A qualitative reputation system for multiagent systems with protocol-based communication

Emilio Serrano
Facultad de Informática
Universidad de Murcia
Murcia, Spain
emilioserra@um.es

Michael Rovatsos
School of Informatics
University of Edinburgh
Edinburgh EH8 9AB, UK
mrovatso@inf.ed.ac.uk

Juan Botia
Facultad de Informática
Universidad de Murcia
Murcia, Spain
juanbot@um.es

ABSTRACT

We propose a novel method for assessing the reputation of agents in multiagent systems that is capable of exploiting the structure and semantics of rich agent interaction protocols and agent communication languages. Our method is based on using so-called *conversation models*, i.e. succinct, qualitative models of agents' behaviours derived from the application of data mining techniques on protocol execution data in a way that takes advantage of the semantics of inter-agent communication available in many multiagent systems. Contrary to existing systems, which only allow for querying agents regarding their assessment of others' reputation in an *outcome-based* way (often limited to distinguishing between "successful" and "unsuccessful" interactions), our method allows for contextualised queries regarding the structure of past interactions, the values of content variables, and the behaviour of agents across different protocols. Moreover, this is achieved while preserving maximum privacy for the reputation querying agent and the witnesses queried, and without requiring a common definition of reputation, trust or reliability among the agents exchanging reputation information. A case study shows that, even with relatively simple reputation measures, our qualitative method outperforms quantitative approaches, proving that we can meaningfully exploit the additional information afforded by rich interaction protocols and agent communication semantics.

Categories and Subject Descriptors

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

General Terms

Algorithms, Theory, Design

Keywords

Trust and reputation, agent communication, data mining

1. INTRODUCTION

Reputation, i.e. the beliefs or opinions generally held about other agents in a society, is one of the main means of evaluating the trustworthiness and reliability of individuals in multiagent systems (MASs). In the trust and reputation literature [5], trust is usually taken to denote the belief that a party will act cooperatively and not fraudulently, while reputation normally refers to trust information propagated through a social network of individuals [6]. The autonomy and heterogeneity of agents in open MASs makes the use of reputation in MASs particularly challenging, and often impedes the use of centralised trustworthy authorities such as the reputation models implemented in some internet-based markets, e.g. Amazon [1] or eBay [2] (although such reputation mechanisms are certainly most popular in the real world [5]). Yet, from the point of view of an agent, a correct assessment of others' reputation may greatly enhance performance, as it can be used to make appropriate decisions regarding which agents to interact with and how to behave in these interactions.

Existing trust and reputation approaches [5, 11, 8, 12] mostly focus on a purely *quantitative* assessment of trust, based on witness reports regarding positive/successful and negative/unsuccessful interaction experiences, usually only making binary (or one-dimensional numerical) distinctions resulting focussing on a single property of interactions that describes the trustworthiness or reliability of the target (i.e. reputation-evaluated) agent. Even when these methods allow for queries with a more "semantic" content (to ask for the reputation of an agent with regard to particular products, types of services, etc) [8], the assessment is always entirely *outcome*-oriented, and allows no assessment of the qualitative properties of the interaction process, i.e. the content and sequence of messages exchanged and physical actions observed.

Adopting this quantitative perspective effectively ignores the interaction mechanisms provided by many multiagent systems, in particular complex structure-rich interaction protocols that use agent communication languages (ACLs) with formal semantics. As opposed to low-level interaction mechanisms in other distributed systems, these languages and protocols attempt to capture shared meaning for messages exchanged in MASs, and the structure and "knowledge-level" assumptions captured in ACLs and interaction protocols is semantically rich and can be used to extract *qualitative properties* of observed conversations among agents.

In this paper, we introduce a novel reputation system based on the *qualitative context mining* approach proposed by Serrano *et al* [10], which allows us to exploit the semantics and structure of agent interactions, in order to produce better, contextualised assessments of reputation that can be tailored to the needs of the reputation-evaluating agent and

Appears in: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, Conitzer, Winikoff, Padgham, and van der Hoek (eds.), 4-8 June 2012, Valencia, Spain.

inform her interaction decisions. Our method is based on extracting succinct models of the evaluated agents' behaviour from previous interaction data. These can be queried by the evaluating agent (whether or not she is the *modelling* agent who has constructed the conversation model) with respect to specific protocols, paths within these protocols, or values of constraint arguments that are part of the protocol definition. Our approach minimises information disclosure among agents: The evaluating agent might request the entire conversation model from the modelling agent (which does not require the modelling agent to share her original interaction data, and thus also limits the bandwidth needed for data exchange) and perform queries herself on it (to avoid sharing definitions of what counts as "trustworthy" or "untrustworthy" to her). Alternatively, the evaluating agent can share these definitions of trustworthiness and query a modelling agent unwilling to transmit her conversation model, and only obtain reputation assessments in return, without access to the full conversation model.

What is more, through experiments in an example e-commerce scenario, we show that our reputation system is capable of effectively utilising the additional information provided by rich interaction protocols and ACLs, and results both in better predictions of future interaction behaviour of evaluated agents, and in improved responsiveness to unexpected changes in others' behaviours. This can be achieved by defining relatively straightforward reputation measures on top of the qualitative reputation assessment mechanism.

The remainder of the paper is structured as follows: Section 2 reviews the qualitative context mining approach suggested in [10] and describes how it is used as a basis for interaction data analysis in our system. In section 3, we introduce the proposed reputation measures that can be defined on top of our qualitative data analysis method. An empirical analysis of our method is presented in section 4. Section 5 discusses related work, and section 6 concludes.

2. MINING AGENT CONVERSATIONS

As described above, our reputation system uses the framework proposed in [10] as a base method for interaction analysis. The context mining approach presented there does not assume a specific protocol or agent communication language for MASs, but represents protocols in a very general way as graphs whose nodes are speech-act like messages placeholders, and whose edges define transitions among messages that give rise to message sequences specified as admissible according to the protocol. The edges are labelled with logical constraints, i.e. formulate logical conditions that the agent using the protocol is able to verify. These act as guards on a given transition, so that the message corresponding to a child node can only be sent if the constraint(s) along its incoming edge from the parent node (the message just observed) can be satisfied.

[10] defines a *protocol model* as a graph $G = (V, E)$ where nodes $v \in V$ are labelled with messages $m(v) = q(X, Y, Z)$, q is a performative and X, Y, and Z are sender/receiver/-content variables, respectively. Edges are labelled with a (conjunctive) list of logical constraints

$$c(e) = \{c_1(t_1, \ldots, t_{k_1}), \ldots, c_n(t_1, \ldots t_{k_n})\}$$

where each constraint $c_i(\ldots)$ has arity k_i, head c_i and arguments t_j. Constraints can be arbitrary logical formulas composed of predicates which may contain constants, functions or variables, with all variables implicitly universally quantified. It is assumed that all outgoing edges of a node result in messages with distinct performatives, i.e. for all $(v, v') \in E$, $(v, v'') \in E$

$$(m(v') = q(\ldots) \wedge m(v'') = q(\ldots)) \Rightarrow v' = v''$$

so that each observed message sequence corresponds to (at most) one path in G by virtue of its performatives. Figure 1 shows an example protocol model in this generic format.

The *semantics* of a protocol model G is based on considering finite paths $\pi = v_1 \stackrel{e_1}{\rightarrow} v_2 \stackrel{e_2}{\rightarrow} \ldots \stackrel{e_{n-1}}{\rightarrow} v_n$ in the graph G (which may include unfoldings of cycles, assuming fresh variable names each time a node is revisited). If $\mathbf{m} = \langle m_1, \ldots, m_n \rangle$ are the ground messages observed in a run, $G(\mathbf{m}) = \langle \pi, \theta \rangle$ returns the (unique) path π that can be traced in G following the observed messages, and θ is the most general unifier of the set

$$\{m_1, \ldots, m_n\} \cup \{m(v_i) | 1 \leq i \leq n\}$$

and $\pi = v_1 \stackrel{e_1}{\rightarrow} \ldots \stackrel{e_{n-1}}{\rightarrow} v_n$. In other words, the pair $\langle \pi, \theta \rangle$ returns the path and variable substitution the message sequence $\mathbf{m}$ corresponds to in protocol model G. While context models are defined in [10] based on an analysis of the logical formula resulting from constraints along a path, for our purposes it is sufficient to consider pairs $\langle \pi, \theta \rangle$ that correspond to message sequences $\mathbf{m}$ of past observed interactions as samples for data mining algorithms.

To explain how we proceed in collecting and processing samples of protocol executions, consider the protocol model shown in figure 1. An execution run using this model will consist of a sequence of messages and constraints satisfied along that path (or, at least, presumably satisfied, assuming that the other agent only utters a message when its preconditions are satisfied) and will be translated to a list of feature-value pairs where the features are variables used in the messages, and the values their respective ground instantiations. In terms of actual data mining methods used, we restrict ourselves here to decision tree learning (we use *J48*, an open source implementation of the C4.5 algorithm [3]). Though [10] compares several other techniques, our system operates on trees like the one shown in the example of figure 2 obtained from the protocol in figure 1. As in our evaluation in section 4, this is derived from a scenario where agents use the protocol to negotiate over cars using a well-known database for car evaluation [4]. In this scenario, the modelling agent (who builds the tree from past data) is a potential customer (role A) who has requested offers from a car selling agent (role B) where T specifies the technical characteristics of the car, including number of doors, capacity in terms of persons to carry, the size of the luggage boot, the estimated safety of the car, price and maintenance cost. We assume that a feature vector for terms is of the form

$$T = (doors, persons, lug\_boot, safety, price, maint)$$

where

$doors \in \{2, 3, 4, 5\text{-}more\}$	$persons \in \{2, 4, more\}$
$maint \in \{v\text{-}high, high, med, low\}$	$safety \in \{low, med, high\}$
$price \in \{v\text{-}high, high, med, low\}$	$lug\_boot \in \{small, med, big\}$

The conversation model shown in figure 2, for example, shows that seller S_8, for instance, performed 44 successful negotiations but also that these involved cars with a low maintenance cost, medium safety, and a low buying price.

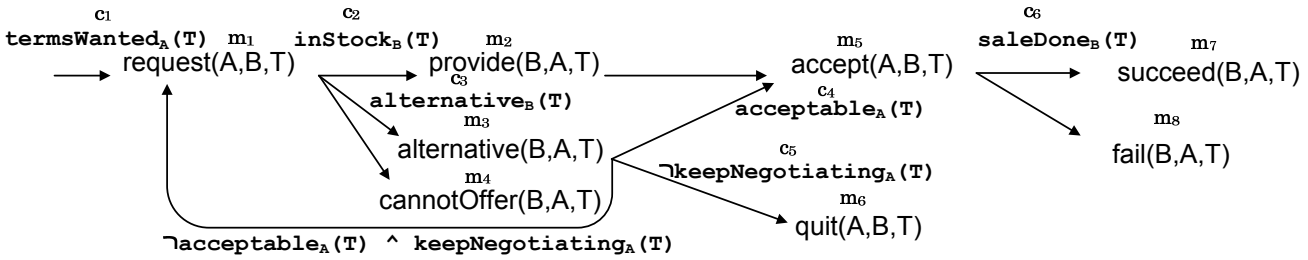

Figure 1: A simple negotiation protocol model: A requests a product with description T (the *terms*) from B. The initial response from B depends on availability: if terms T cannot be satisfied, A and B go through an iterative process of negotiating new terms for the item, depending on the *keepNegotiating*, *acceptable*, and *alternative* predicates (for simplicity, we use a fixed variable T in the diagram, although in the course of a negotiation its value may change). In case of acceptance (which implies payment), B may succeed or fail in delivering the product. Edge constraints are annotated with the variable representing the agent that has to validate them. Additional (redundant) shorthand notation c_i/m_j is introduced. Different out-edges represent XOR if constraints are mutually exclusive.

```
maint = v-high: F (47.0)
maint = high: F (48.0)
maint = med: F (299.0)
maint = low
|   safety = low: F (88.0/2.0)
|   safety = med
|   |   B = S_1: F (17.0/1.0)
|   |   B = S_2: F (15.0)
|   |   B = S_3
|   |   |   price = v-high: S (0.0)
|   |   |   price = high: S (0.0)
|   |   |   price = med: F (3.0)
|   |   |   price = low: S (56.0)
|   |   B = S_4: F (21.0)
|   |   B = S_5: F (13.0)
|   |   B = S_6: F (13.0)
|   |   B = S_7: S (62.0/1.0)
|   |   B = S_8
|   |   |   price = v-high: S (0.0)
|   |   |   price = high: S (0.0)
|   |   |   price = med: F (4.0)
|   |   |   price = low: S (44.0)
```

Figure 2: J48 output for 1000 negotiations. The notation `a =v : S/F` denotes that "if `a` has value `v` the target predicate has value `S/F`". Every leaf includes the number of instances classified in parentheses (the second number appearing to the right of the "/" in some cases is the incorrectly classified instances.

In what follows we shall assume, somewhat informally, that a conversation model has the form of such a tree which can provide path information for (potentially incomplete) sets of variable-value pairs, and denote such tree structures generically as *conversation models CM*. In principle, many other formalisms can be conceived of that achieve the same, such as a relational database, a set of Horn clauses, a Bayesian classifier, etc.

3. REPUTATION SYSTEM

As suggested in the introduction, our reputation system includes an *evaluating agent* a who is trying to assess the reputation of the *target agent* b using a conversation model provided by a *modelling agent* (or *witness*) m, who may, but need not be, the same agent as a.

3.1 Querying the modelling agent

Three modes of reputation calculation are possible in principle: (i) a obtains the entire conversation model from m which has been built by m based on a's definitions of success and failure, and then makes specific queries for specific instances (i.e. lists of variable substitutions) in the model, (ii) a is not granted access to the conversation model, but instead sends only information about its definition of success and failure to m and then m answers particular queries of a regarding specific instances, or (iii) a receives the interaction data from m and builds the conversation model herself.

In our system, we use a method that allows for a uniform treatment of all three cases. This is achieved by splitting the querying process into two steps: *providing path classification*, where a informs m of which paths in the protocol model it considers successful and which are deemed unsuccessful, and m builds its classifier using the methods described in the previous section to build the conversation model; and *instance querying*, where a sends m a specific (though potentially partial) substitution for variables occurring in the model, and m returns a success/failure prediction based on the conversation model previously constructed. With this, whether case (i) or (ii) applies makes no difference from an algorithmic point of view – the same two processing steps are performed regardless of who holds the model. Moreover, since the path classification of a is probably stable over time, whereas instance queries vary (and occur more often), it makes sense to avoid rebuilding the conversation model unless path classification changes, and issue instance queries to the model that only rarely changes (except when m wants to rebuild it based on new data, or is asked for updating it by a). Case (iii) can be basically ignored, as it simply amounts to $a = m$ (in all other cases nothing can be really gained from sending around the entire dataset, methods (i) or (ii) are preferable, at least as long as m is trusted).

Path classification requires that a send m a set of successful paths $\mathcal{S} \subseteq (E \times V)^+$ in protocol model $G = (V, E)$,

and we write $CM(G, \mathcal{S})$ (or simply CM, where G and $\mathcal{S}$ are assumed to be specified) for the conversation model derived by m adding an additional *Outcome* to each path $s \in \mathcal{S}$ with value S (for success) and F (for failure) to all paths $s \notin \mathcal{S}$. The reason we allow for a set of paths to be specified as successful, is that various types of untrustworthy behaviour might occur. In our example protocol, B might claim to provide terms that are not in stock, she might propose alternatives unrelated to the terms proposed by A, might provide terms in the final message unrelated to those accepted by A, or simply offer unacceptable terms such as an excessive price. Even in simpler cases, e.g. when identifying those paths as successful which terminate with a `succeed` message, one may need to specify relatively complex rules that involve entire sets of paths like the following:

if $(\overset{c_1}{\rightarrow} m_1((\overset{\neg c_2 \wedge c_3}{\rightarrow} m_3 \overset{\neg c_4 \wedge c_5}{\rightarrow} m_1)^* \overset{\neg c_2 \wedge c_3}{\rightarrow} m_3 \overset{c_4}{\rightarrow})|$
$(\overset{c_2}{\rightarrow} m_2 \rightarrow))m_5 \overset{c_6}{\rightarrow} m_7)$ **then** $Outcome = S$
else $Outcome = F$

Instances i queried for are lists of attribute-value pairs $i = \{V_1 = g_1, \ldots, V_n = g_n, Outcome = g\}$ for variables V_i occurring in the messages and constraints of protocol model G with ground values g_i from their respective domains in previous interactions, extended by the outcome value $g \in \{S, F\}$ for the queried instance. Querying for i basically amounts to asking "if $V_1, \ldots, V_n$ have values $g_1, \ldots, g_n$, will the outcome of the interaction be g?"

In our example conversation model, an instance query about target agent b concerning a successful outcome in a negotiation after asking for a a car with high safety assessment and low price is:

$$i = \{B = b, safety(T) = high, price(T) = low, Outcome = S\}$$

where we use functions like $safety(T)$, and $price(T)$ to return the respective values of the "terms" variable T. It should be noted that such queries neither need to contain all variables on the paths involved, nor that those paths provided in S need to terminate in leaves. Using CM instead of a simple database of past interaction data provides this flexibility.

3.2 Reputation and reliability

The basic reputation measure used by evaluating agents a in our system is defined as follows:

$$R(CM, i) = \begin{cases} 1 & \text{if } prediction(CM, i) = i.Outcome \\ -1 & \text{else} \end{cases}$$

where $prediction(CM, i)$ returns the classification value (S/F) from the conversation model CM given i. For this, the conversation model CM is used to classify the expected result of the interaction in i and if the predicted class matches the outcomes queried for by i, the prediction 1 (=correct) is returned. Note that, while we have assumed a binary good/bad classification in our formalisation, using a larger number of distinctive labels is straightforward, and even a numerical assessment would be possible using alternative data mining methods (such as a Bayes' Net). Note also that this simple measure already allows a to specify what it views precisely as "trustworthy", and that the same interaction data store can be queried by different evaluating agents easily without a shared notion of reputation. Moreover, G may contain a number of different (independent) protocols, and if different variables or constraints occur across several of these, all past interaction experience will be taken into account when building CM and can be queried simultaneously.

It is straightforward to generalise this measure to return values for a set of target agents $\mathcal{T}$ simply by extending the above function canonically to return a vector of values, taking into account appropriate substitutions:

$$R(CM, V, i, \mathcal{T}) = \langle R(CM, i_{V/b_1}), \ldots, R(CM, i_{V/b_n}) \rangle$$

where $\mathcal{T} = \{b_1, b_2, \ldots, b_n\}$ are the possible target agents and i_{V/b_j} is the extension of the instance query i by the assignment $V = b_j$ and V is the variable in G that refers to the role for which we want to evaluate the reputation of agent b_j. In our example above, if $i = \{maint(T) = low, safety(T) = med, price(T) = low, Outcome = S\}$ and $\mathcal{T} = \{s_1, s_2, s_3\}$, we would obtain $R(CM, V, i, \mathcal{T}) = \langle -1, -1, 1 \rangle$ as a prediction vector for the three agents in the seller's (B's) role. Such a query can be easily used to pick appropriate interaction partners from a set of agents.

To assess the reliability of a prediction provided by the conversation model, we also need to take into account how many past experiences match the query and what proportion of them has been correctly or incorrectly classified according to a rule in the conversation model. Here, it is important to restrict the set of correct/incorrect classifications to those queried by the evaluating agent. For example, assume the queried instance is

$$i = \{B = b, safety(T) = high, price(T) = low, Outcome = S\},$$

and the result of the prediction is S. The result of the reputation query would be $R(CM, i) = 1$, and a possible rule in the tree used for this prediction may have been "if $B = b$ and $safety(T) = high$) then $Outcome = S$". However, the instances that match the antecedent are a superset of those considered by the query, so that the number of correctly classified instances for this rule is an upper bound for those matching the query. To account for this, let $CM(i)$ the set of all rules in CM that match *at least* query i (i.e. they may contain more, but no less attribute-value pairs), and define the reliability of a reputation assessment as

$$r(CM, i) = \begin{cases} \frac{\sum_{\rho \in CM(i)} cci(\rho)}{\sum_{\rho \in CM(i)} ci(\rho)} & \text{if } \sum_{\rho \in CM(i)} ci(\rho) \neq 0 \\ 0 & \text{else} \end{cases}$$

where $ci(\rho)$ are the instances classified by rule ρ, and $cci(\rho)$ returns the number of correctly classified instances by the same rule. In figure 2 these numbers are shown adjacent to the leaves of the tree. This effectively evaluates the confidence of CM in its prediction by calculating the ratio of correctly classified samples that match the query compared to all matching samples in the modelling agent's data set.

3.3 Individual and collective reputation

Next, we can easily combine reputation and reliability to obtain the reputation by *personal experience* and by *group experience* measures used in reputation systems like [8]

$$PE(CM, i) = R(CM, i) \cdot r(CM, i)$$

as the product of reputation and reliability obtained for a simple query. If the instance i does not include an instantiation of the target agent, we can extend this, as before, to sets $\mathcal{T} = \{b_1, \ldots, b_n\}$ of target agents:

$$PE(CM, i, V, \mathcal{T}) = \langle R(CM, i_{V/b_1}) \cdot r(CM, i_{V/b_1}), \ldots, R(CM, i_{V/b_n}) \cdot r(CM, i_{V/b_n}) \rangle$$

Considering $|\mathcal{M}|$ modelling agents $m_1, m_2, \ldots, m_{|\mathcal{M}|}$, each of whom has a respective CM_j at her disposal built using the classification requirements provided by the evaluating agent, reputation by group experience is defined as[1]:

$$GE(\mathcal{M}, i) = \frac{\sum_{1 \leq j \leq |\mathcal{M}|} PE(CM_j, i)}{\sum_{1 \leq j \leq |\mathcal{M}|} r(CM_j, i)}$$

Here the modelling agents are used as witnesses who each provide a personal experience for the target query, and the evaluating agent normalises their individual reports by their respective reliabilities. Again, this can be extended to return a vector of values if the target agent is not specified in i:

$$GE(\mathcal{M}, i, V, \mathcal{T})[k] = \frac{\sum_{1 \leq j \leq |\mathcal{M}|} PE(CM_j, i)[k]}{\sum_{1 \leq j \leq |\mathcal{M}|} r(CM_j, i)}$$

where $1 \leq k \leq |\mathcal{T}|$.

With these, we can now define our main measure of *social reputation* as follows:

$$SR(\mathcal{M}, i) = \xi \cdot PE(CM_a, i) + (1 - \xi) \cdot GE(\mathcal{M}, i)$$

where ξ can be used to weight the impact of personal vs. group experience in the overall judgement. As above, in its vector form covering a set of target agents $\mathcal{T}$, social reputation is defined as

$$\begin{aligned} SR(\mathcal{M}, i, V, \mathcal{T})[j] = & \xi \cdot PE(CM_a, i, V, \mathcal{T})[j] + \\ & (1 - \xi) \cdot GE(\mathcal{M}, i, V, \mathcal{T})[j] \end{aligned}$$

for $1 \leq j \leq |\mathcal{T}|$. Note that ξ is effectively the only parameter introduced in our system that may be specific to a particular implementation. All other elements of the measures introduced above are generic. It should be remarked that as some popular rival approaches [5, 8], we do not include measures in the calculation of SR that take into account how much the witnesses are trusted (in terms of past interactions with them, not assessments of third parties), or the opinion toward a "group" the target agent belongs to. These could be easily defined in our framework, as discussed in section 5. As we show below, we can achieve good predictability without them, by focussing more on the structure and semantics of interactions in analysing past interactions.

4. EVALUATION

To illustrate the usefulness of our approach, we conducted a number of experiments in the simulated car selling domain introduced in section 2. Our scenario contains six preference profiles P_i for customer agents regarding T. These are used to define what cars are considered acceptable by the customers, and are specified as disjunctions of combinations of product properties, e.g.

$$\begin{aligned} P_1(T) = & (\mathit{persons} = \mathit{more} \wedge \mathit{lug\_boot} = \mathit{big} \wedge \mathit{price} = \mathit{low} \wedge \mathit{maint} = \mathit{low}) \vee (\mathit{persons} = \mathit{more} \wedge \mathit{lug\_boot} = \mathit{big} \wedge \mathit{price} = \mathit{med} \wedge \mathit{maint} = \mathit{med}) \vee (\mathit{doors} = \text{5-}\mathit{more} \wedge \mathit{persons} = \mathit{more} \wedge \mathit{price} = \mathit{low} \wedge \mathit{maint} = \mathit{low}) \vee (\mathit{doors} = \text{5-}\mathit{more} \wedge \mathit{persons} = \mathit{more} \wedge \mathit{price} = \mathit{med} \wedge \mathit{maint} = \mathit{med}) \end{aligned}$$

[1]As in the definitions of other measures below, we set this quantity to 0 when the denominator is 0 and omit this case for brevity.

We implement fifty customer agents C_1 to C_{50} with associated profiles $C_i \leftarrow P_{i \bmod 6}$, so that agents C_1 and C_7 use P_1, C_2 and C_8 use P_2, and so on.

Similarly, we specify three seller agent preference profiles Q_j, again specified in terms of T. These describe what types of cars a seller can offer. Additionally, every disjunction is labelled with *tb* or *ub* to indicate in which cases the seller will behave in a trustworthy or untrustworthy way when it negotiates those products. Again, we only show one of these profiles for illustration:

$$\begin{aligned} Q_1(T) = & (\mathit{safety} = \mathit{med} \wedge \mathit{price} = \mathit{low} \wedge \mathit{maint} = \mathit{low}) \rightarrow \mathit{tb} \\ & \vee (\mathit{safety} = \mathit{high} \wedge \mathit{price} = \mathit{low} \wedge \mathit{maint} = \mathit{low}) \rightarrow \mathit{tb} \\ & \vee (\mathit{safety} = \mathit{high} \wedge \mathit{price} = \mathit{med} \wedge \mathit{maint} = \mathit{med}) \rightarrow \mathit{ub} \end{aligned}$$

This profile specifies that the seller will respond positively to a request for terms ($\mathit{safety} = \mathit{med} \wedge \mathit{price} = \mathit{low} \wedge \mathit{maint} = \mathit{low}$), and that she will then also comply with all subsequent steps until the sale is completed. In those cases labelled *ub*, the seller will initially agree to the terms, but will then choose a random "failure" path in her subsequent behaviour. Our system implements 10 sellers S_1 to S_{10}, with associated profiles $S_j \leftarrow Q_{j \bmod 3}$.

4.1 Model construction and reputation measurement

To convert raw sequences of message exchanges to training data samples, we make the following design choices: As far as variables occurring in constraints are concerned, we uniformly record all attributes contained in "terms" descriptions T, including a "?" (unknown) value for those not mentioned in a given execution trace. This is feasible in the given protocol model as the amount of unspecified data is manageable. Our strategy to deal with loops is to only record the last value of every variable occurring in multiple iterations over the `alternative-request` sub-sequence for negotiation, as we are primarily interested in the final offer accepted or rejected by the customer.

The strategy that customer agents follow using our reputation system is explained below:

1. Each customer from $\mathcal{M} \in \{C_1, \ldots, C_{50}\}$ computes
$$SR(\mathcal{M}, i, B, \{S_1, \ldots, S_{10}\})$$
with $\xi = 1/50$ (i.e. equal weight is given to personal experience as to each of the 49 witnesses) and for a query i complying with one of their acceptable preferences in P_i. Each C_i thereby uses her own conversation model, built using only own the agent's own interaction experience.

2. Each customer chooses the seller S_j with the highest positive reputation value and interacts with that agent in the current negotiation. If the prediction of the model does not match the observed interaction experience, the agent re-builds her model from scratch.

3. If there is no such agent, the terms i are updated according to the customer's preferences, and we repeat from 1 (the disjunctive clauses in the P_i profiles are incomplete and can be easily randomly extended to obtain a specific requested car). If no seller with positive reputation can be identified after up to 100 attempts, the customer will interact with a random seller.

It should be observed that we deliberately test our system in a very "heavy" form of its usage, repeating the data mining step over all past interaction data, posing up to 100 queries until a positive prediction is returned, and using each customer agent as an independent modelling agent and witness for every other agent. This approach is chosen to illustrate that even this resource-intensive way of employing our method results in reasonable computation times, as will be shown below. This configuration also allows us to show the workings of our method in the "optimal" case, i.e. when investing a maximum effort of computation.

We compare the prediction accuracy of our system against a number of alternative reputation strategies, measured as the percentage of successful interactions over time:

Random. The seller is chosen randomly – this provides a baseline for the minimum performance that could be achieved without any use of reputation. An optimal strategy is not included, as 100% success constitutes the upper bound of what can be achieved in this scenario (we ensure that there are always sellers in the system who can provide the requested items in a trustworthy way).

Quantitative. The seller is chosen using a distance function based on the number of past successes and failures with them in the customer's personal experience. The function used is $D(s, f) = 1 - (1 + \frac{s}{2f+1})^{-1}$, where s is the number of successes and f the number of failures with a particular seller, and the seller to interact with is chosen with probability corresponding to $D(s, f)$ [9].

Personal experience only. Our reputation system is used as described above but with $\xi = 1$, i.e. the customer only takes her own interaction experiences into account. This method is chosen for comparison to assess the relative importance of witness information as compared to local interaction experience.

Restricted qualitative. Instead of structural and semantic information we use only A, B, and the *Outcome* label (S or F) in combination with the data mining technique. This serves to illustrate the performance of using the same data mining technique without any in-depth information about the content of interactions.

4.2 Static seller behaviour

As figure 3 shows, the results show that after 100 negotiations (2 negotiations per customer) all strategies exhibit similar performance. After 1000 negotiations (20 negotiations per customer) our reputation system greatly outperforms all other strategies, with the social reputation strategy converging much faster to optimal performance than the strategy based on personal experience only. This difference is understandable, as the conversation models combined in the social reputation strategy are based on a much broader variety of data earlier on in the process. However, later convergence of the "personal experience only" strategy also shows that it performs equally well in the long term, provided sufficient data becomes available. The plot also shows that a data mining approach without an analysis of the detailed structure of interactions does not perform any better than the purely quantitative approach, thus proving that the advantage of our method is indeed brought about by the inclusion of qualitative interaction properties rather than by the effectiveness of the data mining algorithm itself.

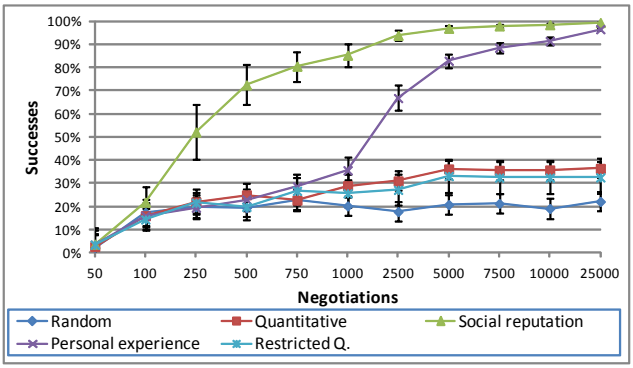

Figure 3: Average number of successful negotiations over number of total negotiations across all customers (100 experiments); error bars show standard deviation

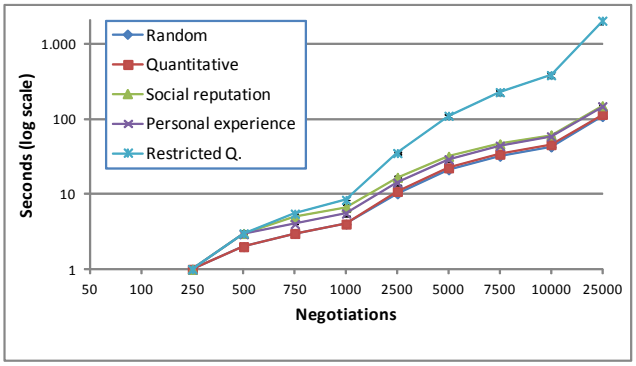

Figure 4: Time per negotiation, log scale (averaged over 100 experiments); the standard deviation across experiments is negligible

The downside of our method is of course increased runtime, at least when, as above, conversation models are rebuilt every time a customer obtains a wrong prediction, which happens very often, while also the datasets over which the models are built increase over time. Figure 4 shows the time taken on average per negotiation, which reaches around 150 seconds after 25000 negotiations for the personal and group experience methods. While this is clearly a shortcoming of our method, it is highly customisable in that the maximum amount of data processed or the frequency with which models are re-built can be adapted as suits the system designer (albeit at the cost of lower accuracy). Also, the runtime per negotiation is still much lower than the over 2000 seconds required by the restricted qualitative approach, which has to rebuild the model very often due to its failures. This also shows that a data mining-based analysis which doesn't take the semantic and structural dimension of communication into account actually combines the worst of both the quantitative (low performance, hence constant need to re-build model) and qualitative (high computational effort to rebuild model) worlds. It would only work well if a given seller behaved well or badly in *every* interaction.

4.3 Dynamic seller behaviour

The ability to respond to dynamic changes in others' behaviours is an important performance characteristic of reputation systems. In our second experiment, we introduce seller agents who suddenly switch their behaviour (from trustworthy to untrustworthy and vice versa) as specified in their original profiles (each rule resulting in *tb* will be modified to *ub* and vice versa). We compare the success rate of the following strategies for responding to dynamic behaviour change against the extreme cases ("no change" in seller behaviour to respond to, and "no strategy" to respond to changes in seller behaviour, i.e. fixed social reputation):

1. *Incongruence detection.* This method is based on erasing all previously collected data samples if a new prediction result is incorrect *and* there is past experience for same instance which provided the correct prediction. The idea behind this is that this should not happen unless evidence shows that the behaviour of the target agent(s) has changed drastically. The method requires that past queries are remembered, and may also lead to removal of many past data samples.

2. *Timestamp weighting.* The second strategy is based on weighting past samples according to their recency during model construction. A weight function $W : \mathbb{N}^2 \rightarrow [0,1] \subset \mathbb{R}$ is employed which uses the current time stamp t and the time t' an instance was observed as a weight $W(t,t')$ for an interaction observed in the past. We use the same weight function as [8], i.e. $W(t,t') = t'/t$ to give more weight to samples closer to t.

3. *Weighted resampling.* Similar to the previous method, this strategy applies a re-sampling step after fixing the weights, i.e. it produces a random subsample of the dataset using sampling with replacement to produce a constant-sized dataset, where the selection probability is proportional to the sample weight [3].

4. *Fixed window.* This strategy simply retains a window with the last 1000 samples for model construction, omitting all previous samples. Another strategy has been added for a window with 500 samples instead of 1000.

The results for the different strategies are shown in figure 5. The plot shows that the incongruence detection strategy achieves the fastest recovery from the intermittent drop in success rate after the seller's behaviour change and manages to return to near-optimal performance very soon. As incongruence recovery strongly relies on an understanding of the structure of qualitative queries, this result illustrates that our reputation system not only manages to exhibit responsiveness (which can aid agents much in adapting to shifting behaviour of malicious agents who try to "massage" them into thinking they are trustworthy) with relatively simple dynamic re-evaluation strategies, but also that the qualitative approach we take is essential to enable such strategies.

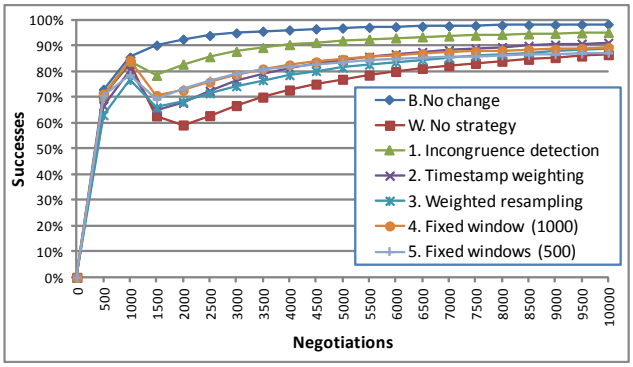

Figure 5: Success probability against number of negotiations, with dynamic seller behaviour change in round 1000 (average over 100 experiments).

5. RELATED WORK

Apart from systems that rely on a purely centralised reputation mechanism such as [1, 2, 13], popular and comparable recent distributed approaches include TRAVOS [11], Referral System [12] and FIRE [5]. All of them use two main information sources to compute reputation values, the *personal experience*, and *witness' experience*. FIRE [5] also uses additional dimensions, i.e. role-based rules and third-party references provided by the target agents. We argue that the major limitation of these approaches is that their definitions of reputation do not depend on the semantics of the domain and the structure of interactions. As a consequence, reputation values are only relevant when all evaluating agents are interested in the same aspect or type of interaction, and the modelling agent(s) calculate reputation based on precisely this information. To use the example of [6], if agents buy something on eBay, trusting a seller agent implies that she will send the right product to the right place at the right time. While some buyers will accept a small delay, they are not able to query the reputation system for such specific details. However, when all providers offer the same product with the same characteristics, as in the trading system used to assess FIRE, these approaches work well.

A notable exception to the lack of semantics in reputation approaches is the REGRET model proposed by Sabater and Sierra [8]. REGRET uses ontologies to detail the type of trust required by the evaluating agent, which can be used to query witnesses. REGRET's contribution is that, since the meaning of trust can be different for each agent, the evaluating agent must be able to ask to what extent witness agents trust the target agent concerning specific aspects of the interactions. Following our eBay example, this means that an agent can query the seller's reputation with regard to getting a low price, quick delivery, etc. Based on these values, the evaluating agent can define its own concept of global trust, e.g. giving less weight to price than to delivery. The main improvement of our approach over REGRET is that reputation is defined as a *model of behaviour* with arbitrarily complex properties, modelled on the basis of the interaction procedures used by the agents in a system. This allows agents to make much more informed decisions based on more fine-grained and flexible queries, makes a priori agreement on a set of specific ontological dimensions of trust across the system unnecessary, and also implies more concise reputation models that are not merely constantly growing databases of past interactions, but store regularities in observed behaviour in succinct data structures.

A limitation that we share with other approaches is that witnesses are assumed to be trustworthy. Although dealing

with untrustworthy witnesses is beyond the scope of this paper, our method provides improved capabilities which could be used to address this issue: When complete reputation models are exchanged between modelling and evaluating agent, the evaluating agent can assess the *long-term* reliability of a model by evaluating its reliability over its own past interaction experiences *prior* to using a prediction provided by this model to make concrete interaction decisions. Contrary to non-qualitative methods, this can be done *without* requiring access to the original interaction data the model was built with. Another possible strategy which illustrates the generality of our approach would be to model interactions with witnesses *themselves* as protocols, and build a trust model for them in much the same way as this is done for target agents.

The obvious weakness of our contribution are its complexity and requirement for additional knowledge. The definition of protocols, application of data mining algorithms, manipulation of conversation models, etc are much more elaborate and less efficient than the application of polynomial-time mathematical operations used in quantitative reputation systems. Possible measures to reduce the number of conversation models created are: (i) the use of data mining techniques which incorporate new experiences without rebuilding the entire model (incremental learning algorithms) [7], and (ii) not creating a new conversation model if this model is not expected to be better than the previous one. With this respect, one way of limiting the amount of computation performed is to rebuilds a conversation model only if a new experience is incorrectly classified by the old conversation model, or if the evaluating agent changes the set of classification rules which determine the classes of the instances before obtaining the conversation model.

6. CONCLUSION

In this paper, we have proposed a novel *qualitative* approach to reputation systems based on mining "deep models" of protocol-based agent interactions. Contrary to most existing methods, the reputation measures we define do not solely rely on the assessment of the predicted *outcome* of an interaction, but take the complex, knowledge- and content-rich structure and semantics of multiagent protocols and agent communication languages into account. On the side of the reputation-evaluating agent, this allows us to introduce more complex, fine-grained, and contextualised queries that can be posed to a reputation-modelling (collection of) witness(es), which results in higher prediction accuracy than quantitative methods as the queries are tailored to the needs of the agent. As a side-effect, our system also allows more intelligent and rationally reasoning agents to exploit the expressiveness our framework affords: As our case study shows, if agents have preferences and objectives specified in a language that can be related to the semantics of a protocol language, the reputation queries can be seamlessly constructed on the basis of their internal beliefs and mental states. On the side of the witness, our method leads to more concise, generalised models of target agents' behaviours, reducing the need to store huge amounts of past interaction data in what would otherwise be a "flat" database of past interactions, allows for disclosure of the model instead of transmission of primary interaction experience (which may also be subject to confidentiality restrictions), and enables different levels of privacy toward a reputation-querying agent without the need to modify the algorithms used to measure reputation. Our empirical results show that our method is capable of exploiting the additional structure and semantics we provide it with, both in terms of achieving higher prediction accuracy (sooner), and in terms of responding to unexpected changes in target agents' behaviours.

In the future, we would like to explore more elaborate data mining techniques, in particular to learn logical theories of the constraint definitions other agents apply from past interaction data, to evaluate our system in larger scenarios with a broader variety of interaction protocols and behaviour types, and to explore issues of trust in witnesses in order to be able to accommodate scenarios where witnesses are not necessarily trustworthy, or might even collude with target agents [2].

7. REFERENCES

[1] Amazon website. http://www.amazon.com.

[2] eBay website. http://www.ebay.com.

[3] R. R. Bouckaert, E. Frank, M. Hall, R. Kirkby, P. Reutemann, A. Seewald, and D. Scuse. *Weka manual (3.7.1)*, June 2009.

[4] A. Frank and A. Asuncion. UCI machine learning repository, car evaluation data set, 2010.

[5] T. Huynh, N. R. Jennings, and N. 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] G. Lu, J. Lu, S. Yao, and J. Yip. A review on computational trust models for multi-agent systems. In *International Conference on Internet Computing*, pp. 325–331. CSREA Press, 2007.

[7] T. M. Mitchell. *Machine Learning*. McGraw-Hill, New York, 1997.

[8] J. Sabater and C. Sierra. Regret: reputation in gregarious societies. *Procs AAMAS'01*, pp. 194–195, 2001.

[9] E. Serrano, A. Quirin, J. A. Botía, and O. Cordón. Debugging complex software systems by means of pathfinder networks. *Information Science*, 180(5):561–583, 2010.

[10] E. Serrano, M. Rovatsos, and J. Botia. Mining qualitative context models from multiagent interactions (extended abstract). *Procs AAMAS'11*, pp. 1215–1216, 2011.

[11] W. T. L. Teacy, J. Patel, N. R. Jennings, and M. Luck. Coping with inaccurate reputation sources: experimental analysis of a probabilistic trust model. *Procs AAMAS'05*, pp. 997–1004, 2005.

[12] B. Yu and M. P. Singh. An evidential model of distributed reputation management. *Procs AAMAS '02*, pp. 294–301, 2002.

[13] G. Zacharia and P. Maes. Trust management through reputation mechanisms. *Applied Artificial Intelligence*, 14(9):881–907, 2000.

[2]Acknowledgments: This research work is supported by the Spanish Ministry of Science and Innovation under the grant AP2007-04080 and in the scope of the Research Projects TSI-020302-2010-129, TIN2011-28335-C02-02 and through the Fundación Séneca within the Program 04552/GERM/06.