing that accounting for them improves the trust and allows the individual to select a better match.

5.2 Experiment Steps

Given a set of eBay transactions for several sellers with:
a. past consumer's ratings,
b. knowledge of the service past buyer's expectations, and
c. knowledge of the current buyer's expectations
We will find that (a) can be enhanced based on the correlation between (b) and (c).
The user expectation will be deduced by individuals participating in the experiment. They will follow the following steps:

- For a given category,
- Read the negative and neutral comments for each seller.
- Deduce what was the expectations not met, and hence complained about, of the user that provided the feedback.
- Provide the following:
  1. Total number of feedback
  2. Number of negative or neutral feedback.
  3. Breakdown of the expectations not met per each negative or neutral feedback.

For example, two sellers were selected. Both sellers provide goods under the 'Cell Phones and PDAs, Bluetooth Wireless Accessories, Headsets - Wireless' category. Both of them have identical PFP of 96.6 percent. How can a new buyer distinguish between them? This is illustrated in Table 1.

By studying the negative feedback left for each sellers by

Table 1: Raw Data

Seller	Positive	Not positive	Positive %
Seller 1	3444	121	96.606%
Seller 2	2498	83	96.784%

other buyers, we attempt to deduce the expectations profile of the user. We select one attribute, for example 'Level of communications'. Notice that the level of communication is not part of the advertised service, rather it is an expectation. Some buyers expect quick response while others do not care that much about it. From the sample of the feedback collected, let's look at the following:

For Seller 1

- Sent out the wrong item. offered an exchange if i sent the item back.
- This earpiece doesn't ring in my ear it has sound but doesn't ring in my ear.
- Slow shipping and did not contact until I contacted. Recommend with caution.
- Didn't come with original packaging, but good all the same.

For Seller 2

- No contact AT ALL, sudden refund placed into paypal acct. Bogus seller.
- Serious communication issues, very slow delivery, hard to deal with.
- Unresponsive, they canceled transaction.
- Bad customer service, slow response, did not deliver the item.

Applying the model we get the results in Table 2.

Table 2: Enhanced Data

Seller	Communication Issues	Enhanced not positive	Enhanced Positive%
Seller 1	100	221	93.970%
Seller 2	80	163	93.874%

We can clearly see that Seller 2 has a lower level of communications, thus, if a new buyer has a high expectations

for the level of communications, it is more likely that the buyer will be disappointed. By applying the enhancement suggested in this work, the score of Seller 2 will be lowered based on the expectations profile and thus, the buyer will select Seller 1. This result is illustrated in Figure 3.

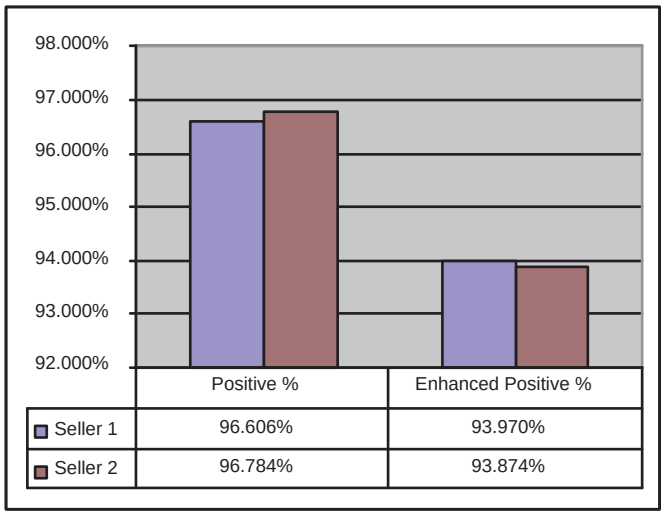

Figure 3: Results

6. CONCLUSION AND FUTURE WORK

In recent years two main developments have emerged, a web of services and trust based approaches to select and compose web services. In this work we have demonstrated that 'User Satisfaction' depends on meeting the 'User Expectations', subsequently, the more satisfied the users are, the highly they will rate the vendor. We have proposed to consider 'User Expectations' when obtaining trust. We outlined how to represent user expectations semantically using RDF. We demonstrated that the semantic approach will provide a dynamic and extensible model. We have provided a model to compute the Expectations Enhanced Trust. This approach has merits which will enable service agents to determine over time the 'best' service selection for a consumer. To further evaluate this approach, we will build an application that will interface with eBay to collect more data. This data can be then queried to provide better examples. We also intend to investigate the possibility of following each buyer to attempt to deduce the complete expectations profile. This will be obtained through the analysis of all the buyer's feedback.

7. REFERENCES

[1] A. Bhattacherjee. Understanding information systems continuance: An expectation-confirmation model. *MIS Quarterly*, 25(3):351–370, September 2001.

[2] Cambridge. Cambridge online dictionary.

[3] C. Castelfranchi, R. Falcone, and G. Pezzulo. 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*, pages 89–96, New York, NY, USA, 2003. ACM Press.

[4] R. Falcone and C. Castelfranchi. Trust dynamics: How trust is influenced by direct experiences and by trust itself. In *AAMAS '04: Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 740–747, Washington, DC, USA, 2004. IEEE Computer Society.

[5] R. Guha, R. Kumar, P. Raghavan, and A. Tomkins. Propagation of trust and distrust. ACM, May 2004.

[6] D. J. Kim, D. L. Ferrin, and H. R. Rao. A study of the effect of consumer trust on consumer expectations and satisfaction: the korean experience. In *ICEC '03: Proceedings of the 5th international conference on Electronic commerce*, pages 310–315, New York, NY, USA, 2003. ACM Press.

[7] P. Lamsal. Understanding trust and security. http://www.cs.Helsinki.FI/u/lampa/papers/UnderstandingTrustAnd 2001.

[8] S. P. Marsh. *Formalising Trust as a Computational Concept*. Doctor of philosophy, University of Stirling, 1994.

[9] Merriam-Webster. Merriam-webster online dictionary.

[10] I. Ray, S. Chakraborty, and I. Ray. Vtrust: A trust management system based on a vector model of trust. In *Lecture Notes in Computer Science*, volume 3803/2005, pages 91–105. Springer Berlin / Heidelberg, 2005.

[11] P. Resnick, K. Kuwabara, R. Zeckhauser, and E. Friedman. Reputation systems. *Commun. ACM*, 43(12):45–48, 2000.

[12] P. Resnick and R. Zeckhauser. Trust among strangers in internet transactions: Empirical analysis of ebays reputation system. Working paper, 2001.

[13] M. P. Singh and E. M. Maximilien. Agentbased trust model involving multiple qualities. pages 25–29. AAMAS, ACM, July 2005.

[14] Y. Wang and M. P. Singh. Trust representation and aggregation in a distributed agent system. In *Proceedings of the 21st National Conference on Artificial Intelligence (AAAI)*. AAAI, AAAI, July 2006.

Learning Task-Specific Trust Decisions

Ikpeme Erete
ikpeme@cc.gatech.edu
Mathematical & Computer Sciences Department
University of Tulsa
Tulsa, Oklahoma, USA

Sandip Sen
sandip@utulsa.edu
Mathematical & Computer Sciences Department
University of Tulsa
Tulsa, Oklahoma, USA

ABSTRACT

We study the problem of agents locating other agents that are both capable and willing to help complete assigned tasks. Each task is defined by a vector of attribute values and an agent has pre-defined capabilities for different task types. An agent incurs a fixed cost for each help request it sends out. To minimize this cost, the performance metric used in this work, an agent should learn based on past interactions to identify agents likely to help on a given task. We compare three trust mechanisms for choosing the order in which other agents are asked for help: success-based, learning-based, and random. We also consider different agent social attitudes: selfish, reciprocative, and helpful. We evaluate the performance of these social attitudes with both homogeneous and mixed groups of trust-decision makers. Our results show that learning-based trust decisions, which consider the current task characteristics, consistently performed better than other schemes. We also observed that the success rate of obtaining help is significantly better for reciprocative agents over selfish agents.

1. INTRODUCTION

To facilitate collaboration and coordination in a multiagent system (MAS) agents need to learn about other agents capabilities, preferences and goals. Previous research on agent learning is limited in scope as they focused on learning tasks descriptions that agents can or cannot solve [13]. In other words, an agent is competent in one particular task type with no other skills for other task types. In reality, however, an agent's expertise can vary for different task types. Our research framework takes into consideration varying levels of agent expertise.

The scenario we study is the following: each agent is assigned the same set of tasks where each task is described by a set of attribute-value pairs. Given a task description, an agent needs to find another agent to help it complete the given task. Each agent can perform only a subset of the task types and have varying competency levels for the task types it can perform. The expertise of an agent for different task types is represented by a decision tree defined over the space of domain attributes. Finding an agent that can help depends on the task at hand and an agent must be able to locate such a collaborator quickly. We assume that the asking agent incurs a fixed cost for each help request and need to minimize this cost by locating, with few requests, an agent capable and willing to help. Agent expertise is not advertised and hence an agent does not directly know the expertise of the other agents in the system. Over time an agent's expertise and social attitude can be learnt and the corresponding expertise model built using past interactions with that agent. The learning goal is to map from any given task to an ordering of the set of agents by their likelihood of helping with that task. We refer to this mapping as a *trust decision mechanism*. An agent would like to quickly form these trust decisions while keeping the learning cost at a minimum.

An agent requesting for help expects that the asked agent would help if it can achieve the corresponding task, but this is not always the case. We refer to an agent's attitude towards help request as its social attitude. In our work, we evaluate these social attitudes: helpful, selfish, and reciprocative. Some other factors that determine if a help request would be honored include the ability of the asked agent to do the task, past interactions between the asking and the asked agent, etc.

We emphasize that the trust decision mechanism and hence the order in which agents are asked for help is instrumental in determining the total asking cost incurred by the asking agent. To minimize this, learning can be used. We compare a heuristic, success-based scheme based on frequency of past helps, with a more sophisticated learning-based scheme, which identifies agents that are likely to help for given task types.

We evaluate and compare the effectiveness of different trust mechanisms and a random agent selection scheme for minimizing asking costs for agents. We varied other parameters that affected the expertise levels of agents and evaluated the corresponding effects on their performance.

The rest of the paper is organized as follows: Section 2 discusses the related work. Section 3 describes the environment modeled and describes in detail the various agents, including their decision making processes, social attitudes, and expertise modeling. Section 4 present experimental results and analysis. Finally, Section 5 presents conclusions and possible future work.

2. ENVIRONMENT AND AGENT MODEL

In this section we present the model of our environment and details about agent properties and strategies. We simulated an environment where agents are assigned a set of tasks. Agents seek help from other agents to perform an assigned task. The asking agent incurs a fixed cost for every help request ("asking cost"). The goal of each agent is to minimize total asking cost. To achieve this an agent must locate, with as few attempts as possible, another agent that can and will provide help with an assigned task.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
AAMAS-TRUST workshop '07 Honolulu, Hawaii USA
Copyright 200X ACM X-XXXXX-XX-X/XX/XX ...$5.00.

The cost to an agent for completing a task is based on the expertise of that agent for that task type. The ability to perform a task is dependent on the agent's expertise, which is modeled by a decision tree. For a given task description and a decision tree, a boolean response is generated. If the response is positive, we use the length of the path from the root to the leaf for that task description as a rough measure of the agent's expertise for that task. The decision to help another agent is based on the agent's social attitude, e.g., agents may be selfish, reciprocative, etc. The expertise of an agent in the environment is private information to that agent. Hence agents need to learn the expertise model of other agents from experience and subsequently use the learnt knowledge to build a trust decision mechanism.

2.1 Environment

The environment is formally defined as $E = (A, T, C_a, E)$, where

- $A = \{a_1, a_2, \ldots, a_n\}$ is the **set of agents**.
- T is the **set of Task description**, where each task $t \in T$ is defined as a set of pairs $\{(a, \nu_a)\}$ of attribute-values. For fairness of evaluation, each agent is assigned the same set of tasks.
- C_a is the **asking cost** per help request.
- E is **set of Expertise** where $e_i \epsilon E$ corresponds to the decision tree representing the capabilities of agent a_i.

2.2 Task Description and Competency

A task is described by a set of attribute-value pairs. Task descriptions are randomly generated based on the attributes and the corresponding domains.

2.2.1 Domain Attributes

The attributes and corresponding values that we have used in our model corresponds to the problem of searching for goods or services:

Tasktype: This indicates the task to be performed. It can be used to represent any goods or services.

Priority: Reflects the time constrains on goods and services.

Quality of service: Indicates the quality of goods or services that is acceptable.

Help duration: The time commitment required for the task.

Advance notice: Time until the task has to be completed.

Quantity: Units of goods/services required.

Help cost: This indicates what an individual is prepared to pay for getting help from another.

Preference: Level of bias or importance for the task.

To expedite experimentation, we have limited the possible alternative attribute values to three for each attribute.

2.2.2 Task Competency

Each agent possesses a given expertise profile which determines if it is able to perform any given task and the corresponding cost. The expertise profile is encoded by a decision tree that determines its varying level of expertise over task types. An agent's decision tree represents the task descriptions that are known to the agent. A path from the root to any leaf in the tree and the outcome of this path determines the expertise of the agent for tasks matching the description specified by the conditions on that path. The outcome of leaf labels is determined probabilistically. Decision trees are effective representations of agent expertise because they can represent, in an accessible format, disjunctions of conjunctions. When a task description is presented to an agent, it simply traverses its decision tree based on the task attributes values and finds a decision at a leaf node: *true*, if it can do the task or *false*, if it can not.

We build these decision trees, representing agent expertise, randomly from attributes and attribute values. The decision-tree construction algorithm presented in Algorithm 1 gives control over two important features of the constructed tree:

Algorithm 1: Decision Tree Construction Algorithm

```
Function DecisionTree(attribs,P_success,P_leaf)
return decision tree
set of attributes with corresponding values: attribs
probability of agent capable of doing task: P_success
probability of a leaf node: P_leaf
{
if P_leaf then
    if flip(P_success) then
        return +
    else
        return -
else
    if no attribute left then
        if flip(P_success) then
            return +
        else
            return -
    else
        att ← Choose-attribute(attribs);
        m ← make-node(att) ;
        attribs ← attribs - att ;
        left node(m) ←
        DecisionTree(value(att,left),
        attribs,P_success,P_leaf);
        mid node(m) ← DecisionTree(value(att,
        mid), attribs,P_success,P_leaf);
        right node(m) ←
        DecisionTree(value(att,right),
        attribs,P_success,P_leaf);
        return m
}
```

Outcome: We use a probability, $P_{success}$, to represent the likelihood that an agent can do an arbitrary task. Using $P_{success}$ we can control the types of task an agent can perform.

Tree depth: We use a probability, P_{leaf}, to control the depth of the trees. A deeper, compared to a shallower, tree implies that an agent is more specialized in tasks.

Given a task description, the combination of the matching path

from the root to a leaf in its decision tree and the corresponding outcome determines the expertise level of an agent for that task type. The cost to an agent for doing a task is inversely proportional to the length of the path in its decision tree matching the task description. For example, if an agent only matches just one feature in a given task description, signifying only general-level expertise in the task, the incurred cost is a maximum; on the other hand, if it matches all the task features, signifying the most and specific expertise for the task, then a minimum cost is incurred[1]. Similarly, the asking agent can calculate the cost it incurs if it had to do the task.

2.3 AGENT DECISIONS

The focus of our work is increasing the efficiency of trust decisions for locating another agent who is capable and willing to help with an assigned task. To emphasize and evaluate the efficiency of trust strategies, we require our agents to seek help for an assigned task even if it has the ability to complete it. Put another way, an agent can always benefit if another agent performs a task on its behalf. An agent's response to a help request is influenced both by its expertise level and its social attitude towards helping. For representing trust decisions, though it may be possible to develop separate models for agent expertise levels and social attitudes, we believe it is sufficient to develop a combined model that predicts an agent's response given a task description. The performance of an agent depends on how effectively it is able to learn accurate help-giving behaviors of other agents in the society.

2.3.1 Social Attitudes

We use three types of agents to represent the social attitudes commonly found in a society:

Helpful agents: These agents help any asking agent irrespective of the attitude of the asking agent towards them. They are susceptible to exploitation.

Selfish agents: These agents request help but never accept help request. They can benefit in the presence of helpful agents by exploiting their benevolence.

Reciprocative agents: These agents use the balance of cost and savings to determine whether to accept a given request for help. The actual probabilistic decision policy used is based on Sen's work [16] and is presented below.

Reciprocity model:

Reciprocity involves the mutual or cooperative exchange of favors or privileges. We modeled this social attitude by using the framework provided by Sen [16]. They proposed an adaptive and tunable reciprocity scheme as shown in Equation 1

The cost incurred in carrying out the task in Equation 1 depends on the expertise of the helping agent and the current task type. The possibility of a reciprocative agent helping another agent is dependent on the outcome of the probability function shown in Equation 1. The use of balance of cost as an input to the function incorporates the influence of past interaction history between the asking and the asked agent on the helping decision. The function shown in Equation 1 is a sigmoidal function where the probability of helping increases as balance increases and decreases with increasing task cost. A higher value of β makes the agent more inclined to help initially while τ can be used to control the steepness of the curve,

[1]This is only a rough, plausible measure for generating task costs to simplify our implementation. Other, more elaborate, knowledge-based cost functions can be used, and should not affect the performance of our trust mechanisms.

$$Pr(i,k,l) = \frac{1}{1 + \exp \frac{C_{ij}^{kl} - \beta \cdot C_{avg}^{k} - B_{ki}}{\tau}} \quad (1)$$

where

$Pr(i,k,j)$: the probability that $agent_k$ will carry out task t_ij for $agent_i$,

C_{ij}^{k}: the cost of carrying out t_{ij} for $agent_k$,

C_{avg}^{k}: the average cost of tasks performed by $agent_k$,

B_{ki}: the balance of cost, i.e., the difference in savings obtained and the extra cost incurred by $agent_i$ from $agent_k$ over all previous interactions,

β, τ: constants with which the social attitude can be tuned.

e.g., make it a linear or a step function (corresponding to deterministic decision threshold).

Over time, a reciprocative agent can adapt to different help attitudes with the other agents in the environment [16]. This characteristic of the above-mentioned helping decision mechanism makes it suitable for our work since in our environment agents would interact with agents of varying social attitudes.

2.3.2 Decision Mechanisms for Soliciting help

Agents need to carefully decide who to ask for help given a task at hand. When presented with a task, an agent can use its trust decision mechanism to decide on the best agent to request help. Every request is associated with a cost called "asking cost" and the agent wants to minimize the total asking cost. The trust-mechanisms for soliciting help we have used are:

Random: An agent using this baseline mechanism randomly decides who to ask for help.

Success-based: This trust-mechanism makes help-soliciting decisions using the history of interactions with other agents. The agent keeps track of the frequency of help received from other agents and orders the agents based on the frequency of help received. Agents in the list are asked in decreasing order of this frequency until one agent agrees to help.

Learning-based: This trust-mechanism model another agent's help-giving decision for each task type from previous interactions with that agent. For a given task, the asking agent uses the model to predict the likelihood of the other agents helping with that task. We use the nearest neighbor algorithm to determine these likelihoods. The asking agent then asks other agents in decreasing order if these likelihoods until one agrees to help.

The latter two approaches depend on experiences from previous interactions in making help-seeking decisions. To enable them to make decisions at the onset of the experiment, when little or no interaction history is available, these decisions are made at random until a fixed number of interactions have occurred with each agent. The learning scheme is then applied to subsequent decisions.

Nearest Neighbor algorithm:

In the nearest neighbor algorithm, past classified training instances are stored as a model and new queries are answered by finding the nearest point(s), using some distance metric, and then

classification of those point(s) are used to compute the classification of the query point. In our domain, past experiences for each positive and negative response from other agents are stored as part of the model and new queries are mapped to most relevant past experiences to predict an outcome. The challenge in this approach is determining an acceptable neighborhood for an input instance. If the neighborhood is too small it would not contain any data points, while if it is too large it may include all the stored data points resulting in a prediction that is not very accurate and is the same everywhere. A solution to this problem is to define the neighborhood to be just large enough to provide a meaningful estimate by including the *k* nearest points. We used an iterative procedure to determine an acceptable neighborhood.

Various distance metrics, *D(m,n)*, used to compute the distance between *m* and any other point *n*, have been studied in literature such as the Hamming distance (defined as the number of features on which *m* and *n* differ), a special case of Mahalanobis distance, and Manhattan distance. We have chosen to use the least mean square metric in determining the distances between points. To use this approach we need to map each attribute value to a real number and then use Equation 2 to evaluate the distance.

$$D(m,n) = \sqrt{\sum_{i=1}^{N} (m_i - n_i)^2} \tag{2}$$

where

m_i and n_i: values for attribute i for the query point m and the stored point n respectively.

N: total number of attributes.

Predicting an agent's response

We utilize both positive and negative response from that agent in the past for modeling an agent's decision mechanism and predicting response. We use the nearest neighbor approach to determine the likelihood of an agent accepting a help request to the current task description. Previous acceptance and denials of help request of the ith agent, $agent_i$, is stored and indexed by the corresponding task descriptions. Given a new task description, *m*, we find the *k* nearest stored points, K, in the task space and the probability that $agent_i$ will agree to help with this task is calculated as

$$Pr(i,m) = \frac{\frac{1}{\sum_{j \in K^+} D(m,j)}}{\frac{1}{\sum_{j \in K} D(m,j)}} \tag{3}$$

where K^+ is the set of positive responses in K. The probability function, Equation 3, is a weighted nearest neighbor algorithm. Therefore, as the number of positive responses from an agent increases and the sum of distances decreases, the predicted probability of help increases. Agents with higher predicted probability for helping are approached before approaching others

3. EXPERIMENTAL RESULTS

We conducted a series of experiments to study the effects of social attitude of an agent and trust mechanisms for modeling other agents' helping decisions. Our goal is to evaluate if agents can effectively learn to select which other agent to trust given an assigned task. We use the "asking cost" as our primary performance metric and the *t*-test to determine if observed performance differences between the algorithms are statistically significant. Our experiments have been run using two environments: a homogeneous environment where all agents use similar trust mechanisms, and a heterogeneous environment where different agents use different trust mechanisms.

3.1 Experimental setup

We simulated an environment with 10 agents, where each agent is assigned the same set of 3000 tasks for fairness of comparison and a value of *k*=5 is used in the nearest neighbor algorithm. The results are averaged over 10 runs.

As indicated earlier, our environment comprises of agents with three different social attitudes. Helpful agents represent an ideal case and we shall use that as our basis of comparison and analysis of the selfish and reciprocative agents.

3.1.1 Homogeneous Environment

We first compare performances of the agent social attitudes using similar trust mechanisms. Next we compare performances of each social attitude using different trust mechanisms and analyze the effects of increasing $P_{success}$. Finally, we vary the percentage of selfish agent in a group of selfish and reciprocative agents.

Effect of Decision mechanism: We derive the following conclusions from results presented in Table 1 with different homogeneous groups as well as mixed groups of reciprocative and selfish agents using different trust decision mechanisms:

- Helpful agents perform better than the other types of agent with each trust decision mechanism. This is because these agents always accept help requests. The drawback to this approach is that it will not be effective in open groups which can contain exploitative agents.
- In groups comprising selfish and reciprocative agents, reciprocative agents identifies the presence of selfish agents and rarely help such agents. The calculated rate of success to asking cost from these tables show that 0.038% of request for help from selfish agents are honored irrespective of the trust mechanism used. The reciprocative agents' success rate is best with learning-based trust. Reciprocative agents are also able to generate significantly more help than selfish agents.
- The failure of selfish and reciprocative agents to match the success rate of helpful agents is because they do not honor all help requests.

With increasing $P_{success}$ all trust schemes do better. This is because as agent expertise increases, so does the number of tasks that can be completed.

The performance of agents depends on how well they are able to learn effective trust policies. In general, all social attitudes except for selfish agents perform better when the trust mechanism is learning-based except for selfish agents. Selfish agents perform better using random trust mechanism. This is due to its social attitude. Selfish agents do not help others and after a while other agents learn not to help them. Within a few interactions with other agents the selfish agents using other trust mechanisms besides random have identified who to request help from. But they fail to realize that other agents have also discovered that they are selfish and would refuse to assist. As agents who originally helped may no longer help, it might be just more effective to ask randomly. The selfish agents can increase savings by using random selection as long as all the other agents have not learnt that the asking agent is selfish.

Effects of varying $P_{success}$: Our next series of experiments is aimed at observing how the performance of agents using similar trust mechanisms varies with increasing expertise levels in the system. We expect that with increasing agent expertise more task descriptions would be solved and this will lead to a reduction in total asking cost.

	Asking Cost		Successes		Help offered	
L-based	*Ave.*	*Std*	*Ave.*	*Std*	*Ave.*	*Std*
Helpful	9740.5	167.2	2493.1	24.9	2493.1	327.2
Sel.	27318.2	82.2	10.6	0.1	0.0	0.0
Reci.	19881.7	2565.8	1134.4	348.7	1137.0	375.4
Mixed	21369.0	2965.7	909.6	414.9	1137.0	375.4

	Asking Cost		Successes		Help offered	
Random	*Ave.*	*Std*	*Ave.*	*Std*	*Ave.*	*Std*
Helpful	13978.6	154.8	2493.1	24.3	2493.1	420.4
Sel.	26957.3	0.7	10.4	0.05	0.0	0.0
Reci.	21828.9	1681.2	1147.6	353.5	1150.2	382.9
Mixed	22854.6	1969.4	920.1	420.3	1150.2	382.9

	Asking Cost		Successes		Help offered	
Success-based	*Ave.*	*Std*	*Ave.*	*Std*	*Ave.*	*Std*
Helpful	13807.5	182.5	2493.1	24.3	2493.1	1118.4
Sel.	27416.5	55.3	10.4	0.2	0.0	0.0
Reci.	20738.9	2682.5	1090.5	402.50	1093.1	428.8
Mixed	22074.5	2996.5	874.5	456.8	1093.1	428.8

Table 1: Averages and Standard Deviation of asking cost, number of requests honored (successes) and number of helps offered, for each trust decision strategy (L-based: Learning-based, Sel.:Selfish, Reci: Reciprocative, Mixed: Sel and Reci.); $P_{success} = 0.2$, $P_{leaf} = 0.5$, Number of Tasks=3000, Number of Agents=10, and Number of Runs=10.

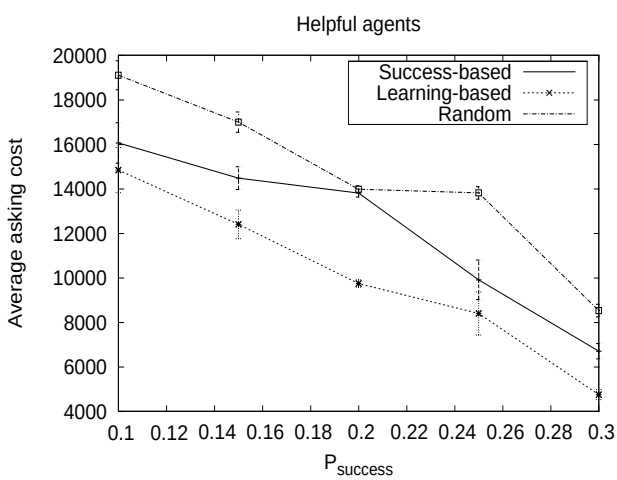

Figure 1: Average Asking Cost for helpful agents with varying $P_{success}$; # Tasks=3000, P_{leaf}=0.5, and # Runs=10.

We plotted results when varying the outcome probability, $P_{success}$, in the range of $0.1 - 0.3$, for each social attitude and different trust decision mechanisms. From Figures 1 and 2, we conclude the following:

- With increasing $P_{success}$ the average asking cost decreases for helpful agents irrespective of the trust mechanism. The $askingcost/success$ is lowest for learning-based, and highest for random.

- With learning and success-based schemes, the performance of selfish agents improve initially with increasing $P_{success}$ but then worsens for $P_{success} > 0.2$. The performance of random trust mechanism for selfish agents is consistent throughout the probability range but is lower than other algorithms.

- Reciprocative agents perform better than selfish agents and random trust schemes perform worse than the others. The performance of learning-based trust mechanism begins to worsen for $P_{success} = 0.25$, though not as markedly as for selfish agents. An increase in $P_{success}$ increases the number of positive responses from a given agent. This results in an increase in the frequency of help request to those agents and corresponding help response. Consequently, there is a race condition between savings in cost and incurred cost, which produces a fluctuating trust relationship between agents. This accounts for the slight performance worsening at higher values of $P_{success}$.

3.1.2 *Varying number of selfish agents in a system of selfish and reciprocative:*

We ran experiments varying the percentage of selfish agents in the group. We initially ran experiments with two selfish and eight reciprocative agents. Subsequently, we varied the percentage of selfish agent from $0\% - 60\%$ to observe and analyze the effects of this change on the performance of all agents.

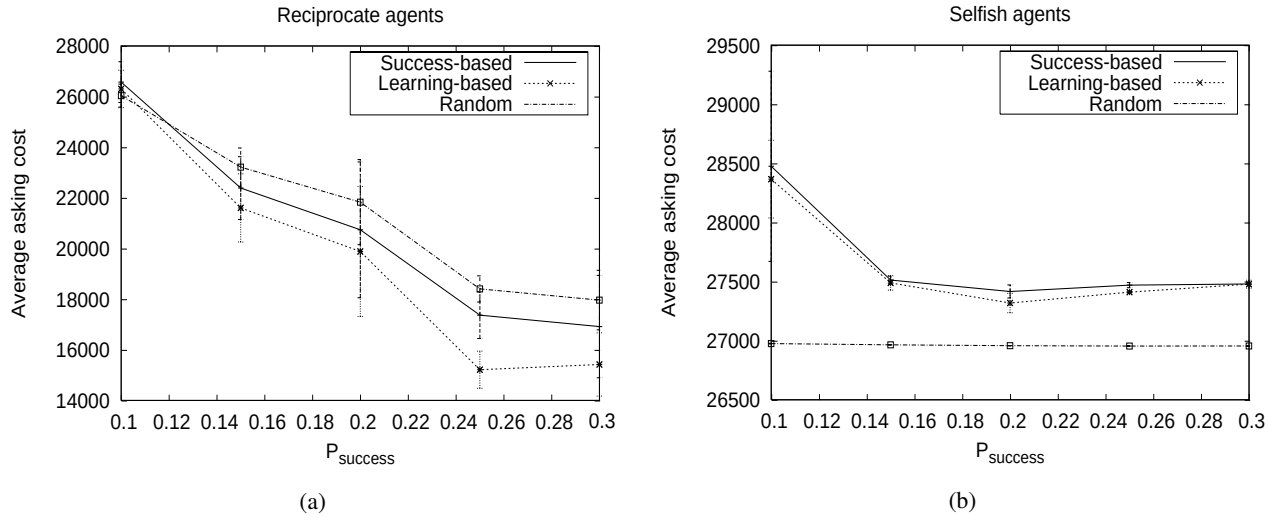

Figure 2: Average Asking Cost for (a) reciprocative and (b)selfish agents with varying $P_{success}$; # Tasks=3000, $P_{leaf} = 0.5$, # Runs=10.

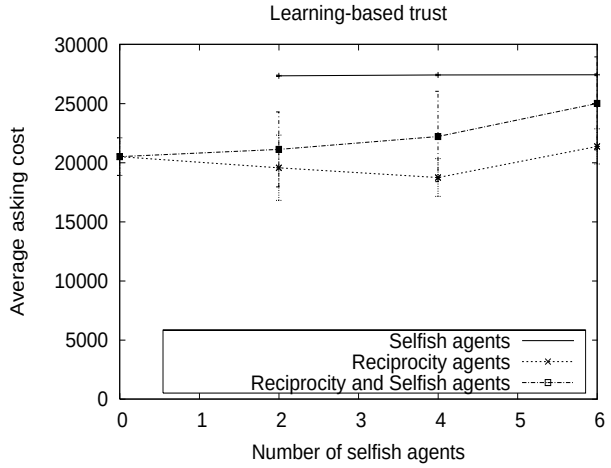

Figure 3: Average Asking Cost with Varying Number of Selfish Agents; $P_{success}$=0.5, # Tasks=3000, # Runs=10, # Agents=10.

From Figure 3 we observe that reciprocative agents are able to consistently perform well until the percentage of selfish agents increase beyond 40% of the agent population, when their performance slightly deteriorate. The primary reason for this trend is that the number of expert agents willing to help given the set of tasks become small. This observation is consistent throughout the various trust mechanism used.

3.1.3 Groups with heterogeneous trust schemes

We also wanted to study how agents would perform if the trust decision mechanisms were not the same for all agents in the system. We ran experiments with 11 helpful agents where the learning and success-based trusting schemes were used by four agents each and the remaining three agents randomly selected agents to ask for help. The results are presented in Table 2. Using the ratio of asking cost to success we find that the learning-based trust schemes performs better than the other algorithms in the system even though they have the lowest number of successes. The differences in success rate, however, are not statistically significant.

4. RELATED WORK

Mulitagent systems are studied to help us understand how agents should behave in the presence of other agents. These interactions might be either cooperative, competitive or simply co-existential in nature. With the need to coordinate their actions, agents should be able to communicate with one another and learn from their interactions. The issue of effective coordination has been an on-going concern in the multiagent systems community. Researchers have worked on developing effective communication protocols and efficient learning algorithms in different social environments. One of the most challenging issues in open multiagent environment is the issue of trust and reputation among agents [5, 14, 17, 19, 11, 7]. Therefore, one of the critical research issues in multiagent systems involves how we learn to trust other agents and how we build and maintain reputations in an agent society.

The application of learning to the problem of trust and reputation management has received increasing attention from researchers in

Fun. Type	Asking Cost		Successes		Help offered		$\frac{Askingcost}{Success}$
	Ave	*Std*	*Ave*	*Std*	*Ave*	*Std*	
Success-Based	13038.6	66.9	2520.5	17.6	1610.2	535.2	5.173
Learning-Based	10608.3	747.8	2440.7	76.6	3518.2	2301.8	4.346
Random	15506.4	386.2	2498.7	40.7	2275.8	1412.7	6.206
Combined	12827.9	2002.5	2485.5	62.6	2485.5	1803.7	5.161

Table 2: Averages and Standard Deviation, Agent= helpful, Function Type= mixed group (4-success-based, 4-learning, 3-random), $P_{success}$=0.2, P_{leaf}=0.5, # Tasks=3000,# Runs=10.

intelligent systems and game theory [2, 4]. We now discuss some learning algorithms that have been used for this purpose in multiagent systems and some representative applications of these learning algorithms.

Fullam [9] has shown how environmental rewards can be used to learn comprehensive trust strategies. Q-learning is a well known reinforcement learning approach that associates actions with expected rewards (that is the difference between decision benefits and decision cost)[18] . In Fullam's work, the loss incurred by interacting with unreliable agent is mitigated by first obtaining reputation information of the agent. Therefore, the agent only losses reputation cost rather the cost of goods/services exchanged, which is much higher than reputation cost. Her work also identifies interdependencies which exist since agents can influence each other by exchanging reputation. To reduce such interdependencies, assumptions are introduced and rewards are attributed to the decision which facilitates reinforcement learning of trust decision strategy.

Other researchers, such as Sen [15], showed that adaptive probabilistic reciprocity strategies can be used to develop and sustain trust and cooperation between self-interested agents. Even though reciprocity does not address the problem of task specific learning we are interested in, it shows how trust relations between agents can be developed and sustained in order to facilitate efficient decision making and identify exploitative agents in a system. Sen's work showed that this probabilistic reciprocity scheme generates stable and cooperative relationships between self-interested agents with a fair distribution of the workload. Such reciprocal exchanges also improve both individual and group performances in the long run.

Our work could be seen as a blend of Fullam and Sen's work. We used the adaptive reciprocity strategy by Sen to develop and maintain trust relationship between agents, with emphasis on task specific relationship. We also used learning to develop trust models which agents use as a basis for initiating interaction with other agents.

In competitive situations such as negotiation and bargaining, agents interact without complete information about others preferences. They do not know what offers are acceptable to other agents or other options that might be available to them. Given such a scenario, an agent can benefit from learning its opponent's decision model, which can be effectively used to optimize its utility. Saha *et al.* [12] study a negotiation framework similar to the screening game [6], a bargaining and negotiation scenario. In the screening game, one agent has full information about the item being negotiated and another agent, without this information, makes an offer which may or may not be acceptable to the first agent. Saha *et al.* use a buyer-seller environment. The buyers' are constrained by incomplete information of sellers' decision policy but are required to make an offer that is equally acceptable to both parties. The authors developed an efficient on-line model learning algorithm using a family of orthogonal polynomials. The algorithm can incrementally develop better models with increasing observations and will converge to the actual model given infinite samples from the decision function. They were able to show that, for most environmental settings, this modeling approach outperforms representative bargaining heuristics such as risk seeking buyers, step buyer, etc.

Carmel and Markovitch [3] used a model-based approach where an opponent's strategy is modeled as a finite automaton and learned from experience. The drawback to this approach is that it is NP-hard to find the smallest finite automata consistent with a set of examples. They propose a heuristic algorithm for learning the minimal automata consistent with the opponent's behavior by using the "Ockham's razor" principle.

Learning to cooperated in iterated games have been studied widely in the game theory and social sciences literature [1, 8, 10]. These approaches, however, typically address repetitions of stylized, abstract games and does not support task descriptions and task-specific interaction decisions which is the focus of the current work.

In most multiagent systems, agents lack complete knowledge of the environment as well as preferences, strategies, etc, of other agents. Even with incomplete information agents must be able to maximize their utility. In our work each agent in the environment have skills that other agents are not initially cognizant of. But these agents must be able to discover models of the capabilities of other agents based on interaction experiences and then use these models to optimize their performances.

5. CONCLUSIONS AND FUTURE WORK

We are interested in enabling agents to learn to form trusted, cooperative relationships with other agents in stable environments. While previous work in MAS addressed forming complementary coalitions, we addressed the problem of quickly identifying collaborators based on specific task attributes. When cost for asking help can be time-consuming or costly, it is necessary for agents to order probable collaborators by the likelihood of their being willing and capable of helping with a given task type. One could develop a "model-based" approach which separately learns, for each agent, its expertise for different task types and its social attitude towards providing help. We preferred a "model-free" approach which combines these two decisions and learns to predict the probability that a given agent will agree to help with a particular task. Such a trust decision scheme can be developed and enriched incrementally from experience and can significantly reduce communication cost and time taken to find useful collaborators in a multiagent society.

We compare our task-attribute based trust decision learning scheme with a success-based approach and a random selection procedure. We conducted series of experiments to observe the performances of these agents in different environment settings and we highlight the following conclusions:

- Helpful agents generally perform better than agents with other social attitudes in homogeneous groups. This difference is highest when the trust mechanism used is learning-based.

The drawback to this social attitude is that it will perform poorly in open groups.

- Selfish agents have very limited success in obtaining help from reciprocative agents and their success percentage is independent of the trust mechanism. These agents tend to perform slightly better when the trust mechanism is random.

- Reciprocative agents are able to perform consistently well in groups with mixed social attitudes until the percentage of selfish agents increases beyond 40%. Their performance improvement with increasing competencies of agents in the society is more than that of selfish agents.

- Learning-based approach dominates all other trust decision mechanisms while reciprocity social attitude dominates selfish.

In this paper we have used myopic selfish agents who never help and are therefore more easily identified and ignored by reciprocative agents. More sophiticated selfish agents may help infrequently to avoid social backlash. Correspondingly more sophisticated reciprocative strategies can be devised to identify such smarter selfish agents. While the space of possible selfish and reciprocative strategies are infinite, we can easily design reciprocative agents with provable, bounded worst-case behavior. The actual system dynamics and agent performances depend on the mix of strategies used in the system.

In the future we intend to use more sophisticated learning mechanisms to reduce the cost of locating collaborators. In particular, more comprehensive learning schemes that can handle continuous and nominal attributes, needs to be used for practical scenarios. We plan to incorporate referrals and other information sources besides direct experience in the trust decision scheme. An interesting avenue would be to evaluate the effect of differing tasks costs from the requesting and asked agents' perspectives. The scale-up properties of our model and the effect of dynamic changes in the agents' capabilities can be studied.

Acknowledgment: US National Science Foundation award IIS-0209208 partially supported this work.

6. REFERENCES

[1] R. M. Axelrod. The evolution of cooperation. In *BASIC Books*, 1984.

[2] B. Banerjee and S. Sen. Selecting partners. In *Game Theory and decision theory in agent-based systems*, pages 29–42. Simon Parsons, Piotr Gmytrasiewicz and Michael Wooldridge, Kluwer, 2002.

[3] D. Carmel and S. Markovitch. Opponent modeling in multi-agent systems. In *Adaptation and Learning In Multi-agent Systems*, volume 1042 of *Lecture Notes in Artifi cial Intelligence*, pages 40–52. Springer-Verlag, 1996.

[4] D. Carmel and S. Markovitch. Exploration strategies for model-based learning in multi-agent systems. *Autonomous Agent and Multi-agent Systems*, 2(2):141–172, 1999.

[5] C. Castelfranchi and R. Falcone. Principles of trust for MAS: Cognitive autonomy, social importance, and quantification. In *Proceedings of the Third International Conference on Multiagent Systems*, pages 72–79, Los Alamitos, CA, 1998. IEEE Computer Society.

[6] A. F. Daughety and J. F. Reinganum. Economics theories of settlement bargaining. *Annual Review of Law and Social Sciences*, 1:35–59, 2005.

[7] S. D.Ramchurn, D. Huynh, and N. R. Jennings. Trust in multiagent system. *The Knowledge Engineering Review*, 19(1):1–25, 2004.

[8] D. Fudenberg and K. Levine. *The Theory of Learning in Games*. MIT Press, Cambridge, MA, 1998.

[9] K. Fullam. Learning complex trust decision strategies. In *Autonomous Agent and Multiagent Systems Conference*, page 1241, 2006.

[10] R. Hegselmann and A. Flache. Understanding complex social dynamics: A plea for cellular automata based modelling. *Journal of Artifi cial Societies and Social Simulation*, 1(3), 1998.

[11] T. D. Huynh, N. R. Jennings, and N. R. Shadbolt. An integrated trust and reputation model for open multi-agent systems. *Journal of Autonomous Agents and Multi-Agent Systems*, 13(2):119–154, 2006.

[12] S. Saha, A. Biswas, and S. Sen. Modeling opponent decision in repeated one-shot negotiations. In *Proceedings of the Fourth International Conference on Autonomous Agent and Multiagents System*, pages 397–403, 2005.

[13] S. Saha, S. Sen, and P. S. Dutta. Helping based on future expectations. In *Proceedings of the Second International Joint Conference on Autonomous Agents and Mulitagents System*, pages 289–296, 2003.

[14] M. Schillo, P. Funk, and M. Rovatsos. Using trust for detecting deceiptful agents in artificial societies. *Applied Artifi cial Intelligence*, 14:825–848, 2000.

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

[16] S. Sen. Believing others: Pros and cons. *Artifi cial Intelligence*, 142(2):179–203, December 2002.

[17] S. Sen and N. Sajja. Robustness of reputation-based trust: Boolean case. In *Proceedings of the First Intenational Joint Conference on Autonomous Agents and Multiagent Systems*, pages 288–293, New York, NY, 2002. ACM Pres.

[18] C. Watkins and P. Dyan. Q-learning. *Machine Learning*, 8:279–292, 1992.

[19] B. Yu and M. P. Singh. Distributed reputation management for electronic commerce. *Compuational Intelligence*, 18(4):535–549, 2002.

Dynamic Trust Formation in Multi-Agent Systems

Dr Robert Ghanea-Hercock
BT Group
BT Labs, Adastral Park
Ipswich, IP53RE
(+44)1473 605486
Robert.ghanea-hercock@bt.com

ABSTRACT

In large-scale multi-agent systems (MAS) the ability of such agents to form coalitions of trusted partners and reputation networks is vital to autonomous operation. This paper investigates the dynamic formation of cooperative communities within a simulated MAS. In particular this work considers the requirements for stable high-trust coalitions to self-organise and survive, while some percentage of agents in the population is defecting. A model of dynamic group formation is presented which enables the rapid formation of a self-organising cooperative agent community. The results presented indicate that by utilizing a self-reinforcing cooperative trust model, a very high degree of resilience to perturbation and defection can be achieved. A number of critical parameters are investigated which indicate a phase transition in the formation and resilience of the cooperative structures that emerge.

Keywords: Autonomous Agents, Trust, Adaptive Behaviour

1. INTRODUCTION

In any distributed MAS whenever an agent receives a message from another agent there is the question of to what extent it can trust the information or intention of that agent. It may be reasonable to assume that the sending agent belongs to a trusted community and has an authenticated identity. However the information it supplies may be unreliable, or the agent may have been compromised, or it may have decided to defect during this transaction. We define defection as any action that is detrimental to the utility of the recipient agent.

The aim of this paper is firstly to investigate the possible strategies a MAS community may adopt in order to build a community of cooperating agents with a high degree of communal trust. The second aspect is to determine what processes or strategies assist in reducing the impact or cost of defecting agents.

The application domains of most interest are those with open communities of agents or where the reputation of the agent has commercial value and is therefore an incentive for agents to distort their reputation in the community. This is of value as many MAS now include open systems such as peer-to-peer applications [Aberer & Despotovic 2001]. In the majority of multi-agent applications there is the need for a distributed community of agents to establish networks of trust and reputation [Yu & Singh 2003].

A significant body of work has also emerged in agent based network authentication systems [Helmer et al. 1998], and reputation development between agent brokers in e-commerce systems [Braynov & Sandholm 1999]. In related work the fundamental question has been raised of what level of resources an agent can afford to allocate to the task of determining another agent's reputation [Ghanea-Hercock 2004].

1.1 The Problem

While agent reputation systems [Abdul-Rahman & Hailes 2000], and [Yu & Singh 2002] have been extensively reviewed there is still a lack of understanding in terms of how processes of trust and reputation evolve dynamically over time. In addition there is an implicit and mistaken assumption that bolting on some classic security mechanisms will make a MAS secure. This work therefore follows the concept of social processes of trust construction as set out by Yu and Singh:

"Social mechanisms complement hard security techniques (such as passwords and digital certificates), which only guarantee that a party is authenticated and authorized, but do not ensure that it exercises its authorization in a way that is desirable to others." [Yu and Singh 2003]

What is required are mechanisms for filtering agents that have defected out of the population, or reducing the frequency with which they are allowed to interact with agents that cooperate. This work assumes that for any agent there is a variable probability that it may defect or

cooperate. The agent decides which action to take at each step based on its expected payoff from that action. This follows the extensive work on spatial iterated prisoners' dilemma and similar research [Nowak & May 1992].

1.2 Trust

This section outlines what is meant by trust in the context of software agents. Much work has addressed the notion of trust in computational systems, such as [Marsh 1999], [Castelfranchi & Falcone 1998], or the early work on evidence based reasoning [Shafer 1976]. One working definition of trust has been suggested by Marsh:

"…trust, (or symmetrically, distrust) is a particular level of the subjective probability with which an agent will perform a particular action, both before he can monitor such action (or independently of his capacity to monitor it) and in a context in which it affects his own action." [Marsh 1999]

A parallel body of work is the study of agent group formation managed by processes of trust, such as that identified by [Griffths & Luck 2003]. They suggest a clan style agent collective process, within which agents experience mutual benefit. A second important approach is that of [Jonker & Treur 1999] which considers the dynamic evolution and adaptation of trust within individual agents. For example:

"*Each event that can influence the degree of trust is interpreted by the agent to be either a* ***trust-negative*** *experience or a* ***trust-positive*** *experience. If the event is interpreted to be a trust-negative experience the agent will lose trust to some degree, if it is interpreted to be a trust-positive, the agent will gain trust to some degree*". [Jonker & Treur 1999]

This work is similarly concerned with how an agent dynamically modifies its own trust perception of social interactions and events. This adaptive shift in an agent's degree of trust forms the basis for the trust algorithm presented in section 3 of this paper.

1.3 Hypothesis

The primary hypothesis of this work is that for large-scale MAS to become resistant to defection there is a critical threshold for the size of cooperating clusters of agents. In order for a set of agents to achieve a stable cooperative cluster they require two parallel processes, as follows:

i. Each agent dynamically updates its internal measure of trust for every agent it interacts with, using the rules defined in section 3.

ii. Each agent requires sufficient internal memory of recent interactions to remember if a specific agent is likely to cooperate or defect.

This concept is elaborated upon in section 2. Of course any MAS first require mechanisms that enable the formation of cooperating groups. In related work [Ghanea-Hercock 2004] used a tag based selection scheme in order to enable agents to determine whether to trust other agents with minimal cost as the criteria. (Where a tag is defined as any unique attribute displayed by an agent [Holland 1993]). (See [Riolo 1997] for a broader discussion of tag processes in agent systems.)

It was proposed that it is economically efficient for most agents to utilise a "*passive trust*" process. By this is meant that an agent preferentially selects which agents to interact with on the basis of a similarity metric applied to the other agent. In contrast if an agent has access to a centralized or global resource, then it would normally be required to perform an active trust assessment of every agent that requests an interaction. In this work no tag selection process is used and the agents use a dynamic trust threshold process to determine which agents to interact with, i.e. trust is reinforced by cooperative interactions, and degraded by defections. This simplifies the model and removes the option for agents to forge useful tags.

Section 2 describes the background theory to trust formation in agent systems. Section 3 covers the specific model developed for this work. Sections 4 and 5 outline the results and conclusion from this model.

2. THE CONSTRUCTION OF TRUST

In general an agent has three methods for establishing to what degree another agent can be trusted. It can request an evaluation by a trusted third party (TTP) or referral agency, or query a number of agents for reputation information. Thirdly it can build an internal model of the trust status of an agent based on its own memory of past interactions and predicted future behaviour.

Any internal assessment normally requires a long term memory of previous interactions with the agent and/or information from N other agents that may have referral information regarding the trust status of the agent in question [As suggested in the work by Yu & Singh 2002]. They also point out the cost issues associated with an agent making such a judgment. The implication is that trust operates on a macro-scale in determining the strength of an interaction process between agents. In the case of a distributed peer-to-peer (P2P) model [Aberer & Despotovic 2001], pervasive computing (e.g. swarm agents [White & Paguerk 1999] or smart dust [Hill et al. 2000]), a centralized TTP may not be available, hence the agents are forced to rely on locally available knowledge.

2.1 Related Work

The primary model for this class of problem has been the spatial iterated Prisoners dilemma. This assumes a local interaction model in which each individual plays the prisoner's dilemma with his or her neighbors, [Nowak and May 1992]. Using a spatial model in which local interactions occur between individuals occupying neighboring nodes on a square lattice, then stable population states for the prisoner's dilemma depends upon the specific form of the payoff matrix. Depending on the payoff matrix values the dynamics of local interaction can lead to a world constantly in flux or equilibrium. In the case of dynamic parameter values, regions occupied predominantly by Cooperators may be successfully invaded

by Defectors, and areas occupied by Defectors may be successfully invaded by Cooperators. In this state, there is no "stable strategy" in the traditional dynamical sense.

Related work on models of social dynamics using cellular automata have adopted a “Bounded Confidence model” [Hegselmann & Krause 2002]. In this model an agent trusts in the opinions of some selected group of agents, (usually neighbouring cells), and adopts the average opinion of the group. Such a model will tend to form emergent clusters of opinion and this process is reflected in the cooperative clusters of high trust that form as a result of the dynamics presented in section 3.

3. DYNAMICS OF TRUST

An agent’s interaction in any complex environment requires a continuous reassessment of the degree of trust it should assign to external agents and events. The underlying assumptions made in this paper are:

a) Trust is assigned as a continuous variable internal to each agent.
b) Positive events result in an agent increasing its degree of trust.
c) Negative events result in an agent decreasing its degree of trust.
d) Agents apply a threshold parameter to determine whether to trust another agent.
e) Agents shift their probability of cooperation or defection based on the *expected behaviour* of the majority of its neighbours, i.e. if the majority of neighbours play defect then each agent will increase the probability that it defects, and the same for cooperation.

3.1 Experimental Model

This section describes the simulation model used to test the hypothesis of the paper. Using the REPAST agent simulation platform [http://repast.sourceforge.net] a population of agents was created with the following attributes:

- Vision – integer range of local cells the agent can perceive and directly interact with.
- Credit parameter – the value of an agent’s current energy parameter.
- Resource – an integer parameter defining a tradable commodity.
- Metabolism – agents consume credit at a rate specified by the metabolism.
- Trust threshold parameter – defined as the degree to which an agent will trust agents within its vision.
- Probability to defect – probability that agent will defect, range [0.0 – 1.0].
- Probability to cooperate - probability that agent will cooperate, range [0.0 – 1.0].
- Mailbox of received messages – an internal list of recently received messages to allow asynchronous exchange between the agents. Also acts as a memory of recent interactions with other agents.

The use of an asynchronous messaging process allows for more flexible interaction patterns, such as agents moving in the environment or irregular spatial topologies. The simulation selects a number of agents at random each time step and calls the execution method of each agent. The rules applied by an agent are:

i) Create a new message requesting a resource and an offered credit price for the resource.
ii) Broadcast the message to all agents within the neighbourhood defined by its vision.
iii) Parse all messages from the mailbox and respond.
iv) For each message, if the predicted trust rating of the sending agent is higher than the internal trust threshold, of this agent then interact, else ignore message.
v) If interaction occurs then select to play cooperate or defect, based on internal probabilities for each.
vi) If agent cooperates the sending agent increases its degree of trust for future interactions.
vii) Else if agent defects the sending agent decreases its degree of trust for future interactions.
viii) If an agents credit or resources is < 0 then reset the agent to an initial random state.

Clearly the iteration of this process can lead to cycles of positive or negative feedback for each agent, which leads to either a global low or high trust regime. This final state depends strongly on the initial conditions, as discussed in the following results section. (It would also be interesting to use a non-uniform network topology to determine whether the topology of the process has a major effect. Related work on trust dynamics by [Yu & Singh 2002] has considered the evolution of trust on small-world type networks.)

3.2 Definitions

The following parameters are applied to each agent:
$T_i(j)$ is the trust rating of agent i for agent j, [0 : 1.0]
α is a fixed increment made to increase $T_i(j)$ if agent j cooperates.
β is a fixed increment made to decrease $T_i(j)$ if agent j defects.

If $\alpha > \beta$ then there is a bias towards cooperative behaviour, which leads to larger clusters of cooperating agents. In the following experiments $\alpha = 0.02$, and $\beta = 0.01$. Stable groups will form when $\alpha = \beta$ but the simulation time required increases by an order of magnitude.

P_d is the probability of an agent choosing to defect
P_c is the probability of an agent choosing to cooperate
γ is an incremental parameter applied to P_d if the majority of an agents neighbours play defect
η is an incremental parameter applied to P_c if the majority of an agents neighbours play cooperate.

Hence the agents become more trusting if the majority of interactions are cooperative and more likely to defect if the majority of recent interactions were defections. For the work in this paper $\gamma = \eta = 0.001$.

T_h is the internal threshold used by the agent to determine whether to interact with other agents.

If $T_h < T_i$ (j) then agent i can interact with agent j else no interaction occurs for this specific transaction.
If the interaction is allowed then each agent selects to cooperate if $P_c > P_d$, or defect if $P_c < P_d$.

Clearly if agents experience a sequence of cooperative interactions then they will tend to form stable high-trust groups and otherwise will move rapidly to a low trust state with no stable groups of trusting agents. In the following results section we see this displayed with strong bi-modal distributions of cooperating and defecting agents; (where the equilibrium state is strongly dependent on the systems initial conditions.)

4. RESULTS

A number of experiments were performed using the specified model in the REPAST agent simulation tool. A population of 400 agents was initialized in a regular grid, with the parameters defined in section 3. The vision and metabolism parameters were both set to 1, the internal threshold parameter T_h is set to a neutral value of 0.5. The simulation colour-codes each agent according to the value of T_h for visualisation purposes, (see figure 1). The minimum group size in this model is a reflection of the cellular neighbourhood dimensions, i.e. nine agents if the vision parameter = 1.

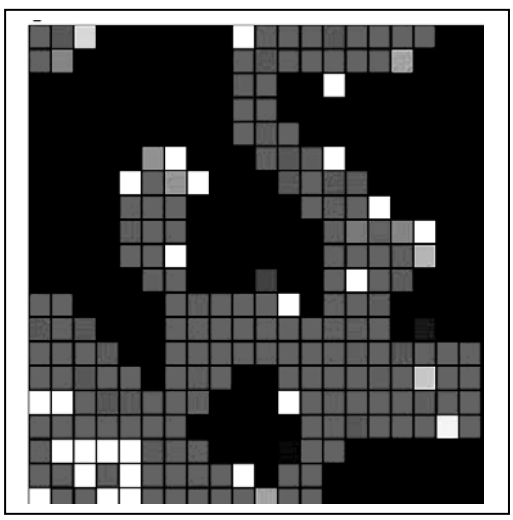

Figure 1 Screenshot from the REPAST simulation, showing an equilibrium state with several clusters of cooperating agents (shades of light grey), surrounded by agents playing defect (black).

A range of experiments were performed using different initial values for the update parameters α and β. In addition, alternative update strategies for the trust and threshold parameters were tested. The evolution of the agents average trust threshold is illustrated in figure 2.

In this set of runs the population converges on quite different trust regimes from almost identical initial conditions, i.e. small variations in the ratios of α and β have a significant impact on the equilibrium state.

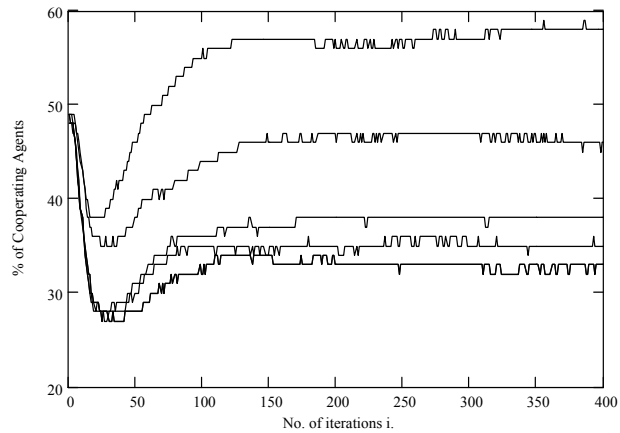

Figure 2 The equilibrium state for a series of five runs of the simulation, showing average percentage of cooperating agents; for a population of 400 agents after 1000 time steps. :(fixed random number seed).

4.1 Group Resilience

The key result obtained from this method of cooperative social group formation is the degree of resilience and resistance to defection achieved. Based on the dynamical model applied in this simulation, the emergence of cooperative clusters was expected and correlates well with related work. A series of experiments were then applied to the set of agents after such clusters had emerged in order to determine the degree to which they could resist defection by internal and external agents.

The response of the system is illustrated in figure 3 with identical initial conditions and a sequence of six runs. For each run a step increase (after some time t) in the percentage of defecting agents was applied by randomly selecting p % of the agents and setting their probability of defection = 1.0. In this sequence p = 25% and the time intervals until the defection increase are: t = 0, t = 20, t = 30, t = 40, t = 70, t = 100.

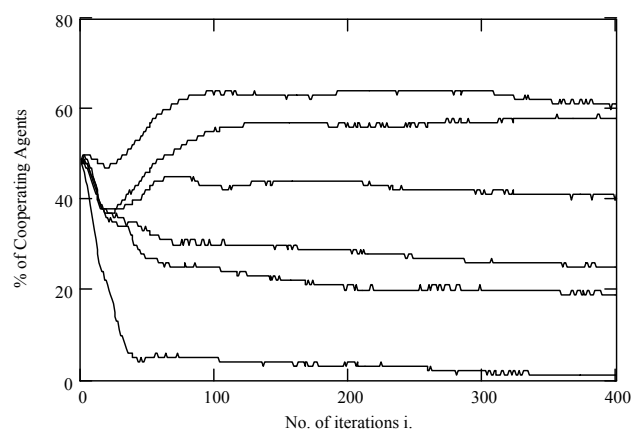

Figure 3. Dynamic evolution of the percentage of cooperating agents after 25% are converted into defecting

agents; at increasing intervals of time from t=0 to t=100 iterations.

What we observe is that a transition occurs at approximately t = 50 iterations, when stable clusters occur which are then resistant to the defection attack that follows. Below this value small clusters of 4-12 cells form but these succumb to the higher defection rate within a few iterations.

This result is summarised in figure 4 which shows the effect averaged over 50 runs of the simulation, as a function of the time to the step increase in defection.

Work by [Ghanea-Hercock 2004] discusses how a related dynamical model i.e. the Ising model [Ising 1925] provides a comparative set of dynamic behaviours for this class of 2-dimenional interacting system. In particular such a model allows an analytical prediction of the critical cluster size threshold, where:

Mean magnetization ≡ avgerage percentage of cooperating agents (i.e. can be compared to aligned spin states)

and:

Spin-spin coupling coefficient J/kT ≡ defection probability.

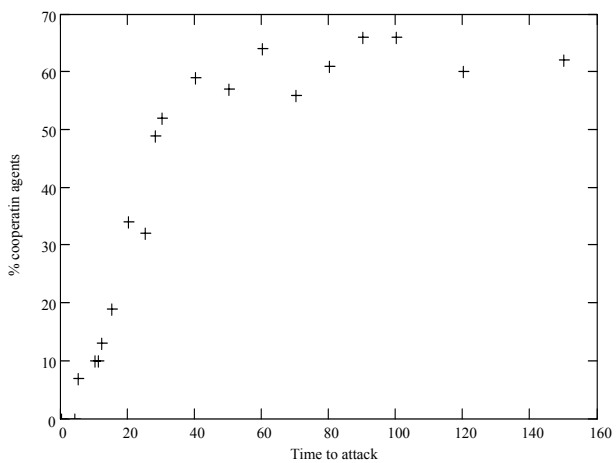

Figure 4. Graph of average percentage of cooperating agents in the population as a function of time to attack: p =25% defection (averaged over 50 runs).

4.2 Future Work

A number of extensions to this work are currently planned. First it would be useful to include the ability to request explicit referrals from other agents. This has been a commonly adopted method in literature on trust in open MAS. It would also be useful to add the ability to bias the predicted trust evaluation of agents using centralized reputation sources when available. Hence an agent could benefit from global and local knowledge sources in making a trust evaluation of another agent. A second area of investigation will be to map the algorithm onto a realistic network topology rather than a regular grid which strongly influences the dynamics exhibited in this class of model.

5. CONCLUSION

The formation of cooperating agent communities is a fundamental requirement for the large-scale deployment of MAS. Previous models of agent security and trust have revolved around assumptions of closed systems with centralized authentication and reputation services. In this work a more flexible approach is adopted that accommodates dynamic variation in the degree of trust allocated to an agent. It also allows an agent to assign some low initial value of trust to a new agent wishing to interact with it that has no prior available reputation.

The key behaviour of the agents to enable them to form cooperative groups is that they shift their probability of cooperation or defection based on the *expected behaviour* of the majority of its neighbours, i.e. if the majority of neighbours play defect then each agent will increase the probability that it defects, and the same for cooperation.

The result is that a group of agents using such a dynamic trust model becomes highly resistant to defection by external or internal agents, once a sufficiently large cooperative group has formed. However, below the critical group size the agents are susceptible to defection strategies, which then dominate the system. Future work will explore how this behaviour maps onto a realistic social network of agents, and investigate the exact mechanisms by which the mutual reinforcement process operates.

It is interesting to speculate that this result appears to reflect the nature of human cooperative group formation and confirms the expression: "*united we stand, divided we fall*".

Acknowledgements

This work was supported by the award of a Business Network Fellowship from the Santa Fe Institute, NM, USA, (http://www.santafe.edu).

6. REFERENCES

[Abdul-Rahman A., and Hailes S., 2000] Supporting Trust in Virtual Communities. In: *Hawaii Int. Conference on System Sciences 33*, Maui, Hawaii, 2000.

[Aberer K., and Despotovic Z., 2001] Managing trust in a Peer2-Peer information system. In Henrique Paques, Ling Liu, and David Grossman, ed., *Proc. of the Tenth International Conference on Information and Knowledge Management* (CIKM01), pages 310-317, New York, November 5-10, 2001.

[Braynov S., and Sandholm T., 1999] Contracting with Uncertain Level of Trust, *Proceedings of the first ACM conference on Electronic commerce*, November 3-5, 1999.

[Castelfranchi C., and Falcone R., 1998] Principles of trust for MAS: Cognitive anatomy, social importance, and quantification. In *Proceedings of the Third International Conference on Multi-Agent Systems* (ICMAS-98), pages 72–79, Paris, 1998.

[Ghanea-Hercock R., 2003] Phobos: An Agent-based User Authentication System. *IEEE Intelligent Systems*, 18(3): 67-73 (2003), June 2003.

[Ghanea-Hercock R., 2004] The Cost of Trust. *Seventh International Workshop on Trust in Agent Societies*, held at Autonomous Agents & Multi-Agent Systems Conference New York, 2004.

[Griffiths N., and Luck M., 2003] Coalition Formation through Motivation and Trust. *International Conference of Autonomous Agents and Multi-Agent Systems*, Melbourne, Australia, 2003.

[Hegselmann R., and Krause U., 2002] Opinion dynamics and bounded confidence: models, analysis and simulation. *Journal of Artificial Societies and Social Simulation* vol. 5, no. 3, http://jasss.soc.surrey.ac.uk/5/3/2.html, 2002.

[Helmer G.G., Wong J.S., Honavar V., and Miller L. 1998] Intelligent agents for intrusion detection. In *Proceedings, IEEE Information Technology Conference*, pages 121-124, NY, September 1998.

[Hill J., Szewczyk R., Woo A., Hollar S., Culler D.E., and Pister K., 2000] System Architecture Directions for Networked Sensors. *Architectural Support for Programming Languages and Operating Systems*, pages 93-104, 2000.

[Holland J., 1993] The Effects of Labels (Tags) on Social Interactions. *Santa Fe Institute Working papers* 93-10-064, Santa Fe, NM, 1993.

[Ising, E. 1925] Beitrag zur Theorie des Ferromagnetismus. Zeitschr.. f. *Physik* **31**, 253-258, 1925. [German].

[Jonker C., and Treur J., 1999] Formal analysis of models for the dynamics of trust based on experiences. In *Multi-Agent System Engineering, Proceedings of the 9th European Workshop on Modeling Autonomous Agents in a Multi-Agent World*, MAAMAW'99. LNAI 1647, 1999.

[Marsh, S., 1999] Formalising Trust as a Computational Concept. Ph.D. Thesis. *Department of Mathematics and Computer Science, University of Stirling*, http://citeseer.nj.nec.com/marsh94formalising.html, 1999.

[Nowak M.A., May R.M., 1992] Evolutionary Games and Spatial Chaos. *Nature*, **359**(6398), 29 October, pp. 826-829, 1992.

[Riolo R., 1997] The Effects and Evolution of Tag-Mediated Selection of Partners in Populations Playing the Iterated Prisoner's Dilemma, *Proceedings of the Seventh International Conference on Genetic Algorithms* (ICGA97), pub. Morgan Kaufmann, San Francisco, CA, ed. Thomas B., 1997.

[Shafer G. 1976] A Mathematical Theory of Evidence. *Princeton University Press*, Princeton, NJ, 1976.

[Sherrington D., Garrahan J. and Moro E. 2000] Statistical physics of adaptive correlation of agents in a market. cond-mat/0010455, 2000.

[Yu B., and Singh M., 2002] An Evidential Model of Distributed Reputation Management. International *Conference of Autonomous Agents and Multi-Agent Systems*, Bologna Italy, 2002.

[Yu B., and Singh M.P., 2003] Detecting Deception in Reputation Management, *Proceedings of Second International Joint Conference on Autonomous Agents and Multi-Agent Systems*, 2003.

[White T., and Pagurek B., 1999] Emergent Behaviour and Mobile Agents, *Proceedings of the workshop on Mobile Agents in the Context of Competition and Cooperation at Autonomous Agents*, 1999.

Trust dynamics, motivational attitudes and epistemic actions

Mario Gómez
Department of Computing Science
University of Aberdeen

Clara Benac Earle and **Javier Carbó**
Department of Computer Science
University Carlos III of Madrid

Abstract

In this paper we built upon our anticipatory trust model to deal with dynamic environments. We propose two mechanisms: first, a forgetting mechanism to deal with situations where new information about other agents is expensive or scarce, and second, a model based on motivational attitudes and epistemic actions to implement adaptive behaviors. In particular, we have developed and tested experimentally a curiosity-driven exploratory behavior that performed better than other exploratory behaviors across a number of different experimental situations involving different kind of changes in the environment.

Introduction

Trust is a universal concept that plays a very important role in social organizations as a mechanism of social control. Therefore, modeling trust in open distributed systems such as agent systems becomes a critical issue since their offline and large-scale nature weaken the social control of direct interactions. Sometimes there are objective and universal criteria to evaluate the quality of interactions (products/services provided by them). In such a case, trust can be inferred from certificates issued by third parties that verify such objective criteria. Unfortunately, there are many application domains in which the evaluation criteria is subjective (books, films, web pages, leisure activities, consulting services, technical assistance, etc.) and can be obtained only through local interactions. As a consequence, when a set of universal objective evaluation criteria is not available, this subjective and local trust is not easily asserted.

Open distributed systems pose a challenge to trust modeling due to the dynamic nature of these systems (e.g., electronic auctions) and the unreliability of self-interested agents, which may be specially malicious in case of competitive domains. However, most of the frameworks proposed so far assume that the behavior of agents do not change over time, which in many cases is a very unsafe assumption. Although trust modeling frameworks would eventually detect the changes in the environment, they tend to be reactive instead of proactive. In order to adapt more swiftly to the changes in the environment, agents must incorporate anticipatory mechanisms, ie. ability to deal with the future by mental representations or specific forms of learning.

Anticipatory behavior is an interdisciplinary topic attracting attention from computer scientists, psychologists, philosophers, neuroscientists, and biologists. There are fundamental cognitive psychology experiments supporting the existence of anticipatory behavior in animals and humans, and some frameworks to model anticipatory learning and behavior have been already proposed (Butz, Sigaud, & Gérard 2003). Recently, a growing number of researchers have identified and emphasized the importance of anticipation as the basis of models of animal learning and behavior. Anticipatory behavior can be seen as a mechanism for devising hypotheses that make predictions about future events, conducting experiments to corroborate them and subsequently using the knowledge they have gained to perform useful behaviors. Each experiment takes the form of an expectancy or prediction that is either fulfilled, so corroborating the effectiveness of the hypothesis, or it is not fulfilled, thus weakening or denying the hypothesis. Anticipatory principles seem interesting in the context of trust modeling for they define a continuing process of discovery and refinement. This process allows an agent to progress throughout its lifetime, incrementally developing its structures, and so matching its behavior patterns to its environment.

In previous work, we have proposed an anticipatory framework to model trust in open environments(Gomez, Carbo, & Benac 2006) which uses the pattern of discrepancy between information and experience as a mechanism to anticipate the behavior of other agents. Usually, the discrepancy between information and experience is deemed as a source of dishonesty: if an agent a_1 has said that it provides a quality of service with value q, and another agent a_2 has observed a value r, then a_2 views $q-r$ as a degree of dishonesty, a source of distrust. Instead of using this approach, we have proposed a trust model which does not assume a concrete cognitive model for other agents, instead the discrepancy between information and experience as a valuable source of information to learn about other agent's behavior so as to anticipate their actions: $q-r$ is not used as a source of dishonesty, instead, it is used to estimate the quality of service to be obtained in the future. The result is a framework that fosters a swiftly adaptation to the changes in the environment by anticipating those changes before actually

Copyright © 2007, American Association for Artificial Intelligence (www.aaai.org). All rights reserved.

observing them.

Although we have already defined the basic components of trust, and how to aggregate these components to obtain a global trust assessment(Gomez, Carbo, & Benac 2006), we have not explained yet all its dynamics aspects. Typically, trust models describe the way trust is updated as more interactions occur; however, what happens when an agent is not interacting with other agents? Must an agent stick to its trust beliefs after some time without obtaining new evidence or information about other agents? Most of the trust models proposed in the literature skip this question: either because it is assumed that trust beliefs are continually updated with new information, or because it is assumed that agents do not change over time and thus there is no need to modify trust even in absence of new information. However, in some environments the interaction required to obtain information about other agents is expensive or scarce and thus it is not appropriate to assume that an agent is continually obtaining new information. Furthermore, in many situations assuming that the behavior of agents does not change over time is very unsafe. Therefore, we need a mechanism to address this problem.

In this paper we extend and complement our trust framework to deal with dynamic environments in two ways: first, we introduce a forgetting mechanism to address the absence of information; second, we propose a model based on motivational attitudes and epistemic actions to implement more adaptive and proactive behaviors. The structure of the paper is as follows: the following section briefly reviews our anticipatory trust model, then we explain an extension to deal with the absence of information. A complementary framework based on motivational attitudes is described afterwards, followed by an experimental evaluation of the attitudinal framework. Finally, there is a section to sum up our contributions and future work.

Trust Model

A general definition of trust embraces many dimensions, for instance: to have belief or confidence in the honesty, goodness, skill or safety of a person, organization or thing (Cambridge Advanced Learners Dictionary). Herein we adopt an approach centered on the skill dimension: trust as a confidence in the competence, utility or satisfaction expected from other agent concerning a particular context. More specifically, we approach trust as a belief that estimates the quality of service expected from a particular agent, based on both direct experience using that service, and information obtained from other agents. We have chosen the term quality of service –henceforth abbreviated QoS– to sum up our view of trust because it encompasses the notions of utility and interaction that is characteristic of open distributed systems, including multi-agent systems and electronic institutions, grid computing, web services,etc.

Typically, a trust model considers two main sources of information: direct experience, which results in what is usually referred to as direct trust or interaction trust; and indirect experience, which is often referred to as witness-information, "word of mouth" or reputation. In our model we keep this distinction between direct and indirect experience, but in addition, we distinguish between the information provided by third party agents about other agents, what we call *recommendations*, and the information provided by an agent about itself, what we call *advertisements*. All in all, our model builds trust upon three components, namely: Direct Trust (DT), Advertisements-based Trust (AT), and Recommendations-based Trust (RT).

In order to adapt quicker to the dynamic and uncertain nature of an open environment, an agent can anticipate or have expectations (not necessarily rational) about the possible consequences of its actions. In our model, the anticipatory component is based on the pattern of discrepancies that is observed between predictive information and actual experience. This anticipatory component is included in those beliefs that use some kind of indirect information about the Quality of Service (QoS) to be provided by an agent, namely the AT and RT.

To simplify the dynamics of a multi-agent system, we use a discrete time model made up of time steps. A time step represents the minimal time period an agent requires to take decisions, act, and perceive the result of its actions. We use t to denote a particular time step in the past, T for the current time step, $T+1$ for the next time step, and ΣT for an aggregation of historic beliefs until time step T.

To handle uncertainty and ignorance, we use two measures of confidence, namely: *intimacy* and *predictability*. Intimacy is a measure of confidence based on the number of data (or interactions) used to calculate a belief, while predictability is a measure of confidence based on the dispersion or variability of the data. In our model, all the components of trust have attached a measure of confidence made up of intimacy and predictability. Depending on the way we combine these measures of confidence, it may have the meaning of a conjunction or a disjunction, thus, for the sake of making the model as general as possible, we propose to use t-norms and t-conorms to represent these alternative ways of combining the confidence values [1].

The components of our trust model are formally defined below.

Direct Trust($DT_j^{\Sigma T}$): assesses the QoS provided from agent a_j until time step T inclusive, based on direct experience.

$$DT_j^{\Sigma T} = \frac{\sum_{t=0}^{T} \varphi(t,T) pDT_j^t}{\sum_{t=0}^{T} \varphi(t,T)} \quad (1)$$

where $pDT_j^t : \mathbb{R} \rightarrow [0,1]$ is the partial DT obtained for a_j in time step t, and $\varphi(T,t) : \mathbb{N} \rightarrow [0,1]$ is a forgetting function used to weight each partial belief according to its age (number of time steps since a belief was obtained, T-t).

Direct Trust Confidence($DTC_j^{\Sigma T}$): assesses the reliance of DT as an estimator of the QoS provided by agent

[1] Although we have implemented the min and the product t-norms and its conjugated t-conorms (max and algebraic sum respectively), in practice we are using only the product; a comparative study of different operators could be the subject of future research

a_j.

$$DTC_j^{\Sigma T} = ITM_j^{DT} \otimes (1 - \upsilon_{dt}(pDT_j^t)) \quad (2)$$

where $ITM_j^{DT} \in [0,1]$ is the intimacy for DT (Sabater & Sierra 2001), a growing function in [0,1] over the number of pDTs used to compute DT, $\upsilon_{dt} \in [0,1]$ is a measure of the variability of pDT_j^t, and $\otimes$ is a *T-norm* operator.

Advertisements-based Trust(AT_j^{T+1}): assesses the QoS expected from agent a_j in the next time step $(T+1)$, based on the advertisements from a_j.

$$AT_j^{T+1} = \left\{ \begin{array}{ccc} 1 & if & pAT_j^{T+1} + \Delta AT_j^{\Sigma T} \geq 1 \\ 0 & if & pAT_j^{T+1} + \Delta AT_j^{\Sigma T} \leq 0 \\ pAT_j^{T+1} + \Delta AT_j^{\Sigma T} & if & 0 < pAT_j^{T+1} + \Delta AT_j^{\Sigma T} < 1 \end{array} \right\} \quad (3)$$

where $pAT_j^{T+1} : \mathbb{R} \rightarrow [0,1]$ is the most recent advertisement from a_j, and $\Delta AT_j^{\Sigma T}$ (AT-Discrepancy) is the discrepancy between advertisements and experiences obtained in the past (until time step T inclusive).

AT-Discrepancy $\Delta AT_j^{\Sigma T}$: measures the discrepancy between the past advertisements made by agent a_j and the experiences obtained.

$$\Delta AT_j^{\Sigma T} = \frac{\sum_{t=0}^{T} \varphi(t,T)(pDT_j^t - pAT_j^t)}{\sum_{t=0}^{T} \varphi(t,T)} \quad (4)$$

where $pAT_j^t : \mathbb{R} \rightarrow [0,1]$ is the Partial AT for agent a_j, and time step t, and $\varphi(t,T)$ is a time forgetting function.

Note that $\Delta AT_j^{\Sigma T} \in [-1,1]$, since $pDT_j^t, pAT_j^t, \varphi(t,T) \in [0,1]$ by definition Positive values of $\Delta AT_j^{\Sigma T}$ means that the experiences obtained from agent a_j were better than advertised, negative values have the opposite meaning, and zero means that the experiences matched perfectly with the advertisements.

AT Confidence(ATC_j^{T+1}): assesses the degree of reliance on the AT as an estimation of the QoS to be obtained from agent a_j in the next time step.

$$ATC_j^{T+1} = ITM_j^{AT} \otimes (1 - \upsilon_{at}(\Delta AT_j^t)) \quad (5)$$

where ITM_j^{AT} is the intimacy for AT, $\Delta AT_j^t = pDT_j^t - pAT_j^t$ is the partial discrepancy observed between AT and DT in time step t, $\upsilon_{at} \in [0,1]$ is a measure of the variability of ΔAT, and $\otimes$ is a T-norm operator.

As we have done for DT and AT, we define both partial and historic Recommendations-based Trust (RT). However, RT must handle the fact that there are potentially many providers of information (recommenders) about any other agent. As a result, we have to distinguish between the trust component due to the recommendations provided by a single agent and the trust component due to the recommendations provided by several agents, what we call *combined recommendation*.

Recommendations-based Trust(RT_{jk}^{T+1}): assesses the QoS expected from agent a_j in the next time step $(T+1)$, based on the recommendations from agent a_k.

$$RT_{jk}^{T+1} = \left\{ \begin{array}{cc} 1 & pRT_{jk}^{T+1} + \Delta RT_{jk}^{\Sigma T} \geq 1 \\ 0 & pRT_{jk}^{T+1} + \Delta RT_{jk}^{\Sigma T} \leq 0 \\ pRT_{jk}^{T+1} + \Delta RT_{jk}^{\Sigma T} & 0 < pRT_{jk}^{T+1} + \Delta RT_{jk}^{T+1} < 1 \end{array} \right\} \quad (6)$$

where $pRT_{jk}^t : \mathbb{R} \rightarrow [0,1]$ is the partial Recommendations-based Trust for agent a_j obtained from agent a_k, and $\Delta RT_{jk}^{\Sigma T}$ (RT-Discrepancy) is the discrepancy between past recommendations and experiences about agent j.

RT-Discrepancy($\Delta RT_{jk}^{\Sigma T}$): measures the discrepancy between the past recommendations byagent a_k about agent a_j , and the experiences obtained using that service, until time step T inclusive.

$$\Delta RT_{jk}^{\Sigma T} = \frac{\sum_{t=0}^{T} \varphi(t,T)(pDT_j^t - pRT_{jk}^t)}{\sum_{t=0}^{T} \varphi(t,T)} \quad (7)$$

where $pRT_j^t : \mathbb{R} \rightarrow [0,1]$ is the partial Recommendations-based Trust for agent a_j, and time step t, and $\varphi(t,T)$ is a time forgetting function.

Note that $\Delta RT_{jk}^{\Sigma T} \in [-1,1]$, since $pDT_j^t, pRT_j^t, \varphi(t,T) \in [0,1]$ by definition Positive values of $\Delta RT_j^{\Sigma T}$ means that the experiences obtained from agent a_j were better than recommended, negative values have the opposite meaning, and zero means that the experiences matched perfectly the recommendations.

Combined Recommendations-based Trust(cRT_j^{T+1}): assesses the QoS expected from agent a_j in the next time step, based on both historic information and the most recent recommendations about that service.

$$cRT_j^{T+1} = \frac{\sum_{k=1}^{N_k} (RT_{jk}^{T+1} \times RTC_j^{T+1})}{\sum_{k=1}^{N_k} RTC_{jk}^{T+1}} \quad (8)$$

where RT_{jk}^{T+1} is the Recommendations-based Trust about agent a_j based on agent a_k's recommendations, and RTC_{jk}^{T+1} is the confidence on that belief as an estimation of the QoS to be obtained froma_j in $T+1$.

The Combined Recommendations-based Trust aggregates the recommendations obtained from several agents. Simmilarly, the confidence on cRT is defined as an aggregation of the confidences on every recommendation.

RT Confidence(RTC_j^{T+1}): assesses the degree of reliance on the Recommendations-based Trust (RT_{jk}^{T+1}) obtained from agent a_k, as an estimation of the QoS to be obtained from agent a_j in the next time step.

$$RTC_{jk}^{T+1} = ITM_j^{RT} \otimes (1 - \upsilon_{rt}(\Delta RT_{jk}^t)) \quad (9)$$

where ITM_j^{RT} is the intimacy for RT, $\Delta RT_{jk}^t = pDT_j^t - pRT_{jk}^t$ is the partial discrepancy observed between DT and RT in time step t, $\upsilon_{rt} \in [0,1]$ is a measure of the

variability of ΔRT_{jk}^{t}, and $\otimes$ is a T-norm operator.

Combined RT Confidence($cRTC_j^{T+1}$)**:** assesses the degree of reliance on the Combined Recommendations-based Trust as an estimation of the QoS to be obtained from agent a_j in the next time step.

$$cRTC_{jk}^{T+1} = \bigoplus^{k}(RTC_{jk}^{T+1}) \tag{10}$$

where $\bigoplus^k$ denotes the aggregation of the confidence associated to each recommender ($RTC_{jk}^{\Sigma T}$) using a *T-conorm* operator.

We have already defined the components of trust according to the source of information. Now we provide a global measure of trust that integrates the three components into a single belief: the Global Trust.

Global Trust(GT_j^{T+1})**:** assesses the QoS expected from agent a_j during the next time step, using all the sources of information.

$$GT_j^{T+1} = \frac{DT_j^{\Sigma T} \times DTC_j^{\Sigma T} + AT_j^{T+1} \times ATC_j^{T+1} + cRT_j^{T+1} \times cRTC_j^{T+1}}{DTC_j^{\Sigma T} + ATC_j^{T+1} + cRTC_j^{T+1}} \tag{11}$$

where $DT_j^{\Sigma T}$ is the DT for agent a_j; AT_j^{T+1} is the Anticipatory AT; cRT_j^{T+1} is the Combined Recommendations-based Trust, and $DTC_j^{\Sigma T}$, ATC_j^{T+1}, RTC_j^{T+1} are the confidences associated to DT, AT and cRT respectively.

Global Trust Confidence(GTC_j^{T+1})**:** assesses the reliance on the Global Trust GT_j as an estimation of the QoS to be obtained in the next time step.

$$GTC_j^{T+1} = DTC_j^{\Sigma T} \oplus ATC_j^{T+1} \oplus cRTC_j^{T+1} \tag{12}$$

where $\oplus$ is a *T-conorm* operator.

Remark that Global Trust and Global Trust Confidence can be used either independently or combined into a single value (eg. $GT \times GTC$), depending on the specific application domain.

Trust dynamics

In the former section we have described the components trust model basically from an static viewpoint. Although the dynamic dimension is somewhat embraced in the notions of forgetting function and intimacy, the strongly dynamic nature of real open systems advocates for a more complex model to deal with the changes in the environment. Herein, we first review the dynamics in our previous model, and then describe a dynamic extension of the intimacy to deal situations involving scarce or expensive information..

Due to the dynamic nature of open environments, it is common sense to assume that old beliefs, based on old perceptions, are less reliable that those beliefs based on recent perceptions. In consequence, several trust models, including ours, use a *forgetting function* to combine partial beliefs obtained at different times by weighting each belief according to its age. A concrete example of a forgetting function follows, and is depicted in Figure 1.

$$\varphi(t,T) = \begin{Bmatrix} 0 & T-t \geq \phi \\ \cos(\frac{\pi}{2\phi} \cdot (T-t)) & 0 < T-t \end{Bmatrix}$$

where ϕ is a parameter that establishes the maximum age for an experience to be relevant (to influence current beliefs).

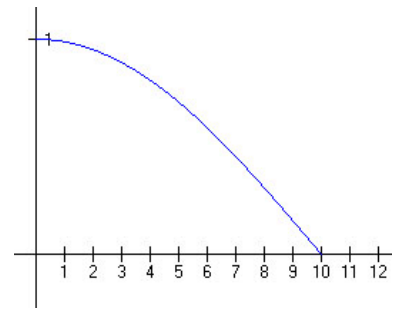

Figure 1: Forgetting function with $\phi = 10$

Nevertheless, the use of a forgetting function to weight each primitive component of a belief according to its age is not enough to account for a rapidly changing environment, actually, it is intended to aggregate the elementary components of a belief (single perceptions), given more importance to the more recent ones, but, what happens when there is no new information? In our previous model of trust there is an implicit assumption: that an agent has information and experience to update its beliefs each time step. However, the former assumption is quite simplistic and idealistic as to be applicable to the real world; on the contrary, in most real situations information is scarce or expensive, specially valuable information. Therefore, in general, we cannot assume that an agent will obtain information and experience continuously and exhaustively, instead we should devise a model to handle scarce information.

We have stated that each trust belief has associated a confidence value with two components: intimacy and predictability. Intimacy, as defined in (**?**; **?**), encompasses dynamics in a very simplistic way: the intimacy is an ever growing function based on the number of data or interactions, until a maximum value is reached, and then intimacy becomes a constant. However, we claim that in order to deal with dynamic environments, intimacy must should also decrease when there is no new information, to reflect the fact that an agent's knowledge become obsolete if it is not updated periodically.

To give a formal definition of the intimacy we need first to define a function that indicates whether a new perception has been obtained for a specific component of trust. Let Φ be any of the basic components of trust, i.e. $\Phi \in \{DT, AT, RT\}$; let $p\Phi$ be the partial value of Φ for a single time step, i.e. $p\Phi \in \{pDT, pAT, pRT\}$; and let $AV(p\Phi, T)$ be a boolean function that is true when $p\Phi$ is instantiated in time step T. Then, the intimacy for Φ in time step T, denoted by $ITM^{\Phi}(T)$, is defined recursively as a function of the intimacy at the previous time step $ITM^{\Phi}(T-1)$, and

$AV(p\Phi, T)$

$$ITM^T(T) = \begin{cases} \Gamma^\Phi & if \quad T = 0 \\ \Delta(ITM^\Phi(T-1)) & if \quad AV(p\Phi, T) = true \\ \Theta(ITM^\Phi(T-1)) & if \quad AV(p\Phi, T) = false \end{cases}$$

where $\Gamma^\Phi \in [0, 1]$ plays the role of a prejudice, that is, a default value used to asses one's beliefs in absence of other source of knowledge; Δ is a growing function $\Delta : [0, 1] \rightarrow [0, 1]$, and Θ is a decreasing function $\Theta : [0, 1] \rightarrow [0, 1]$.

In our experiments we have tried different functions to increment and decrement the intimacy, both additive and multiplicative, and in particular, for the experiments described later in this paper we have used functions of the following type:

$$\Delta(X) = X + \iota\,(1 - X)$$

$$\Theta(X) = \delta\, X$$

where ι and δ are coefficients in $[0, 1]$ (in addition we ensure that $ITM^\Phi \in [\Gamma^\Phi, 1]$).

Figure 2 depicts the kind of functions we are using; the functions at the top of the figure are examples of intimacy increment functions with different ι values (ranging from 0.2 to 0.5) , while the functions at the bottom are intimacy decrement functions with different δ values (ranging from 0.6 to 0.9). As new information is being received each turn, intimacy grows, tending to 1, and when no new information is received, the intimacy decreases. In general, we think that the increment pace for the intimacy should be considerably faster than the decrement pace. The idea here is that the occurrence of new information has a much stronger impact on the intimacy than the absence of information; which is suggested by the way human memory works. Anyway, we are not stating here that some specific functions are better than others to model the dynamics of the intimacy; instead, we advocate experimenting with different functions, as far as the dynamics of the intimacy follow this growing/falling mechanism based on the presence/absence of new information.

Motivational attitudes and epistemic actions: a curiosity-driven model for exploratory behavior

In open systems agents may enter, exit or change their behavior at any moment. Thus, if an agent needs the services provided by other agents or can obtain benefits by delegating some task, and there is scarce information about the others, then that agent is motivated to conduct some kind of exploratory behavior to keep its knowledge updated. When an agent does not interact with other agents its social knowledge may become obsolete, not reflecting the current state of the world. Although an agent can use reputation information to know other agents without direct interaction (since reputation usually is more abundant and cheaper than direct interaction), reputation itself is exposed to the same risk of becoming obsolete as time goes by. Therefore, some kind of exploratory behavior is mandatory if an

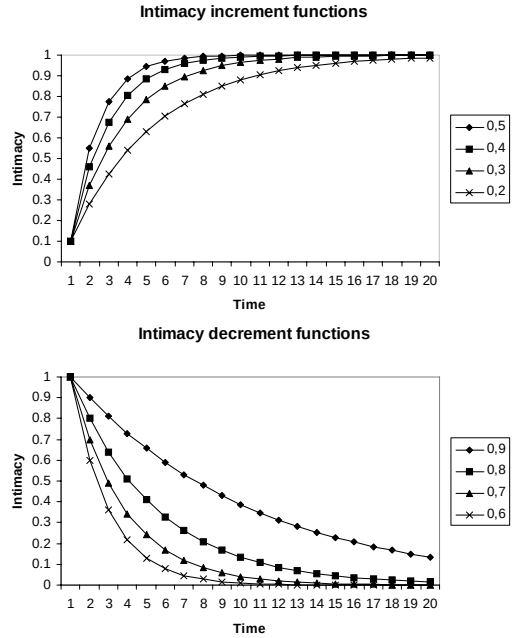

Figure 2: Example of intimacy increment and decrement functions

agent wants to know which are the best providers of a service at any moment. Hence, we have devised a mechanism to implement exploratory behaviors based on the use of *epistemic actions*, ie. actions aimed at acquiring knowledge from the world. To be more specific, we propose a framework to implement trust-oriented exploratory behaviors based on curiosity-driven epistemic actions(Lorini & Castelfrnachi 2004)[2].

Herein we focus on the skill dimension of trust, defined as the QoS expected from an agent; therefore, we approach trust as a belief that aims at taking decisions on whether to request a service to other agents. In this context, we can classify service requests according to the underlying purpose of the request, namely *pragmatic* or *epistemic*: on the one hand, an agent may request a service to another agent because it expects to obtain some utility based on its previous knowledge; on the other hand, an agent may request a service just because it is willing to know that agent better. Note that the ultimate goal of knowing other agents is probably to maximize some global utility by exploiting that knowledge, but in the short term, a purely epistemic action is performed just to know, even with an associated cost, and not because a real benefit is expected to follow immediately from that action.

In order to develop intelligent exploratory behaviors, we have developed a model based on motivational attitudes. However, instead of adopting the binary-logic approach that

[2]Curiosity-driven epistemic action are driven by the meta-goal of getting new knowledge, in opposition to goal-driven epistemic actions, which aim at testing a specific belief involved in the processing of a given goal.

is usual in the agent community –that of logic beliefs, desires and intentions–, we adopt a continuous approach to representing motivational attitudes, using variables in $[0, 1]$. In particular, we propose a model comprising four basic attitudes as the main forces driving the epistemic behavior of an agent, namely: *necessity*, *knowledge/ignorance*, *satisfaction*, and *curiosity*.

Necessity Assessment of the potential benefit that an agent could obtain by requesting a service to other agents; it is a measure of the lacking of skill, that is to say, the degree to which an agent is not utterly competent.

Knowledge/Ignorance Belief about the degree of knowledge (or absence of it) an agent has about the QoS provided by a society of agents.

Satisfaction/Dissatisfaction The satisfaction expresses the degree to which an agent is happy with the utility it expects to obtain from other agents in relation to its estimated necessity.

Curiosity Represents the attitude of an agent towards learning about other agents; it is defined as a function of the knowledge and the satisfaction: the curiosity is directly proportional to the ignorance and the dissatisfaction.

These attitudinal beliefs are defined below, but first we need to introduce another belief, the *expected utility*, which assesses the relative utility an agent expects from another agent compared with itself.

Definition 1 ***(Expected utility)***
The expected utility U_j is the difference between the QoS an agent a_i expects from another agent a_j and the QoS that a_i expects from itself.

$$U_j = (GT_j - myGT) \cdot A_j$$

where GT_j is the Global Trust for agent a_j , $myGT$ denotes the self-assessment of an agent about the QoS it is able to produce (equivalent to GT_i), and A_j is the ratio of service requests accepted by a_j.

The expected utility, as any other belief in our model, has attached a measure of confidence(cU_j. In particular, U_j is given the same confidence value attached to GT_j, since the confidence on $myGT$ is assumed to be maximum (i.e. the null element for the conjunction: $cGT_j \otimes cGT_i = cGT_j$).

Definition 2 ***(Necessity)***
The necessity N is the difference between the maximum QoS potentially possible (a priori, without knowledge of other agents) and the QoS that an agent believes it is able to provide by itself.

$$N = 1 - myGT$$

Definition 3 ***(Knowledge)***
The Knowledge K if the proportion of intimacy an agent has compared with the maximum it is able to obtain.

$$K = \frac{\sum_{j=1}^{n} ITM_j^{GT}}{n-1}$$

where ITM_j^{GT} is the intimacy of GT_j, and n is the maximum number of agents that an agent is able or willing to know (with small agent societies n could be the total number of agents, but with many agents n must be limited).

The ignorance is the complementary to the knowledge, that is $1 - K_i$

Definition 4 ***(Satisfaction)***
The Satisfaction S is the complement of the difference between the lack of skill and the expected utility to be obtained from others.

$$S = 1 - (N - maxU) = (1 + maxU - N)$$

where $maxU$ represents the maximum utility an agent expects to obtain from others, and is defined as the $U_j \times cU_j$ that verifies $\forall i \neq j \mid U_i \times cU_i < U_j \times cU_j$

Complementarily, dissatisfaction is defined as $1 - S$

Definition 5 ***(Curiosity)***
The Curiosity C is the conjunction of ignorance and dissatisfaction.

$$C = (1 - S)(1 - K) = N - maxU(1 - K).$$

As stated above, the main purpose of defining these attitudes is to drive the exploratory behavior of an agent, for instance, an agent with a high curiosity is highly motivated to know other agents, and thus, it would expend more resources performing epistemic actions.

Intuitively, an agent would have curiosity to know other agents either because it is ignorant about other agents, or because it is not satisfied with the QoS provided by the agents it already knows. An agent would not be curious when it thinks it does not need the services being provided by other agents (eg. because it is already competent), or because it is quite satisfied with the service quality it is currently obtaining or expecting to obtain; in either cases it is not worth expending resources to know new agents.

The use of curiosity and the other motivational attitudes may be used not only to drive the exploratory behavior of an agent, but also to decide on pragmatic actions. For example, the necessity or the satisfaction may be used to take decisions on the amount of delegation to incur. An example of such an strategy is included in the empirical evaluation.

However, the use of curiosity is not the only approach to develop an exploratory behavior. Next section describes the empirical results obtained when comparing a curiosity-based strategy against several alternative strategies to explore the environment, using the ART testbed as the experimental platform.

Empirical evaluation

We have chosen the ART Testbed (Fullam *et al.* 2005) to test our test model. The ART Testbed is a simulator of the *art appraisals* domain whose goal is twofold: to serve as a competition forum in which researchers can compare their technologies against objective metrics, and as an experimental tool, with flexible parameters, allowing researchers to perform customizable, easily-repeatable experiments. In the art appraisal domain, agents act as painting appraisers with varying levels of expertise in different artistic eras (e.g. classical, impressionist, postmodern). Clients request appraisals for paintings from different eras. Appraisers can use both their own opinions and opinions purchased to other agents,

so as to make more accurate appraisals. Appraisers estimate the accuracy of the opinions they send by the cost they choose to invest in generating an opinion, though they may lie about the estimated accuracy of their opinions. Appraisers receive more clients, and thus more profit, for producing more accurate appraisals. Appraisers may also purchase reputation information from other agents. The decisions about which opinion providers and reputation providers to trust strongly impact the accuracy of their final appraisals. In competition mode, the winning agent is selected as the appraiser with the highest bank account balance, which depends basically on the ability of an agent to (1) estimate the value of its paintings most accurately and (2) purchase more valuable information.

Experiments setup

In order to understand the mapping between the ART Testbed and our model, we will explain first the way opinions about the value of a painting are generated by the simulator. Each agent is assigned by the simulator a random expertise level s^*, using an uniform distribution in {0.1, 0.2, ...,1}. The opinions of an agent on the true market value of a painting are generated by the simulation such that opinion errors adhere to a normal distribution with mean equals to 0 and a standard deviation s defined as:

$$s = (s^* + \frac{\alpha}{C_g})t_k \tag{13}$$

where s^* is the *expertise level* of an agent for a particular era, α is a parameter named Sensing-Cost-Accuracy, C_g is the cost invested by an agent to generate an opinion, and t_k is the true value of a painting, its market price. Note that C_g is a variable, since it is specified by an agent for each particular opinion, ranging from 0 to any positive value. From the definition of s follows that the so called expertise level s^* is actually the minimum standard deviation of opinions' error, and thus, $s^* = 0.1$ constitutes the best case (the most expert an agent can be), since it implies a narrow distribution of errors, i. e., more accurate opinions.

We map our trust model to the ART Testbed domain as follows:

- The partial DT for an agent a_j is mapped to the perceived expertise for a_j on a given era, which is a function of the observed accuracy of a_j's appraisal opinions for that era. Since during a single time step, an agent may have several paintings to appraise per era, pDT_j^T is defined as a function of the average error of agent a_j's opinions, noted $\overline{\epsilon_j}$. However, opinion errors can not be used straightforwardly as a measure of trust. First of all, we must make errors relative to the true price of a painting so as to make them comparable. The relative error of an opinion from about the value of a painting k is defined as follows

  $$\epsilon_{jk} = \frac{|o_{jk} - t_k|}{t_k}$$

  where o_{jk} is the value of painting k according to a_j's opinion, and t_k is the true value of that painting. The absolute value of the difference is used because we are measuring the magnitude of the error, not the sign.

  In addition, one must invert the measure of error and enclose it in [0,1] to become a measure of trust: small errors impling a high pDT, while large errors imply a low pDT. We are using the following function to calculate pDT:

  $$pDT_{ij}^T = \left\{ \begin{array}{cc} 1 & \overline{\epsilon_{ij}} \leq LB_\epsilon \\ 0 & \overline{\epsilon_{ij}} \geq UB_\epsilon \\ \frac{UB_\epsilon - \overline{\epsilon_{ij}}}{UB_\epsilon - LB_\epsilon} & 0.1 < \overline{\epsilon_{ij}} < 1 \end{array} \right\}$$

  where LB and UB are the lower and upper bounds expected (or enforced) from opinion errors. Values falling out of these limits are rated either as the best value (when $\overline{\epsilon_{ij}} < LB$) or as the worst value (when $\overline{\epsilon_{ij}} > UB$). The lower bound for opinion errors is imposed by the simulation, and is 0.1, the minimum expertise level. The maximum expertise level is 1, but there is no upper bound due to the way s is calculated (see Equation 13): the component $\frac{\alpha}{C_g}$ can be extremely big because C_g (the price invested to generate an opinion) can be arbitrarily small. A reasonable upper limit for $\frac{\alpha}{C_g}$ can be established depending on the expected domain of C_g in the context of an specific combination of simulation parameters; for instance, we can assume than an agent will never expend more money generating opinions than the fee f paid by clients per appraisal. In particular, in our experiments we have been using 0.1 or 0.2 as reasonable upper limits for $\frac{\alpha}{C_g}$, which implies $UB = 1.1$ or $UB = 1.2$ respectively.

- Advertisements are mapped to `certainties`. Certainties are provided by agents when their opinions are requested. These values represents the quality –the accuracy– of the opinions that agents are able and willing to provide. A certainty is a value in [0,1], but it cannot be mapped directly to the partial AT (pAT). The point is that an agent may and will probably receive multiple certainties per time step and era, due to several paintings belonging to the same era. As a consequence, we have to aggregate all the certainties for the same era, using for instance the mean.

- Recommendations are mapped to `reputations`. There is a direct mapping between a single reputation and what we call partial RT (pRT), since a reputation is a value in [0,1] that represents the assessment on the opinions accuracy of other agents for a specific era.

- The three former mappings go from the application domain to our model, they are input mappings. But we also have to establish an output mapping, from our global notion of trust, to the global notion of trust in the ART Testbed, which are the weights send by each appraiser to the simulation engine at the end of each time step. In particular, we use the Global Trust × Global Trust Confidence ($GT \times GTC$) to calculate those weights.

We want to test the potential advantages ad drawbacks of curiosity-based exploratory strategies by comparing different different strategies across several scenarios, ranging from static and reliable situations, to dynamic and unreliable ones. Although a proper evaluation of exploratory strategies would ideally involve a much larger agent population than supported by the current version of the ART Testbed,

we have tried to overcome such limitations by introducing strong changes in the dynamic of opinion prices and other aspects of agent behavior.

In particular, we have tested the following strategies:

1. Curiosity-driven: the maximum number of agents to know each time step is a function of the curiosity: $maxAgentsToKnow = Round(Curiosity * K)$
2. Eager strategy: this strategy tries to know every unknown agent (where unknown means that the intimacy is below a minimum level) each time step.
3. Lazy strategy: tries to know only one of the unknown agents each time step.
4. Non forgetting: this strategy is quite different from the others, since it avoids the use of a forgetting mechanism, ie. intimacy never decreases (persistency = 1). This agent performs epistemic actions eagerly during the first time steps, trying to know everybody, but as soon as an agent is known (intimacy reaches certain level) it will not be the target of an epistemic action again.

Furthermore, we have tested a variation of the curiosity-driven strategy that in addition uses the motivational attitudes to decide on pragmatic actions: it uses the satisfaction to decide how much collaboration to request from other agents: the less the satisfaction, the more the collaboration an agent is willing to request to others.

We have performed two basic types of experiment:

- Type I) Experiments involving **dynamic prices.** In these experiments we simulate a market-like model, with alternating periods of inflation and deflation. Actually, the price to buy an opinion is fixed in the ART Testbed, but market dynamics can still be simulated by changing the cost agents invest in the opinions the provide to other agents: investing more money implies an improvement on the the QoS, and thus it is equivalent to a price reduction (better relation between quality and price), and viceversa. In these experiments we run games involving agents implementing all the strategies we want to compare (two agents per strategy).
- Type II) Experiments involving **double-personality** agents. These are a special kind of agents, namely *autists* and *reputation abusers*, which have a certain probability of changing between two modes of behavior, a normal mode and an anomalous mode. While playing the anomalous behavior, autists deny any kind of interaction with other agents, and the reputation abusers deceive other agents by not providing opinions after been paid. In these experiments each simulation includes only one of the strategies we want to compare (lazy, eager, etc.), competing against and a combination of autists and reputation abusers.

Results

In order to evaluate our proposals we have compared the results obtained by the different strategies stated above along four dimensions:

- The **global performance**, which is evaluated by the *bank balance* at the end of a simulation. This measure is a combination of the performance of agents generating appraisals, and the efficiency expending money and resources to be competent. Because of the tensions between competence and efficiency, the bank balance is very dependent on the parameters of a simulation; for example, the relation between the client fee and the opinion price have a strong impact on the global performance.
- The **accuracy**, which is evaluated either using the *average appraisals error* for an entire simulation, or the total number of appraisals, that is highly correlated with the *average appraisals error* (since appraisals are distributed among agents according to their relative competence)
- The **efficiency**, evaluated by the net income obtained per appraisal, and calculated as the final bank balance divided by the total number of appraisals received during a simulation. This measure gives and idea of how much money has been necessary to obtain an appraisal. Ideally, we want the efficiency to represent the behavior of an appraiser as a provider of appraisals to its ultimate clients, but actually this measure mixes things, since it depends also on the behavior of an agent as a provider of opinions and reputations to other appraisers. Anyway, we have controlled this factor by ensuring that all the agents implement the same policies as providers of opinions to other appraiser.

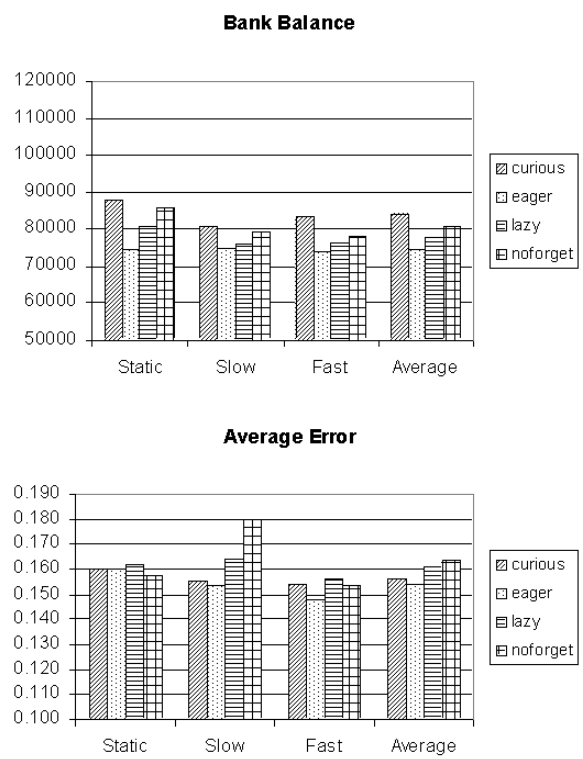

Figure 3: Type I Experiments

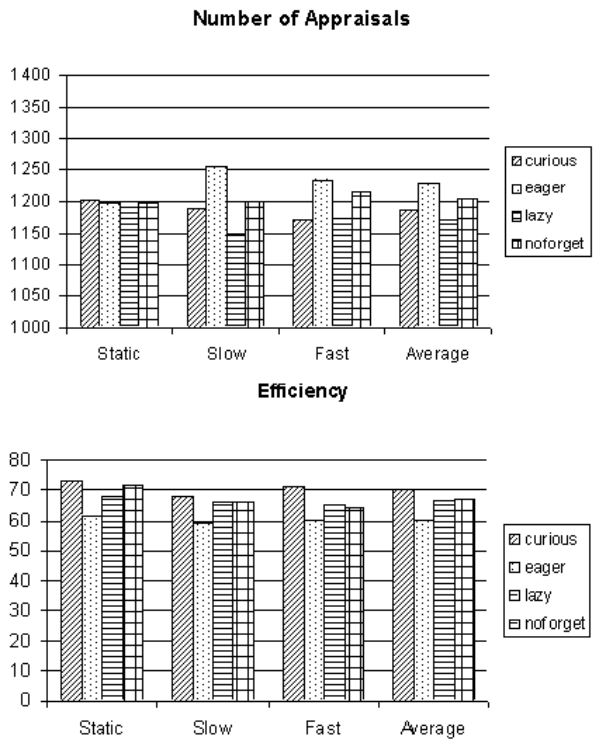

Figure 4: Type I Experiments

Figures 3 and 4 summarize the results for Type I experiments. There are three experimental situations involving different price dynamics, from left to right: *static* prices, *slow* price dynamics, and *fast* price dynamics. The bars with diagonal lines represent the average score for agents implementing the exploratory strategy number 1 (curiosity-driven), dotted bars for type 2 (eager), horizontal lines for type 3 (lazy), and a grid pattern for type 4 (non-forgetting). Clearly, in these experiments the most successful agent in terms of the overall performance is the curiosity-driven strategy, since it obtains the highest bank balance across all the situations considered. However, it does not mean that the curiosity-driven agent is more accurate generating appraisals; actually, the most accurate strategy (the one with lower average errors and thus a higher number of appraisals) is the eager, which is clearly the one with the worst overall performance (the lowest bank account). Therefore, these results could only be explained in terms of efficiency, which is supported by the results shown in Figure 4: the curious strategy is clearly the most efficient one, while the eager strategy is the worst.

Figures 5 and 6 summarize the results of Type I experiments, using an alternative curious-driven strategy that uses motivational attitudes to decide on pragmatic actions too, and not only on epistemic actions: the less satisfied an agent is, the more opinions is willing to demand from other agents. The results are similar to those obtained for the curious agent analyzed previously, but now the curious agent is being much more efficient. As a consequence, although now the curious strategy is obtaining slightly worse accuracies, the improved efficiency results in a much better overall performance.

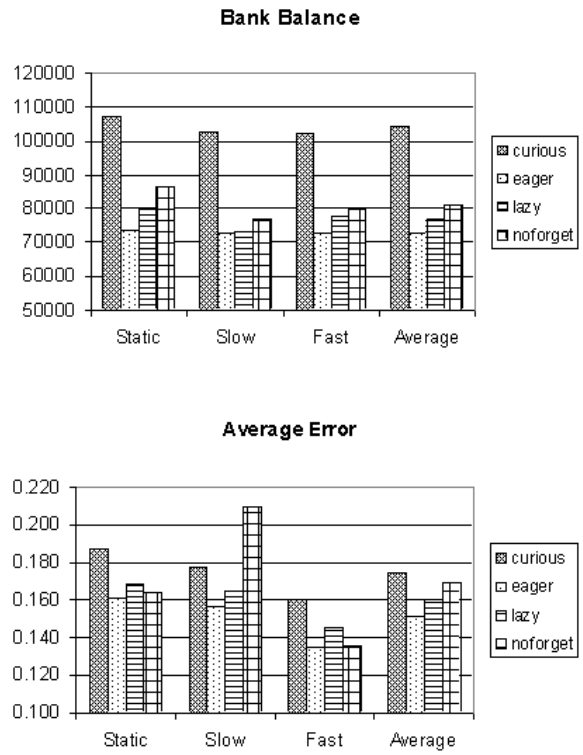

Figure 5: Experiments Type I (bis

Figures 7 and 8 sum up the results for the Type II experiments. Instead of degrees of dynamic prices, these experiments are performed across situations of increasing difficulty (*easy*, *medium* and *hard*) involving double-personality agents, with the difficulty being modeled as the probability of change between two modes of behavior. Each of the simulations in this experiment involves 2 agents using of the strategies being compared, 2 autists, and 2 reputation abusers. Now we are are including together in the same graphics the two versions of the curiosity-driven strategy (cur denotes the first strategy described and curB the alternative version). The overall results are quite similar to those obtained in the Type I experiments: the alternative curious is the one obtaining better overall results, followed by the curious, and the eager agent is one obtaining the worst results. However, in these experiments the curious agents are obtaining relatively better results in terms of accuracy, with average errors in some cases better than the other agents, though the results are not utterly consistent across the three levels of difficulty considered.

To finish this section, Figure 9 shows the results obtained by repeating the type I experiments with the alternative version of the curious agent, and across situations involving a different relation between the client fee and the opinion

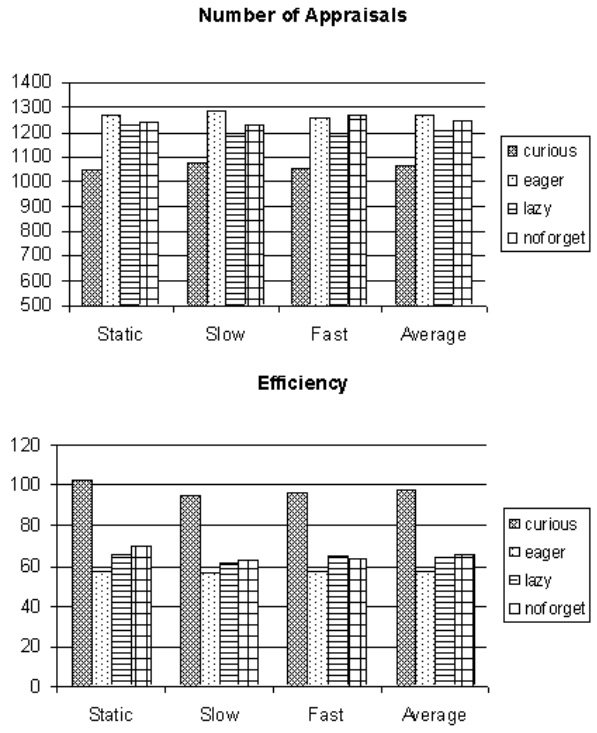

Figure 6: Experiments Type I (bis)

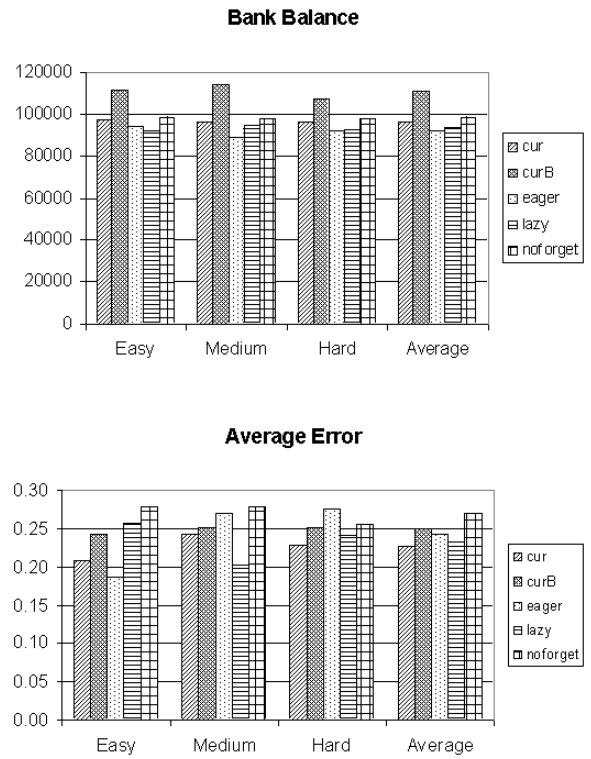

Figure 7: Experiments type II (a)

price. Since the cornerstone of the kind of attitudinal strategies being evaluated is the efficiency, it seems interesting to compare the overall performance of this model changing the relation between the benefits obtained as a consequence of obtaining accurate appraisals, and the cost of obtaining such accurate appraisals. To evaluate this, we have fixed the client fee at 100 and performed experiments with the opinion price between 10 and 30. As expected, the relative performance of the different strategies compared has changed, for instance, with opinion price = 10 all the agents have obtained a similar balance of benefits. Note, however, that the curious agent is the only one that has been able to maintain a similar level of efficiency across all the experimental situations.

Conclusions and future work

In this paper we extend the and refine the trust model proposed in (Gomez, Carbo, & Benac 2006) to address two issues:

- First, we address the problem of absence of information that results when the information required to update trust is expensive or scarce, and propose a more complex forgetting mechanism to address this problem (and not just a forgetting function to weight the historic elements of a belief). Our approach states that the confidence on trust must decay as time goes by without obtaining new information.
- And second, we propose a framework based on motivational attitudes that enables the implementation of adaptive strategies: necessity, knowledge, satisfaction, and curiosity. To test the utility of the proposed framework, two strategies have been implemented and tested across a number of experimental situations: on the one hand, we have experimented with curiosity-driven exploratory policies: on the second hand, we have added a mechanism to limit the amount of cooperation requested from other agents based on the degree of satisfaction.

We have not found any reference in the literature addressing the first issue. More specifically, existing frameworks do not reduce the confidence on trust for a lack of new information: either because it is assumed that agents do not change over time and thus there is no need to update trust for this reason (Teacy *et al.* 2006), or because the question is simply skipped, perhaps implicitly assuming that information is cheap and abundant and thus trust confidence is continually updated. However, in some environments the interaction required to obtain information about other agents is expensive or scarce and thus it is not appropriate to assume that an agent is continually obtaining new information. Furthermore, in many situations assuming that the behavior of agents does not change over time is very unsafe. Therefore, we need a mechanism to address this problem.

Concerning the second issue, there are some frameworks involving a socio-cognitive model of agents and including motivational attitudes; however, most of these frameworks

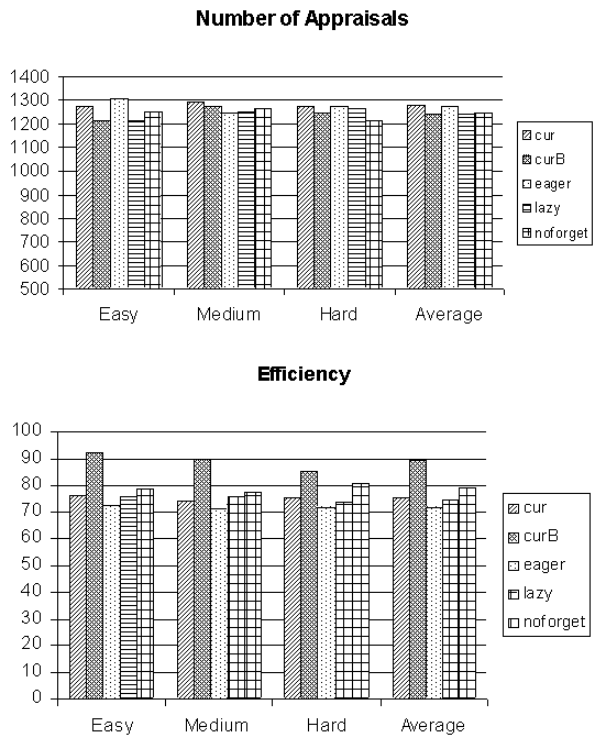

Figure 8: Experiments type II (b)

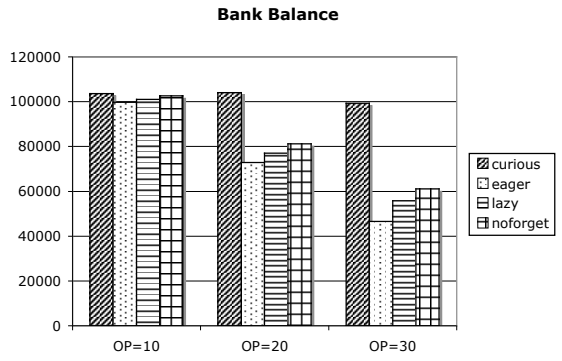

Figure 9: Experiments with different opinion prices

are logic-oriented, based on the BDI paradigm (Falcone & Castelfranci 2001), rather than quantitative. In addition, in these BDI frameworks the motivational attitudes are typically used to model other agents (Braynov & Sandholm 2002; Dasgupta 2000), rather than modeling the truster itself; however, there are some exceptions, as the AFRAS trust model (Carbo, Molina, & Davila 2003), which includes other beliefs to emotively characterize the truster such as *shyness*, *egoism*, or *susceptibility*. All in all, the concept of epistemic actions and the motivational attitudes supporting curiosity-driven strategies are, to our knowledge, quite new to the field.

The proposed attitudinal framework has been evaluated using a modified version of the ART Testbed that enables a greater level of control over the experimental conditions, thus reducing the number of experiments to be performed in order to generalize the results. In particular, we have tested several models that exploit the forgetting mechanism (eager, lazy, curious) against a model that do not uses it (non-forgetting). The results obtained have shown a great potential for implementing more adaptive behaviors that result in a more efficient use of resources, both in terms of epistemic actions (curiosity-driven exploratory behaviors)and pragmatic actions (satisfaction-driven). The most serious drawback of the proposed framework is due to its strong dependence on specific environment properties such as the relation between the cost of epistemic actions and the utility of discovering new providers; for example, in the ART Testbed the results can vary dramatically according to the relation between the Client-Fee and Opinion-Cost parameters (Figure 9).

All in all, it is difficult to conduct an appropriate assessment of exploratory behaviors with the ART Testbed ,mainly because of the limitation in the number of agents participating, and also because it does not allow new agents to enter or leave the system dynamically. To compensate these limitations our experiments have included several modes of behavioral change, including price dynamics and personality changing agents, but an evaluation across larger societies of agents is targeted as a very convenient thing to do in the future. Another interesting line of research would involve the introduction of mechanisms for learning the parameters of behavioral strategies, so as to find the combination of parameters that maximizes the efficiency. Finally, the curiosity-driven exploratory behavior tested in our experiments is still quite simple and barely intelligent, since curiosity is defined globally, instead of for each agent in particular. Actually, once the maximum number of agents to know is decided according to the curiosity, the current criterium to select specific agents to know is based solely on the level of intimacy, trying to know first the agents with lower intimacy levels; that way we are ensuring that all the agents are eventually known. A more intelligent behavior would incorporate other criteria to select the agents to know, and not only the intimacy, for instance, knowing first the agents that have a better reputation; this constitutes a third topic to address as part of our future work.

Acknowledgements

Funded by projects CICYT TSI2005-07344, CICYT TEC2005-07 186 and CAM MADRINET S-0505/TIC/0255

References

Braynov, S., and Sandholm, T. 2002. Trust revelation in multiagent interaction. In *Workshop on The Philosophy and Design of Socially Adept Technologies, Minneapolis*, 57–60.

Butz, M.; Sigaud, O.; and Gérard, P., eds. 2003. *Anticipatory Behavior in Adaptive Learning Systems, Foundations, Theories, and Systems*, volume 2684 of *Lecture Notes in Computer Science*. Springer.

Carbo, J.; Molina, J.; and Davila, J. 2003. Trust management through fuzzy reputation. *International Journal of Cooperative Information Systems* 12(1):135–155.

Dasgupta, P. 2000. Trust as a commodity.

Falcone, R., and Castelfranci, C. 2001. Social trust: A cognitive approach. In Castelfranchi, C., and Tan, Y.-H., eds., *Trust and Deception in Virtual Societibes*. Kluwer Academic Publishers, pp 55-90. 55–90.

Fullam, K.; Klos, T.; Muller, G.; Sabater, J.; Schlosser, A.; Topol, Z.; Barber, K. S.; Rosenschein, J.; Vercouter, L.; and Voss, M. 2005. A specification of the agent reputation and trust (art) testbed: Experimentation and competition for trust in agent societies. In *The Fourth International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS-2005)*, 512–518.

Gomez, M.; Carbo, J.; and Benac, C. 2006. An anticipatory model of trust. In *Anticipatory Behavior in Adaptive Learning Systems*.

Lorini, E., and Castelfrnachi, C. 2004. The role of epistemic actions in expectations. In *Second Workshop of Anticipatory Behavior in Adaptive Learning Systems (ABIALS 2004)*.

Sabater, J., and Sierra, C. 2001. Regret: a reputation model for gregarious societies. In *Fourth Workshop on Deception, Fraud and Trust in Agent Societies*, 61–69.

Teacy, W. L.; Patel, J.; Jennings, N. R.; and Luck, M. 2006. Travos: Trust and reputation in the context of inaccurate information sources. *Journal of Autonomous Agents and Multiagent Systems* 12:183–198.

Guaranteed Security in Trust and Reputation Systems

Reid Kerr and Robin Cohen

David R. Cheriton School of Computer Science
University of Waterloo
Waterloo, Ontario, Canada N2L 3G1
{rckerr, rcohen}@cs.uwaterloo.ca

Abstract

In this paper, we present a framework for evaluating the security of trust and reputation systems for electronic marketplaces populated with buying and selling agents. We argue that current systems to model trust are vulnerable to various attacks; to provide protection from such attacks, systems must be designed not only to predict cheating by agents seeking to deceive one another, but also to cope with agents who are intentionally trying to circumvent the trust/reputation system. Our proposed framework offers a method for researchers to understand the security of their systems, and to provide precise guarantees of the degree of provable security that these systems offer. We focus in particular on characterizing buyer security—the properties that must hold for buyers to feel secure from cheating sellers. We develop a set of security 'levels', benchmarks that may be used in the evaluation and comparison of system security. We demonstrate the viability of our proposed framework by presenting a specific monetary-based trust system known as Trunits, along with an analysis that shows that Trunits does provide a guaranteed level of security for buyers.

Introduction

Much research has been conducted in the area of trust and reputation in multiagent systems [2, 3, 5, 6, 8, 9, 10, 11, 12, 13, 14]; a common focus has been modeling the trustworthiness of buying and selling agents in electronic marketplaces. Many existing proposals consist of predictive models, aimed at determining whether agents are untrustworthy and proposing methods for agents to make effective decisions in the face of possible dishonesty. It is our position that trust and reputation systems should be provably secure, making precise and proven claims as to the degree of protection they provide and the circumstances under which such protection holds. The goal of this approach is twofold: a) to provide a basis for users to confidently choose whether or not to adopt or participate in a system; b) to make explicit the security limitations of systems, and the reasons for such limitations, to allow meaningful progress towards meeting the needs of users. A provably secure system allows users to place their trust in a system, removing an obstacle to its use.

We propose a framework for the consideration of security in trust and reputation systems, based on uncompromising guarantees of protection when required conditions are met. Further, we identify a number of 'levels' of security, varying in the strength of conditions required for the security guarantee to hold. We then demonstrate how a trust system can be provably secure for a buyer, by presenting the monetary-based Trunits trust model, and discussing the protection that it offers for its buying agents.

A common theme of work in trust and reputation in multiagent systems is to increase the likelihood of selecting a trustworthy business partner. Unfortunately, this may not be strong enough to inspire the confidence of potential users or adopters of such systems—anything less than complete trustworthiness of agents raises doubts about the attractiveness of using the system. Some trust and reputation systems attempt to provide incentives for agents to be honest (e.g., [2]), but one must still ask: Under what circumstances will the incentive hold? Will the incentive always be sufficient?

If systems aren't provably secure, the potential for vulnerabilities exists; if vulnerabilities exist in trust and reputation systems, self-interested agents will exploit these in order to maximize profit. We have identified a catalogue of vulnerabilities in trust and reputation systems, displayed and briefly described in Table 1. We then provide a chart to record the vulnerabilities to which we believe each of a small sample of trust systems is open, shown in Table 2.

In some cases, we can relate the presence of vulnerabilities to design choices made by system designers. Systems in which agents make use of the recommendations of other agents, and where each opinion may be used as support for multiple transactions simultaneously [4, 9, 13, 14] tend to be vulnerable to Reputation lag. Systems in which the transaction's impact on an agent's trustworthiness rating is not tied to the value of the transaction [4, 5, 8, 9, 13, 14] tend to be vulnerable to Value imbalance. The potential for Ballot-stuffing and Bad-mouthing can arise in several ways when agents rely on the advice of others. Opportunities can

occur when a system does not protect against users cooperating to undermine the system, or when users can freely create many new accounts, with feedback from these new accounts being weighted similarly to that from established accounts [4, 9, 13]. Re-entry tends to be a problem in systems that treat unknown users preferentially to disreputable users [4, 5, 8, 9, 11, 12, 13], making it beneficial to create a new user account after one's reputation is damaged. Systems that rely on direct experience [5, 8, 11, 12] tend to be vulnerable to the Initial window; the presence of the Re-entry problem magnifies this vulnerability, allowing a dishonest agent to repeatedly take advantage of the window. Most systems tend to be vulnerable to the Exit problem, unless they have taken specific steps to provide an attractive alternative to cheating for agents that decide to leave the market.

Table 1. Descriptions of vulnerabilities

Vulnerability	Brief Description
Reputation lag	Agent can engage in virtually unlimited cheating transactions before reputation is updated to reflect his dishonesty
Value imbalance	Agent can build reputation on small transactions, then cheat on large ones
Ballot-stuffing/ Bad-mouthing	Reputation is artificially improved/damaged by registering large numbers of unfair ratings
Re-entry Problem	After his reputation is destroyed, the cheating agent can enter the market under an alias, effectively shedding his history
Initial window	An agent is vulnerable to an unknown agent, until experience is gained with the new agent
Exit problem	If an agent plans to leave the market, he can cheat freely without repercussions

Table 2. Vulnerabilities found in some existing systems

System	Vulnerabilities
eBay feedback system [4]	Reputation lag, Value imbalance, Ballot-stuffing/Bad-mouthing, Re-entry, Exit
General/Situational Trust [8] Multidimensional Trust [5]	Value imbalance, Re-entry, Initial window, Exit
Tran and Cohen [11, 12]	Re-entry, Initial window, Exit
Sporas/Histos [14]	Reputation lag, Value imbalance, Exit
REGRET [9]	(In market scenario) Reputation lag, Value imbalance, Ballot-stuffing/Bad-mouthing, Re-entry, Exit
Yu and Singh [13]	Reputation lag, Value imbalance, Ballot-stuffing/Bad-mouthing, Re-entry, Exit

This analysis highlights the need to consider the issue of system security carefully. Motivated by work in the field of cryptography, which seeks to deliver provable protection to users, we seek methods for ensuring the provable security of trust and reputation systems.

If our aim is to ensure that a marketplace is secure, we must first characterize what we mean by security. It is our position that a secure system for marketplaces is one where participants are protected from harm (at least, harm due to 'dishonest' behaviour, rather than from legitimate competition). Thus, we define security in terms of a set of 'safety properties': conditions that, if proven to hold for the system, ensure participants within the system will not be harmed by dishonesty. This is distinct from other notions of security, such as the prevention of unauthorized access.

In the marketplace scenario, there are three identifiable 'stakeholders' who participate directly in the market, each with their own requirements: buyers, sellers, and the market operator. In this paper, we focus on the security of buyers, since protecting buyers from cheating sellers is a predominant focus of current research. The other groups are discussed only briefly in section 5, to be addressed in more detail in future work.

The Security Framework

In our framework, we distinguish between two transaction states. An *agreed transaction* consists of the terms to which both parties have agreed: $t_A = (g_p, v, d_p, A_b, A_s, \ldots)$, where g_p is the good promised, v is the value (agreed price) of the good, d_p is the date/time promised, A_b is the buying agent, and A_s is the selling agent. (The ellipsis indicates that there may be other system- or market-dependent parameters.) This might be viewed as a promise or a contract; it may also be viewed as the transaction at the point both parties have struck a deal, but have not yet acted, so the honesty or dishonesty of the transaction is undetermined.

A *delivered transaction* is one where the selling agent has provided the goods to the buyer, but the buyer has not yet rated the seller: $t_D = (t_A, g_d, c, d_d, \ldots)$, where t_A is the agreed transaction, g_d is the good delivered, c is the cost incurred by the seller in providing and delivering the good, and d_d is the date/time of delivery. We consider a delivered transaction t_D to be *honest* if it fulfills the seller's commitments—g_d satisfies g_p, d_d satisfies d_p, etc.—and denote it by the predicate $honest(t_D)$. (The details of how a good or promise is specified are left to the system designer. For instance any $d_d \leq d_p$ may be considered honest.)[1] We consider a transac-

[1] Note that buyers will often also close off a given transaction by computing a rating for the seller. When ratings are elicited from buyers may vary by system, however. Thus, we refrain from defining a 'rated transaction' for generality.

tion where an agent (intentionally) fails to fulfill his commitment (whether by providing a good that does not meet the commitment, or by not providing a good at all) to be an instance of *cheating*, or *dishonesty* on the part of the seller.

It is possible that a buyer could cheat by withholding payment after receipt of the goods. We base our framework, however, on the common policy that a buyer must pay before goods are shipped.

For brevity, we make use of 'accessor' functions that return the value of individual transaction parameters. These functions have the same name as the parameter that they return.

An agent attempting to cheat may act alone, or as part of a coalition. We denote such a coalition G; an agent acting alone is equivalent to the case where $|G| = 1$.

We term a set of transactions a *schedule*. Let T_D represent a schedule of delivered transactions. For any T_D, there is a corresponding T_A consisting of the same transactions with the delivery parameters removed. Note that for any T_A, there are possibly many T_D, since each transaction in T_A might be executed honestly or dishonestly. *Executing* a transaction refers to delivering a good (that either meets of fails to meet the advertised promise), or consciously deciding not to deliver the good at all.

For any coalition of sellers G, consider T_D where $t_D \in T_D \Leftrightarrow A_s(t_D) \in G$. For each transaction in the set, the sellers in G may choose to execute the transaction honestly or dishonestly. We denote $C \subseteq T_D$ as the *cheating set*, the subset that is executed dishonestly. The coalition may have a choice of many different cheating sets for any given schedule; choosing C is a strategic choice. (We stop short of saying that C is a strategy, however, since the coalition might also strategically choose the composition of T_D by choosing the transactions into which they will enter.)

Not all schedules can actually be executed. For example, an agent that cheats repeatedly might not continue to find buyers for its products; although it might be possible to formulate a schedule that includes continued future business, such a schedule may be impossible under the trust system. Continuing the example, if trustworthiness is rated in the interval [0, 1], and an agent's score has dropped to 0, he may not be able to engage in further transactions, even though he has inventory. We define the predicate *feasible*(t, T) to denote that a transaction t can actually be executed within the schedule T, in the system under consideration. *feasible*(T) denotes that every transaction in T is feasible. We do not define feasibility further, since it will be system- or market-specific.

As we will see, profitability is a key concern when considering the security of trust systems. The profit to the seller on an individual transaction is the selling price minus the cost, or $P(t_D) = v(t_D) - c(t_D)$. The profit to a coalition on the entire set of transactions is

$$P(G, T_D) = \sum_{\{t_D \in T_D | A_s(t_D) \in G \wedge A_b(t_D) \notin G \wedge feasible(t_D, T_D)\}} P(t)$$

Buyer Security

A buyer who engages in no transactions suffers no direct harm from those transactions. A buyer who enters into a transaction (assuming the common pay-before-delivery policy) becomes vulnerable at the moment that they pay for a good. From this point, the seller is in control, and the buyer may be harmed by receiving an inferior good, or no good at all.

A seller may be harmed, for example, by unfair feedback from buyers. For a seller to wish to be honest, he may need confidence that buyers will provide fair reviews. We do not address means to ensure the fairness of buyers here, however, since this is an element of seller security. To establish that a system is buyer secure, then, may require the assumption that buyers are honest. We discuss how to address this problem, and lift this assumption, in the Discussion and Future Work section.

In our framework, for a system to be secure, we do not hold the seller responsible for the buyer's complete satisfaction. Instead, she need only deliver the good that she was 'supposed' to give. When a seller offers a good for sale, she provides information about that good. This information constitutes the basis for the buyer's understanding of the good they will receive. Should they purchase the good, they would expect that it corresponds to each claim made in the offer. This, then, is the basis for our notion of buyer security: a buyer will be secure under a trust system if

$$\forall t_D \in T_D \;\; honest(t_D)$$

Note that 'buyer security' directly addresses issues such as the value imbalance and reputation lag problems. Other forms of dishonest behaviour may not be relevant to the buyer. For example, ballot stuffing would be precluded by this property if used to lure buyers into cheating transactions, but not if used by a seller to steal sales from another seller; the later is an issue of seller security.

Levels of buyer security

We might term the previous property, should it hold, as *full buyer security*—i.e., it is impossible for a seller to cheat a buyer. Unfortunately, this property would be extremely hard to guarantee in practice. For example, one might envision a trusted third party who receives both payment from the buyer and the good from the seller, and only forwards payment to the seller after inspecting the good to ensure it fulfills the agreement. Such a system might offer great security, but is unlikely to be practical or scalable. [3]

While it may not be feasible for a system to guarantee this property, it may be possible to achieve it under certain con-

ditions, when certain assumptions hold for the marketplace. By limiting the guarantee to those circumstances where the assumptions hold, we effectively weaken the guarantee, allowing us to specify levels of security that are weaker than the ideal. We specify these properties in the form of an implication:

$$(assumption_1 \wedge \ldots \wedge assumption_m) \Rightarrow \forall t_D \in T_D \;\; honest(t_D)$$

The assumptions denote limitations in the system, which prevent it from delivering on the unconditional guarantee. This does not mean that the system is useless, however. For each assumption, there are two primary approaches to dealing with it:

1. External: It may be possible to ensure that an assumption actually holds for the marketplace in question. If the property can be verified to hold for the marketplace, or if some mechanism external to the trust/reputation system can be used to guarantee the property, then the system will function adequately despite the presence of the assumption.
2. Internal: It may be possible to modify the system to remove the assumption as a requirement for safety. Such modification may yield a more robust system, capable of working under a smaller set of assumptions. Thus, the presence of an assumption can provide important guidance for future research, allowing meaningful progress to be made.

Through the use of these techniques, the goal would be to arrive at a system for which every remaining assumption can be ensured to hold in the marketplace—such a system would be secure for that marketplace. It is our contention that clearly stating assumptions aids understanding of the security delivered, and the limitations of this security, as well as easing comparisons between possible models.

Rational-agent secure

While we may not be able to guarantee that every sale is executed honestly, we may be able to design the system so that it is in a seller's best interest to be honest. Such incentive-based approaches depend on agents being rational profit-maximizers—operation of the system depends on agents reliably choosing what is best for them. We believe this to represent an important and high level of security, stated as:

$$\text{selling agents are rational} \Rightarrow \forall t_D \in T_D \;\; honest(t_D)$$

More formally, denoting a coalition of selling agents as G:

$$[\forall G \; rational(G)] \Rightarrow \forall t_D \in T_D \;\; honest(t_D)$$

Recall that this entire statement is a specified property of a system. It does not state that the implication holds in all cases; rather, for a system to be considered Rational-agent secure, it must be proved that under the system, if selling agents are rational then all transactions are honest.

Since rational sellers are profit maximizers, the property above can be restated as:

$$[\forall G \; \forall T_{D1}, T_{D2} \; [\, P(G, T_{D1}) > P(G, T_{D2}) \Rightarrow T_{D1} \text{ is selected}]] \\ \Rightarrow \forall t_D \in T_D \;\; honest(t_D)$$

For a system to be rational-agent secure, sellers must be able to understand that honesty is the most profitable policy. Under some systems this may require considerable computation. For example, determining that honesty maximizes profit may require the computation of an entire tree of possible future outcomes, which may be beyond the capabilities of the agent. Where this may be an issue, the set of assumptions should include the computational capacity required of the agents.

Just as rational-agent security is quite a strong guarantee, it may also be difficult to achieve. We consider several lower levels of security, derived by adding weakening conditions to the rational-agent secure property. The assumptions described below are not mutually exclusive, nor can they be ordered in terms of security. Systems requiring one or more of the following assumptions may be useful for certain scenarios, or may be only of research interest, as a stepping stone to a more secure method.

Rational single-agent secure

Ideally, a system would make the buyer secure regardless of collusion between agents. However, collusion is notoriously difficult to combat. A lower level of security might protect agents only from sellers who are not part of a coalition:

$$[\forall G \; rational(G) \wedge |G| = 1] \Rightarrow \forall t_D \in T_D \;\; honest(t_D)$$

Rational single-seller-only secure

Under some systems, a seller might be able to execute attacks by acting as a buyer for some transactions, and as a seller for others. As a weaker extension of single-agent security, a system might be secure when sellers cannot act as buyers. (A seller might be able to open another account to use as a buyer, but that is an instance of collusion.)

$$[\forall G \; rational(G) \wedge |G| = 1 \wedge \forall t_D \in T_D \; A_B(t_D) \notin G] \\ \Rightarrow \forall t_D \in T_D \;\; honest(t_D)$$

Rational infinite-transaction secure

The exit problem is an extremely difficult one to combat, and it may be difficult to prevent dishonest sales once sellers have exhausted finite inventories. However, a system may make it more attractive for a seller to continue to do business than to exit at any point. Such a system may prevent the exit problem, but requires agents to be able to engage in infinite transactions (e.g., the seller never runs out of inventory, there are always buyers willing to purchase the product, etc.):

$$[\forall G \; rational(G) \wedge \forall T_D \; [honest(T_D) \wedge feasible(T_D) \Rightarrow \\ (\exists t \; honest(T_D \cup t) \wedge feasible(T_D \cup t) \;\;)\,]] \\ \Rightarrow \forall t_D \in T_D \;\; honest(t_D)$$

Of course, a buyer may require protection in other ways—that the market operator won't take her money, that her personal information won't be sold, etc. However, these issues fall outside the traditional role of a trust/reputation system, and it is difficult to conceive of a trust/reputation system preventing behaviour that occurs outside of the marketplace itself, or that controls the behaviour of its operator.

It may seem very difficult to use these standards in the analysis of many models, particularly those that are predictive in nature. It is worth reiterating, however, that unless proofs of such properties can be rendered, these systems are of unknown security at best; from our survey in section 1, it appears likely they are insecure.

Security analysis of Trunits

Having outlined a framework for establishing security guarantees and enumerated a number of important 'levels' of security, we provide no guidance in the construction of such proofs. The reason is simple—proof methods are likely to vary greatly depending on the nature of the system used. Instead, we provide an analysis of our particular model of trust, Trunits, for two reasons. First, it provides an example of the analysis of a trust model using our proposed framework. Second, we believe that it serves as one example of how trust might be modeled with an eye towards providing provable levels of security to adopters and participants.

The Trunits Model

Trunits is a model of trust—a buyer will engage in sales with a seller in which there is a certain degree of trust. It differs from most trust models, however, in that it attempts not to predict behaviour, but rather to ensure good behaviour by making honesty the most profitable strategy. We provide a brief description of Trunits here; a more detailed treatment can be found in [7].

The 'Trunits' model is inspired by the concept of money. Before the advent of money, goods and services were exchanged by bartering. This placed several limitations on trade; here, the most relevant was the requirement for buyers and sellers to interact directly, to exchange goods of comparable value. A primary function of money is to overcome this requirement.

Money is an abstract 'substance', representing quantities of value. Money flows in a transaction, mirroring the flow of value in a barter transaction: the value of the money stands in for the value of a good. Money frees the traders from the requirement that goods move in both directions—value gained from one trader can be 'spent' with another.

The key problem with trust in our new breed of marketplaces is that buyers and sellers usually do not have direct relationships, so trust cannot form naturally. Since we seek to overcome the requirement for a direct relationship—to allow trust gained from one trader to be 'spent' with another—it seems natural to consider the use of abstract trust units, or 'trunits', to play the same basic role in which money has been so successful.

As with money, the movement of trunits should mirror that of trust in a direct relationship. This movement, however, is very different from that of value. While the flow of value is an exchange process, we see the 'movement' of trust as a *risk* process, and suggest a model based on this view. We focus on trust of the seller as the primary issue:

- Before a buyer will purchase something from a seller, the buyer must have sufficient trust in the seller. The degree of trust required is dependent on a number of factors; the price of the item is likely a major one.
- After purchasing the good, the buyer will evaluate it.
  - If the good met her expectations (i.e., it was at least as good as was advertised by the seller), then the seller is likely to gain more of her trust.
  - If the good did not meet her expectations, then the seller is likely to lose some of her trust.

Based on this view, we suggest a model that makes use of abstract units of trust, where trust of a seller is not tied to a specific buyer:

- The seller has some quantity of trunits, representing all of the trust gained from all buyers to date. For a buyer to consider buying from a seller, the seller must possess a sufficient degree of trust, i.e., must hold sufficient trunits. The required number of trunits is tied to the price of the good.
- After purchasing the good, the buyer will evaluate it, relative to her expectations.
  - If the good met her expectations, then the seller gains some additional quantity of trunits.
  - If the good did not meet her expectations, then the seller loses some quantity of trunits.

As a seller executes honest transactions, his trunit balance grows, allowing future profitable transactions. In contrast, dishonest sales curtail future transactions. This provides the fundamental incentive for honesty. The number of trunits gained is proportional to the size of the sale. Honest execution of small transactions will allow a seller to continue making small sales, and to grow his sales volume, but will not allow him to immediately jump to disproportionately large sales for which he has not demonstrated trustworthiness.

We propose a 'basic Trunits mechanism', based on this model. When an agent wishes to make a sale, we require him to put up a quantity of trunits to 'cover' the sale[2].

[2]Under Trunits as currently described, trunits are kept with a 'market operator' who administers the system. If this market

These trunits represent the trust that the seller is risking by engaging in a transaction. We require that the number of trunits risked be directly tied to the value of the transaction, using the formula:

$$V = r\tau$$

where V is the value (selling price) of the transaction, τ is the number of trunits, and r is the required *risk ratio*[3], a (positive) parameter set by the market operator. The trunits are put into escrow with the market operator, pending completion of the transaction. Upon completion, if the buyer rates the transaction as unsatisfactory, then the seller loses the τ trunits placed in escrow. If, on the other hand, the buyer rates the transaction as satisfactory, then the τ trunits are returned to the seller, along with some additional quantity of trunits related to the value of the transaction, for a total of:

$$(1+p)\ \tau = (1+p)\ V/r$$

where p is a *premium* or *reward* of additional trust for acting in an honest manner, a (positive) parameter set by the market operator. In the basic mechanism presented here, the same values of r and p are used for all traders and transactions. In this implementation of Trunits, we do not allow a buyer to sell to himself (i.e., to the same user account), in order to gain trunits through 'honest' transactions.

From a buyer's perspective, no evaluation or computation is required prior to purchasing to determine if a seller is trustworthy—if the seller possesses enough trunits for a transaction, then *by definition*, she is trustworthy *for that transaction*. The market operator will not allow a transaction to be executed unless the seller has sufficient trunits. From a seller's perspective, honesty results in a growing trunit balance and the ability to engage in more sales in the future, while dishonesty will reduce the potential for future sales.

One issue encountered with this model is the 'start-up' problem: how does an agent acquire an initial quantity of trunits? In our current work, we investigate several options that appear to be safe, including loaning agents an initial quantity of trunits to be secured by a cash bond, and (where identities can be established with certainty, preventing Re-entry) simply providing new agents with an initial quantity. A third alternative, allowing agents to purchase trunits on an open market, is discussed briefly in section 5.

Note that, while Trunits is a model of trust, it is also a mechanism, designed to encourage honest behaviour. Moreover, the incentive provided by the mechanism can be calculated with precision (as outlined below), serving as a basis for rational decision making.

Buyer Security in Trunits

We seek to verify that the essential buyer security property, $\forall t_D\ honest(t_D)$ will hold. As an incentive mechanism, Trunits relies on agent rationality to ensure desirable behaviour: we target rational-agent security. The basic function of Trunits is that an agent makes more money if he fulfills his commitment, so he tries to do so. For this incentive to hold, the agent must *actually be able* to fulfill his commitments. If he is unable to do so successfully (e.g., poor quality control) he might find it more profitable to cheat, rather than incurring the cost associated with honestly executing a transaction, and still getting a bad rating. Thus, we assume that agents can control quality in order to meet commitments if they choose to do so. (We can actually relax this assumption under Trunits, instead specifying with precision an acceptable range in the degree of control, but omit these details for brevity.)

The basic Trunits mechanism regulates the behaviour only of sellers, so on its own, it cannot provide provable security in the face of coalitions of both buyers and sellers. Thus, we attempt to prove that Trunits provides rational single-seller-only security. Further, since Trunits is based on buyer feedback controlling future sales, we must assume that buyer honesty is ensured through some parallel mechanism. Finally, basic Trunits provides no direct impediment to a seller cheating as she exits the market, should she exhaust her ability to honestly sell goods (e.g., if she has run out of inventory). However (as will be shown below), there is a strong incentive not to exit the market, so our analysis is conducted under the assumption that the infinite-transaction property holds, where the agent can engage in infinite honest sales if desired. (By 'infinite', we mean both that the seller's activity is unbounded in duration, and that the seller's capacity for sales at any given moment is unbounded as well.)

This analysis is based on the assumption that selling cost is a fixed fraction c of selling price. This assumption is not unreasonable; many companies determine selling prices by applying percentage markups to cost, and in many industries the markup used is consistent among sellers. While we do not believe that this constraint is required for Trunits to be secure, it has been assumed in order to simplify analysis.

Security guarantee of Trunits

What we seek to prove, then, is:

Trunits is in use $\wedge$
selling agents are rational $\wedge$ (A)
selling agents act alone $\wedge$ (B)
selling agents can engage in infinite honest transactions $\wedge$ (C)

operator is an identifiable entity (e.g., in a centralized implementation), this market operator is considered to be a trusted third party.

[3]While this relationship determines the number of trunits required to secure a sale, note that it does not imply an 'exchange rate'; trunits cannot be directly traded for money in this manner under the mechanism presented.

buying agents are honest ∧ (D)
selling agents can reliably meet commitments if willing ∧ (E)
cost c is a constant percentage of selling price (F)
$\Rightarrow \forall t_D \; honest(t_D)$

Specified more formally:

[Trunits is in use ∧
$\forall G \; \forall T_{D1}, T_{D2} \; [\, P(G, T_{D1}) > P(G, T_{D2}) \Rightarrow T_{D1}$ is selected] ∧
$\forall G \; |G| = 1$ ∧
$\forall T_D \; [honest(T_D) \wedge feasible(T_D) \Rightarrow$
$(\, \exists t \; honest(T_D \cup t) \wedge feasible(T_D \cup t) \;\;)\,]$ ∧
buying agents are honest ∧
selling agents can reliably meet commitments if willing ∧
cost c is a constant percentage of selling price]
$\Rightarrow \forall t_D \in T_D \; honest(t_D)$

A note on feasibility

The feasibility of a schedule is important to its profitability, so impacting the analysis of Trunits. A transaction under Trunits begins when the agreement is made, and ends when the buyer has rated the seller. Let *start*(i) represent the start time of transaction i, and *end*(i) its time of completion. Let τ_{init} represent the seller's initial trunit balance, and τ_i the trunits required for transaction i. Since every transaction i requires an outflow of trunits when it begins, but only honest transactions have inflows (plus reward) at completion, the balance of trunits available at any given *time* is:

$$\tau_{bal}(time) = \tau_{init} - \sum_{i \in T_d | start(i) \leqslant time} \tau_i + (1+p) \times \sum_{i \in T_d \setminus C | end(i) \leqslant time} \tau_i$$

If, at any *time*, $\tau_{bal} < 0$, then some transaction(s) starting before *time* required more trunits than were available (i.e., the transaction(s) would not have been allowed). A feasible schedule (from the standpoint of the constraints imposed by Trunits), then, is one for which τ_{bal} is never less than 0 for all sellers of transactions in the schedule. Consider any T_D and cheating set C, where $C \subseteq T_D$. Note that the addition of a transaction (that is a member of T_D) to C (i.e., changing an honest transaction to a dishonest one) does not change the number of 'outflow' trunits, but does reduce the number of 'inflow' trunits. Thus, the addition of a transaction to C never increases τ_{bal}, but will lower it (specifically, after *end*). This means that the addition of a transaction to C might result in a previously feasible transaction becoming infeasible. Conversely, the removal of a transaction from C only increases the number of trunits available, so it cannot render a feasible transaction infeasible.[4]

Proving the guarantee

Since rational sellers choose the most profitable option, our goal is to show that, for any arbitrary schedule, profit is maximized by executing each transaction in the schedule honestly. First, we consider only finite schedules. Consider any honest, feasible schedule T_D, and a schedule T_D' with the same set of agreed transactions T_A. T_D' has the non-empty cheating set C. Since we have assumed that sellers act alone, we omit the G from our profit formula. For each computation below, we denote each T_D by its corresponding schedule of agreed transactions and its cheating set. Thus, we seek to show that for any non-empty $C \subseteq T_A$,

$$P(T_A, C) < P(T_A, \varnothing).$$

Note that by our assumptions, the seller acts alone; further, the mechanism does not allow him to sell to himself. This means that all cash and trunit flows come only through transactions with buyers. Note, too, that the seller can meet his commitments if he chooses to do so, and that buyers are honest—this means that if a seller intends to fulfill a transaction honestly, he will receive the trunit flows due him.

The profit function for Trunits requires further consideration, regarding the value of accumulated trunits. At the end of the schedule, the seller will have earned some profit, and will have some quantity of remaining trunits (denoted τ_{bal} (*exit*), where *exit* is the time at which the last transaction is completed). While cheating might increase profit earned during the schedule, it would reduce the number of leftover trunits—since trunits can be used to earn future profits, this is a reduction in value gained by the seller. To measure this value, we introduce one additional transaction that occurs after *exit*. In this transaction, the seller uses all remaining trunits to cheat, as he is free to do. We do not mean to suggest that this is what the seller will or should do. (As we will show below, if he is rational he would continue to make honest trades beyond the end of the schedule.) Instead, we use this to determine the value he can *assuredly* gain from his trunits, and effectively set a lower bound on the future profits that could be earned with them. Thus, for every schedule, the total profit will be the sum of the profit from honest sales, the profit from cheating sales, and the revenue from the 'final cheat' after the schedule has completed:

$$\begin{aligned} P(T_A, C) &= (1-c)r \sum_{i \in T_A \setminus C} \tau_i + r \sum_{i \in C} \tau_i + r\,\tau_{bal}(exit) \\ &= (1-c)r \sum_{i \in T_A \setminus C} \tau_i + r \sum_{i \in C} \tau_i + r\left(\tau_{init} - \sum_{i \in T_A} \tau_i + (1+p) \sum_{i \in T_A \setminus C} \tau_i\right) \end{aligned}$$

(In fact, if the schedule is infeasible, the profit will be less than this, because some of the transactions will not be permitted to occur. Thus, this represents an upper limit on the profitability of the schedule.) Now, consider the same schedule, but with two different sets of cheating transactions C_1 and C_2, where $C_1 \subset C_2$, (i.e., C_2 may be thought of as the result of adding cheating transactions to C_1). If the delivered schedule using C_1 is feasible, the one using C_2 may be either feasible or infeasible. To compare profits from each schedule, we subtract the profit of the second from that of the first:

[4] While feasibility depends on the timing of transactions, we do not specify temporal parameters for the schedules (T_A/T_D); instead, the timing of each transaction can be specified within the transaction itself.

$$
\begin{aligned}
P(T_A, C_1) - P(T_A, C_2) &= \left[(1-c)r\sum_{i\in T_A\cup C_1}\tau_i + r\sum_{i\in C_1}\tau_i + r\left(\tau_{init} + \sum_{i\in T_A}\tau_i\right)(1-p)\sum_{i\in T_A\cup C_1}\tau_i\right] \\
&\quad - \left[(1-c)r\sum_{i\in T_A\cup C_2}\tau_i + r\sum_{i\in C_2}\tau_i + r\left(\tau_{init} + \sum_{i\in T_A}\tau_i\right)(1-p)\sum_{i\in T_A\cup C_2}\tau_i\right] \\
&= (1-c)r\left(\sum_{i\in T_A\cup C_1}\tau_i - \sum_{i\in T_A\cup C_2}\tau_i\right) + r\left(\sum_{i\in C_1}\tau_i - \sum_{i\in C_2}\tau_i\right) \\
&\quad - r(1-p)\left(\sum_{i\in T_A\cup C_1}\tau_i - \sum_{i\in T_A\cup C_2}\tau_i\right) \\
&= (1-c)r\left(\sum_{i\in C_2\setminus C_1}\tau_i\right) + r\left(\sum_{i\in C_2\setminus C_1}\tau_i\right) - r(1-p)\left(\sum_{i\in C_2\setminus C_1}\tau_i\right) \\
&= (1-c+p)r\sum_{i\in C_2\setminus C_1}\tau_i
\end{aligned}
$$

Given that $(1 - c)$, p, and r must all be greater than 0, as must all trunit values in the sets (and hence in the summation), this subtraction yields a positive number. (Further, note that if C_2 yields an infeasible schedule, then its profit will be reduced, increasing the result of the subtraction.) This means that if $C_1 \subset C_2$, the profit using C_1 must be higher than that of C_2. Given that the empty set is a subset of every set, for any finite T_A and non-empty $C \subset T_A$, $P(T_A, C) < P(T_A, \varnothing)$.

The exit problem

The analysis above shows that for any finite schedule, profit is maximized through honesty, but for the last 'cheating exit' transaction. Ideally, the seller will never want to make such an exit; we now relax the finite schedule constraint, consistent with our stated assumption. Consider any arbitrary feasible schedule T_A. A rational seller will maximize profit by executing every transaction honestly, so the profit formula simplifies to:

$$P(T_A) = r(1-c+p)\sum_{i\in T_A}\tau_i + r\tau_{init}$$

Instead of cheating on exit, the seller might consider executing one more honest transaction t. Assuming that the new transaction yields a feasible schedule (and since every sale in T_A is honest, it must be possible to add a feasible transaction), the new profit is:

$$
\begin{aligned}
P(T_A \cup \{t\}) &= r(1-c+p)\sum_{i\in T_A\cup\{t\}}\tau_i + r\tau_{init} \\
&= r(1-c+p)\left(\sum_{i\in T_A}\tau_i + \tau_t\right) + r\tau_{init}
\end{aligned}
$$

Since all of r, p, $(1 - c)$, and τ_t are positive, $P(T_A \cup t) > P(T_A)$, meaning that for any given schedule, it is more profitable for the seller to add profitable transactions. (Note that adding dishonest transactions does not increase the profit—cheating within the schedule is no more profitable than during the 'cheating exit'.)

The result implies that to maximize profit, the seller should never cheat, but should continue to sell items indefinitely.

In summary, for any schedule, profit is maximized by executing every transaction honestly, and continuing to add honest transactions to infinity. Thus,

[Trunits is in use $\wedge$
$\forall G\ |G| = 1 \wedge$
$\forall T_D$ [$honest(T_D) \wedge feasible(T_D) \Rightarrow$
$(\exists t\ honest(T_D \cup t) \wedge feasible(T_D \cup t))$] $\wedge$
buying agents are honest $\wedge$
selling agents can reliably meet their promised specifications $\wedge$
cost c is a constant percentage of selling price]
$\Rightarrow \forall T_A[\forall C \subseteq T_A, |C| > 0 \Rightarrow P(G, T_A, \varnothing) > P(G, T_A, C)]$

This yields:

$[\forall G\ \forall T_{D1}, T_{D2}\ [P(G, T_{D1}) > P(G, T_{D2}) \Rightarrow T_{D1}$ is selected$]] \wedge$
$\forall T_A[\forall C \subseteq T_A, |C| > 0 \Rightarrow P(G, T_A, \varnothing) > P(G, T_A, C)]$
$\Rightarrow \forall T_A(T_A, \varnothing)$ is selected $\Rightarrow \forall t_D \in T_D\ honest(t_D)$

Essentially this means that since it will not be profitable for a rational seller agent to cheat, they will execute every transaction honestly. Thus, the Trunits mechanism can provide the user with a guarantee of security, at this level.

The (labelled) list of assumptions given above identifies the limitations of the basic Trunits mechanism. Understanding these, how can we be sure that the mechanism will be secure? Rationality of agents (A) is a fundamental assumption of most work in mechanism design, and a limitation that we likely must accept; it is not an unreasonable expectation of sellers in a marketplace, however. Sellers acting alone (B) speaks to the issue of collusion, a difficult problem with which the trust and reputation community continues to struggle. Since it is unlikely that we can safely assume that agents won't collude, we must devote effort to extending the mechanism to make it collusion-proof. The need for infinite transactions (C) can be addressed through enhancements to the Trunits mechanism; this is discussed in more detail in section 5. The requirement for agents to be honest (D) speaks directly to the absence of any system to address the trustworthiness of buyers. Extension of Trunits, or the use of a parallel system to ensure buyer honesty, is required to address this limitation. The assumption that sellers can control quality (E) can actually be refined to specify the degree of control required, as noted. Finally, the assumption that cost is a fixed percentage of selling price (F) is a special case of an internal limitation. This assumption is not likely required for the desired property to hold, but has been added to ease analysis. It might be eliminated with more detailed consideration.

The Trunits mechanism, as presented, provides security against reputation lag (since trunits must be placed in escrow during the course of a transaction, they cannot be used to cheat multiple buyers at once) and value imbalance (since the number of trunits earned in honest sales is proportional to the value of the sale), and some degree of protection against the exit problem (since there is an incentive for sellers to remain in the market). Note too that re-entry does not diminish the security of buyers (since there is no advantage for a seller who has lost his trunits to re-enter

under a different name), and that no initial window of opportunity exists (since buyers are not relying on their own experience to choose partners.) These arguments are outlined in greater detail in [7]. However, success against a large number of catalogued vulnerabilities is not enough for any system to be considered secure—our catalogue is almost certainly not exhaustive. In contrast, our analysis above gives us a clear picture of exactly what guarantee basic Trunits provides, and under what conditions—Trunits has achieved a specific provable level of security. The ability to do so appears to be linked to the fact that the incentive for honesty is explicit, and thus measurable. Based on this guarantee, informed decisions can be made about whether this mechanism is appropriate for a given scenario. Moreover, clear directions have been identified for future research, in overcoming these restrictions.

Discussion and Future Work

This paper has argued that it is necessary to explicitly consider security in the development of trust and reputation systems, and has proposed a framework for doing so. This approach is contrasted with that of current researchers in the area of trust for multiagent systems in electronic marketplaces, which focus predominantly on developing methods for predicting untrustworthiness.

Existing methods of modeling trust and reputation are subject to vulnerabilities; the gravity of the existence of such vulnerabilities should not be underestimated. To a rational agent who knows that a trust/reputation model is in use, the model constitutes part of the agent's environment as much as the operational rules of the market do. Just as we would expect this agent to maximize profit within the operational rules, we should expect he might do so with this expanded understanding of the environment—a profit maximizing strategy might very well involve exploiting a vulnerability.

To be considered secure, a system with potential vulnerabilities would need to demonstrate that it is secure against an exhaustive list of such vulnerabilities (which is difficult to obtain). Moreover, in the process of adjusting the system to remove vulnerabilities, the potential for introducing new vulnerabilities exists. In contrast, proving a system to be secure according to our framework enumerates a complete set of assumptions to be addressed.

It can be argued that when a predictive model is being used, if an agent is rational and knows that trustworthiness is being modeled, it will affect his decisions—he might be inclined to try to act honestly, to maximize future profit. In this case, a predictive model might also be viewed as a de facto incentive mechanism. This view makes our framework especially relevant for those working on predictive approaches.

While we argue that security is critical to the adoption of trust and reputation systems, we do not mean to suggest that other goals are unimportant. Indeed, improving predictive accuracy, maximizing social welfare, etc., are all worthwhile objectives. It is our contention, however, that a complimentary consideration of security needs to be included as well.

We focused in this paper on characterizing security for the buyer, but to be complete a system should also offer security to sellers, and to the market operator. We present a very brief outline of seller and market security here, to be expanded in future work.

Seller security is more difficult to define than buyer security. Buyers wish to receive goods as promised, and the required protection can be specified within individual transactions. By comparison, the primary goal of sellers is profit, which can be attacked in a variety of indirect ways. The example of ballot-stuffing [1] illustrates this point, wherein a coalition artificially inflates the attacker's rating in order to steal sales from the victim. Such activity can certainly cause damage, but this damage is more difficult to isolate than an unfulfilled commitment, and hence safety properties are more difficult to formulate. This is complicated by the fact that a trust system cannot guarantee certain levels of profit or revenue, since these will be affected by quality of marketing, legitimate competitive activity, etc. We suggest that for a system to be secure for the seller, the dishonest activity of any attacker should not reduce the seller's revenue (from agents other than the attacker(s)). Other approaches will also be considered.

To protect the market operator, or to ensure the continuing operation of the market where no operator can be identified (e.g., in some peer-to-peer systems), a proposed set of security requirements might consist of the following. Dishonest activity should not cause costs to be incurred by the 'market', because violation of this property would allow attacks to render the 'market' insolvent. Operation of the trust system should be budget balanced, or profitable. Finally, dishonest activity by participants should not cause the market to fail.

Our analysis revealed that Trunits is buyer secure, under specific conditions. Future work will pursue both the removal of some of these conditions through enhancements to the system, and providing protection for other market participants. First, since Trunits regulates only the behaviour of sellers, it does not provide seller security; further, it depends on the honesty of buyers to deliver buyer security, rather than providing a means to ensure buyers are honest. Trunits might incorporate protection from dishonest buyers directly. Alternatively, it is important to note that the model does not preclude the parallel use of another system for this purpose. For example, a mechanism like that of Jurca and Faltings [6] might be considered.

Trunits provides weak protection against the exit problem, by providing an incentive for sellers to stay in the market. This protection requires the assumption of unlimited future transactions for sellers, however, which is unrealistic. In our current work we are studying the treatment of trust as a tradable commodity (i.e., allowing agents to buy and sell trust). This notion, while counterintuitive, seems to allow us to cope with the exit problem without sacrificing the other beneficial properties of the model. Under this system, selling unneeded trunits can be made more profitable than using them to cheat, providing a strong incentive for honesty whether the seller exits the market or not. This approach also provides a natural solution to the 'start-up problem' noted above, since agents can purchase initial quantities of trunits.

In summary, the security framework presented in this paper offers a new direction for researchers in the area of trust and reputation to promote confidence in their models for real users. The Trunits model also provides a promising direction for designing electronic marketplaces in a way that offers guarantees of security to buyers.

References

[1] Bhattacharjee, R. and Goel, A. 2005. Avoiding ballot stuffing in eBay-like reputation systems. In *Proceeding of the 2005 ACM SIGCOMM Workshop on Economics of Peer-To-Peer Systems* (Philadelphia, Pennsylvania, USA, August 22 - 22, 2005). P2PECON '05. ACM Press, New York, NY, 133-137.

[2] Braynov, S. and Sandholm, T. 2002. Incentive compatible mechanism for trust revelation. In *Proceedings of the First international Joint Conference on Autonomous Agents and Multiagent Systems: Part 1* (Bologna, Italy, July 15 - 19, 2002). AAMAS '02. ACM Press, New York, NY, 310-311.

[3] Dellarocas, C. 2002. Goodwill Hunting: An Economically Efficient Online Feedback Mechanism for Environments with Variable Product Quality, Lecture Notes in Computer Science, Volume 2531, 238 – 252.

[4] eBay. http://pages.ebay.com/help/feedback

[5] Griffiths, N. 2005. Task delegation using experience-based multi-dimensional trust. In *Proceedings of the Fourth international Joint Conference on Autonomous Agents and Multiagent Systems* (The Netherlands, July 25 - 29, 2005). AAMAS '05. ACM Press, New York, NY, 489-496.

[6] Jurca, R. and Faltings, B. 2003. An incentive compatible reputation mechanism. In *Proceedings of the Second international Joint Conference on Autonomous Agents and Multiagent Systems* (Melbourne, Australia, July 14 - 18, 2003). AAMAS '03. ACM Press, New York, NY, 1026-1027.

[7] Kerr, R. and Cohen, R. 2006. Modeling Trust Using Transactional, Numerical Units. In *Proceedings of the Conference on Privacy, Security and Trust* (Markham, Ontario, Canada). PST '06.

[8] Marsh, S. *Formalising Trust as a Computational Concept.* PhD thesis. University of Stirling. 1994.

[9] Sabater, J. and Sierra, C. 2001. REGRET: reputation in gregarious societies. In *Proceedings of the Fifth international Conference on Autonomous Agents* (Montreal, Quebec, Canada). AGENTS '01. ACM Press, New York, NY, 194-195.

[10] Sabater, J. and Sierra, C. 2005. Review on Computational Trust and Reputation Models. *Artif. Intell. Rev.* 24, 1 (Sep. 2005), 33-60.

[11] Tran, T. and Cohen, R. 2002. A Learning Algorithm for Buying and Selling Agents in Electronic Marketplaces. In *Proceedings of the 15th Conference of the Canadian Society For Computational Studies of intelligence on Advances in Artificial intelligence* (May 27 - 29, 2002). R. Cohen and B. Spencer, Eds. Lecture Notes In Computer Science, vol. 2338. Springer-Verlag, London, 31-43.

[12] Tran, T. and Cohen, R. 2004. Improving User Satisfaction in Agent-Based Electronic Marketplaces by Reputation Modelling and Adjustable Product Quality. In *Proceedings of the Third international Joint Conference on Autonomous Agents and Multiagent Systems - Volume 2* (New York, New York, July 19 - 23, 2004). International Conference on Autonomous Agents. IEEE Computer Society, Washington, DC, 828-835.

[13] Yu, B. and Singh, M.P. 2002. Distributed Reputation Management for Electronic Commerce. In *Computational Intelligence*, 18(4): 535-549, 2002.

[14] Zacharia, G., Moukas, A., and Maes, P. 1999. Collaborative Reputation Mechanisms in Electronic Marketplaces. In *Proceedings of the Thirty-Second Annual Hawaii international Conference on System Sciences-Volume 8 - Volume 8* (January 05 - 08, 1999). HICSS. IEEE Computer Society, Washington, DC, 8026.

How to talk about reputation using a common ontology: from definition to implementation

Isaac Pinyol
Artificial Intelligence Research Institute
Campus UAB, Bellaterra
Barcelona, Spain
ipinyol@iiia.csic.es

Jordi Sabater-Mir
Artificial Intelligence Research Institute
Campus UAB, Bellaterra
Barcelona, Spain
jsabater@iiia.csic.es

Guifré Cuní
Artificial Intelligence Research Institute
Campus UAB, Bellaterra
Barcelona, Spain
cuni@iiia.csic.es

ABSTRACT

The field of multiagent systems has experienced an important growth and evolution in the past few years. Due to the agent autonomy and their need for cooperation, special attention has been paid to reputation mechanism. Several reputation models have appeared in literature offering solutions to this problem. However, each one uses their own concepts, terminology and ways to represent evaluations that make impossible an hypothetical transmission of social evaluations between agents using different reputation systems. In this paper we describe and present the implementation of an ontology of reputation as well as an ontology mapping mechanism that can be used for these dialogs. When transmitting social evaluations, agents will map them into elements of this common ontology, that the receiver agents will translate to elements understandable by their particular reputation system.

1. INTRODUCTION

The field of multiagent systems has experienced an important growth and evolution in the past few years. These systems can be seen as virtual societies composed of autonomous agents where there is a need to interact with other members of the society to achieve their goals. As in human societies, these interactions are not solely restricted to direct *trades*, but also include simple exchanges of information[1]. A scenario like this arises intrinsically the problem of partners selection via the detection of good or bad potential partners, or how agents evaluate the credibility of received information. Internet has shown us at least one solution to that problem, one based on electronic certificates and encrypted digital signatures that make us *trust* the site we are visiting and increase the credibility of information that we gathered. Notice that this *metainformation*, these certificates, are generated by the same owners. However, human societies along its history have been using other mechanisms that take advantage of exchanging information about the goodness of other members' performances. They are the trust and reputation mechanisms, a very powerful social control artifacts that have been studied from different perspectives, such as psychology (Bromley [3], Karlins et al. [15]), sociology (Buskens [4]), philosophy (Plato [18], Hume [14]) and economy (Marimon et al. [16], Celentani et al. [8]).

As described in Conte and Paolucci [9], reputation-based systems can be seen as a spontaneous and implicit norm-based system for social control. Every society has its own rules and norms that participants should follow to achieve a *well-fare* society. Social control mechanisms should be able to exclude not normative participants, but not even designed institutions can do that because they would need to keep a control over each transaction, and it is totally out of the question in distributed environments populated by autonomous entities. Here is where reputation-based systems can help to solve the problem. The social control they generate emerge implicitly in society, since non normative behaviors would tend to generate bad reputation that agents will take into account when selecting their partners, and therefore it may cause exclusion of the society due to social rejection. The idea of not having a totally explicit social control is crucial in multiagent systems and the last decade the interest on these mechanisms has considerably increased in the field. As a consequence numerous reputation models have appeared in the literature.

E-Commerce sites already use some of them (eBay [10], Amazon [2], OnSale [17]). These models consider reputation as a centralized global property of each particular agent (in these cases, sellers and buyers) calculated from the ratings that the system has received from users. These *reputation* values may be taken into account by potential buyers while selecting sellers. More sophisticated models ([1], [13], [21], [26], [5], [22], [19]) consider reputation as a subjective and contextualized property. Therefore every agent has its own reputation system that provides evaluations of other agents calculated from external communication and direct experience, giving the agent its own vision of the society. Furthermore, other models (see [6], [20]) take into account social information when providing these evaluations.

Once the actual situation is set, the beauty of such diversity clashes against an hypothetical virtual society whose agents do not use the same model. Since all of them use their own nomenclature, representation of the evaluations and even ways to interact, there is no way to establish communication between two agents using different reputation models that exchanges information about social evaluations. If the

[1]We understand as direct trade an interaction between agents (typically two) that involves cooperation or offering some service, and where it is possible to define a previous contract with the expectations that have the different parts. This is different from exchanging information, since in that case, there is no contract to be accomplished

source knew which model is using the recipient it might *convert* his/her own representation to the other one, but there is no reason to think that agents will know the internal functionalities of other participants.

This work gives an implementable solution to this problem. We propose a common ontology for reputation that allows the communication of social evaluations among agents that are using different reputation models. The implementation is based on a common API[2] that will work as a middleware between the common ontology and each particular model. An special emphasis is put on how to deal with the differences on the representation of the social evaluation values.

In Section 2 we explain the related work in reputation ontologies that exists in the literature. In Section 3, we describe in detail the proposed ontology and its main elements. Then, in Section 4 we define how we deal with the problem of value representation in social evaluations. In Section 5, we explain the design of the API interface. Finally, in Section 6, we show how the ontology can be mapped to several representative reputation models.

2. RELATED WORK

There is not much work done on this specific topic. When talking about ontologies for reputation there are in the literature two main works.

On one hand, in Casare et al. [7] is proposed a functional ontology the goal of which is to put together in a very conceptual level all the knowledge about reputation. It is based on the concepts defined in the Functional Ontology of Law [24]. This approach is interesting from a theoretical point of view because offers an structured definition of reputation and its related concepts, including processes of transmission, that could be used, as they claim in [7], as a *meta-model* of reputation concepts that could be mapped into existent models. In fact, in [25] is proposed a basic agent architecture to allow interoperability between agents using different reputation models that incorporates their functional ontology in this way. However, it is still a very conceptual approach that does not give solutions for the problems that in a possible real implementation we could find, like representation types for the evaluations, ontology mapping mechanisms etc...

On the other hand, as part of the European project eRep [12] a set of terms about reputation concepts has been defined. They can be seen in the *wiki* of the project [11]. The aim of this effort is to define an ontology that all partners participating in the project could use as a consensual starting point. This ontology describes in detail all the elements participating in social evaluations, as well as the processes of transmitting them. It also defines the main decisions concerning reputation that agents may take. This ontology is based on the cognitive theory of reputation defined in the book by Conte and Paolucci [9] and the Repage model [19].

The ontology presented in this article is a subset of the one used in eRep. Basically we are taking the elements of the eRep ontology concerning agents' beliefs that deal with social evaluations (see Section 3) adapting them to allow a direct and easy implementation. The main difference between the resulting ontology and the ontologies in eRep [12] and Casare et al. [7] is the focus on the implementation with an special emphasis on the representation of the social evaluation values (see Section 4), something not present in the previous approaches. For a description of the rationale behind each element of the ontology we refer the reader to the article about the Repage model [19].

[2]Standard acronym for Application Programming Interface

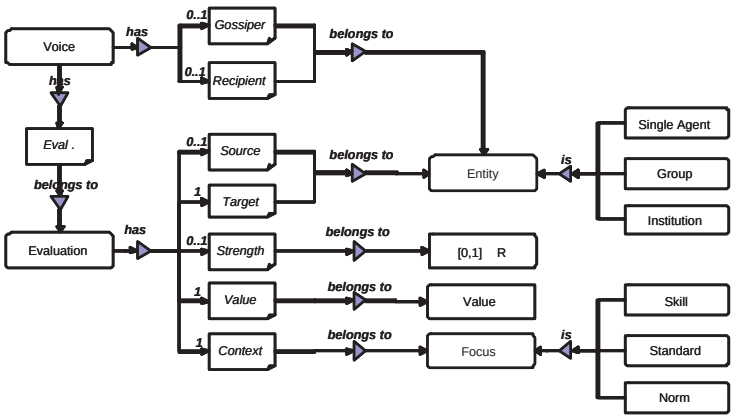

Figure 1: The main classes and components of a social evaluation and voice

3. THE ONTOLOGY

3.1 Preliminaries

In order to explain the elements of the ontology it is necessary to get in touch with some concepts defined in [9]. This cognitive theory keeps an essential difference between *image* and *reputation*. Both are social evaluations, evaluations that concern other participants in the society, individuals like single agents, or supra individuals like groups or collectives, but while *image* refers to evaluations that agents take as certainty, *reputation* refers to other's evaluations and therefore is considered a meta-belief, that is, a belief about other's belief. This brings us some important consequences, since accepting a meta-belief does not imply to accept the nested belief. Assuming that target agent A has some given *reputation* means that it is reputed with more or less goodness, and that such evaluation circulates on the society, but not necessary implies to share the evaluation itself. So, *reputation* refers to what is said, not what is true.

In the following subsections 3.2 and 3.3 we explain the elements of this ontology, from the characterization of a social evaluation to a taxonomy of what we understand as evaluative belief, that is, agent's beliefs that include some social evaluation. For a more formal definition of the ontology we refer to the work performed in the eRep project [12]. Here the objective is to provide a less formal description of the elements.

3.2 The social evaluation and voice

In Figure 1 we show the main elements involved in the definition of a social `Evaluation` and `Voice`. In the following paragraphs we describe each of them.

3.2.1 Entity

An `Entity` is any element of the society susceptible of being evaluated or having an active part in the generation or diffusion of evaluations. From the point of view of our theory an `Entity` can participate in the process of reputation in four different ways. On one hand, an `Entity` being evaluated is playing the social role of `Target`, meanwhile the one generating it, the role of `Source`. On the other hand, an `Entity` that spreads an `Evaluation` plays the role of `Gossiper`, and the one receiving it the role of `Recipient`.

3.2.2 Focus

The class `Focus` is the context of the evaluation. The goodness or badness of some entity's `Evaluation` is towards to an specific `Norm`, `Standard` or `Skill`. Agents can evaluate the same `Target` agent from different points of view. For instance, we can have a bad *image* of Agent A as a chess player, but a very good *image* of the same agent as a soccer player.

3.2.3 Value and Strength

The class `Value` contains the goodness or badness of the `Evaluation`, it is how good or bad is the `Target` entity in the context defined by the element `Focus` by the `Source` entity. In Section 4 we describe how we have represented this value. The `Strength`, represented with a bounded real belonging to the interval $[0, 1]$ is a subjective measure set by the `Source` that indicates how reliable is the evaluation, being 1 the maximum reliability. For instance, agent A may have had only one direct interaction with agent B, getting very good results. Then, agent A may generate a very good evaluation of this agent, but because he/she had only one interaction, agent A may not be totally sure of this `Evaluation`, specially when is communicating this result to another agent. The `Strength` is simply a measure that agents may use in order to be more accurate in their `Evaluations`.

As we will see, the `Strength` value is closely related to the uncertainty conversion (CU) described in Section 4.3. Both refer to some uncertainty associated to the `Value` class, but meanwhile the `Strength` is a subjective value that agents deliberately set, CU is intrinsic to how the `Value` is represented and the history of performed conversions, as will be shown in Section 4.3. It is a decision of each agent to use and combine these indicators or simply ignore them.

3.2.4 Evaluation and Voice

Finally, the class `Evaluation` encapsulates all the elements that participate in a social evaluation. It includes two instances of the class `Entity` playing the role of `Source` and `Target`, the `Context`, that belongs to the class `Focus`, and finally the `Value` of the evaluation. In the literature we found reputation and trust models that represent the `Value` with a simple boolean (good, bad), with a bounded real or even with fuzzy sets or probability distributions. The choice of this representation is one of the most delicate issues when developing a common ontology that should take into account a whole variety of representations, or at least, the most common ones. We discuss this problem and the solution we propose in Section 4.

Once defined the class `Evaluation`, we introduce the class `Voice`, that includes the necessary elements to represent the spreading of an `Evaluation`. A `Voice` is defined as a 'report on reputation'. For instance, "It IS SAID that John is good at playing soccer" is an example of a `Voice`. Apart from the `Evaluation` itself, it has two attributes belonging to `Entity` that identify the `Gossiper` and the `Recipient` of the `Voice`.

3.3 Evaluative Belief

Not all the `Evaluations` and `Voices` that agents have in their belief set have an specific meaning. I may have my own `Evaluation` of somebody, but I may have as well an `Evaluation` that somebody else gave me. Both are `Evaluations`, but their semantics are completely different. Taking into account this, the ontology describes a taxonomy of evaluative beliefs, beliefs that contain some `Evaluation`. Figure 2 shows the graphical representation of the taxonomy and the attributes that the leaf classes have. As a root we find the class `EvalBelief`, representing social evaluations that agents may have in their belief set. As showed in Figure 2 we divide the social evaluations in two categories, the classes `SimpleBelief`, a belief that the holding agent acknowledge as true, and `MetaBelief`, a belief about others' belief (in other words, a representation of other's mind). Let's describe each one of the bottom classes.

3.3.1 Image

An agent holding it, believes in the `Evaluation` that contains the class as attribute. In other words, an `Image` contains the believed opinion of the agent about a given `Target` with respect to a given `Focus`. The important point here is that the agent believes that this `Evaluation` is true.

3.3.2 Direct Experience

This class refers to the `Evaluation` that an agent creates from a single interaction or experience with another `Entity`. After an interaction, the generated outcome (the objective result of the transaction) is subjectively evaluated by the agent. This evaluation depends on the current mental state of the agent. The `Evaluation` is associated with a particular transaction, represented in the class by the attribute `IdTrans`[3] .

3.3.3 Shared Voice

An agent holding a `Shared Voice` will have the certainty that a perfectly identified set of `Entities` will acknowledge the existence of the `Voice` included in the class. Following the previous example, the fact that agent A, B, C and D inform agent X that "It IS SAID that John is good at playing soccer" is understood as a `Shared Voice`, since a set of agents (A,B,C,D) share the same `Voice` about "John". This class only refers to what the set of agents have informed, it is not a representation of what they believe. Therefore, a `Shared Voice` cannot be considered a meta-belief.

3.3.4 Shared Image

In this case, an agent holding a `Shared Image` believes that a perfectly identified set of `Entities` have as a belief the `Evaluation` included in the object. Clearly, this concept is considered a meta-belief in the sense that is a belief of other's belief, even though the set of `Entities` is known.

[3] In Figure 2 this attribute is presented belonging to the class `Real`. It simply means that `IdTrans` $\in \mathbb{R}$.

3.3.5 Reputation

In our approach, `Reputation` is a generalization and loss of reference of the `Shared Voice`. An agent holding a `Reputation` believes that most of the entities would acknowledge the existence of the `Voice` included in the class. It refers to what a target agent "IS SAID to BE" by most of the population or group. From the point of view of the holder agent, it is understood as a belief of others' belief in the sense that the holder agent believes that most of the population believe certain evaluation. For instance, taking again our example, to acknowledge that most of the people say that "John is good at playing soccer", can be understood as to believe that most of the people believe that "John is good as a soccer player", but this does not imply to really believe that John is good at it. Again, we consider `Reputation` as a meta-belief.

4. VALUE REPRESENTATION AND CONVERSIONS

One of the most important aspects of the reputation models are the value representations and semantics they used for the evaluations. In literature we find numerous models each of them using a different way to represent the value of an evaluation, from a simple boolean value indicating good or bad, to probability distributions and fuzzy sets. For instance, the eBay model uses a system of colored stars to show the reputation of a seller that could be seen as a simple real number between 0 and 100.000, meanwhile the Repage model uses a probability distribution over the discrete set Very Bad, Bad, Neutral, Good, Very Good. . When designing a common ontology on reputation it is crucial to have a consensual representation of the evaluations, since they are the key for the wished understandability between agents using different reputation models.

To decide a representation that everybody has to use is not an easy issue. For instance, if we took a simple boolean representation, models using a real number for evaluations would loose a lot on information when using the ontology. For this reason and after checking the most popular reputation models we decided to allow different types of representation (Boolean, Real, Discrete Set and Probability Distribution) providing automatic transformation functions between types, to allow connectivity between agents using different models that use different representations. The representation type is encapsulated in the class *Value* of the ontology.

In this section we explained these four types in detail and its semantics, as well as the transformation functions we designed.

4.1 Type definitions

- **Boolean Representation** (BO): In this case, evaluations take two possible values, good or bad. We define true as Good, and false as Bad.

- **Bounded Real Representation** (RE): Here, the value is a real number included in the bounded interval $[0, 1]$ where 0 is the worst evaluations, 1 the best evaluation and 0.5 the absolute neutral evaluation. The curve we have chosen indicating the level of goodness/badness is completely linear, from 0 to 1.

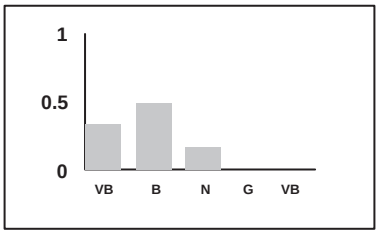

Figure 3: The graphical representation of the prob. distribution of (0.3,0.5,0.2,0,0)

- **Discrete Sets Representation** (DS): In this case, the value belongs to the following sorted discrete set {*Very Bad, Bad, Neutral, Good, Very Good*} ({VB,B,N,G,VG} from now on). Its semantics is intrinsic on the definition of each element of the sorted set.

- **Probabilistic Distribution Representation** (PD): Finally, this last representation applies a probability distribution (PD) over the sorted discrete set seen in the DS representation.

  Let L be the vector $[VB, B, N, G, VG]$ where $L_1 = VB, L_2 = B$ and so on. If X is a probability distribution over L then we define X_i as the probability of being evaluated as L_i. Given X is a probability distribution we have that $\sum_{i=1..5} X_i = 1$. For instance, we could have the distribution $(0.3, 0.5, 0.2, 0, 0)$ meaning that with probability of 0.3 the agent evaluated is *Very Bad*, with 0.5 that is *Bad* and with 0.2 that is *Neutral*. Graphically it can be represented as shown in figure 3

4.2 Transformation Functions

As we stated, we offer automatic transformations between the types we explained in the previous subsection. Some of them require justified arbitrary decisions. Table 1 summarizes all these transformations, however, in the following subsection we explain in detail each of them.

4.2.1 Transformations From PD

This is the most expressive type,being the only one offering probabilities. Because of that, it is difficult to find a direct transformation to the rest of types. Let's check each possible transformation:

→ *To Boolean (BO)*

In the BO we only have two values, false/true. The idea is that a PB value will converge to a *Good* if the probability distribution *tends* to values $\{Good, VeryGood\}$, and *Bad* if it *tends* to the values $\{VeryBad, Bad\}$. In our context, the word *tend* implies that we need an operation capable to transform a probability distribution element to an unidimensional number with some range, in where a threshold could tell us whether to transform the PD value to true or false. This operation is what we have called the center of mass (CM) of a PD element, $CM : PD \rightarrow [0, 1] \in I\!R$. It returns a bounded real number $\in [0, 1]$ indicating in terms of average how *GOOD* (converging to 1) or *BAD* (converging to 0) is the evaluation value represented in a probabilistic distribution. Of course 0.5 would be the absolute neutral.

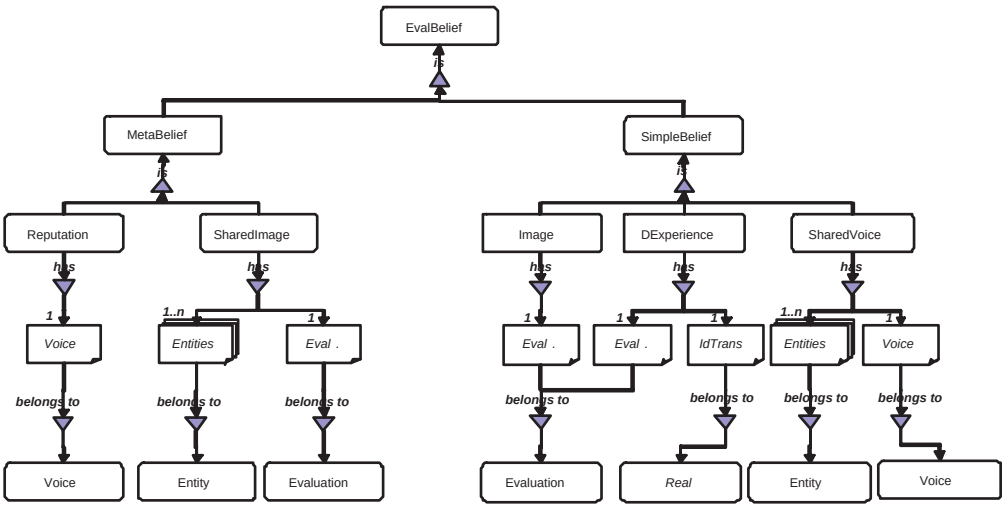

Figure 2: The taxonomy, membership relations and components of evaluative beliefs

Then, it is easy to think that values over 0.5 would indicate mostly good, and below mostly BAD. The value 0.5 will be our threshold. Let $X \in PB$, the function CM is defined as follows:

$$CM(X) = \frac{1}{10}\sum_{i=1}^{5}(2i-1)X_i \tag{1}$$

Then, to transform a given PB value X to a boolean is enough to evaluate the following boolean expression[4]: $CM(X) >= 0.5$

→ *To Real (RE)*

Once defined the center of mass function, let $X \in PB$ the transformation to a RE is: $CM(X)$

→ *To Discrete Set (DS)*

Notice that due to the semantics of the bounded real type (RE), the interval $[0,1]$ could be mapped into the ordered discrete set type (DS) $\{VB, B, N, G, VG\}$ in an easy way, keeping the semantics in the transformation. Let's define the function $R : [0,1] \in I\!R \rightarrow \{VB, B, N, G, VB\}$ that do this mapping as follows: Let $X \in RE$, then

$$R(X) = \begin{cases} VB & \text{if } 0 \leq X \leq 0.2; \\ B & \text{if } 0.2 < X \leq 0.4\ ; \\ N & \text{if } 0.4 < X \leq 0.6\ ; \\ G & \text{if } 0.6 < X \leq 0.8\ ; \\ VG & \text{if } 0.8 < X \leq 1\ . \end{cases} \tag{2}$$

Then, we have already seen how to transform an element in type PD to type RE. Then we can apply the R function over the result element in type RE, obtaining an element of type DS. Given that, the full transformation of an element $X \in PD$ would be calculated by the expression $R(CM(X))$.

[4]The decision of including 0.5 as a good evaluation is totally arbitrary, but consistent in all the transformations

4.2.2 Transformations From DS

→ *To Boolean (BO)*

In this case, the semantics of $TRUE$ in a boolean representation implies a possible condition of G or VG in a discrete set representation, and the $FALSE$ to a VB or B. Following the same decision we made in the previous subsection, the neutral value N should be considered $TRUE$ as well. Therefore, the transformation is complete. In order to do it more formal, we define the function $S : \{VB, B, N, G, VB\} \rightarrow [1,5] \in I\!N$ that returns the index position of a given element in the sorted set $\{VB, B, N, G, VG\}$, and it is defined as follows:

Let $X \in DS$, then

$$S(X) = \begin{cases} 1 & \text{if } X = VB; \\ 2 & \text{if } X = B\ ; \\ 3 & \text{if } X = N\ ; \\ 4 & \text{if } X = G\ ; \\ 5 & \text{if } X = VG\ . \end{cases} \tag{3}$$

Then, the transformation to BO is calculated with the expression $S(X) \geq 3$.

→ *To Real (RE)*

For this transformation we need to realize that function R (equation 2) divides the interval $[0,1]$ into five parts, each of them assigned to one of the values of the type DS. For instance, all elements between 0.2 and 0.4 are mapped in the element B (bad) of DS. Then, given an element of type DS, the possible real value equivalent should be included in the interval defined in function R. Then, for instance, a VB value as real would be in the interval $(0.2, 0.4]$.In fact, whatever value in the interval would be fine, however, we decided to pick the one just in the middle,0.3 that would reach the less error if we pick randomly one number from the interval. To formalize the transformation we decided to use a function that gives this central point.We define the function $C : [1,5] \in I\!N \rightarrow \{0.1, 0.3, 0.5, 0.7, 0.9\}$.Let $X \in [1,5]$ the function C is defined as $C(X) = \frac{2X-1}{10}$. Having it, we can

describe the transformation.

Let $X \in DS$, its transformation to RE would we calculated with the expression $C(S(X))$.

→ *To Probabilistic Distribution (PD)*

This case is quite simple, since a DS can be seen as a particular case of a PD, assigning the probability of 1 to the corresponding element of the set. For this reason we define the function $B : [1,5] \in I\!N \rightarrow PD$, that creates a PD element assigning a probability of 1 to the corresponding element and zero to the rest. Let $i \in [1,5] \in I\!N$ the function B is defined as:

$$B(i) = \{X \in PD : \forall_{r \neq i} X_r = 0 \& X_i = 1\} \tag{4}$$

Then, let $X \in DS$, its transformation to PD is calculated with the expression $B(S(X))$. Notice that the possible uncertainty that can have the source DS value, will be reflected in the strength and therefore will be transmitted as such to the strength of the target PD value. Therefore, it is not necessary to consider this uncertainty in the transformation of the value.

4.2.3 Transformations From RE

→ *To Boolean (BO)*

Following the same reasoning used in subsection 4.2.1, let $X \in RE$, the transformation between a RE type to a BO type is calculated evaluating the expression $X \geq 0.5$.

→ *To Discrete Set (DS)*

Let $X \in RE$ and having defined the function R (eq 2), the transformation to a DS is calculated using the expression $R(X)$.

→ *To Probabilistic Distribution (PD)*

The idea for converting an element $X \in RE$ to a PD type is to generate a PD element which center of mass is equal to X. It is obvious that there are infinite possible combinations. We decided to pick only two contiguous elements of the PD set and assign the corresponding probabilities in order to achieve the desirable center of mass.

Let's consider that i_1 and i_2 are the two index positions of the elements of PD that we choose to create the PD value. We decided the elements would be contiguous. To calculate them we use the function $R' : [0,1] \in I\!R \rightarrow [1,5] \in I\!N$ defined as

$$R'(X) = \begin{cases} 1 & \text{if } 0 \leq X \leq 0.2; \\ 2 & \text{if } 0.2 < X \leq 0.4\ ; \\ 3 & \text{if } 0.4 < X \leq 0.6\ ; \\ 4 & \text{if } 0.6 < X \leq 0.8\ ; \\ 5 & \text{if } 0.8 < X \leq 1\ . \end{cases} \tag{5}$$

This equation simply indicates to which interval belongs certain RE element[5]. We find i_1 as follow:

$$i_1 = min\{R'(X), R'(max\{X - 0.1, 0\})\} \tag{6}$$

[5]Notice that the equality $R'(X) = S(R(X))$ is hold

The index i_2 is the next following number, taking into account that the maximum allowed number is 5

$$i_2 = min\{5, i_1 + 1\} \tag{7}$$

Let $X \in RE$, then we need to find two probabilities, z and y, such that $z + y = 1$ and its center of mass (considering a PD element) is the original X, it means, that the following equation holds:

$$X = C(i_1)z + C(i_2)y \tag{8}$$

Solving the equation we have that $z = 1 - y$ and $y = \frac{X - C(i_1)}{C(i_2) - C(i_1)}$. Having this, we use the function $B' : [1,4] \in I\!N x[0,1] \in I\!R \rightarrow PD$ defined as

$$B'(i,p) = \{X \in PD : \forall_{r \neq i,i+1} X_r = 0 \& X_i = p \& X_{i+1} = 1-p\} \tag{9}$$

For instance, $B'(3, 0.3)$ returns as a PD element [0,0,0.3,0.7,0], $B'(1, 0.8)$ returns [0.8,0.2,0,0,0]. Finally, we have all the element to calculate the transformation from a given $X \in RE$ to a PD:

$$B'(i_1, 1 - \frac{X - C(i_1)}{C(i_2) - C(i_1)}) \tag{10}$$

4.2.4 Transformations From BO

In this case we start our explanation from the most expressive type (PD) .

→ *To Probabilistic Distribution (PD)*

In this point, it is important to notice that knowing one of the two possible values of the boolean representation, implies certain conditions that help to decide the transformations. Having only a $FALSE$ we can say that whatever representation to be transformed, the transformed value should be in the side of the bad evaluations. In fact, with the type of probabilistic distribution, because it works with probabilities, we could think that in the $FALSE$ value for instance, there is the same probability of being Very Bad, and Bad, and less probability although some , of being Neutral. The same happens with the value TRUE. Then, we define the following constants $B_F = [2/5, 2/5, 1/5, 0, 0]$ and $B_T = [0, 0, 1/5, 2/5, 2/5]$ belonging to PD. The transformation function from a BO to PD is then quite simple. Let $X \in BO$, the conversion is:

$$\begin{array}{l} B_T : X \\ B_F : \neg X \end{array} \tag{11}$$

→ *To Discrete Set (DS)*

Here we should decide which values of the considered bad evaluations or good evaluations correspond to the $FALSE$ and $TRUE$ values respectively. However, once fixed the constants B_F and B_T to represent FALSE and TRUE values in probabilistic distribution, we have to take them as a base to decide the transformation. The idea is that if B_F represents a $FALS$E, its center of mass (function CM) has to indicate the position in the interval $[0,1]$ that the FALSE value represents, and having it, the function R would determine which element of the discrete set represents the $FALSE$ value. The same reasoning can be made for the $TRUE$

value. Then, let $X \in BO$, the transformation to DS is:

$$\begin{array}{l} R(CM(B_T)) : X \\ R(CM(B_F)) : \neg X \end{array} \tag{12}$$

Actually, the $FALSE$ value goes to B, and $TRUE$ goes to G.

→ *To Real (RE)*

If we have used the function CM in the previous transformation, it is clear that to keep consistency, having $X \in BO$ the transformation should be:

$$\begin{array}{l} CM(B_T) : X \\ CM(B_F) : \neg X \end{array} \tag{13}$$

4.3 Conversion Uncertainty

As we stated, in a society where participants may be using different kinds of reputation and trust models, the necessity of exchanging social evaluations to achieve their goals may drive in a situation where an agent that uses a boolean representation needs to communicate with one that uses probabilistic distribution, and then, a conversion of representations must take place. However, type conversions carry lose of precision and addition of uncertainty. As an example, some evaluation represented as a boolean that is Bad, when is converted to a real representation may have an evaluation from 0 to 0.5 (not included), when is converted to discrete set, it may be one of these elements $\{VB, B\}$ etc... This factor of uncertainty that is added when we convert a value to a more expressive representation is what we call Conversion Uncertainty (CU), and is an information that the recipient should know.

4.3.1 Entropy of the representations

In order to calculate the CU we decided to use the information theory approach introduced by Shannon [23]. In this context, the entropy of a random variable X ($H(X)$) can be understood as the *uncertainty* of X, and is defined as

$$H(X) = -\sum_{x \in X} p(X = x) \log(p(X = x)) \tag{14}$$

From Shannons's theory we can define the conditional entropy as follows:

$$H(X|Y = x) = -\sum_{x \in X} p(X = x|y = Y) \log(p(X = x|Y = y)) \tag{15}$$

and finally,

$$H(X|Y) = -\sum_{y \in Y} p(Y = y) H(X|Y = y) \tag{16}$$

Now, we consider each one of the representations as discrete random variables. Without lose of generality we can discretize the Real representation using two digits (in base 10), having a hundred possible values. The fact of using 100 divisions for the interval and not a bigger amount is because we think that a greater precision is completely unnecessary (and even counterproductive) given the nature of the measure that is represented with this value, that is, a measure of a social evaluation. At the same time, taking into account the hundred possible values of a Real number, we can count the number of elements of the Probabilistic Distribution representation considering all possible combinations of distribution values that need to achieve the unit[6].Let A be this number, the following equation is hold:

Type	Entropy
BO	1.00
DS	2.31
RE	6.64
PD	22.19

Table 2: Entropies of the type representation

	BO	DS	RE	PD
BO	0	1.29	5.64	21.19
DS	0	0	4.32	19.89
RE	0	0	0	15.55
PD	0	0	0	0

Table 3: CU values

$$A = \sum_{i=0}^{4} \binom{5}{i} \binom{100}{4-i} = 4780230 \tag{17}$$

Each random variable has as elements each possible element of the representations and it's probability distribution is totally equiprobable. Then, we define the conversion uncertainty of the source random variable X to the target random variable Y as

$$CU(X, Y) = H(Y|X) \tag{18}$$

In other words, CU is the increment of uncertainty produced when a value is represented in X and is converted to a value of type Y that is more expressive. There is a set of candidate values that makes conditional entropy increase. The values of the entropy of each type is showed in table 2. See appendix A for the details of the calculus.

The CU values for each conversion is showed in table 3. Each row is the source and each column is the target.

4.3.2 CU usage. An example

An example will illustrate the usage of the CU value. Let's suppose agent A is using a Boolean representation, and generates and sends an evaluation to agent B that uses a discrete set representation. Agent B would reach the evaluation with a CU value of 1.29. If agent B send the same evaluation to agent C that uses a probabilistic distribution, agent C would receive the evaluation with a CU value of 1.29 (the base value coming from the communication) plus 19.89 (from the type conversion between DS to PD), it means, a CU value of 21.18. The idea is that the uncertainty of the evaluations is accumulative, without allowing loops (if the evaluation goes back to an agent using a representation type that have already been used in some transformation there is no addition of uncertainty)

[6]This is a combinatorial problem related to the famous Balls and Bins problem

	BO	RE	DS	PD
X:BO	X	$CM(B_T) : X$ $CM(B_F) : \neg X$	$R(CM(B_T)) : X$ $R(CM(B_F)) : \neg X$	$B_T : X$ $B_F : \neg X$
$X : RE$	$X \geq 0.5$	X	R(X)	$i_1 = min\{R'(X)$ $R'(X - 0.1)\}$ $i_2 = min\{5, i_1 + 1\}$ $B'(i_1, 1 - \frac{X - C(i_1)}{C(i_2) - C(i_1)})$
X:DS	$S(X) \geq 3$	$C(S(X))$	X	$B(S(X))$
X:PD	$CM(X) \geq 0.5$	$CM(X)$	R(CM(X))	X

Table 1: Conversion table

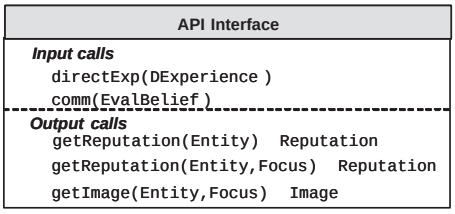

Figure 4: The first level of the API interface

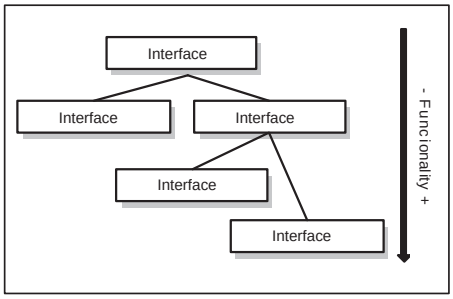

Figure 5: API interface hierarchy

5. THE API DESIGN

So far, we have defined an ontology that allows agents to communicate and reason on reputation concepts independently of their reputation model they are using. However, we need a module that translates the elements of the ontology to elements understandable by each particular reputation model, an ontology mapping. We defined then an API as an interface with a set of common operations whose inputs and outputs are elements of the ontology, and must be implemented for each particular model. For simple models, it will be enough to implement the operations for the first level of the API interface (see Figure 4). For more complex models, new operations may be needed to use all the expressiveness, getting a hierarchy of interfaces (see Figure 5).

- **Input calls**

  In the first level of the hierarchy (see Figure 4) we defined two functions needed to feed most of the trust and reputation systems that currently exist. A direct experience is the main source to build up an image. Then, an agent that interacts with another agent signing a contract would generate an outcome that would be the direct experience of the agent on this particular transaction. This fact will be notified to the reputation model used by the agent, if it is a subjective model, through the API function *directExp*. The other main way to feed reputation models is through communications from other entities. Then, when a communications is received the model will be fed using the API function *comm*. Notice that in principle, whatever evaluative belief can be communicated. The implementation of each API interface will be in charge of translating these entrances from elements of the ontology to elements understandable by the reputation model, and therefore, decide what means in each case the fact of communicating, for instance, an image, reputation or whatever evaluative belief.

- **Output calls**

  In this first level we decided to specify only three functions that give support to the most common queries that any agent may require from a reputation model (see Figure 4): knowing the reputation and the image a of certain entity. The reason we describe two versions of *getReputation* is due to the existence of some classical reputation models, like eBay [10], that use implicit context information, and in terms of the ontology means that there is no *Focus*. Again, here the API will query the reputation system in its own understandable language, that will respond with certain value that the same API will translate to the respective object of the ontology. Notice that the main point is that the decision making module of the agent always reason over the concepts described by the ontology. That would allow communication between agents that use different reputation models (if they have an implementation of the respective API).

6. IMPLEMENTATION AND EXAMPLES

We decided to implement several well known reputation and trust models and its respective API interfaces to prove the effectiveness of our approach. In this section we briefly describe these models and the first level of the API interface.

6.1 eBay model

eBay site [10] is one of the most concurred (if not the most) online marketplace in the world with more that 50 million registered users. As we stated above, eBay reputation model considers reputation as a public and centralized value that

is not dependent on the context [7]. In this case, users can rate the seller they have bought items from after each transaction, with values of +1, 0 , -1. The reputation value of the sellers is calculated with the sum of all the ratings over the last six month.

- **API input calls**

  Taking into account that this is a centralized model, it is fed only with external ratings that in the ontology are communication of direct experiences. From our approach, the fact of rating with +1, 0, -1 a given transaction is a subjective evaluation of this transaction that coincide with the class *DirectExperience*. Then, the API needs to implement the function *comm(DExperience)*. The representation value we have chosen in this case is the discrete set, with the following equivalence: $\{-1 \leftrightarrow VB, 0 \leftrightarrow N, +1 \leftrightarrow VG\}$

- **API output calls**

  The model provides to the users the plain sum of all the ratings for each seller with a colored stars system. According to the definition in the theory of reference used in this work (see Section 3), this value can be seen as sellers' reputation, and therefore, the API function to be implemented should be *getReputation(Entity)* $\rightarrow$ *Reputation* The best way to represent the value of the Reputation object clearly is the bounded real. Since eBay punctuation goes from 0 to 100000, a simple normalized transformation to the interval [0,1] seems to be enough. However, notice that the colored stars representation does not follow a linear curve. From a semantic point of view and in our value representation, 0 means very bad reputation, 0.5 neutral reputation, and 1 very good reputation, with a totally linear function. In eBay, having more that 10 point is already considered a good reputation. The next step in the scale is more than 100 points (with a different colored star), and the next is more than 500. In conclusion there is no lineal relation between the punctuation and the semantic representation of the stars. Then, it is necessary a transformation from the ontology representation value to the eBay scale. A possible transformation function is described in the following equation:

$$H : [0, 100000] \rightarrow [0, 1] \tag{19}$$

$$H(X) = \begin{cases} 0 & \text{if } X < 10; \\ 1 & \text{if } X > 100000; \\ \frac{log(X)-0.5}{8} + 0.5 & otherwise. \end{cases} \tag{20}$$

  In this case, no strength value is taken into account, considering every reputation value with the maximum possible strength.

6.2 Sporas model

This model [27] considers reputation, like eBay model, as a public and centralized value. In this case though, only the most recent ratings between two users are considered. Furthermore, users with very high reputation values experience much smaller changes than with low reputation after each update due to the aggregation function. In fact, the model is parameterizable with different parameters such like the range of the reputation values or the number of ratings to consider for the calculus. For the integration of the model with the API, we need to know how reputation values are presented and its semantics, as well as the rating values. Like in eBay, the system is fed by communication of direct experiences of the users rating sellers. Then the API function *comm(DExperience)* needs to be implemented. Because users query the system asking for reputation values the function *getReputation(Entity)* $\rightarrow$ *Reputation* needs to be implemented as well. In both cases, the representation value of the evaluation clearly is the bounded real, since the reputation value as well as the rating measures are numbers belonging to the interval defined by the range, following a totally linear curve: that is, the minimum value of the range is the worst reputation value (and rating value) and the maximum value of the range is the best one, with a linear gradient from the minimum to the maximum. Then a simple normalized transformation can be done.

[7] In fact, the context is determined by the environment where this model is used: an e-Auction market

6.3 Abdul-Rahman and Hailes Model

This model [1] uses the term *trust*, understanding it as a distributed and subjective value. It means that every agent has its own reputation system as a submodule of the architecture of the agent. In this case, social evaluations take into account the context, the focus element in terms of the ontology. The model is fed by two sources: direct experiences and third party communications of direct experiences. The representation of the evaluations is done in terms of the discrete set {*vt (very trustworthiness), t (trustworthiness), u (untrustworthiness), vu (very untrustworthiness)*}. Then, for each agent and context the system keeps a tuple with the number of past own experiences or communicated experiences in each category. For instance, agent A may have a tuple of agent B as a seller like $(0, 0, 2, 3)$, meaning that agent A has received or experienced 2 results as untrustworthiness and 3 as very untrustworthiness. Finally the *trust* value is computed taking the maximum of the tuple values. In our example for agent A, agent B as a seller would be very untrustworthy. In case of tie between *vt* and *t* and between *u* and *vu* the system gives the values U^+ (mostly trustworthy) and U^- (mostly untrustworthy) respectively. In any other tie case the system returns U^0 (neutral).

- **API input calls**

  As we described, this model is fed by communications of direct experiences from other agents, and direct experiences of the same agent. In this case then we need to implement the functions *comm(DExperience)* and *directExp(DExperience)*. Since in the model direct experiences are evaluated with one of the four categories, {vt,t,u,vu} seems logical to represent it in the ontology with the discrete set, using the following equivalences: vt $\leftrightarrow$ VG, t $\leftrightarrow$ G, u $\leftrightarrow$ B, vu $\leftrightarrow$ VB.

- **API output calls**

  The *trust* measure that the model provides, in terms of the ontology, is close to the concept of *image*, because agents accept as true the measure. Then, we need the function *getImage(Entity,Focus)* $\rightarrow$ *Image*. Here, we decided to use as a representation value of the image

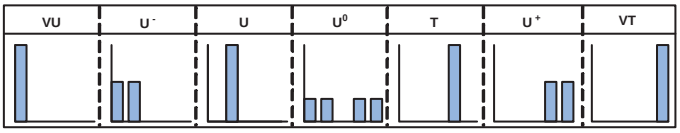

Figure 6: Transformation of Abdul-Rahman and Hailes model trust values to probabilistic distribution(PD)

evaluation a probabilistic distribution type, since the semantics of U^+, U^+ and U^0 involve more than one *trust* measure[8]. In this model, when the answer is *vu* means that for sure the agent is very untrustworthy, meaning that with a probability of 1 the agent is very bad (in terms of the ontology). In the same way when the model achieves the value U^- means that with the same probability, the agent is untrustworthy or very untrustworthy, that can be translated into probabilistic distribution as having a probability of 0.5 of being VB and 0.5 of being B. The total transformation table is showed in Figure 6.

6.4 Repage

Repage [19] is a distributed computational system based in the cognitive theory of reputation described in [9]. Consequently the main elements of the presented ontology appears and its semantics is very similar. In this case, the model keeps as well a difference between Image and Reputation. The memory consists of a set of connected predicates (beliefs) conceptually organized in layers, where in the bottom we find non evaluated predicates, like communications from other partners and contracts (before a direct interaction) and fulfillments (the result of the direct interaction). From communicated reputations of the same target and context, a shared voice predicate is generated that when it is strong enough, ends up as a reputation predicate. Similarly, communicated images and third party images[9] of the same target and context generate what is called a shared evaluation predicate, that with the outcome predicate obtained from direct interactions (through the subjective evaluation of fulfillments) may generate an image predicate over the target in the context. The representation value of social evaluations uses the probabilistic distribution we have described in this paper together with a strength value, a real bounded number. We refer to [19] for a more detailed description of all these concepts and the aggregation functions used to add up different probabilistic distribution values.

- **API input calls**

  This model is fed by communications of images, reputations and third party images, and by direct experience using the mechanism of signing a contract, generating a fulfillment of the transaction and evaluating it generating an outcome. Communication issues require the implementation of the functions *comm(Image)* and *comm(Reputation)*. Notice that third party images are images where the source and the gossiper are not the same. However, the API function *comm* always carries the origin of the communication, the gossiper. If it is not the same that the source of the evaluation of the image, it is a third party image.

  Direct interaction requires a small change. Notice that for each interaction, there is a contract predicate, a fulfillment predicate and an outcome predicate, being basically the goodness or badness of the fulfillment respect the original contract. This calculus is done inside the model. However, in order to use this ontology, the concept of *outcome* has to be seen as a direct experience object, and therefore, needs to be calculated outside the model, in the agent part. The situation is not critical though. Already in Repage, outcome predicates are totally context dependent, and for each situation have to be implemented. Then the API function *directExp(DExperience)* will represent the insertion of an outcome in the Repage system. Because this model uses already a probabilistic distribution representation of the evaluations, there is no reason to not use the same type in the direct experience object.

- **API output calls**

  The system basically provides images and reputations of a given agent in certain context. Then, the main API operations to implement are *getImage(Entity,Focus)* → *Image* and *getReputation(Entity,Focus)* → *Reputation*. The mapping from the model to the elements of the ontology is direct, besides minor structural points. Furthermore, there is no need to readjust evaluations values representation, since we can keep using the same probabilistic distribution.

6.5 Conclusions on the API implementation part

What we have showed in this section is a small set of representative reputation models and its mapping to the ontology through an implementation of a set of operations from an API interface. In this work we only implemented the first level of the API hierarchy of each model, but more levels could be implemented in function of the necessities and possibilities of each model. For instance, Repage model [19] incorporates the concept of *Shared Evaluation* (see subsection 6.4) that can be mapped to the class *SharedImage*. In this second level of the Repage API hierarchy we could incorporate functions to access to these elements.

7. CONCLUSIONS AND FUTURE WORK

In this work we have presented an implementable solution for the communication of social evaluations between agents using different reputation models that is required in multi-agent paradigms. This solution offers an ontology of reputation that agents can share as a common *language* to exchange reputation concepts. The communication through a common ontology requires an ontology mapping linking the own ontology used by each reputation model with this common one. We implemented this mapping using a hierarchy

[8]For instance, U^- is done when there is a tie between *vu* and *u*, meaning that both options have a 50% of probability

[9]A third party image communication is a communicated image where the source of the image is not the source of the communication. For instance, agent A may communicate to somebody that for agent C, agent D is BAD

of API interfaces that offer set of primitive function to be implemented for every specific model.

This approach allows to have a library of APIs that users could use when designing decision making modules without having to think in the peculiarities of each reputation model, and having total freedom for using different reputation models in different agents that participate in the same society. As a future work, we plan to study also the use of other entropy measures like differential entropy or relative entropy and mutual information for the calculation of the CU.

Another line of future research is the use of what is called dynamic ontology alignment. Notice that the presented ontology is static. It means that for every new reputation model we would need a new API implementation and, even when currently there is no model that could use all the expressiveness of this ontology (except may be Repage [19]), it is not a guarantee in the future. The main idea behind the dynamic ontology alignment is that by using a very basic but commond set of elements that two agents share, they can build at runtime a common ontology specific for that interaction. In our domain, it would mean they were able to understand, for instance, the concepts of goodness or badness of the other agents. We expect to design mechanisms based on ontology alignment theory to achieve this objective.

8. ACKNOWLEDGMENTS

This work was supported by the European Community under the FP6 programme (eRep project CIT5-028575, OpenKnowledge project FP6-027253 and SocialRep project MERG-CT-2005/029161). Jordi Sabater-Mir enjoys a Ramon y Cajal contract from the Spanish Government.

9. REFERENCES

[1] A. Abdul-Rahman and S. Hailes. Supporting trust in virtual communities. In *Proceedings of the Hawaii's International Conference on Systems Sciences, Maui, Hawaii*, 2000.

[2] Amazon. *Amazon Auctions.* http://auctions.amazon.com, 2002.

[3] D. B. Bromley. *Reputation, Image and Impression Management.* John Wiley & Sons, 1993.

[4] V. Buskens. The social structure of trust. *Social Networks*, (20):265—298, 1998.

[5] J. Carbo, J. Molina, and J. Davila. Trust management through fuzzy reputation. *Int. Journal in Cooperative Information Systems*, pages in–press, 2002.

[6] J. Carter, E. Bitting, and A. Ghorbani. Reputation formalization for an information-sharing multi-agent sytem. *Computational Intelligence*, 18(2):515—534, 2002.

[7] S. J. Casare and J. S. Sichman. Towards a functional ontology of reputation. In *Proceedings of the Fourth International Joint Conference on Autonomous Agents and Multi Agent Systems, Utrecht, The Netherlands*, pages 505–511. ACM Press, 2005.

[8] M. Celentani, D. Fudenberg, D. K. Levine, and W. Psendorfer. Maintaining a reputation against a long-lived opponent. *Econometrica*, 64(3):691—704, 1966.

[9] R. Conte and M. Paolucci. *Reputation in artificial societies: Social beliefs for social order.* Kluwer Academic Publishers, 2002.

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

[11] eRep. *eRep Wiki.* http://megatron.iiia.csic.es/mediawiki/index.php, 2006.

[12] eRep. *eRep:Social Knowledge for e-Governance.* http://megatron.iiia.csic.es/eRep, 2006.

[13] B. Esfandiari and S. Chandrasekharan. On how agents make friends: Mechanisms for trust acquisition. In *Proceedings of the Fourth Workshop on Deception, Fraud and Trust in Agent Societies, Montreal, Canada*, pages 27—34, 2001.

[14] D. Hume. *A Treatise of Human Nature (1737).* Oxford: Clarendon Press, 1975.

[15] M. Karlins and H. I. Abelson. *Persuasion, how opinion and attitudes are changed.* Crosby Lockwood & Son, 1970.

[16] R. Marimon, J. P. Nicolini, and P. Teles. Competition and reputation. In *Proceedings of the World Conference Econometric Society, Seattle*, 2000.

[17] OnSale. *OnSale.* http://www.onsale.com, 2002.

[18] Plato. *The Republic (370BC).* Viking Press, 1955.

[19] J. Sabater, M. Paolucci, and R. Conte. Repage: Reputation and image among limited autonomous partners. *Journal of Artificial Societies and Social Simulation*, 9(2), 2006.

[20] J. Sabater and C. Sierra. Regret: A reputation model for gregarious societies. In *Proceedings of the Fourth Workshop on Deception, Fraud and Trust in Agent Societies, Montreal, Canada*, pages 61—69, 2001.

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

[22] S. Sen and N. Sajja. Robustness of reputation-based trust: Boolean case. In *Proceedings of the first international joint conference on autonomous agents and multiagent systems (AAMAS-02), Bologna, Italy*, pages 288—293, 2002.

[23] C. Shannon. A mathematical theory of communication. *Bell System Thecnical Journal*, 27:379—423, 1948.

[24] A. Valente. *Legal Knowledge Engineering - A modelling approach.* IOS Press, 1995.

[25] L. Vercouter, S. Casare, J. Sichman, and A. Brando. An experience on reputation models interoperability based on a functional ontology. In *IJCAI '07: Proceedings of the Twentieth International Join COnference on Artificial Inteligence, Hyderabad, India*, pages 617—622, 2007.

[26] B. Yu and M. P. Singh. Towards a probabilistic model of distributed reputation management. In *Proceedings of the Fourth Workshop on Deception, Fraud and Trust in Agent Societies, Montreal, Canada*, pages 125—137, 2001.

[27] G. Zacharia. Collaborative reputation mechanisms for online communities. Master's thesis, Massachusetts Institute of Technology, September 1999.

APPENDIX

A. CALCULUS OF CU

A.1 $CU(BO, DS) = 1.29$

Considering True as t and False as f:

$$CU(BO, DS) = H(DS|BO) \tag{21}$$

Knowing that $p(BO = t) = p(BO = f) = \frac{1}{2}$ we can write that

$$CU(BO, DS) = \frac{1}{2}H(DS|BO = t) + \frac{1}{2}H(DS|BO = f) \tag{22}$$

At this point, when $BO = t$ and following our semantic interpretation we know that it may refer to one value of the set $\{N, G, VG\}$, and if $BO = f$ of the set {VB,B}. Then $P(DS = \{VB\}|BO = f) = P(DS = \{B\}|BO = f) = 1/2$ (zero in other values of DS) and $P(DS = \{N\}|BO = t) = P(DS = \{G\}|BO = t) = P(DS = \{VG\}|BO = t) = 1/3$ (zero in other values of LL). Then, following the previous equations and developing the entropy formula we have that

$$H(DS|BO = t) = -3\frac{1}{3}\log(\frac{1}{3}) \approx 1.58 \tag{23}$$

$$H(DS|BO = f) = -2\frac{1}{2}\log(\frac{1}{2}) = 1 \tag{24}$$

finally, computing the equation 22 we have

$$CU(BO, DS) = 0.79 + 0.5 = 1.29 \tag{25}$$

A.2 $CU(BO, RE) = 5.64$

Here, knowing that $BO = t$ our semantic indicates that as a real, it could be a value from 0.50 and 1, then $\forall_{i\in[0,1]}p(RE = i|BO = t) = p(RE = i|BO = f) = 1/50$ and therefore,

$$H(RE|BO = t) = H(RE|BO = f) = -50\frac{1}{50}\log(\frac{1}{50}) \approx 5.64 \tag{26}$$

$$CU(BO, RE) = 5.64 \tag{27}$$

A.3 $CU(BO, PD) = 21.19$

Having in mind the total number possible elements in PD (see equation 17), we know that $BO = t$ implies that whatever representation of PD will tend towards a good evaluation, it means that the probability of being good is higher that the opposite. That eliminates exactly 50% of all the representations, and therefore

$$\forall_{i\in PD}p(PD = i|BO = t) = p(PD = i|BO = f) = \frac{2}{A} \tag{28}$$

$$H(PD|BO = t) = H(PD|BO = f) = -\frac{A}{2}\frac{2}{A}\log(\frac{2}{A}) \approx 21.19 \tag{29}$$

$$CU(BO, PD) = 21.19 \tag{30}$$

A.4 $CU(DS, RE) = 4.32$

Following the same reasoning:

$$CU(DS, RE) = H(RE|DS) \tag{31}$$

$$CU(DS, RE) = \sum_{i\in\{vb,b,n,g,vg\}} \frac{1}{5}H(RE|DS = i) \tag{32}$$

Notice that in this case, the difference between a Real and DS is that the first is continuous and the second discrete. Then, dividing the $[0, 1]$ interval into five identical parts, and assigning each of them into a value of DS we have the problem almost done. In this situation, each value of DS correspond to a 20 values of Real, and therefore,

$$\forall_{i\in\{vb,b,n,g,vb\}}\forall_{j\in[0,1]}p(RE = j|DS = i) = \frac{1}{20} \tag{33}$$

Then,

$$\forall_{i\in\{vb,b,n,g,vb\}}H(RE|DS = i) = -20\frac{1}{20}\log(\frac{1}{20}) \approx 4.32 \tag{34}$$

and then,

$$CU(DS, RE) = 4.32 \tag{35}$$

A.5 $CU(DS, PD) = 19.89$

The key in all the calculus is in the fact that each element of DS may correspond to a set of elements of PD whose center of mass is included in the interval corresponding to the function defined in R'. In the same way we have discretized the interval $[0, 1]$ in five parts, for each of these intervals we have a total of $\frac{A}{5}$ elements of PD with a center of mass that points inside the interval. Therefore, we can establish the following statement:

$$\forall_{i\in\{vb,b,n,g,vb\}}\forall_{j\in PD}p(PD = j|DS = i) = \frac{5}{A} \tag{36}$$

and,

$$\forall_{i\in\{vb,b,n,g,vb\}}H(PD|DS = i) = -\frac{A}{5}\frac{5}{A}\log(\frac{5}{A}) \approx 19.89 \tag{37}$$

then,

$$CU(DS, PD) \approx 19.89 \tag{38}$$

A.6 $CU(RE, PD) = 15.55$

Following the same reasoning than in the previous point, the number of elements of PD whose center of mass is the one being converted is approximately $\frac{A}{100}$, and therefore,

$$\forall_{i\in[0,1]}\forall_{j\in PD}p(PD = j|RE = i) = \frac{100}{A} \tag{39}$$

and,

$$\forall_{i\in\{vb,b,n,g,vb\}}H(PD|RE = i) = -\frac{A}{100}\frac{100}{A}\log(\frac{100}{A}) \approx 15.55 \tag{40}$$

then,

$$CU(RE, PD) = 15.55 \tag{41}$$

The ART of IAM: The Winning Strategy for the 2006 Competition

[ART TESTBED SPECIAL SESSION]

W. T. Luke Teacy, Trung Dong Huynh, Rajdeep K. Dash, Nicholas R. Jennings and Jigar Patel
School of Electronics and Computer Science
University of Southampton
Southampton, SO17 1BJ, United Kingdom
{wtlt,tdh,rkd,nrj}@ecs.soton.ac.uk
jpatel@mckinsey.com

Michael Luck
Department of Computer Science
King's College
London, WC2R 2LS
United Kingdom
michael.luck@kcl.ac.uk

ABSTRACT

In many dynamic open systems, agents have to interact with one another to achieve their goals. Here, agents may be self-interested, and when trusted to perform an action for others, may betray that trust by not performing the actions as required. In addition, due to the size of such systems, agents will often interact with other agents with which they have little or no past experience. This situation has led to the development of a number of trust and reputation models, which aim to facilitate an agent's decision making in the face of uncertainty regarding the behaviour of its peers. However, these multifarious models employ a variety of different representations of trust between agents, and measure performance in many different ways. This has made it hard to adequately evaluate the relative properties of different models, raising the need for a common platform on which to compare competing mechanisms. To this end, the ART Testbed Competition has been proposed, in which agents using different trust models compete against each other to provide services in an open marketplace. In this paper, we present the winning strategy for this competition in 2006, provide an analysis of the factors that led to this success, and discuss lessons learnt from the competition about issues of trust in multiagent systems in general. Our strategy, IAM, is Intelligent (using statistical models for opponent modelling), Abstemious (spending its money parsimoniously based on its trust model) and Moral (providing fair and honest feedback to those that request it).

1. INTRODUCTION

Trust constitutes an important facet of multiagent systems research since it provides a form of distributed social control within highly dynamic and open systems whereby agents form opinions about other agents based on their past interactions, as well as from reports of other agents [8]. As a result, a number of trust models and strategies have been proposed in order to deal with distinct aspects of the interactions between agents (e.g. to deal with lying agents [1], to model and learn the behaviour of other agents [10, 9] and to fuse information from disparate sources and models [5]). This, in turn, has rendered it hard to compare trust strategies since the underlying problem addressed by these strategies has been different. Therefore, in order to provide a common platform on which "researchers can compare their technologies against objective metrics" the Agent Reputation and Trust (ART) Testbed Competition has been proposed [3].

The ART Testbed simulates an environment consisting of service providers that compete in order to provide information services. There is a fixed total number of clients who are apportioned between the service providers according to the comparative quality of their service provision. Each of these information providers needs to spend money in order to gain information. Furthermore, they can improve their quality of service by requesting (against a payment) the other agents for information. However, it is not necessary that the requested agents will provide good information. In fact, as a result of the competition between the agents, it is quite likely that the agents will provide bad information. Thus, within this competition, trust and reputation become important metrics with which to measure the reliability of other agents. This is because trust and reputation measure the certainty with which each of the other agents provides good opinions and reputation information.

In this paper, we describe the *winning* strategy, **IAM** (our research group name), of the 2006 competition. The ART testbed competition received 17 entries and our entry proved to be highly successful, beating its closest competitor in the final by approximately 28%. The aim of this paper, therefore, is to describe the various facets of the strategy which contributed to the effectiveness of the IAM agent.

The paper is organised as follows. In Section 2 we analyse the decision problems faced by a generic agent within the competition. We then go on to describe the two main com-

ponents of the IAM agent, namely its trust model and its strategy. In Section 3, we describe how the IAM agent fuses information from the disparate sources within the competition for the purposes of its trust model. Section 4 then details the spending and earning strategy of the IAM agent according to the trust model. Finally, we discuss the strategy and the competition in Section 5 and conclude in Section 6.

2. DESIGNING A COMPETITION ENTRANT

In this section, we first provide a brief description of the ART testbed competition. We then analyse the general problem faced by an agent within the competition which gives rise to the broad outlines of the design of a generic agent.

2.1 Competition Overview

The ART competition consists of a number of art appraising agents that provide appraisals about the value of the art objects of their clients, who pay them to do so. A game, within this competition, consists of a predetermined (but unknown to the agents) number of iterations. At the beginning of each game, each agent is assigned a *privately known* expertise vector that determines the variance of its error in appraising a painting when spending a certain amount of money. The vector is over the ten eras from which the paintings can come. The higher the expertise of the agent, the lower the variance of its error in assessing the painting's true value. The agent can then spend a certain amount of money gathering *opinions* from other agents about the paintings it has been tasked with appraising. Since the other agents can provide spurious opinions, the agent also needs to build a trust model to reflect its belief of how other agents will act when providing opinions. In order to aid it to build this trust model, the agent can also buy *reputation* information from other agents, but again this is open to dishonest behaviour on the part of the reputation provider. Having decided which set of agents it will buy the opinions from, the agent then provides the game simulator with *weights* that determine how the different opinions are fused in order to provide the final *appraisals* of the paintings. This then completes one iteration of the game. The accuracy of this appraisal as compared with that of the other competing agents then determines the client share of the next iteration i.e. the number of clients each agent obtains in the next round. Thus, in each round, the agent makes money from providing appraisal to its clients and spends money in ensuring that the appraisal of the art objects are as accurate as possible. The winner of the competition is the agent with the highest bank balance at the end of the game. A much more detailed description of the competition can be obtained in [3] or at `http://www.lips.utexas.edu/art-testbed/`.

2.2 The Agent Decision Model

We now describe the decision problem faced by a generic agent at a given round in a game, which can be represented as shown in Figure 1. Note that this model is part of our analysis of the problem in which we are identifying the main functional blocks that competition agents would need to a more or less sophisticated degree. Though they may not implement the model in this way, they still require to have all this functionality in whatever architecture they do have.

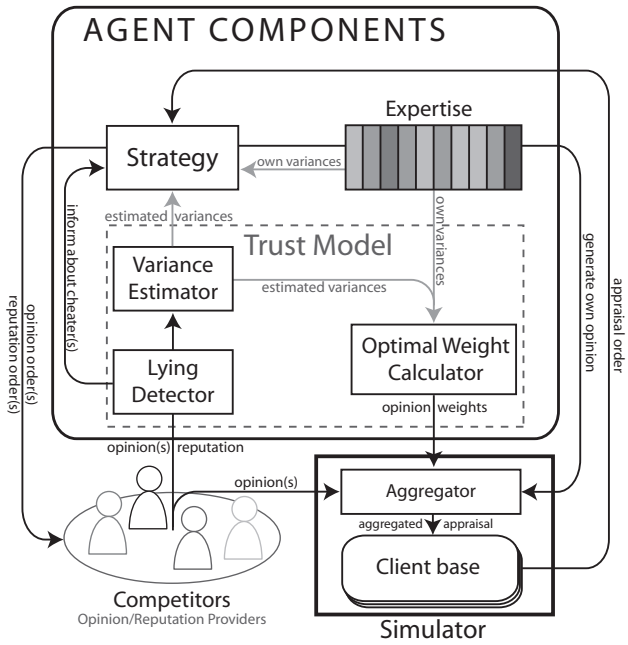

Figure 1: Agent design for the ART game

As such, it can be observed that in order to be a successful agent within the ART Testbed Competition, there are two main issues to consider. Firstly, we need to consider how to model the performance (i.e. accuracy) of each opinion provider in order to use their opinions accordingly. This is shown by the trust model component in Figure 1. Secondly, we need to formulate the strategy of the agent which will in turn regulate how its spends and earns money as well as how it provides information to its competitors. The expertise of the agent is a variable fixed by the simulator at the beginning of the game. As a result, an agent knows about its own variances in generating opinions and this will in turn influence the strategy of the agent. We now describe these two main parts of a generic agent design.

The first part is concerned with the trust model which itself consists of three main parts: (i) the lying detector which detects malicious agents, (ii) the variance estimator which estimates how much variance there is in the error of the opinions of other agents (i.e. their performance), and (iii) the optimal weight calculator which provides the optimal weights to provide to the simulator. These weights determine how the opinions an agent requested from its competitors are fused with its own opinion to generate the final appraisal and thereby its client share in the next iteration.

Now a large part of the quality of the trust model will rest on the reliability of the information the agent has gathered. The agent receives two main types of information. Firstly, it receives opinions about the paintings from its competitors (by requesting and paying for them) and it can receive opinions from the simulator by spending a certain amount of money. Secondly, it can obtain information about the behaviour of other agents, termed reputation information, by paying its competitors. Now, the quality of information obtained from the various sources is dependent on the

amount of money the agent spends on these various information providers as well as the behaviour of these sources. The provision of this information is what the strategy component of the agent determines.

An agent can spend money on requesting opinions and reputation information from its competitors, as well as generating its own opinion. Furthermore, an agent can make money by providing opinions and reputation information, as well as by providing appraisals to its client base. Thus, a successful agent needs to formulate a strategy that prudently spends money on obtaining opinions and information whilst ascertaining that its overall appraisal of the art painting of its client is comparatively good. Furthermore, the strategy determines how it interacts with its requesters when requested for opinions and reputation information.

Having briefly described the ART competition and how to construct a generic competition agent for it, we now describe the specific components of the IAM strategy. We will first describe the trust model component of IAM in the next section and then go on to detail its strategy in the subsequent section.

3. THE IAM AGENT'S TRUST MODEL

In this section, we show how, by combining its own opinion with those of others, the IAM agent can generate an overall appraisal of a painting that is more reliable than any single opinion on which it is based. That is, for a given painting with true value v, we wish to derive a combined estimate, $\overline{e}$, which is a function of the agent's own estimate, e_0, and the reported third party estimates $\{e_1, \ldots, e_q\}$. Moreover, our aim is to perform this combination such that the mean squared error, $E[(\overline{e} - v)^2]$, is smaller than that of each individual appraisal in $\{e_0, \ldots, e_q\}$. Thus, by reporting $\overline{e}$ during the competition, the IAM agent can increase its market share, due to the increased accuracy of its appraisals, and hence increase its revenue.

How this can best be achieved depends on the reliability of each agent's individual appraisals per era and, more specifically, on the variance, $var(e_i)$, of each estimate, e_i. Although, in the competition, each agent knows the variances of its own private appraisals, agents are not obliged to reveal these variances, nor are they obliged to reveal their private appraisals truthfully.

For these reasons, we must estimate the variances associated with each agent, and consequently adopt a three-step approach to combining appraisals. First, under the assumption that an agent reveals its appraisals truthfully, we use a Bayesian analysis to estimate an agent's variance, by modelling the factors which determine an agent's opinions. Second, we calculate a lower bound on the probability that an agent is lying about its appraisals, and use this to discard the opinions of potentially malicious agents. Finally, based on the estimated variances of agents believed to be truthful, we derive the optimal method for combining appraisals, under the constraints imposed by the competition. In the following subsections, we examine each of these steps in more detail.

3.1 Calculating Optimal Weights

The problem of combining estimates of an unknown value is one that has received much attention in the literature, for example [4] and [6]. In general, however, it can be viewed as an optimisation problem, in which we need to find a function $\overline{e} = f(e_0, \ldots, e_q)$ of several estimates, $\{e_0, \ldots, e_q\}$, such that $\overline{e}$ has minimal mean squared error (MSE).

In many cases, the optimal function for achieving this is non-linear. However, one of the restrictions imposed by the competition is that estimates must be combined by the testbed, on an agent's behalf. This is achieved according to Equation 1, in which $\{w_0, \ldots, w_1\}$ are weights applied to the individual opinions, which an agent can specify:

$$\overline{e} = \sum_{i=0}^{q} w_i \cdot e_i \tag{1}$$

Clearly, this function is linear, which constrains our search for an optimal combination. Moreover, optimality results for functions of this form are well known, and are generally referred to as Best Linear Unbiased Estimates (BLUE) [7]. Here, by unbiased we mean that the expected value of the combined estimate is equal to the value being estimated, or equivalently in this case, $E[\overline{e}] = v$. According to the design of the competition, the private appraisals of each agent are themselves unbiased estimates of v, which, although not essential, does simplify the problem of deriving a BLUE. Specifically, if we assume for the moment that private appraisals are truthfully revealed and that each $var(e_i)$ is known, a BLUE combination of appraisals can be achieved by setting each w_i as follows:

$$w_i = \frac{1/var(e_i)}{\sum_{i=0}^{q} 1/var(e_i)} \tag{2}$$

Unfortunately, the variance of each e_i is unknown during the competition, and so the optimal weights must be estimated, given an agent's past experience and knowledge of the environment. However, this task is simplified because each w_i only depends on the relative proportions of the variances. That is, if we multiply or divide each variance by an arbitrary constant then the weights remain unchanged, for example:

$$w_i = \frac{c/var(e_i)}{\sum_{i=0}^{q} c/var(e_i)} \tag{3}$$

$$w_i = \frac{c/var(e_i)}{c\sum_{i=0}^{q} 1/var(e_i)} \tag{4}$$

$$w_i = \frac{1/var(e_i)}{\sum_{i=0}^{q} 1/var(e_i)} \tag{5}$$

This has important implications for our strategy, when we consider how variances are assigned in the competition. Specifically, the variance of each agent's private opinion, for a given era, is generated by the testbed according to Equation 6, where s_i is the component of the ith agent's expertise vector specifying its knowledge of the painting's era, C_i is the amount of money spent by the ith agent to generate the appraisal, and α is a constant shared by all agents that during the competition was set to 0.5:

$$var(e_i) = v^2 \left(s_i + \frac{\alpha}{C_i} \right)^2 \tag{6}$$

Although, in general, s_i and C_i are unknown, the value of v^2 is constant for all i, and so is irrelevant. Therefore, rather than estimate $var(e_i)$ directly, it is sufficient to estimate the value of $var(e_i/v)$, which from the general properties of random variables is equal to:

$$var(e_i/v) = \left(s_i + \frac{\alpha}{C_i}\right)^2 \tag{7}$$

This is a much easier problem because we know that $var(e_i/v)$ will remain the same, even if v changes. In the following section we show how, using this fact, an agent can estimate $var(e_i/v)$ and so choose appropriate weights for each appraisal it receives.

3.2 Estimating Truthful Variances

From the previous section, we know that, for $i > 0$, $var(e_i/v)$ depends on two unknown parameters, s_i and C_i, and consequently must be estimated. One well founded way to achieve this is to use Bayesian analysis to marginalise over the unknown parameters [2]. That is, given s_i and C_i we know that $var(e_i/v)$ is equivalent to Equation 7, and from this, we can then calculate the *marginal* variance, $E[var(e_i/v)]$, as follows:

$$E[var(e_i/v)] = \sum_{s_i \in \mathcal{S}} \sum_{C_i \in \mathcal{C}} P(s_i, C_i) \left(s_i + \frac{\alpha}{C_i}\right)^2 \tag{8}$$

Here, $P(s_i, C_i)$ is the joint probability of the unknown parameters, $\mathcal{S}$ & $\mathcal{C}$ are their respective domains, and based on this, optimal weights can be estimated using:

$$w_i = \frac{1/E[var(e_i/v)]}{\sum_{j=0}^{q} 1/E[var(e_i/v)]} \tag{9}$$

In the case of an agent's own variance, the values of s_0 and C_0 are known, and so $E[var(e_0/v)]$ simplifies to $var(e_0/v)$. For third party opinions, on the other hand, we must model the probability distribution of s_i and C_i, based on available evidence. To this end, we assume that domains $\mathcal{S}$ and $\mathcal{C}$ are defined as follows, based on an analysis of the testbed software and competition rules:

$$\mathcal{S} = \{0.1, 0.2, \ldots, 1\} \tag{10}$$

$$\mathcal{C} = \{1, 2, \ldots, 10\} \tag{11}$$

This means that we can record $P(s_i, C_i)$ using a 10x10 conditional probability table (CPT) which we can update in light of an agent's experiences, and we can use this to calculate $E[var(e_i/v)]$. For example, in absence of evidence to the contrary, we may assume that all possible values of s_i and C_i are equally likely, resulting in the CPT shown in Table 1. Substituting the appropriate entries into Equation 9, this gives us a marginal variance of 0.5848. Moreover, if we hold the same beliefs for all agents then $w_i = 1/q$ for each i:

$$w_i = \frac{1/E[var(e_i/v)]}{\sum_{j=0}^{q} 1/E[var(e_i/v)]} = \frac{1/0.5848}{\sum_{j=0}^{q} 1/0.5848} \tag{12}$$

$$= \frac{1/0.5848}{q/0.5848} = \frac{1}{q} \tag{13}$$

Over time, the entries in the CPT are updated to reflect both an agent's experiences with different appraisers, and the possibility that an agent may change its strategy between episodes. This is achieved by performing two operations at the end of each timestep: (1) update the CPT in light of any received opinions, and (2) add time dependent noise to account for dynamic behaviour.

	s_i									
C_i	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
1	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01
2	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01
3	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01
4	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01
5	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01
6	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01
7	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01
8	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01
9	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01
10	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01

Table 1: Joint CPT for C_i and s_i and its initial values

To define these steps, we must first specify how we model dynamic behaviour in truth telling opinion providers. We know from the competition rules that s_i does not change for the duration of a game, but C_i can change if the opinion provider changes its policy. To account for this, we define C_i^t as the value of C_i at time $t \in \mathbb{Z}^+$. For each possible value of C_i, we then define transition probabilities that specify the conditional probability $P(C_i^{t+1} = c_1 | C_i^t = c_0)$ where c_1 and c_0 belong to $\mathcal{C}$. For example, we may define transition probabilities such that:

$$p(C_i^{t+1} = c_1 | C_i^t = c_0) = \begin{cases} \frac{9}{10} & \text{if } c_0 = c_1 \\ \frac{1}{90} & \text{otherwise} \end{cases} \tag{14}$$

thus assigning a probability of 0.1 to a provider changing C_i on any given time step, with equal probability of that transition being to any other possible state.

With this in mind, we can now consider the CPT update rules. First, if by the end of a time step we know that the true value of a painting is v^t, and that the opinion provided by an opinion provider is e_i^t, then we perform step 1 by calculating the posterior CPT using Bayes rule as follows:

$$P\left(s_i, C_i^t | p_i^t\right) = \frac{P\left(p_i^t | s_i, C_i^t\right) P\left(s_i, C_i^t\right)}{\sum_{s \in \mathcal{S}} \sum_{c \in \mathcal{C}} P\left(p_i^t | s, c\right) P\left(s, c\right)} \tag{15}$$

Here, p_i^t is defined as e_i^t/v^t; $P(s, C_i^t)$ is the prior probability of the parameters, taken from the current CPT; and $P(p_i^t | s, C_i^t)$ is the data likelihood function. The latter is determined by the competition testbed, which generates estimates from a Gaussian distribution with mean v^t and variance $var(e_i^t)$. By substituting into the Gaussian p.d.f., we thus have:

$$p(p_i^t | s_i, C_i^t) = \frac{1}{\sqrt{2\pi(s_i + \alpha/C_i^t)^2}} \exp\left[-\frac{(p_i^t - 1)^2}{2(s_i + \alpha/C_i^t)^2}\right] \tag{16}$$

Finally, in step 2, we prepare the ground for the next time step by marginalising over C_i^t using the transition probabilities. This is achieved by updating the CPT for timestep

$t+1$, as follows:

$$P(s_i, C_i^{t+1}) = \sum_{C_i^t \in \mathcal{C}} P(C_i^{t+1}, s_i, C_i^t) \quad (17)$$

$$= \sum_{C_i^t \in \mathcal{C}} P(C_i^{t+1} | s_i, C_i^t) P(s_i, C_i^t) \quad (18)$$

which, assuming we do not specify different transition probabilities depending on s_i, is equal to:

$$P(s_i, C_i^{t+1}) = \sum_{C_i^t \in \mathcal{C}} P(C_i^{t+1} | C_i^t) P(s_i, C_i^t) \quad (19)$$

3.3 Dealing with Liars

So far, we have shown how optimal weights can be estimated for agents that reveal their appraisals truthfully. However, as this is not necessarily the case, we need to be able to identify lying behaviour among agents, and act appropriately. This serves two purposes: (1) it identifies potentially fictitious appraisals, which are then eliminated from consideration before the combined appraisals are calculated; and (2) it informs the IAM agent's spending strategy, and how it behaves toward agents it believes to be malicious (see Section 4).

To this end, we test the hypothesis that an agent's last k opinions are truthful, based on what we know about how an agent's private opinions are generated. Here, k can be set to the total number of opinions received from an agent, or it can be set to the number of opinions received in the last n timesteps, to allow for the possibility that an agent may change its strategy over time. More specifically, according to our assumptions, the maximum mean squared error for a truthful opinion is:

$$max[mse] = v^2 \left(max[s_i] + \frac{\alpha}{min[C_i]} \right)^2 \quad (20)$$

$$= v^2 \left(1 + \frac{0.5}{0.1} \right)^2 \quad (21)$$

$$= 2.25 v^2 \quad (22)$$

On this basis, the random variable e_i/v is normally distributed with mean 1 and variance 2.25. According to the standard properties of normally distributed random variables, if the last k opinions are all generated in this way, then the following statistic has a chi-squared distribution with k degrees of freedom:

$$Q_k = \frac{1}{2.25} \sum_{i=1}^{k} (e_i/v_i - 1)^2 \quad (23)$$

This allows us to place an upper bound on the probability that an agent's last k opinions are truthful, by comparing the actual value of Q_k to the quantiles of the chi-squared distribution.

For example, the distribution of Q_k for $k = 5$ is shown in Figure 2, in which the vertical lines mark the 0.75, 0.95 and 0.99 quantiles respectively. Now suppose that, as illustrated, Q_k is recorded as 18. As this occurs beyond the 0.99 quantile, this means that, even if all k opinions were generated with the highest possible variance for truthful opinions, the probability of Q_k occurring by chance is less than 0.01. In the next section, we show how, by making such comparisons, an agent can make appropriate decisions about how to interact with its peers.

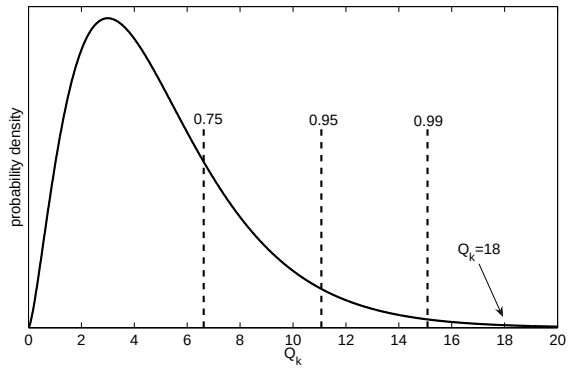

Figure 2: Chi-squared distribution with 5 degrees of freedom

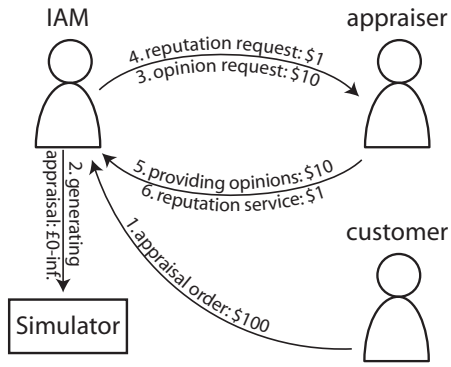

Figure 3: Cash flows from/to the (IAM) agent

4. THE IAM AGENT'S STRATEGY

This section describes the strategy that the IAM agent adopts in order to maximise its final earnings (i.e. income less spending) in the ART game. First, we need to look at the cash flows to and from our agent in order to identify where decisions need to be made to optimise the agent's spending and earnings (see Figure 3). In brief, there are three sources and three sinks of cash for an agent participating in the competition. Figure 3 also depicts the magnitudes of these six flows of cash as decided by the competition. We will now discuss each of these cash flows in turn. We will denote with $+$ those flows that generate income for the agent and by $-$ those where the agent needs to spend.

1. Appraisal orders (+): At the beginning of each round in the game, the simulator generates a number of orders for each agent. The agent earns $100 per appraisal order regardless of how it will fulfill that order (even if it decides not to provide an appraisal). Since the amount earned per order is fixed, an agent can only increase its earnings from this flow by increasing its number of orders in successive rounds. This can only be achieved if it provides a better service than its competitors (i.e. providing more accurate appraisals than

those of its competitors). This in turn requires the spending of money on 2, 3, and 4 below.

2. Generating own appraisals (−): The IAM agent can spend an arbitrary amount of money (representing the time cost associated with the appraising activity) to generate its own opinion for a particular order. The more money it spends, the more accurate the appraisal will be. However, this accuracy is limited to a predefined value according to the agent's expertise in a particular art era. More mathematically, the standard deviation of the error of the opinion generated by the simulator, to which the appraiser with expertise s_i has provided C_i is given by:

$$std(e_i) = \sqrt{var(e_i)} = \left(s_i + \frac{\alpha}{C_i}\right) v \qquad (24)$$

where α is a parameter set by the competition and v is the true value of the painting. Thus, it can be observed that deciding on the expenditure of money in generating one's own opinion is not straightforward and this will be investigated in greater detail in Section 4.1.

3. Opinion requests (−): An agent can also request the opinion(s) of other agents on a particular appraisal order. Each opinion request has a fixed cost of \$10. Note that opinion providers (i.e. other agents in the game) are not obliged to provide their opinions after having received the money. In addition, they can provide opinions of low accuracy either due to having very low expertise on the concerned art era and/or deciding to spend little money on generating their opinions. Thus, the selection of opinion providers and the calculation of the weights to be provided to the simulator in order to aggregate opinions are challenging problems which an agent faces when buying opinions. Section 4.2 discusses how the IAM agent tackles these problems.

4. Reputation requests (−): In order to help agents find out about good opinion providers, the ART testbed allows agents to contact one another to ask for their evaluation about the appraising performance of a particular agent in a particular era[1].

5. Providing opinions (+): An agent can also earn by providing opinions to other agents (i.e. its competitors) when requested. The earnings for such a service are fixed at \$10 per opinion request. This service can provide extra income to boost the IAM agent's profits and the strategy for this is specifically looked at in Section 4.3.

6. Reputation service (+): Similarly to providing its opinions, the IAM agent can also provide to other agents the reputation value of a particular agent in the game for \$1 per reputation request.

[1]The performance of an agent in an art era is not necessarily equivalent to its expertise in that era. The actual performance of an agent is also determined by the amount of money it decides to spend on generating appraisals for others in addition to its expertise. For this reason, an agent can provide different levels of quality of service to different partners and, thus, the reputation values of the same agent provided by different partners can be different (assuming that the partners provide their honest evaluation as the reputation values).

From the cash flow analysis above, since providing the appraisal service to customers earns relatively much more money than providing opinions or reputation values, our strategy focuses on maximising the customer base of the IAM agent by trying to provide a highly accurate appraisal service. However, the more accurate an agent wants its appraisals to be, the more money it needs to spend (see the money flows 2, 3, and 4), and, as a result, the less profit it makes from a job. Meanwhile, spending less will result in less accurate appraisals, and in the ART game, this means fewer customers and thus less income. This is a classic tradeoff between profit and customers. Therefore, it is important to find the balance between these; i.e. spend wisely to generate accurate appraisals and to retain a high profit at the same time. In more detail, before a final appraisal is produced for a particular order, the IAM agent needs to make the following decisions:

1. how much should it spend on generating its own appraisal given its expertise on the art era of the order,

2. whether it needs to ask for external opinions for a given order and, if so, which opinion providers it should choose and

3. whether it needs to ask for the reputation values of opinion providers and how these values can be used to help it makes the decisions in (2).

Regarding using reputation values provided by other agents, since the game rules do not specify the semantics of a reputation value, except that it is a number in the range $[0, 1]$, interpreting their real meanings is very difficult (because there is no commonly shared semantics). Moreover, given the small number of participating agents in a game (5), the behaviours of the other agents can be learnt fairly quickly and thus reputation values are not of much use. For these reasons, the IAM agent does not request reputation values from other participants (and saves a small amount of money by doing so). Thus, it relies solely on the variance estimator (Section 3.2) to estimate the performance of an opinion provider.

In the remainder of this section, Sections 4.1 and 4.2 will address the first two issues above. In addition, Section 4.3 looks at how the IAM agent can increase its earnings by providing opinions and reputation values to other agents in a game.

4.1 Generating its Own Appraisals

An agent's own appraisal for a painting is generated by the simulator from a Gaussian distribution whose mean is the painting's true value (v) and whose deviation is determined by the agent's expertise (s_i) and the money it spends (C_i) as in Equation 24. Therefore, given that s_i is predetermined by the simulator, an agent can only directly affect the accuracy of its appraisals by changing the amount of money it is willing to spend to generate the appraisals (i.e. C_i). However, since the painting's true value v is factored out when the accuracy of an appraisal is calculated [3], for the sake of simplicity, it is equivalent to considering the standard

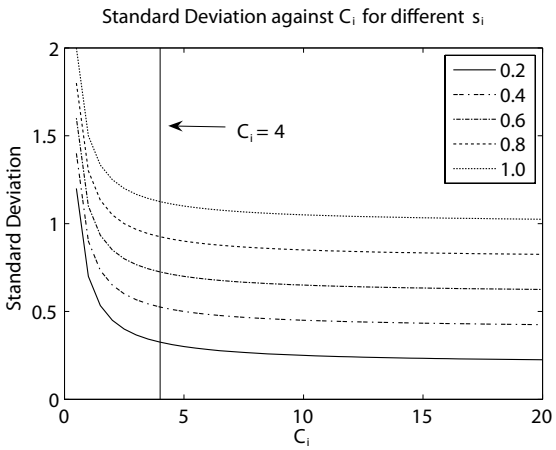

Figure 4: The cost vs. deviation trade off

deviation of an agent's appraisals to be the following:

$$std(e_i/v) = \sqrt{var(e_i/v)} = s_i + \frac{\alpha}{C_i} \tag{25}$$

Although the more an agent spends, the less the deviation and the more accurate its appraisals, from Equation 25, it is clear that the lower bound of the deviation is s_i and it might not be worth spending more money when $std(e_i/v)$ is close enough to s_i. In this respect, Figure 4 shows the values of $std(e_i/v)$ corresponding to different amounts of spending on C_i for different values of s_i in $\{0.1, 0.2, \ldots, 1.0\}$. Coincidentally, all the deviation curves in Figure 4 level off from spending of \$4 upwards, despite the value of s_i. This means, in any case, \$4 is a balance point between spending and the accuracy gained from it. Spending more than \$4 only yields a very marginal decrease of the deviation (or a marginal increase of appraisal accuracy). Given this, the IAM agent only spends \$4 for generating its own appraisals in all situations (regardless of its expertise).

4.2 Ordering External Opinions

In this section, we consider, given a particular appraisal order and having known our own expertise s_i in the order's art era, whether external opinion(s) should be sought and, if so, from whom. From the findings in the previous sections, since it is relatively cheap (\$4 compared to the order fee of \$100) to generate its own appraisals which are reasonably close to its expertise, the IAM agent always uses its own appraisals regardless of whether it needs external opinions or not because this would provide a safety catch should other agents decide to cheat by not providing their opinions as agreed. Having estimated the performance of each opinion provider (i.e. its variance, see Section 3.2), the IAM agent can evaluate the benefit of having an opinion from a particular opinion provider to justify the decision of requesting an opinion from it. In other words, the IAM agent needs to calculate the expected variance of the final appraisal[2] resulting from combining its own appraisal with that of the potential opinion provider. Given that in producing the final appraisals, the simulator combines all the submitted appraisals as in Equation 1 and the weight for each appraisal w_i is defined as in Equation 5, the variance of the combined appraisal is given by:

$$var(\overline{e}) = \sum_{i=0}^{q} w_i^2 \cdot var(e_i) \tag{26}$$

where q is the number of external opinions (e_i) the IAM agent receives and e_0 is its own appraisal. Replacing w_i from Equation 5 gives:

$$\begin{aligned} var(\overline{e}) &= \sum_{i=0}^{q} \left(\frac{1/var(e_i)}{\sum_{j=0}^{q} 1/var(e_j)} \right)^2 \cdot var(e_i) && (27) \\ &= \left(\frac{1}{\sum_{j=0}^{q} 1/var(e_j)} \right)^2 \sum_{i=0}^{q} 1/var(e_i) && (28) \\ &= \frac{1}{\sum_{j=0}^{q} 1/var(e_j)} && (29) \end{aligned}$$

Factoring out the painting's value v from both sides of Equation 29 gives us:

$$var(\overline{e}/v) = \frac{1}{\sum_{i=0}^{q} 1/var(e_i/v)} \tag{30}$$

Although we do not know $var(e_i/v)$ for other agents, we can estimate it (as shown in Section 3.2), and thus we can estimate $var(\overline{e}/v)$:

$$E[var(\overline{e}/v)] = \frac{1}{\sum_{i=0}^{q} 1/E[var(e_i/v)]} \tag{31}$$

where $E[var(e_0/v)] = var(e_0/v) = s_0 + \alpha/C_0$ (because we know our own parameters) and $E[var(e_i/v)]$ can be calculated as in Equation 8 for other agents.

Knowing how to calculate the combined variance of opinions, the IAM agent can start selecting agents it will ask for opinions by performing the following four steps. First, all other agents are sorted in ascending order, according to their estimated variances in the art era of the order (i.e. best performance first). Second, the list of agents whose opinions will be used is initialised to contain (only) the IAM agent itself (because it always uses its own appraisal). Third, the IAM agent then considers adding each of the agents into the list according to their performance order by calculating the expected combined variance of the final appraisal assuming that agent is in the list. Finally, if the new combined variance is just insignificantly improved (i.e. improvement is less than 15%[3]) then the selection process stops; otherwise, that agent is put into the list and the process continues with the next agent.

After the selection process is finished, the IAM agent sends opinion requests to agents selected in the list. However, since the variance estimator assumes that agents do not cheat when providing their appraisals, this process does not filter out cheating agents. Therefore, before the selection process begins, all the other agents are evaluated based on their previous interactions using the procedure as described in Section 3.3 to calculate the probability that an agent has cheated in the past. If the cheating probability is over 0.6, then the opinion provider is classified as a *potential cheater* and its opinions will not be taken into account in our agent's

[2]The expected variance of the final appraisal reflects the expected accuracy of the appraisal.

[3]This improvement threshold is hand-picked based on our experiments.

final appraisal. However, it will still be asked for opinions in order to confirm whether it is actually cheating. If the cheating probability is over 0.95, the opinion provider is classified as a *cheater* and no future interaction with it will be made. Moreover, an agent is also classified as a cheater if it failed to provide an appraisal after having confirmed that it will provide an opinion.

4.3 Earning from Being Honest

Beside earning money from appraising orders by the simulator, the IAM agent's strategy is to maximise its income via providing opinions and reputation values. Its philosophy in doing so is to provide an honest and reliable service to all agents in order to maintain a good business relationship with them, which hopefully results in a regular source of income. Moreover, providing a low quality service or cheating other agents might initiate retaliatory behaviours from them. This will have a detrimental effect on the IAM agent's own appraising business when it requires external opinions to fulfil orders in whose art era it is not an expert. Although the general strategy here is to provide a good service, the IAM agent is not naïvely benign. Rather, it looks for reciprocity from its partners (who are also its competitors). Therefore, if a cheater is identified, the IAM agent will use a retaliatory policy toward it. Following this general strategy, the IAM agent's strategies for the earning cash flows 5 and 6 (i.e. providing opinions and reputation values for other agents, see Figure 3) are discussed next.

4.3.1 Providing opinions:

The IAM agent aims to develops its reputation as an honest agent in order to earn (good) money from providing opinions to others. Thus, it always provides honest certainty values (taking into account the money it will spend) and honest appraisals (i.e. spending the same amount of money it said it would spend). In more detail, the IAM agent always spends \$4 to generate appraisals for others (the same amount it would spend to generate appraisals for its own orders, see Section 4.1). When requested for its certainty assessment on its appraisals, the IAM agent will present the certainty value (denoted by cv) based on its expertise in the art era of that particular request and its intended spending to generate an appraisal (i.e. $C_i = \$4$):

$$cv = 1 - \frac{(1 + \alpha/C_i)s_i}{1.5} \tag{32}$$

where s_i is the expertise of the IAM agent in the concerned art era and 1.5 is the maximum deviation of the Gaussian distribution generating an agent's appraisal value (given by Equation 25 in the worst case scenario where the minimum expertise $s_i = 1.0$ and the minimum spending $C_i = \$1$).

In case it detects that an agent cheats by providing false opinions to it or by taking the opinion fee without providing an opinion, it will retaliate against the cheating agent in all future opinion transactions by spending only a fractional amount of money (\$0.01) to generate appraisals that are extremely skewed (due to the ensuing high deviation) for the cheating agent.

4.3.2 Providing reputation values:

Agent	Affiliation	Revenue	Cost	Profit
IAM	University of Southampton	149812	18299	131583
Neil	Nanyang Technological University	116764	13741	103023
Frost	Bogazici University	120753	18176	102577
Sabatini	Universidad Carlos III de Madrid	127137	25726	101411
Joey	University of Nebraska-Lincoln	111985	19506	92479
	mean	*125290*	*19076*	*106215*

Table 2: Average revenue, profit & cost at end of final round games

Agent	Opinion Costs	Opinion Generation Costs	Reputation Costs
IAM	59.46	40.54	0.00
Neil	6.80	92.03	1.16
Frost	31.65	68.35	0.00
Sabatini	38.85	61.15	0.00
Joey	0.00	100.00	0.00

Table 3: Breakdown of cost percentages.

As previously mentioned, there is no defined semantics for reputation values except that their range is $[0, 1]$. We therefore define the reputation value (r_i) to provide based on the estimated variance of an agent, which we believe best reflects the performance of that agent as so far perceived:

$$r_i = \max\{1 - \frac{\sqrt{var(e_i/v)}}{1.5}, 0\} \tag{33}$$

where 1.5 is again the maximum deviation given the worst case scenario and $\sqrt{var(e_i/v)}$ is the estimated deviation of that agent's appraisals. In this service, the IAM agent produces random reputation values in $[0, 1]$ for cheating agents.

5. COMPETITION RESULTS

In this section, we give an in depth discussion of the competition results, concentrating on its final round in particular. This consisted of 10 games with 60 time steps each, in which the top 5 agents from a series of preliminary games were pitted against each other (there were 17 agents entered into the competition). At the end of each game, the total profit earned by each agent was recorded, and then averaged over all 10 games to produce the final scores. The results of this process are shown in Table 2, which includes the average revenue and cost for each agent over the 10 games. In particular, this shows that the IAM agent won the competition by a 28% profit margin over its nearest competitor, earning the highest revenue of any agent, with below average costs.

A more detailed breakdown of each agent's cash flow is shown in Tables 3 and 4, in terms of the types of payment that contributed to the costs and revenue in the final round games. Here, client, opinion and reputation payments refer to the total amount of each agent's revenue that was due to payments for client appraisals, opinion generation and reputation respectively; opinion costs refer to the amount spent

Agent	Client Payments	Opinion Payments	Reputation Payments
IAM	96.09	3.89	0.03
Neil	98.63	1.37	0.00
Frost	98.45	1.52	0.03
Sabatini	88.25	11.72	0.03
Joey	96.96	3.00	0.04

Table 4: Breakdown of revenue percentages.

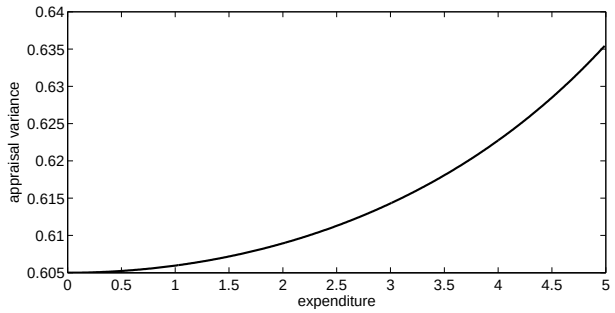

Figure 5: Opinion Spending

requesting third party opinions; opinion generation costs are those spent on generating opinions for third parties; and reputation costs are those spent requesting reputation about an agent's peers.

From these results, it is clear that largest proportion of each agent's revenue was due to payments for client appraisals, followed by opinion payments and reputation in that order. This is partially a consequence of the comparatively high price of client appraisals ($100 compared to $10 for opinions and $1 for reputation). However, even taking this into account, it is clear that very little reputation was exchanged between any of the agents, reinforcing the idea that reputation is only useful in large populations with clear semantics.

On the other hand, opinion transactions did feature more prominently in the final rounds, and played an important role in determining the order of the leader table. Although this is not reflected directly in agent revenues, opinion costs accounted for the majority of the overhead in the IAM agent's winning strategy. To understand the reason for this, suppose that two agents can co-operate by spending a total of $10 on their opinions to appraise a painting. Assuming that, for both agents, $s_i = 1$, $\alpha = 0.5$, and the difference between their individual expenditure is y, then the best achievable variance for the appraisal (relative to a painting value) is:

$$var(\bar{e}/v) = \left(\frac{1}{[1 + 1/(10 + y)]^2} + \frac{1}{[1 + 1/(10 - y)]^2} \right)^{-1} \tag{34}$$

Based on this equation, the appraisal variance resulting from difference values of y is plotted in Figure 5. This shows that the lowest possible variance is achieved when $y = 0$ or, in other words, when the agents divide the $10 equally between their respective opinions. This can easily be shown to be true in general, and explains why it was beneficial for the IAM agent to spend significantly on third party opinions,

Agent	MSE of Opinions Reported to IAM	No. Opinions Sent to IAM
Neil	6.20	439
Frost	142533.43	179
Sabatini	0.41	10601
Joey	1338173.81	270

Table 5: Total number of opinions received by the IAM agent compared to average opinion MSE.

Agent	Revenue	Cost	Profit
IAM	150132	18879	131253
Neil	139870	13757	126113
Joey	133417	19506	113911
Sabatini	127814	46470	81344
Frost	120833	42381	78452

Table 6: Alternative scores based on recorded number of opinion transactions.

rather than investing solely in its own direct assessments. On the other hand, agents such as Neil and Joey, who spent little or no income on third party opinions, did make it to the final round. This is perhaps an indication that, if opinions are not used wisely based on solid statistical models, then the simple strategy of using only an agent's direct experience can produce better results.

Concentrating more specifically on the IAM agent's performance, its success in the competition can be further explained by three other factors. First, by spending $4 on opinion generation, it managed to spend less per client on opinion generation than any of the other four finalists, while still maintaining close to minimal opinion errors. Second, the IAM agent further reduced its costs by only buying opinions from reliable agents. This is made apparent when we compare the total number of opinions received by IAM from the other competitors, compared to the MSE of their reported opinions (Table 5). This shows a clear correlation, with IAM relying heavily on Sabatini's accurate opinions, while mostly rejecting opinions from the other competitors, who provided misleading opinions. Third, by estimating optimal weights for received opinions based on standard statistical theory, the IAM agent was able to achieve the lowest MSE for its appraisals compared to the other finalists, and so was awarded the highest overall client market share.

In addition, the IAM agent's share of the opinion market was second only to Sabatini. This is partially because the IAM agent itself was the highest consumer of third party opinions in terms of expenditure, most of which was spend on Sabatini's expertise. However, there does appear to be a discrepancy in the competition data between the number of opinions received by each agent, and the number of opinions paid for. More specifically, all five finalists sent more opinions than they were paid for, perhaps due to inadequate checks for payment, before opinions where sent. Without further investigation, it is unclear whether the cause of this lies with the competition testbed, the agents themselves, or some combination of both. Nevertheless, if funds had been exchanged for all the opinion transactions that took place,

then this would have had a dramatic effect on the competition, and would have resulted in the alternative scores shown in Table 6. What this shows is that both Frost and Sabatini were sent significantly more opinions than they paid for and, had they been expected to pay for all these opinions, they would have been placed at the bottom of the leader board. However, we emphasise that this does not suggest that these alternative scores are correct, or that the official scores are inaccurate. This is particularly true if the extra opinions were unsolicited, because it would be unfair to make agents pay for unwanted opinions. Even so, this does reveal an interesting anomaly that is perhaps worth further investigation.

6. CONCLUSIONS

From the design of our successful agent for the ART competition, three lessons can be drawn that are applicable to trust assessment within multi-agent systems in general.

The first lesson concerns the conditions under which an agent can benefit by pooling information from other agents about their shared environment. In the ART competition, this aggregation of information occurs at two levels, namely at appraising art objects and at evaluating the trustworthiness of other agents. We first consider the appraisal of art objects. *In this case, an agent can benefit from third party information, if it is easier to establish such information as reliable, than it is to obtain equivalent verifiable information directly.* This lesson is especially true with regard to art appraisal, due to the way in which the MSE of an agent's opinions is calculated. That is, beyond a certain point, only marginal increases in accuracy can be achieved by spending more money on the opinion of an individual agent (Section 4.1). Thus, it is generally more economical for an agent to purchase opinions from a number of third parties than it is to invest heavily in its own opinion. The success of the IAM agent can be partially attributed to this conclusion, when we compare its performance to other agents, such as Joey, who invested little or no income on third party opinions.

The second lesson is the counter of the first, establishing when it is not beneficial to try pooling information. For example, in the competition, there is little apparent advantage to reputation sharing, as is reflected by the negligible number of reputation transactions that occurred in the final rounds. This can be attributed to the small number of agents that participated in each round (typically 5 agents), each of which had sufficient opportunity and funds to interact with all of their peers in every time-step. As a result, participants could gain reliable first-hand experience of their peers, as easily as they could gather reputation. Furthermore, the semantics of reputation were not well defined, increasing the difficultly of its interpretation and assessment. For example, suppose that reputation from one agent consists of estimated standard deviations for opinions, while for another it consists of estimated variances. Both statistics have the same range of values, $[0, \infty)$, and in both cases, lower values indicate more reliable opinions. However, their correct interpretation is still different: a variance of 10 implies a much more reliable opinion that one with a standard deviation of 10. Thus, to assess the reliability of reputation, it would first be necessary to learn the function used to create it. However, this does not imply that reputation sharing is worthless in general. Rather, the lesson learnt is that *reputation is most valuable in cases where direct experience is relatively more difficult to acquire, and in which the semantics are clearly defined.*

The final lesson is that, *even if third party opinions carry useful information, this value is wasted if opinions are not properly applied using well founded statistical techniques.* Although trust can be viewed as a sociological concept, and inspiration for computational models of trust can be drawn from multiple disciplines, the problem of combining estimates of unknown variables (such as trustee behaviour) is fundamentally a statistical one. Thus, any mechanism for making predictions based on reputation that ignores statistical theory, does so at its peril. This claim may partially explain the relative success of agents, such as Joey and Neil, that spent little or no income on third party opinions. By their presence in the final rounds, these agents suggest that, unless appropriate statistical techniques are used (such as those used by the IAM agent), it is perhaps better to ignore reputation completely, in favour of a simpler solution.

7. REFERENCES

[1] R. K. Dash, S. D. Ramchurn, and N. R. Jennings. Trust-based mechanism design. In *Proc. 3rd Int. Conf. on Autonomous Agents and Multi-Agent Systems*, pages 748–755, 2004.

[2] M. DeGroot and M. Schervish. *Probability & Statistics*. Addison-Wesley, 3rd edition, 2002.

[3] K. K. Fullam, T. B. Klos, G. Muller, J. Sabater, A. Schlosser, Z. Topol, K. S. Barber, J. S. Rosenschein, L. Vercouter, and M. Voss. A specification of the agent reputation and trust (art) testbed: experimentation and competition for trust in agent societies. In *Proc. 4th Int. Conf. on Autonomous agents and multiagent systems*, pages 512–518. ACM Press, 2005.

[4] C. Genest and J. V. Zidek. Combining probability distributions: A critique and an annotated bibliography. *Statistical Science*, 1(1):114–135, 1986.

[5] T. D. Huynh, N. R. Jennings, and N. R. Shadbolt. An integrated trust and reputation model for open multi-agent systems. *Journal of Autonomous Agents and Multi-Agent Systems*, 13(2):119–154, 2006.

[6] J. Kittler, M. Hatef, R. P. W. Duin, and J. Matas. On combining classifiers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(3):226–239, 1998.

[7] R. H. Myers, D. C. Montgomery, and G. G. Vining. *Generalized Linear Models With Applications in Engineering and the Sciences*. Wiley, 2002.

[8] S. D. Ramchurn, T. D. Huynh, and N. R. Jennings. Trust in multi-agent systems. *The Knowledge Engineering Review*, 19(1):1–25, March 2004.

[9] W. T. L. Teacy. *Agent-Based Trust and Reputation in the Context of Inaccurate Information Sources*. PhD thesis, School of Electronics and Computer Science, University of Southampton, 2006.

[10] W. T. L. Teacy, J. Patel, N. R. Jennings, and M. Luck. TRAVOS: Trust and reputation in the context of inaccurate information sources. *Autonomous Agents and Multi-Agent Systems*, 12(2):183–198, 2006.

Secure Migration of Mobile Agents based on Distributed Trust

Martijn Warnier Michel Oey Reinier Timmer Frances Brazier
Intelligent Interactive Distributed Systems
VU University, Amsterdam
The Netherlands
{warnier,michel,rjtimmer,frances}@cs.vu.nl

Abstract

Integrity of mobile agents in open environments in which no guarantees can be provided on the integrity of the hosts on which they run, is an open challenge. This paper presents a method with which tampering can be detected. This method is based on the notion of distributed trust; trust distributed over the hosts involved.

Keywords: *Agents Systems, Mobile Agents, Agent Security, Migration Path Integrity*

1. Introduction

Distributed agent systems provide a powerful paradigm for building large scale distributed systems [5, 6]. Mobility and autonomy are attractive features of such systems. In dynamic environments such as e-government, e-health and e-commerce applications mobile agents provide a means to process data locally, respecting intellectual property rights [2].

Mobility and autonomy, however, also provide new challenges, especially if security is of key concern. The challenge is to preserve integrity of an agent and the data it collects in open environments: environments that are not under the control of an agent's owner.

In closed environments in which all hosts are trusted this is not a problem: all hosts are known to behave correctly. Similarly, in more open systems, where agents only migrate to hosts they trust, the integrity of the agent and its data are not at significant risk. The use of a separate trust infrastructure, e.g., a X509-based PKI [4], can, for example, be used to establish which hosts can be trusted. However, mobile agents can also be deployed in open systems. In such dynamic circumstances it is not always feasible to determine the trustworthiness of hosts in advance.

In such open environments the *malicious host problem* forms a serious threat to the integrity of an agent. It is closely related to the intrinsic features of mobile agent systems: agents are executed on hosts that can view, alter their state, or even delete the agent altogether. This leads to a number of security problems:

1. Protecting the integrity of the migration path
2. Protecting the integrity of the agent itself
3. Ensuring confidentiality of the agent's data and (binary) code
4. Ensuring integrity of the agent's control flow

The migration of an agent from one host to another is called a *migration step*. A *migration path* is a sequence of multiple migration steps that identifies all the hosts, in order, an agent has visited.

The integrity of the migration path (item 1, above) is the basis for detecting malicious hosts and/or preventing them from doing any harm. For example, a number of techniques [1, 3, 10, 9] have integrity of agents migration paths as a premise, and can be used to detect tampering with the agent (items 2 & 4). Confidentiality (item 3) can be ensured by using encryption of sensitive data.

The main focus of this paper is the detection of breaches of integrity in migration paths of mobile agents. This paper does not directly address the general problem of protecting the agent from malicious hosts that alter the agent's code or data. However, as stated before, techniques to detect whether an agent has been altered on the basis of its migration path.

How migration works is the topic of the first section. Some known solutions for integrity protection are discussed in the context of their strengths and weaknesses in this context. The next section introduces a new technique with which the integrity of an agent's code and data can be derived on the basis of its migration path. This method is based on distributing trust among the individual hosts of an agent platform. The paper ends with a discussion and conclusions.

2. Agent Migration

This paper assumes that a secure distributed mobile agent system provides the following basic properties: an agent runs on one single host at a time, is aware of its current host, has the ability to *migrate* to other hosts in the system. In addition this paper assumes that the environment provides a public-key infrastructure with which agents and hosts can be authenticated. The host on which an agent is initialized, is assumed to be trusted by the agent's owner. This host can be traced by all other hosts at any arbitrary moment in time. Hosts are assumed to have full control over the agents they run. The consequence of this assumption is that hosts are assumed to be able to read and alter information stored inside agents.

Agents preferably only migrate to trusted hosts. However, a trust relationship does not always give full guarantees on the correct behavior and intentions of hosts. An agent's migration path provides a means to detect breaches of integrity.

The simplest form of migration in a secure agent system requires sending and receiving hosts to mutually authenticate themselves using a PKI. The integrity of a migrating agent is ensured by having the sending host create a (digital) signature of an (hash of the) agent's code. This signature is transmitted together with an agent's code, and data (including state). The receiving host can then verify the integrity of an agent's code before re-initializing the agent process.

3. Migration Paths

There are roughly two different approaches for recording agent migration paths: (i) use a centralized trusted third party (TTP) to authorize and keep track of migration paths or (ii) store (a signature of) migration paths *inside* the agents themselves [11]. Both approaches are discussed. The following notation is used in this paper: capital letters A,B,C,... denote hosts, small letters x,y,z,... denote agents, arrows ($\rightarrow$) represent migration steps between hosts and $[x]_A$ denotes the signature of agent x by host A.

3.1. Centralized Trusted Third Party

A centralized approach requires all migration paths to be registered. Both sending and receiving hosts register each and every migration step before and after migration. The trusted third party authorizes the migration and stores the migration step in its database together with the commitments of the hosts. This makes it possible to prove, at a later point in time, that both parties had agreed with the migration step.

This approach has the advantage that it is relatively easy to monitor hosts over a period of time. Thus, for example, if agents tend to disappear on one specific host, this host is known to be unreliable, possibly malicious. A centralized trusted third party may prevent agents from migrating to untrusted and/or unreliable hosts (simply by not authorizing the migration).

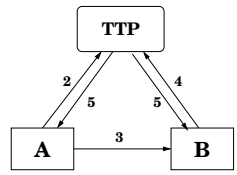

Figure 1. Agent migration using a central authority (trusted third party).

Example 1 below gives a more detailed explanation of a migration step, using a TTP, by an agent from host A to host B.

Example 1
A detailed example of the migration of agent x from $A \rightarrow B$ using a trusted third party. The migration protocol starts when:

1. *host A suspends and signs agent x: $[x]_A$*
2. *host A reports to the trusted third party (TTP) that agent x will migrate from A to B. A sends $[x]_A$ along with the report.*
3. *host A sends agent x to host B*
4. *host B receives x and computes $[x]_B$ which it sends to the TTP.*
5. *the TTP verifies that A and B have both signed the same agent. If the verification passes, the TTP notifies both A and B that the migration has succeeded, and adds this migration step to the migration path it keeps for agent x.*
6. *host B starts the suspended agent*[1].

Figure 1 shows the messages that are sent between the parties. The numbers used in the figure correspond to the numbers above.

Note that this approach does not prevent hosts from conspiring to forge migration paths. For example, malicious hosts can simply decide to migrate an agent between themselves without using the trusted third party. However, after such an illegal migration, the agent cannot migrate to

[1] Note for the sake of clarity, some details are omitted from this and following examples. This protocol has, for instance, not been secured against replay attacks. To solve this problem, other techniques, such as adding freshly generated random numbers to each signature must be applied.

a non-malicious host without being noticed. The trusted third party, having not been informed of the previous migration step, will immediately detect something is wrong and will therefore not authorize the migration. The trusted third party, however, can not detect whether an agent has migrated from a malicious to multiple malicious hosts returning to the first malicious host (a cycle), before moving to a non-malicious host using the trusted third party.

Note also that a centralized approach such as this creates a central point of failure, and a potential performance bottleneck (i.e., a scalability problem). This problem can be partially solved by using an agent's initial host as its trusted third party[2]. This is known as a *home based approach*. The agent platform Mansion uses a dedicated service: the Agent Location Service [12] that implements this use of the initial host as a trusted third party.

3.2. Signature Chain

A method that stores an agent's migration path *together* with an agent's code and data does not suffer the drawbacks of a single centralized server. The stored migration path can be protected against tampering using digital signatures [4]. A chain of such signatures can protect a whole migration path of an agent. In this method, the migration process includes that a host signs a migration step together with all (signed) previous migration steps (the chain, signed by other hosts).

Example 2
A detailed example of the migration step of agent x from $B \rightarrow C$, where B received agent x from A earlier using signature chaining. Agent x's signature chain at host B is $[x, A \rightarrow B]_A$.

1. *host B suspends and signs agent x and signs $B \rightarrow C$ along with the already existing signature chain: $[x, B \rightarrow C, [x, A \rightarrow B]_A]_B$. This signature will become a new link in the signature chain.*
2. *host B sends the agent with the old signature chain, along with the new link of the chain to C.*
3. *host C checks the signature chain stored with agent x and the new link it has received from B.*
4. *if the verification of the signatures was successful, host C adds the new entry to the agent's signature chain and starts the suspended agent x.*

As signatures can only be generated by individual hosts and verified by all other hosts, this method ensures breaches of integrity of the migration path are noticed. The method, however, has some drawbacks: for long chains verifying the complete chain of signatures becomes computationally intensive. A more serious problem is that a malicious host can remove arbitrary cycles from a migration path. If an agent (accidentally) visits the same malicious host for a second time, the malicious host can remove the part of the migration path between the first time it was hosted by the malicious host and the second. The malicious host can then reuse the old signature chain in the agent for other purposes. This cannot be detected by other hosts nor by an agent's owner. Finally, if agents disappear, e.g., by having been killed by a malicious host, outside hosts are not capable of identifying where something went wrong.

Note that with respect to cycles there is a difference between the trusted third party approach and the signature chaining approach. In both approaches a malicious host can remove cycles from the migration path. In the trusted third party approach, malicious hosts could remove the migration path between conspiring malicious hosts ending with the initial malicious host. In signature chaining a malicious host can remove a similar cycle which may also have included non-malicious hosts.

This paper proposes a distributed approach to integrity protection of migration paths that can be used to detect unauthorized modification. The advantages of a distributed approach are that the solution scales better in a distributed system and does not have a single point of failure.

4. Secure Migration using Distributed Trust

In essence, the proposed approach for integrity preservation of migration paths entails the distribution of trust to several hosts on a migration path. This approach assumes that agents are not allowed to migrate to the host on which it currently runs (irreflexivity), and that the migration path can be recorded *together with* an agent's code and data. Tampering of the migration path can be detected by either the agent owner or one ore more receiving hosts.

Briefly, the algorithm works as follows: Suppose agent x migrates along the path $A \rightarrow B \rightarrow C$. Each migration step is recorded with the agent. Each step is signed by the host from which it's originated. When agent x migrates from B to C, host B asks host A to sign the migration step ($B \rightarrow C$). The resulting signature is stored with the agent. When host C receives the agent and the signatures, it confirms receipt to host A. Example 3 below provides a detailed application of the algorithm to a migration step.

Example 3
A detailed example of the migration step $B \rightarrow C$ of agent x. Host B received agent x from host A earlier, using a distributed trust algorithm. The sequence number (hop count) used in the migration step $A \rightarrow B$ is n with $n \in \mathbb{N}$:

[2] If all agents are started from the same location these two approaches are equivalent in terms of scalability.

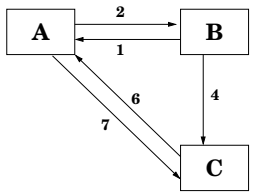

Figure 2. Agent migration using distributed trust.

1. *host B suspends agent x and asks host A to sign $(x, B \rightarrow C, n+1)$ (a* waiver*).*

2. *host A checks whether the seqnr $(n+1)$ corresponds to the seqnr (n) corresponding to the agent being on host A. If all is correct, host A signs the waiver and sends it back to host B. Host A remembers that agent x has migrated to host C with seqnr n+1.*

3. *host B adds the signature $[x, B \rightarrow C, n+1]_A$ to agent x.*

4. *host B sends the agent to host C.*

5. *host B records that agent x has migrated from itself to host C with seqnr $n+1$, and waits for C to request further information. Host C will contact host B whenever host C ships agent x to another host.*

6. *host C receives agent x, verifies the migration chain stored in the agent, and contacts host A to acknowledge the receipt of the agent with seqnr $n+1$.*

7. *host A then checks whether the info from C corresponds to the information it has recorded previously. If there is a mismatch, it notifies host C of this, so C can refuse the agent; otherwise, C can start agent x. After this step, assuming the migration from $B \rightarrow C$ was successful, host A can delete all info concerning agent x.*

Figure 2 shows the messages that are sent between the parties, the numbers used in the figure correspond with the numbers above.

Each *waiver* is issued only once, and can be used only once because the receiving host will 'consume' the waiver by contacting the host that issued the waiver. In Example 3, host C contacts host A to confirm receipt of the agent. Consequently, a malicious host cannot send an agent to different hosts, as it can only acquire one waiver for the agent. Furthermore, a malicious host cannot send the agent twice to the same host (i.e., a replay attack) by reusing a waiver obtained before, because the waiver would have already been used. In other words, host B cannot tamper with the migration path.

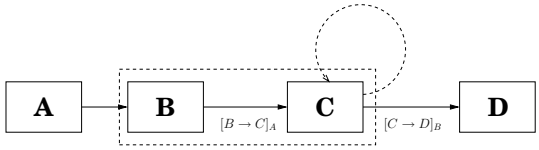

Figure 3. Conspiring hosts altering the migration path.

The migration path cannot be tampered with by a single malicious host, because the entries are linked together via host and sequence number: i) hosts can not cut out a middle piece, because the sequence numbers will not match, ii) hosts can not replace a middle piece, because the signatures in the chain of hosts will not match, iii) hosts can not cut out or replace the tail of a migration chain (including cycles), as the next receiving host will check the migration with the previous host in the migration chain, using both the host and the sequence number. The essence of this approach is that the responsibility of a migration is spread over two hosts: the previous host and the next receiving host, i.e., A and C, in the migration chain $A \rightarrow B \rightarrow C$.

The proposed migration protocol can also withstand two conspiring malicious hosts trying to manipulate the migration path between them. For example, suppose that agent x migrated along the path: $P \rightarrow A \rightarrow Q \rightarrow B$. Now, suppose that B conspires with A to try to remove host Q from the migration path pretending that the agent migrated directly from A to B. However, host P, the predecessor of A, will not issue the waiver $[x, A \rightarrow B]_P$: it has already issued the waiver $[x, A \rightarrow Q]_P$ and will refuse to issue another one.

Unfortunately, it is still possible for two adjacent conspiring malicious hosts to remove a cycle from the migration path. For example, consider an agent that migrates through malicious hosts B and C along the path $A \rightarrow B \rightarrow C \rightarrow \ldots \rightarrow C \rightarrow D$, as depicted in Figure 3 (note, the migration loop at C). In this case, C can remove the cycle from the migration path with the help of B before sending the agent to D. C simply has B issue another waiver for the migration $C \rightarrow D$. Note that C can only remove a cycle if the agent actually returns to C, which the agent typically cannot be forced to do. If cycles are created, then, contrary to signature chaining, two adjacent conspiring malicious hosts are always necessary to remove a cycle.

The main reason two malicious hosts can still alter the migration path is that the receiving hosts assumes that at least one of these is not malicious. To guard against n malicious hosts, this scheme can be extended to incorporate at least the $n+1$ preceding hosts in handing out the waivers.

5. Discussion and Conclusions

This paper introduces a mechanism to ensure that breach of integrity in migration paths of mobile agents in large scale distributed agent systems will be detected. This approach distributes trust over three hosts during each migration step. The combination of sequence numbers with signatures guarantees that one or more hosts can detect if part of the migration path, including cycles, has been removed.

Spreading trust over multiple hosts in an agent system clearly has benefits in terms of scalability and it strengthens the security mechanism, since a 'single point of failure' no longer exists. Orthogonally, a dedicated trust model that can distinguish the –relative– trustworthiness of hosts in multiple agent systems can be of much value as well.

The approach works well in situations with only one (unknown) malicious host in a migration path, or in environments with multiple malicious hosts that do not conspire together. However, if multiple malicious hosts conspire together the situation becomes more complex. A possible solution is to enforce that agents alternate between trusted and untrusted hosts [9] in their migration path. In the extreme situation where only the initial host can be trusted this solution is equivalent to 'two-hop boomerang agents' [7], in which agents always return to their initial host after each migration step to another host. Another possible solution is to disallow cycles in migration paths altogether.

The mechanisms needed to instrument this approach are currently being implemented in in the agent platform AgentScape [8].

Acknowledgments

This research is conducted as part of the ACCESS project[3] funded by the NWO TOKEN program. The authors thank Stichting NLnet for their support and Benno Overeinder for useful comments on earlier drafts of this paper.

References

[1] E. Bierman and E. Cloete. Classification of malicious host threats in mobile agent computing. In *Proceedings of the 2002 annual research conference of the South African institute of computer scientists and information technologists on Enablement through technology*, pages 141–148. South African Institute for Computer Scientists and Information Technologists Republic of South Africa, 2002.

[2] D. de Groot, M. Boonk, F. Brazier, and A. Oskamp. Issues in a mobile agent-based multimedia retrieval scenario. In *Proceedings of The 4th Workshop on the Law and Electronic Agents (LEA 2005)*, pages 33–43, June 2005.

[3] G. Karjoth, N. Asokan, and C. Gülcü. Protecting the computation results of free-roaming agents. *Personal Technologies*, 2(2):92–99, 1998.

[4] C. Kaufman, R. Perlman, and M. Speciner. *Network Security, PRIVATE Communication in a PUBLIC World*. Prentice Hall, 2nd edition, 2002.

[5] D. Kotz and R. Gray. Mobile Agents and the Future of the Internet. *Operating Systems Review*, 33(3):7–13, 1999.

[6] M. Luck, P. McBurney, and C. Preist. *Agent Technology: Enabling Next Generation Computing (A Roadmap for Agent Based Computing)*. AgentLink, 2003.

[7] J. Ordille. When agents roam, who can you trust? In *In the Proceedings of the First Annual Conference on Emerging Technologies and Applications in Communications*, pages 188–191, 1996.

[8] B. Overeinder and F. Brazier. Scalable middleware environment for agent-based internet applications. In *Proceedings of the Workshop on State-of-the-Art in Scientific Computing (PARA'04)*, volume 3732 of *LNCS*, pages 675–679, Copenhagen, Denmark, 2004. Springer.

[9] V. Roth. Mutual protection of co-operating agents. In J. Vitek and C. Jensen, editors, *Secure Internet programming: security issues for mobile and distributed objects*, volume 1603 of *LNCS*, pages 275–285. Springer-Verlag, 2001.

[10] T. Sander and C. Tschudin. Protecting Mobile Agents Against Malicious Hosts. *Mobile Agents and Security*, 60, 1998.

[11] A. Saxena and B. Soh. Authenticating mobile agent platforms using signature chaining without trusted third parties. In *In the Proceedings of the 2005 IEEE International Conference on e-Technology, e-Commerce and e-Service, (EEE'05).*, pages 282–285, 2005.

[12] G. van 't Noordende, F. Brazier, and A. Tanenbaum. Security in a mobile agent system. In *Proceedings of the First IEEE Symposium on Multi-Agent Security and Survivability*, Philadelphia, 2004.

[3] http://www.iids.org/access

Towards More Effective E-Marketplaces: A Novel Incentive Mechanism

Jie Zhang
David R. Cheriton School of Computer Science
University of Waterloo
Waterloo, ON, Canada N2L 3G1
j44zhang@uwaterloo.ca

Robin Cohen
David R. Cheriton School of Computer Science
University of Waterloo
Waterloo, ON, Canada N2L 3G1
rcohen@uwaterloo.ca

Abstract

In the context of electronic commerce, the problem of unfair ratings arises when modeling the trustworthiness of a selling agent relies (partially) on propagation of ratings provided by buying agents that have personal experience with the seller. Extreme diversity of open and dynamic electronic marketplaces causes difficulties in handling unfair ratings in trust management systems. To ease this problem, we propose a novel incentive mechanism for eliciting fair ratings of sellers from buyers. In our mechanism, buyers model other buyers, using an approach that combines both private and public reputation values. In addition, however, sellers model the reputation of buyers. Reputable buyers provide fair ratings of sellers, and are likely considered trustworthy by many other buyers. In marketplaces operating with our mechanism, sellers will offer more attractive products to satisfy reputable buyers, in order to build their reputation. In consequence, our mechanism creates incentives for buyers to provide fair ratings of sellers, leading to more effective e-marketplaces where honest buyers and sellers can gain more profit.

Introduction

In multiagent systems for electronic commerce, trust plays an important role. It provides a form of social control and allows agents in e-marketplaces to reason about reliability, capability and honesty of others, in order to choose the best business partners. Researchers have been working at designing frameworks to model the trust and reputation of agents. A modeling of the trustworthiness of a selling agent can be based on a buying agent's past personal experience with the seller. However, for a new buyer or a buyer without any personal experience with the seller, evaluation of the seller's trustworthiness is often determined by examining the ratings for the seller from other buyers (Sabater & Sierra 2005). The problem of unfair ratings may then arise. Buyers may provide unfairly high ratings to promote the seller. This is referred to as "ballot stuffing" (Dellarocas 2000). Buyers may also provide unfairly low ratings, in order to cooperate with other sellers to drive a seller out of the marketplace. This is referred to as "bad-mouthing".

Besides the problem of unfair ratings, rating submission is voluntary in most trust management systems. Buyers do not have direct incentives to provide ratings because, for example, providing reputation ratings of sellers requires some effort (Jøsang, Ismail, & Boyd 2005; Miller, Resnick, & Zeckhauser 2005). Providing fair ratings for a trustworthy seller may also decrease the chance of doing business with the seller because of competition from other buyers.

To address these two problems, researchers have been working on developing incentive mechanisms. The aim is to encourage honesty in the reporting from buyers, in order to diminish concerns about unfair ratings. Two types of mechanisms have been developed, side payment mechanisms (Jurca & Faltings 2003; Miller, Resnick, & Zeckhauser 2005), and credibility mechanisms (Papaioannou & Stamoulis 2005; Jurca & Faltings 2004). Side payment mechanisms offer side payment to buyers that fairly rate results of business with sellers. In these mechanisms, providing fair ratings for business results is a Nash equilibrium. Credibility mechanisms measure agents' credibility. The credibility of two participants (a buyer and a seller, for example) in their business will be decreased if their ratings about the business result are different. Buyers will provide fair ratings in order to keep up their credibility.

We first develop a model (a personalized approach) that addresses unfair ratings but with more flexibility for buyers to weight their value in private and public reputation ratings of other buyers (advisors). Our aim is to develop improved methods for modeling trustworthiness of advisors by tracking ratings provided according to their related time windows. In so doing, our approach is able to avoid the situation where advisors may untruthfully rate sellers for a large number of times (known as "flooding") and deal with changes of agents' behavior. Our method is able to cope with large numbers of unfair ratings.

Equipped with the richer method for modeling trustworthiness of advisors in terms of private and public reputation, we then propose a novel incentive mechanism. Our mechanism does not rely on side payment. Instead, buyers are encouraged to be truthful in order to gain more numbers of profitable transactions. This idea is supported by the work in the field of evolutionary game theory, such as the work of Gintis et al. (Gintis, Smith, & Bowles 2001). They argue that an agent's altruism in one context signals "quality" of the agent that will benefit from increased opportunities in other wider contexts. Specifically, if the system is such that the provision of truthful reputation feedback makes agents more likely to choose to undertake transactions with the reporting

agent then the reporting agent would benefit for its feedback through a greater number of profitable transactions.

Our personalized approach provides the promising first step for our work. It allows buyers to effectively model the trustworthiness of other buyers. We then use this approach to create a social network of buyers. Each buyer in the society retains a neighborhood of the most trustworthy buyers, as advisors. In our mechanism, we also allow sellers to explicitly model the reputability of buyers, based on the number of neighborhoods they belong to in the society. Buyers that always provide fair ratings of sellers are likely to be neighbors of many other buyers and can be considered reputable. This is also supported by Gintis et al. (Gintis, Smith, & Bowles 2001) through the model of a multi-player game. They argue that agents reporting honestly provide benefit to others and will further be preferred by others as allies. These agents will be able to attract a larger number of audience to witness their feedback (also known as increasing "broadcast efficiency"). Sellers in our system will increase quality and decrease prices of products to satisfy reputable buyers. This therefore creates an incentive for buyers to provide fair ratings of sellers.

The rest of the paper is organized as follows. We first introduce the setting of the electronic marketplace in which our incentive mechanism can operate. We then formalize our mechanism and demonstrate some examples. We also describe our simulations and experimental results. After that, we introduce some related work and contrast approaches of other researchers with our work. Finally, we present conclusions and future work.

E-marketplace Setting

The electronic marketplace environment we are modeling is populated with self-interested agents. Selling agents sell products to buying agents and try to maximize their profit and buyers try to gain good products in terms of, for example, high quality and low prices. There is also a central server, which collects and maintains information about buyers and sellers, including, for example, ratings of sellers. Through this central server, buyers can collaborate and share ratings of sellers. Sellers can also make use of information about buyers maintained by the central server, in order to distinguish them.

The buying and selling process is operated as a procurement (reverse) auction where the auctioneer is a buyer and bidders are sellers.[1] More specifically, a buyer sends to the central server a request containing information about the product it wants to buy. The information includes the buyer's evaluation criteria for the product, which is a function of price and non-price features of the product (delivery time, for instance). In this way sellers are able to know the buyer's values of their products. The central server forwards the request to sellers. We assume that sellers have registered to the central server. Sellers that are interested in selling the product to the buyer will join the procurement auction by submitting bids that describe their settings for prices of the product and values of corresponding non-price features. The auction[2] is similar to Request For Quote (RFQ) introduced by Shachat and Swarthout (Shachat & Swarthout 2003), except that RFQ is an English auction and we use a first-price sealed auction for the purpose of saving communication costs of agents. As also pointed out, an RFQ auction is equivalent to a first-price sealed auction.

The buyer determines the winner of the auction whose product described in its bid has the highest valuation based on the buyer's evaluation criteria. The buyer then pays the winning seller the amount, which is the price in the seller's bid. The winning seller is supposed to deliver the product to the buyer after it receives the payment. However, it may decide to alter the quality of the product actually delivered to the buyer, or not to deliver the product at all. The buyer finally submits a rating to the central server to report the result of the current business with the seller. We assume that a buyer can examine the quality of the product it purchases only after it receives the product. We also assume that there is no complete contract or legal verification to protect buyers from dishonest sellers.

Incentive Mechanism

To formalize the proposed incentive mechanism, we consider the scenario that in an electronic marketplace a buyer B wants to buy a product p. It sends the request to the central server. The request contains information of the buyer's evaluation criteria for a set of non-price features $\{f_1, f_2, ..., f_m\}$, as well as a set of weights $\{w_1, w_2, ..., w_m\}$ that correspond to each non-price feature. Each weight represents how much its corresponding non-price feature is worth. A higher weight for a non-price feature implies that the buyer cares more about the feature. The buyer also provides information in its evaluation criteria about the conversion from descriptive non-price feature values to numeric values (for example, 3 year warranty is converted to the numeric value of 10 on a scale of 1 to 10).[3] We define the function $D()$ to denote such conversion. Inspired by (Boutilier, Sandholm, & Shields 2004), we also use a quasi-linear function to represent the buyer's valuation for the product as follows:

$$V(p) = \sum_{i=1}^{m} w_i D(f_i) - P(p) \quad (1)$$

where $P(p)$ is the price of the product p.

The central server forwards the request to sellers in the marketplace. Sellers $\overline{S}$ that are interested in selling the product to the buyer can submit their bids containing their setting

[1] We use this special setting for demonstrating our proposed approach. However, our incentive mechanism is generally applicable to marketplaces where sellers may alter quality and prices of their products to satisfy honest buyers.

[2] Note that alternative auctions can also be deployed, such as English auction with Bidding Credits (EBC) (Shachat & Swarthout 2003). However, the study of an alternative auction is outside the scope of this paper.

[3] In this paper, we focus on non-price features that are still objective - e.g. delivery time. Handling subjective features is left for future work.

for prices of the product, as well as values for non-price features. The buyer B will then determine the winning seller of the auction, which it can do business with.

In the sections that follow, we first describe how social network of buyers can be created by using our personalized approach. We then formalize how a seller should bid for the buyer's request, by considering the reputation of the buyer modeled based on the social network topology. Finally, we formalize how a buyer should determine the winner of the auction.

Social Network of Buyers

Our mechanism allows the central server to maintain for each buyer a fixed number of neighbors from which the buyer can trust and ask advice about sellers' trustworthiness. The central server models the trust value a buyer has of another buyer (an advisor) through a personalized approach. We first represent private reputation values, based on what is known about the advisors' ratings for sellers with which the buyer has already had some experience. Next, we describe how to construct a public model of trustworthiness of advisors based on common, centrally held knowledge of sellers and the ratings provided by advisors, including the trust ratings of sellers totally unknown to the buyer. We then outline how both private and public models can be combined, in order to obtain a value for the trustworthiness of each possible advisor.

The personalized approach[4] allows the central server to evaluate the private reputation the buyer B has of an advisor A by comparing their ratings for commonly rated sellers $\{S_1, S_2, ..., S_l\}$. For one of the commonly rated sellers S_i ($1 \leq i \leq l$ and $l \geq 1$), A has the rating vector $\overline{r_{A,S_i}}$ and B has the rating vector $\overline{r_{B,S_i}}$. A rating for S_i from B and A is binary ("1" or "0", for example), in which "1" means that the seller delivers the product and the valuation of the product is not less than that described in its bid, and "0" otherwise.[5] In this case, the rating of "1" will be considered as a positive rating, and "0" will be considered as a negative rating. The ratings in $\overline{r_{A,S_i}}$ and $\overline{r_{B,S_i}}$ are ordered according to the time when they are provided. The ratings are then partitioned into different elemental time windows. The length of an elemental time window may be fixed (e.g. one day) or adapted by the frequency of the ratings to the seller S_i, similar to the way proposed in (Dellarocas 2000). It should also be considerably small so that there is no need to worry about the changes of sellers' behavior within each elemental time window. We define a pair of ratings (r_{A,S_i}, r_{B,S_i}), such that r_{A,S_i} is one of the ratings of $\overline{r_{A,S_i}}$, r_{B,S_i} is one of the ratings of $\overline{r_{B,S_i}}$, and r_{A,S_i} corresponds to r_{B,S_i}. The two ratings, r_{A,S_i} and r_{B,S_i}, are correspondent only if they are in the same elemental time window, the rating r_{B,S_i} is the most recent rating in its time window, and the rating r_{A,S_i} is the closest and prior to the rating r_{B,S_i}.[6] We then count the number of such pairs for S_i, N_{S_i}. The total number of rating pairs for all commonly rated sellers, N_{all} will be calculated by summing up the number of rating pairs for each commonly rated seller as follows:

$$N_{all} = \sum_{i=1}^{l} N_{S_i} \tag{2}$$

The private reputation of the advisor is estimated by examining rating pairs for all commonly rated sellers. We define a rating pair (r_{A,S_i}, r_{B,S_i}) as a positive pair if r_{A,S_i} is the same value as r_{B,S_i}. Otherwise, the pair is a negative pair. Suppose there are N_f number of positive pairs. The number of negative pairs will be $N_{all} - N_f$. The private reputation of the advisor A is estimated as the probability that A will provide reliable ratings to B. Because there is only incomplete information about the advisor, the best way of estimating the probability is to use the expected value of the probability. The expected value of a continuous random variable is dependent on a probability density function, which is used to model the probability that a variable will have a certain value. Because of its flexibility and the fact that it is the conjugate prior for distributions of binary events, the beta family of probability density functions is commonly used to represent probability distributions of binary events(see, e.g. the generalized trust models BRS (Jøsang & Ismail 2002) and TRAVOS (Teacy *et al.* 2005)). Therefore, the private reputation of A can be calculated as follows:

$$\alpha = N_f + 1, \beta = N_{all} - N_f + 1$$

$$R_{pri}(A) = E(Pr(A)) = \frac{\alpha}{\alpha + \beta}, \tag{3}$$

where $Pr(A)$ is the probability that A will provide fair ratings to B, and $E(Pr(A))$ is the expected value of the probability.

When there are not enough rating pairs, A's public reputation will also be considered. The public reputation of A is estimated based on its ratings and other ratings for the sellers rated by A. Each time A provides a rating $r_{A,S}$, the rating will be judged centrally as a fair or unfair rating. We define a rating for a seller as a fair rating if it is consistent with the majority of the ratings of the seller.[7] We consider only the ratings that are within the same time window as $r_{A,S}$, and we only consider the most recent rating from each advisor within any time window. In so doing, as sellers change their behavior and become more or less trustworthy to each advisor, the majority of ratings will be able to change.

[4]This approach was first introduced in (Zhang & Cohen 2006).

[5]We could extend our approach to accept ratings in different ranges representing how much more or less the valuation of the product that is delivered compares with that described in the seller's bid. Accordingly, the Dirichlet family of probability density functions would be used to represent probability distributions of ratings.

[6]We consider ratings provided by B after those by A in the same time window, in order to incorporate into B's rating anything learned from A during that time window, before taking an action. According to the solution proposed by Zacharia et al. (Zacharia, Moukas, & Maes 1999), by keeping only the most recent ratings, we can avoid the issue of advisors' "flooding" the system.

[7]Determining consistency with the majority of ratings can be achieved in a variety of ways, for instance averaging all the ratings and seeing if that is close to the advisor's rating.

Suppose that the advisor A totally provides N_{all}^A ratings. If there are N_f^A number of fair ratings, the number of unfair ratings provided by A will be $N_{all}^A - N_f^A$. In a similar way as estimating the private reputation, the public reputation of the advisor A is estimated as the probability that A will provide fair ratings. It can be calculated as follows:

$$\alpha' = N_f^A + 1, \beta' = N_{all}^A - N_f^A + 1$$

$$R_{pub}(A) = \frac{\alpha'}{\alpha' + \beta'}, \tag{4}$$

which also indicates that the more the percentage of fair ratings advisor A provides, the more reputable it will be.

To estimate the trustworthiness of advisor A, we combine the private reputation and public reputation values together. The private reputation and public reputation values are assigned different weights. The weights are determined by the reliability of the estimated private reputation value.

We first determine the minimum number of rating pairs needed for B to be confident about the private reputation value it has of A. The Chernoff Bound theorem (Mui, Mohtashemi, & Halberstadt 2002) provides a bound for the probability that the estimation error of private reputation exceeds a threshold, given the number of rating pairs. Accordingly, the minimum number of pairs can be determined by an acceptable level of error and a confidence measurement as follows:

$$N_{min} = -\frac{1}{2\varepsilon^2} ln \frac{1-\gamma}{2}, \tag{5}$$

where ε is the maximal level of error that can be accepted by B, and γ is the confidence measure. If the total weight of all rating pairs is larger than or equal to N_{min}, buyer B will be confident about the private reputation value estimated based on its ratings and the advisor A's ratings for all commonly rated sellers. Otherwise, there are not enough rating pairs, the buyer will not be confident about the private reputation value, and it will then also consider public reputation. The reliability of the private reputation value can be measured as follows:

$$w = \begin{cases} \frac{N_{all}}{N_{min}} & \text{if } N_{all} < N_{min}; \\ 1 & \text{otherwise.} \end{cases} \tag{6}$$

The trust value of A will be calculated by combining the weighted private reputation and public reputation values as follows:

$$Tr(A) = wR_{pri}(A) + (1-w)R_{pub}(A) \tag{7}$$

It is obvious that the buyer will consider less the public reputation value when the private reputation value is more reliable. Note that when $w = 1$, the buyer relies only on private reputation.

For a new buyer, the central server randomly assigns to it some other buyers with high public reputation as candidates for its neighbors. The new buyer then randomly selects some candidates as its neighbors. The neighbor list will be updated periodically. Each time, the most trustworthy candidates will be selected as neighbors. The candidate list is also updated periodically. Each time, a small portion of buyers is chosen randomly as candidates from all buyers with high public reputation values.

Seller Bidding for Buyer's Request

A seller $S \in \overline{S}$ that is interested in selling p to B submits a bid to the central server. It sets the price and values for the non-price features of the product p, depending on how much instant and expected future profit it can earn from selling p to the buyer B. The instant profit is the profit earned by the seller from the current transaction if it wins the auction. We define the seller's instant profit from selling the product p to the buyer B as follows:

$$U(p) = P(p) - C(p) \tag{8}$$

where $C(p)$ is the cost for the seller to produce the product p with certain values for the non-price features in its bid. The expected future profit the seller can earn depends on the reputation of the buyer. A reputable buyer in this case is one of the neighbors of many other buyers. Cooperating with reputable buyers will allow the seller to build its reputation and to be known as a trustworthy seller by many buyers in the marketplace. It will then be able to obtain more opportunities of doing business with buyers and gain more profit in the future.

To gain profit from each possible transaction, the seller may not include in its bid the true cost of producing product p with certain non-price features. Therefore, it is reasonable to assume that $P(p) > C(p)$. We define the potential gains of the buyer from the transaction as follows:

$$V'(p) = \sum_{i=1}^{m} w_i D(f_i) - C(p) \tag{9}$$

where f_i, $D()$, and w_i are defined earlier in the "Incentive Mechanism" section. We also define the distribution function for $V'(p)$ as $F(V')$, to show the possible values for $V'(p)$.

As argued in (Shachat & Swarthout 2003), a symmetric Bayes-Nash equilibrium can be derived. The equilibrium bidding function of the seller can be derived as follows:

$$P^*(p) = C(p) + \frac{\int_{V_L - C_H}^{V'(p)} F(x)dx}{F(V')} \tag{10}$$

where V_L is the lower bound of the value for the non-price features of p and C_H is the higher bound of the cost for the seller to produce p. We assume $V_L \geq C_H$ to ensure that the value of a seller's product always exceeds its cost.

By taking into account the reputation of the buyer B, the seller has the expected future profit from winning the current auction. It will reduce the instant profit and gain more chance to win the auction if the minimum expected future profit is no less than the loss of the instant profit. The bidding function of the seller in Equation 10 then should be changed to be:

$$P^*(p) = C(p) + \frac{\int_{V_L - C_H}^{V'(p)} F(x)dx}{F(V')} - V_D(R) \tag{11}$$

where $V_D(R)$ is the valuation of discount for the buyer B

with reputation $R(B)$.[8] Comparing Equations 10 and 11, the bidding price of the seller in Equation 11 will be decreased if $V_D(R)$ is greater than 0. The buyer's valuation for the product p will then be increased, according to Equation 1. The seller will be more likely to win the auction. It is also obvious that if the bidding price is fixed, the values of the product's non-price features in the seller's bid will be increased.

As discussed earlier, our mechanism allows the central server to maintain for each buyer a list of neighbors that it trusts the most. A seller can then model the reputation of a buyer based on the number of its neighborhoods (other buyers that include the buyer in their neighbor lists). The seller S periodically acquires neighbor list information of buyers from the central server. It then counts for each buyer the number of neighborhoods. Suppose that there are N_B other buyers considering the buyer B as one of their neighbors. The reputation of B can be calculated as follows:

$$R(B) = \begin{cases} \frac{N_B}{\theta} & \text{if } N_B < \theta; \\ 1 & \text{otherwise.} \end{cases} \tag{12}$$

The value of θ depends on the total number of buyers in the marketplace [9]. The buyer will be considered as reputable if $R(B)$ is no less than a threshold δ. The buyer will be considered as disreputable if its reputation is no larger than a threshold γ ($0 < \gamma < \delta < 1$).

There may exist collusion where dishonest buyers treat each other as neighbors and form a dishonest social network. This problem can be addressed within a centralized architecture. In this case, the seller is allowed to model the trustworthiness of a buyer by checking its ratings provided to the central server by the buyer. If the buyer has provided unfair ratings for the seller, the buyer will be considered untrustworthy by the seller. The seller can maintain a trustworthy buyer list and not enter into auctions of untrustworthy buyers. Trustworthy buyers always provide fair ratings for the seller. Based on the assumption that a trustworthy buyer's neighbors are also likely trustworthy, the seller would then use the list as a basis to find other trustworthy buyers by searching the social network of buyers. From the list of all possible trustworthy buyers that the seller can find, the seller then can correctly model the reputation of a buyer, using Equation 12.

Buyer Choosing Winning Seller

After receiving sellers' bids, the buyer B will then determine the winner of the auction. The winner of the auction is the seller whose bid includes the highest valuation of the product p that it is willing to offer, which can be formalized as follows:

$$S_{win} = arg \max_{S \in \overline{S}} V(p) \tag{13}$$

[8]The amount of discount offered to a buyer depends on how much future profit the seller can gain from conducting the current business with the buyer. We will formalize this discount in our future work after we have better insight into how best to model the expected future profit.

[9]For the examples in this paper, we equate θ with number of buyers. Developing more sophisticated measurements of θ is left for future work.

The buyer chooses the winner of the auction among only sellers that are considered to be trustworthy. As an important component of our proposed marketplace model, the buyer models trustworthiness of a seller by also using a personalized approach. It models private reputation of the seller based on its own ratings for the seller. If the buyer does not have enough personal experience with the seller, it will ask for its neighbors' ratings of the seller. It then can derive a public reputation of the seller from ratings provided by them. The trustworthiness of the seller will be modeled by combining the weighted private and public reputation values. The use of forgetting and discounting factors is included in this part of the model, in an effort to provide a somewhat richer modeling of agents. This is an extension of the personalized method for modeling advisors described in the "Social Network of Buyers" section.

Suppose that B has the rating vector $\overline{r_{B,S}}$, which contains all the ratings provided by B for the seller S. The ratings in $\overline{r_{B,S}}$ are ordered from the most recent to the oldest according to the time when they are submitted. The ratings are then partitioned into different elemental time windows $\{T_1, T_2, ..., T_n\}$. We then count the number of positive ratings $N^B_{pos,i}$ and the number of negative ratings $N^B_{neg,i}$ in each time window T_i. The private reputation of the seller S can be estimated through the beta family of probability density functions as follows:

$$R_{pri}(S) = \frac{\sum_{i=1}^{n} N^B_{pos,i}\lambda^{i-1} + 1}{\sum_{i=1}^{n}(N^B_{pos,i} + N^B_{neg,i})\lambda^{i-1} + 2} \tag{14}$$

where λ ($0 \leq \lambda \leq 1$)is a forgetting rate. The forgetting rate is also introduced by Jøsang and Ismail (Jøsang & Ismail 2002) to deal with possible changes of the seller's behavior over time because old ratings will be given less weight than more recent ones. Note that when $\lambda = 1$ there is no forgetting, and when $\lambda = 0$ only the ratings that are within the current time window T_1 will be considered.

If the buyer B does not have enough personal experience with the seller S, it will also consider ratings provided by its neighbors. The buyer sends a request to the central server to ask for all the ratings provided by its neighbors $\{A_1, A_2, ..., A_k\}$ for the seller S. We also partition these ratings into different elemental time windows. Suppose that the neighbor A_j provided $N^{A_j}_{pos,i}$ positive ratings and $N^{A_j}_{neg,i}$ negative ratings within the time window T_i. These ratings will be discounted based on the trustworthiness of the advisor, so that the ratings from less trustworthy advisors will carry less weight than ratings from more trustworthy ones.

Jøsang (Jøsang 2001) provides a mapping from beliefs defined by the Dempster-Shafer theory to the beta function as follows:

$$\begin{cases} b = \frac{N^{A_j}_{pos,i}}{N^{A_j}_{pos,i} + N^{A_j}_{neg,i} + 2} \\ d = \frac{N^{A_j}_{neg,i}}{N^{A_j}_{pos,i} + N^{A_j}_{neg,i} + 2} \\ u = \frac{2}{N^{A_j}_{pos,i} + N^{A_j}_{neg,i} + 2} \end{cases} \tag{15}$$

where b, d and u represent belief, disbelief and uncertainty parameters, respectively. In our case, b represents the probability that the proposition that the seller is trustworthy is true, and d represents the probability of the proposition is false. Note that $b + d + u = 1$ and $b, d, u \in [0, 1]$. As also pointed out in (Jøsang & Ismail 2002) and (Yu & Singh 2003), beliefs and disbeliefs can be directly discounted by the trustworthiness of the advisor as follows:

$$\begin{cases} b' = Tr(A_j)b \\ d' = Tr(A_j)d \end{cases} \tag{16}$$

From Equations 15 and 16, we then can derive a discounting function for the amount of ratings provided by the advisor A_j as follows:

$$\begin{cases} D^{A_j}_{pos,i} = \frac{2Tr(A_j)N^{A_j}_{pos,i}}{(1-Tr(A_j))(N^{A_j}_{pos,i}+N^{A_j}_{neg,i})+2} \\ D^{A_j}_{neg,i} = \frac{2Tr(A_j)N^{A_j}_{neg,i}}{(1-Tr(A_j))(N^{A_j}_{pos,i}+N^{A_j}_{neg,i})+2} \end{cases} \tag{17}$$

where $Tr(A_j)$ is the trustworthiness of the advisor A_j.

In the same way as estimating the private reputation, the public reputation of the seller S can be calculated as follows:

$$R_{pub}(S) = \frac{[\sum_{j=1}^{k}\sum_{i=1}^{n} D^{A_j}_{pos,i}\lambda^{i-1}] + 1}{[\sum_{j=1}^{k}\sum_{i=1}^{n}(D^{A_j}_{pos,i} + D^{A_j}_{neg,i})\lambda^{i-1}] + 2} \tag{18}$$

The ratings provided by the advisors will be also discounted by the forgetting factor λ.

The trustworthiness of the seller S is estimated by combining the weighted private and public reputation values as follows:

$$Tr(S) = w'R_{pri}(S) + (1 - w')R_{pub}(S) \tag{19}$$

The weight w' is determined by the reliability of the estimated private reputation value as follows:

$$w' = \begin{cases} \frac{N^B_{all}}{N_{min}} & \text{if } N^B_{all} < N_{min}; \\ 1 & \text{otherwise.} \end{cases} \tag{20}$$

where N^B_{all} is the total number of ratings provided by B for the seller. N_{min} represents the minimum number of ratings needed for the buyer B to be confident about the private reputation value it has of S, which can be determined based on Equation 5.

The seller will be considered to be trustworthy only if $Tr(S)$ is no less than a threshold δ'. The seller S will be considered to be untrustworthy if its trust value is no larger than a threshold γ' ($0 < \gamma' < \delta' < 1$).

If there are no trustworthy sellers submitting bids, the winner of the auction will be selected among the sellers with trust values that are between δ' and γ'. Our idea of selective tendering is also supported by Kim's investigation results demonstrated in (Kim 1998). Kim states that public tendering could foster opportunism of quality reduction by bidders; in contrast, selective tendering depending on bidders' trustworthiness may avoid such opportunism.

Examples

In this section, we use some examples to demonstrate how our mechanism works.

Buyer's Neighbor List

We first provide an example to demonstrate how the central server models trust values a buyer B has of other buyers and chooses the most trustworthy ones as B's neighbors. In this example, we assume that each buyer can have at most one neighbor.

Consider the case where there are three other buyers (advisors) A_x, A_y and A_z. Each of them has rated only the five sellers (S_1, S_2, S_3, S_4, and S_5). Table 1 lists the ratings provided by each advisor A_j ($j \in \{x, y, z\}$) for the five sellers. The symbol "T" represents a sequence of time windows, in which T_1 is the most recent time window. To simplify the demonstration, we assume that each advisor provides at most one rating within each time window. We also assume that those are the only ratings provided by them.

Table 2: Ratings Provided by the Buyer B

T	T_1	T_2	T_3	T_4	T_5
S_1	1	1	1	1	1
S_2	1	1	1	1	-
S_3	1	1	1	-	-
S_4	1	1	-	-	-
S_5	1	-	-	-	-

As can be seen from Table 2, the buyer B has also provided some ratings for the five sellers. The buyer B might have not provided any rating for some sellers within some time window. For example, it has provided only one rating for the seller S_5, which is in the time window T_1. We assume that the ratings provided by B are after those provided by A_x, A_y and A_z if they are within the same time window.

We compare the ratings provided by A_x, A_y and A_z in Table 1 and ratings provided by B in Table 2. The buyer B has the same number of rating pairs with each advisor ($N_{all} = 15$). However, B has different numbers of N_f positive rating pairs with A_x, A_y and A_z, which are listed in Table 3. Accordingly, as can be seen from Table 3, the private reputation values of A_x, A_y and A_z are different, in which the private reputation value of A_x is the highest and that of A_z is the lowest. The result indicates that the advisor A_x is most likely to provide fair ratings and have similar preferences with the buyer B, whereas A_z most likely will lie and have different preferences with B.

According to Table 1, the total number of ratings provided by each advisor is the same ($N^{A_j}_{all} = 25$). We also count the number of fair ratings each advisor provides. A rating here is considered as a fair rating when it is consistent with the majority of ratings for the seller within a same time window. Consider the case where all of the five sellers are reputable and the majority of ratings are fair. In this case, a rating of "1" provided by an advisor will be considered as a fair rating, whereas a rating of "0" will be considered as an unfair rating. From the advisors' ratings listed in Table 1, we can

Table 1: Ratings of Sellers Provided by Advisors

A_j	A_x					A_y					A_z				
T	T_1	T_2	T_3	T_4	T_5	T_1	T_2	T_3	T_4	T_5	T_1	T_2	T_3	T_4	T_5
S_1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0
S_2	1	1	1	1	1	0	1	0	1	1	0	0	0	0	0
S_3	1	1	1	1	1	1	0	1	0	0	0	0	0	0	0
S_4	1	1	1	1	1	1	0	0	1	0	0	0	0	0	0
S_5	1	1	1	1	1	1	1	0	0	1	0	0	0	0	0

Table 3: Private and Public Reputation Values of Advisors

A_j	A_x	A_y	A_z
N_f	15	8	0
α	16	9	1
β	1	8	16
$R_{pri}(A_j)$	0.94	0.53	0.06
$N_f^{A_j}$	25	12	0
α'	26	13	1
β'	1	14	26
$R_{pub}(A_j)$	0.96	0.48	0.04

see that ratings provided by the advisor A_x are all fair, the advisor A_z always lies, and some of the ratings provided by the advisor A_y are unfair. Table 3 lists the number of fair ratings provided by each advisor and the corresponding public reputation value of it. From Table 3, it is clear that the advisor A_x is most likely to provide fair ratings, and the advisor A_z most likely will lie.

Table 4: Trustworthiness of Advisors

ε	0.1	0.15	0.2
N_{min}	115	51	29
w	0.13	0.29	0.52
$Tr(A_x)$	0.957	0.954	0.950
$Tr(A_y)$	0.487	0.495	0.506
$Tr(A_z)$	0.043	0.046	0.05

To combine private reputation and public reputation, the weight w should be determined. The value of w depends on the values of ε and γ, and the number of rating pairs N_{all}, which is the same for every advisor in our example. Suppose we have a fixed value, 0.8 for γ, which means that the confidence value should be no less than 0.8 in order for the buyer to be confident with the private reputation values of advisors. In this case, the larger the value of ε the buyer sets, the more confident it is with the private reputation values of advisors, which also means that the more weight it will put on the private reputation values. Table 4 lists different acceptable levels of errors, their correspondent weights of private reputation values, and different results of trust values. It clearly indicates that A_x is the most trustworthy. As a result, the buyer B will choose A_x as its neighbor. In the examples that follow, we set $\varepsilon = 0.2$ and $\gamma = 0.8$. The trustworthiness of A_x is then 0.95.

Buyer Choosing Winning Seller

We then use an example to demonstrate how the buyer B models trustworthiness of sellers by considering ratings of sellers provided by its neighbors, and how it selects the winning seller to do business with. Suppose that the buyer B has two non-price features for the product p that it wants to buy. The buyer specifies a weight for each non-price feature and the information about the conversion from descriptive non-price feature values to numeric values, as presented in Table 5. To prevent it from doing business with possibly dishonest sellers, the buyer B models trustworthiness of sellers and selects trustworthy ones to do business with. Suppose that the four sellers S_6, S_7, S_8 and S_9 are all willing to sell the buyer the product p and have submitted their bids. We also suppose that the buyer B previously has not done business with any one of them. Therefore the buyer B has no ratings for these sellers. The private reputation of S_6, S_7, S_8 and S_9 can be calculated according to Equation 14 as follows:

$$R_{pri}(S_6|S_7|S_8|S_9) = \frac{0+1}{(0+0)+2} = 0.5$$

Table 6: Ratings of Sellers Provided by A_x

T	T_1	T_2	T_3	T_4	T_5
S_6	0	0	0	1	1
S_7	-	-	-	-	-
S_8	1	1	1	1	1
S_9	1	1	1	1	0

Table 7: Discounted Amount of Ratings of Sellers

T	T_1	T_2	T_3	T_4	T_5
$D^{A_x}_{pos,i}(S_1)$	0	0	0	0.93	0.93
$D^{A_x}_{neg,i}(S_1)$	0.93	0.93	0.93	0	0
$D^{A_x}_{pos,i}(S_2)$	0	0	0	0	0
$D^{A_x}_{neg,i}(S_2)$	0	0	0	0	0
$D^{A_x}_{pos,i}(S_3)$	0.93	0.93	0.93	0.93	0.93
$D^{A_x}_{neg,i}(S_3)$	0	0	0	0	0
$D^{A_x}_{pos,i}(S_4)$	0.93	0.93	0.93	0.93	0
$D^{A_x}_{neg,i}(S_4)$	0	0	0	0	0.93

The buyer B then considers ratings of sellers provided by its neighbor A_x. The ratings of the sellers provided by

Table 5: Buyer B's Evaluation Criteria for p

Features	Delivery Time			Warranty		
Weights	0.4			0.6		
Descriptive values	1 week	3 days	1 day	1 year	2 years	3 years
Numerical values	3	5	10	3	5	10

the advisor A_x are listed in Table 6. Note that the advisor A_x does not have ratings for the seller S_7 because A_x has not done business with S_7. The amount of positive or negative ratings provided by A_x within each time window will be discounted by using Equation 17. The discounted amount of positive and negative ratings of sellers is listed in Table 7. For example, the discount amount of positive ratings of seller S_6 in time window T_4 is calculated to be 0.93.

In this example, we set λ to be 0.9, which means that the buyer B does not have much forgetting. According to Equation 18, the public reputation of the sellers can be calculated as follows:

$$R_{pub}(S_6) = \frac{\sum_{i=4}^{5} 0.93 * 0.9^{i-1} + 1}{\sum_{i=1}^{5} 0.93 * 0.9^{i-1} + 2} = 0.39$$

$$R_{pub}(S_7) = 0.5, \quad R_{pub}(S_8) = 0.83, \quad R_{pub}(S_9) = 0.72$$

Because the buyer B has not done business with any of the sellers before, the weights of the private reputation of the sellers are all 0. The trustworthiness of the sellers can be calculated by using Equation 19 as follows:

$$Tr(S_6) = 0 * 0.5 + (1 - 0) * 0.39 = 0.39$$

$$Tr(S_7) = 0.5, \quad Tr(S_8) = 0.83, \quad Tr(S_9) = 0.72$$

We set the threshold δ' to be 0.7. In this case, only the sellers S_8 and S_9 will be considered as trustworthy sellers by the buyer B.

We suppose that the sellers S_8 and S_9 may have different costs of producing the product p with certain features. The bid submitted by the seller S_8 specifies that S_8 will deliver the product with 3 year warranty in three days and the price of the product is 4. The bid submitted by the seller S_9 specifies that S_9 will deliver the product with 2 year warranty in three days and the price of the product is also 4. The values of the product p in their bids are calculated as follows:

$$V(p, S_8) = 0.4 * 5 + 0.6 * 10 - 4 = 4, \quad V(p, S_9) = 1$$

The value of the product in the bid of S_9 is lower than that of S_8. Seller S_8 will be selected as the winner. Buyer B pays S_8 the price of 4. Later on, seller S_8 delivers the product. Suppose that the seller delivers the product with 3 year warranty in three days; we say that the seller is trustworthy in this transaction. Buyer B will submit a rating of "1" to the central server.

Seller Bidding for Buyers' Requests

In this example, we illustrate how a seller S_{10} models reputation of buyers and specifies its bids for buyers' requests according to their reputation values. Suppose that there are 6 buyers, $\{B_1, B_2, B_3, B_4, B_5, B_6\}$. They request the same product p with two non-price features. The weight for each non-price feature and the information about the conversion from descriptive non-price feature values to numeric values are as presented in Table 5. The seller S_{10} needs to decide how to bid for each buyer's request. It models the reputation of each buyer.

Table 8: Neighbors of Buyers

Buyer	Neighbors		
B_1	B_2	B_5	B_6
B_2	B_4	B_5	B_6
B_3	B_4	B_5	B_6
B_4	B_3	B_5	B_6
B_5	B_3	B_4	B_6
B_6	B_3	B_4	B_5

Assume that each buyer is allowed to have only 3 neighbors in this example. The neighbors of each buyer are listed in Table 8. We count the number of neighborhoods for each buyer as follows:

$$N_{B_1} = 0, \quad N_{B_2} = 1, \quad N_{B_3} = 3$$

$$N_{B_4} = 4, \quad N_{B_5} = 5, \quad N_{B_6} = 5$$

If we set θ to be 6, we then calculate the reputation of each buyer according to Equation 12 as follows:

$$R(B_1) = 0, \quad R(B_2) = 0.17, \quad R(B_3) = 0.5$$

$$R(B_4) = 0.67, \quad R(B_5) = 0.83, \quad R(B_6) = 0.83$$

We set δ to be 0.8 and γ to be 0.3. Then, the buyers B_5 and B_6 are considered as reputable buyers, and B_1 and B_2 are disreputable buyers.

Table 9: Profit Gained by Different Buyers

Buyers	Features of Product			Profit
	Warranty	Delivery Time	Price	
B_1, B_2	1 year	1 week	5	-2
B_3, B_4	2 years	3 days	4	1
B_5, B_6	3 years	1 day	3	7

According to the reputation of each buyer, seller S_{10} specifies its bid for each buyer's request. The non-price and price features in each bid and profit that each buyer can gain are

listed in Table 9. From this table, we can see that the reputable buyers B_5 and B_6 are able to gain the largest profit and the disreputable buyers B_1 and B_2 can gain the smallest profit.

Experimental Results

We carry out experiments to examine each expectation of our mechanism. We also measure profit gained by different buyers and sellers. The expectation is that reputable buyers and sellers that are considered as trustworthy by many buyers will be able to gain more profit.

We simulate a marketplace operating with our mechanism in the period of 20 days. The marketplace involves 100 buyers. These buyers have different numbers of requests. Every 10 of them has a different number (from 2 to 20) of requests. In our experiments, we assume that each buyer will submit a rating for each of its requests. Therefore, buyers that have larger number of requests will provide larger number of ratings. We also assume that there is only one product in each request and each buyer has a maximum of one request each day. For the purpose of simplicity, we also assume that the products requested by buyers have the same non-price features. After they finish business with sellers, buyers rate sellers. 50 buyers provide unfair ratings. Every 10 of them provides different percentages (from 10% to 50%) of unfair ratings. Initially, we randomly assign 5 buyers to each buyer as its neighbors.

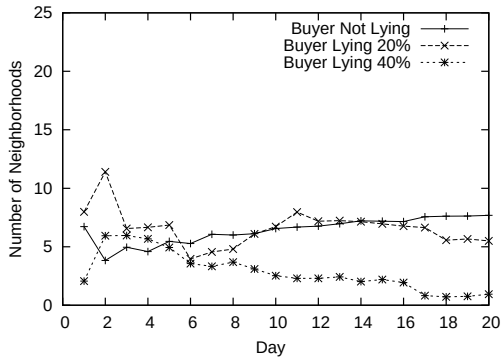

Figure 1: Reputation of Different Buyers

There are 10 sellers in total in the marketplace. Each 2 sellers acts dishonestly in different percentages (0%, 25%, 50%, 75% and 100%) of their business with buyers. One half of the sellers model reputation of buyers and adjust prices of products according to buyers' reputation. Another 5 sellers do not model reputation of buyers. They offer the same price for products requested by buyers. We assume that all sellers have the same cost for producing the products because all products have the same non-price features.

We first measure the reputation of buyers that provide different numbers of unfair ratings. The results are shown in Figure 1. In our experiments, the reputation of a buyer is represented by the number of the buyer's neighborhoods. From this figure, we can see that the buyers providing the smaller number of unfair ratings will have the larger reputation values. Due to the randomness of the initial setting for our experiments, buyers providing more unfair ratings may have larger reputation values at the beginning. But their reputation will continuously decrease after approximately 10 days, as can be seen from Figure 1. After approximately 14 days when our marketplace converges, the buyers providing more unfair ratings will have smaller reputation values. We also measure reputation of buyers that have different numbers of requests. Results are shown in Figure 2. Buyers having more requests (that have provided more ratings) will have larger reputation values. Similarly, reputation values of buyers change stochastically at the beginning. But when the marketplace converges, the buyers having fewer requests will have the smaller reputation values.

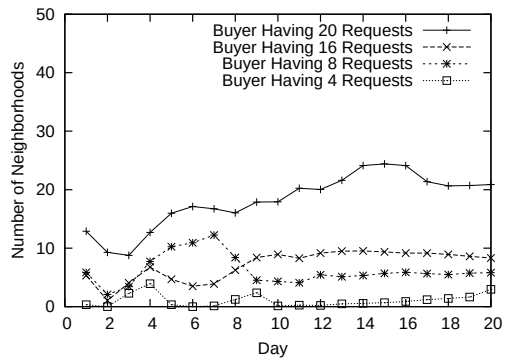

Figure 2: Reputation of Different Buyers

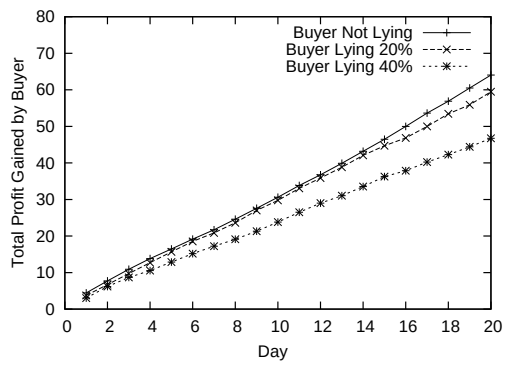

Figure 3: Profit Gained by Different Buyers

After each day, we measure total profit gained by buyers that provide different numbers of unfair ratings. The profit gained by a buyer from buying a product is the valuation of the product received from its business partner. From Figure 3, we can see that buyers providing fewer unfair ratings will gain more total profit. Note that we do not measure total profit gained by buyers that have different numbers of requests. It is essential that the more requests the buyer has, the more profit it will be able to gain. In summary, it is better off for buyers to provide more fair ratings. Also note that the profit difference of different types of buyers is fairly small. It is because buyers have at most 20 requests in total.

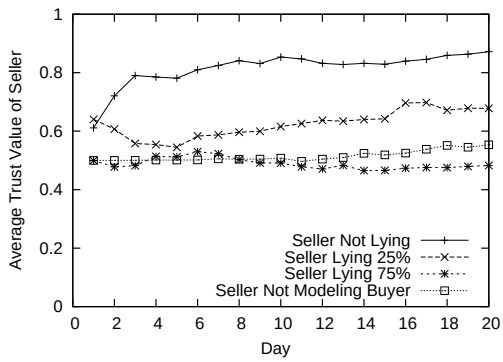

Figure 4: Average Trust Value of Different Sellers

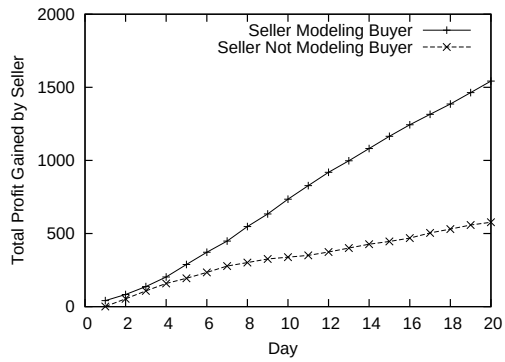

Figure 6: Total Profit Gained by Different Sellers

We compare average trust values of different sellers. The average trust value of a seller is calculated as the sum of a trust value each buyer has of the seller divided by the total number of buyers (100 in our experiments). As shown in Figure 4, results indicate that sellers being dishonest more often will have smaller average trust values. The sellers that do not model reputation of buyers and adjust their prices of products according to buyers' reputation will also have smaller average trust values. From Figure 4, we can see that their average trust values are nearly 0.5. It is because that they do not have much chance to do business with buyers and will not have many ratings. A seller without any ratings will have trust value of 0.5 (for example, the seller S_7 in the "Examples" section). Similarly, the sellers being dishonest in 75% of their business also will not have much chance to do business with buyers and will have a trust value of nearly 0.5.

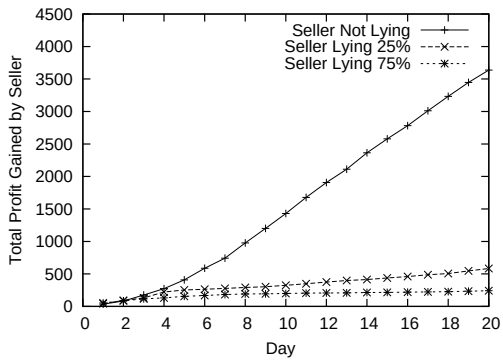

Figure 5: Total Profit Gained by Different Sellers

We also compare total profit gained by different sellers. Results are shown in Figures 5 and 6. From Figure 5, we can see that sellers being honest more often will gain more profit. Therefore, it is better off for sellers to be honest. We can also see that the profit difference between the honest sellers and the sellers lying 25% is much larger than that between the sellers lying 25% and the sellers lying 75%. The reason is that we set the threshold δ' to be very high ($\delta' = 0.8$). The sellers lying 25% will not be considered as trustworthy sellers, therefore will have small chance to be selected as business partners by buyers. Results in Figure 6 indicate that sellers are better off to model reputation of buyers and adjust prices of products according to buyers' reputation, in order to gain more profit.

Related Work

There are other incentive mechanisms eliciting fair ratings. One type of such mechanisms is side payments (Dellarocas 2002; Jurca & Faltings 2003; Miller, Resnick, & Zeckhauser 2005). We survey three side payment mechanisms. They are different, for example, in terms of which party pays to honest buyers and/or in ways of evaluating the truthfulness of buyers' ratings. Another type of incentive mechanisms is credibility mechanisms (Papaioannou & Stamoulis 2005; Jurca & Faltings 2004) where only honest agents have their credibility in the marketplace enhanced. We point out some shortcomings of these methods and provide a contrast between our approach and those of other researchers.

Side Payment Mechanism

Dellarocas (Dellarocas 2002) proposes "Goodwill Hunting" (GWH) as a feedback mechanism for a trading environment based upon the argument that truthful feedback will benefit the community as a whole. This mechanism elicits truthful feedback from buyers by offering rebates of buyers' membership fee if the mean and variance between the buyers' and sellers' perception of quality of their transactions are consistent across the entire buyer community. Buyers may behave badly before they exit from the market. To solve this problem, part of the membership fee will be refunded only at the end of the period on the basis of buyers' behavior.

In the incentive compatible mechanism proposed by Jurca and Faltings (Jurca & Faltings 2003), a set of broker agents called R-agents, can sell and buy ratings of sellers to and from other ordinary agents. These ordinary agents first buy ratings from broker agents. After they finish doing business with sellers, they can sell ratings of the sellers back to the broker agents from which they bought ratings. To balance payoffs, ordinary agents are only allowed to sell ratings of a seller if they have previously bought reputation ratings of the seller. An agent will get paid only if a rating of a seller

they provide is the same as the next rating of the same seller provided by another agent. A simple two agents case in an iterated Prisoner's Dilemma environment proves that the optimal strategy for an agent is to report truthfully because it will get paid with probability of at least 0.5.

Miller et al. (Miller, Resnick, & Zeckhauser 2005) introduces a mechanism which is very similar to that proposed by Jurca and Faltings (Jurca & Faltings 2003). In the mechanism, there is a center that maintains buyers' ratings. The center rewards or penalizes each buyer on the basis of its ratings and ensures that the mechanism at least breaks even in the long run. More specifically, a buyer providing truthful ratings will be rewarded and get paid not by broker agents but by the buyer after the next buyer. To balance transfers among agents, a proper scoring rule is used to determine the amount that each agent will be paid for providing truthful feedback. Scoring rules used by the center (i.e. the Logarithmic Scoring Rule) make truthful reporting a Nash equilibrium. Furthermore, proper scalings of scoring rules and collection of bonds or entry fees in advance ensure budget balance and incentives of the mechanism.

In summary, side payment mechanisms offer side payment to buyers that fairly rate results of business with sellers. However, they do not work well if the majority of buyers elect to provide unfair ratings because each of these dishonest buyers will receive a reward. This means that honest buyers that will not be giving similar ratings as many others, will not be rewarded and will be discouraged from being honest in the future. Moreover, this approach assumes that buyers act independently, and therefore has difficulty with the situation where buyers collude in giving unfair ratings. In contrast, in our mechanism, sellers can view the ratings provided by buyers and can in this way detect dishonesty. Since sellers also only reward reputable buyers, buyers that collude in providing dishonest ratings will not profit. In addition, honest buyers will not be adversely affected by collusion in the marketplace; with our personalized approach for modeling the trustworthiness of advisors, each buyer can rely on private knowledge to detect dishonest buyers and will limit their neighborhood of advisors to those that are determined to be trustworthy.

Credibility Mechanism

Instead of giving instant payment to agents that provide truthful ratings, credibility mechanisms measure agents' credibility or non-credibility according to their past ratings. It is believed that agents are more likely to conduct business with credible other ones.

One credibility mechanism is introduced by Papaioannou and Stamoulis (Papaioannou & Stamoulis 2005) for eliciting truthful ratings in peer-to-peer systems. Besides reputation information, each peer also stores a non-credibility value and a binary punishment state variable. After each transaction between two peers, they submit a rating indicating whether the transaction is successful or not. If both of them agree with the result of the transaction, their non-credibility values will be decreased. Otherwise, their non-credibility values will be increased and they will be punished. They will be forced not to conduct any transactions for a period determined by each of their non-credibility values.

A slightly different credibility mechanism called "CONFESS" is proposed by Jurca and Faltings (Jurca & Faltings 2004) for the online hotel booking industry. In this mechanism, a seller first reports its behavior. If it claims having cooperated, the buyer is then asked to submit a rating. If the buyer also reports that the seller has cooperated, it is sure that the seller has cooperated. Otherwise, both of them will be punished by decreasing their credibility as untruthful reporters because in this case at least one of them is cheating.

In credibility mechanisms, the credibility of two participants (a buyer and a seller, for example) in their business will be decreased if their ratings about the business result are different. Buyers will provide fair ratings in order to keep up their credibility. In these mechanisms, honest agents will be unfairly punished if they meet with a dishonest agent because they will not agree when they rate the results of their transactions with the agent. These honest agents will not gain credibility even if they provide good services. In addition, credibility mechanisms cannot deal with the situation where buyers and sellers collude to increase each other's credibility. Because our mechanism allows buyers to maintain a list of trustworthy other buyers as their neighbors, a buyer can make an informed decision about which sellers to do business with. If a buyer were to accept the advice of another agent that is colluding with a seller and then be disappointed with the purchase, the advisor would be considered untrustworthy and would not impact any future decisions. In addition, all buyers have incentives to be honest, in order to enjoy the rewards offered by the honest sellers of the marketplace, if they maintain their position in many neighborhoods of the social network.

Conclusions and Future Work

In this paper, we propose a novel incentive mechanism to elicit fair ratings of sellers in electronic marketplaces. In our mechanism, a buyer maintains a neighbor list of other buyers that always provide fair ratings. We allow sellers to see how they have been rated by buyers and to model the reputation of buyers based on the social network created using our personalized approach. Reputable buyers are likely to be neighbors of many other buyers. Sellers then increase quality and/or decrease prices of products to buyers that are determined to be reputable. Hence, buyers are better off providing truthful feedback and becoming neighbors of as many other buyers as possible. Sellers are also kept honest, because buyers are modeling the fairness of ratings provided by other agents, when forming their neighbor lists of other buyers. Sellers are motivated to provide quality service to reputable buyers, in order to progressively build their reputation in the social network. The above expectations are upheld in our model and shown by our experiments. In summary, our mechanism is able to create a more effective electronic marketplace for buyers and sellers to do business with each other. In such an environment, honesty is promoted amongst buyers and sellers, and both honest parties participating in business are able to gain more profit. This also engenders trust of buying and selling agents from their human owners.

Our mechanism allows sellers to model reputation of a buyer based on the number of other buyers including the buyer in their neighbor lists. In future work, we will consider a more comprehensive approach for modeling buyers' reputation. The reputation of buyers that include the buyer in their neighbor lists could also be taken into account. How best to form neighborhoods in the marketplace is another open question for research. We will also need to further study the properties of our social network, for example, the proper size of each neighbor list reflecting the population of buying and selling agents in the marketplace and how actively buying agents rate selling agents. Larger neighborhood size will increase the computation of maintaining and updating buying agents' neighbor lists, and may decrease the accuracy for predicting selling agents' trustworthiness from feedback provided by neighbors. Smaller neighborhood size may increase the accuracy, but will have higher chance the neighbors have insufficient experience (Herlocker, Konstan, & Riedl 2002).

Another topic for future work is to examine marketplaces where the identity of buyers is shielded from the seller, to prevent sellers from trying to cheat less reputable buyers (that do not have much impact on the seller's reputation). For instance, the seller could submit bids for certain classes of buyers to the central server and indicate its value for the reputation of each buyer. The central server could then deliver the appropriate bid to the buyer trying to purchase from this seller and keep the buyer's identity protected.

We will also develop more extensive experiments to validate our model. We are particularly interested in determining how robust our model is in coping with various types of collusion, including buyers colluding with sellers in providing unfairly high ratings and buyers colluding with other buyers in giving unfairly low ratings to sellers. It would also be useful to examine the case where some agents may vary their behavior widely, or where agents may enter and leave the marketplace.

References

Boutilier, C.; Sandholm, T.; and Shields, R. 2004. Eliciting bid taker non-price preference in (combinatorial) auctions. In *Proceedings of the Nineteenth National Conference on Artificial Intelligence (AAAI-04)*.

Dellarocas, C. 2000. Immunizing online reputation reporting systems against unfair ratings and discriminatory behavior. In *Proceedings of the 2nd ACM Conference on Electronic Commerce*.

Dellarocas, C. 2002. Goodwill hunting: An economically efficient online feedback mechanism for environments with variable product quality. *J. Padget and et al., editors, Agent-Mediated Electronic Commerce IV. Designing Mechanisms and Systems, volume LNCS 2531, Springer Verlag* 238–252.

Gintis, H.; Smith, E. A.; and Bowles, S. 2001. Costly signaling and cooperation. *Journal of Theoretical Biology* 213:103–19.

Herlocker, J.; Konstan, J. A.; and Riedl, J. 2002. An empirical analysis of design choices in neighborhood-based collaborative filtering algorithms. *Information Retrieval* 5(4):287–310.

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

Jøsang, A.; Ismail, R.; and Boyd, C. 2005. A survey of trust and reputation systems for online service provision. *(to appear). Decision Support Systems*.

Jøsang, A. 2001. A logic for uncertain probabilities. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems* 9(3):279–311.

Jurca, R., and Faltings, B. 2003. An incentive compatible reputation mechanism. In *Proceedings of the IEEE Conference on E-Commerce, Newport Beach, CA, USA*.

Jurca, R., and Faltings, B. 2004. Confess: An incentive compatible reputation mechanism for the online hotel booking industry. In *Proceedings of the IEEE Conference on E-Commerce, San Diego, CA, USA*.

Kim, I.-G. 1998. A model of selective tendering: Does bidding competition deter opportunism by contractors? *The Quarterly Review of Economics and Finance* 38(4):907–925.

Miller, N.; Resnick, P.; and Zeckhauser, R. 2005. Eliciting informative feedback: The peer-prediction method. *Management Science* 51(9):1359–1373.

Mui, L.; Mohtashemi, M.; and Halberstadt, A. 2002. A computational model of trust and reputation. In *Proceedings of the 35th Hawaii International Conference on System Science (HICSS)*.

Papaioannou, T. G., and Stamoulis, G. D. 2005. An incentives' mechanism promoting truthful feedback in peer-to-peer systems. In *Proceedings of IEEE/ACM CCGRID 2005 (Workshop on Global P2P Computing)*.

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

Shachat, J., and Swarthout, J. T. 2003. Procurement auctions for differentiated goods. *Working Paper, IBM Research Labs*.

Teacy, W. T. L.; Patel, J.; Jennings, N. R.; and Luck, M. 2005. Coping with inaccurate reputation sources: Experimental analysis of a probabilistic trust model. In *Proceedings of 4th International Autonomous Agents and Multi Agent Systems (AAMAS 2005)*.

Yu, B., and Singh, M. 2003. Detecting deception in reputation management. In *Proceedings of International Autonomous Agents and Multi Agent Systems (AAMAS)*.

Zacharia, G.; Moukas, A.; and Maes, P. 1999. Collaborative reputation mechanisms in electronic marketplaces. In *Proceedings of the 32nd Hawaii International Conference on System Sciences (HICSS-32)*.

Zhang, J., and Cohen, R. 2006. A personalized approach to address unfair ratings in multiagent reputation systems. In *Proceedings of the Fifth International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS'06) Workshop on Trust in Agent Societies*.