Identifying Factors for Managing Trust in Open Multi-Agent Systems

Sadra Abedinzadeh\(^1\), Samira Sadaoui\(^2\)

\(^1\)\(^2\) Department of Computer Science, University of Regina, 3737 Wascana Parkway, Regina, SK, S4S 0A2, Canada
\(\text{sadra}@\text{cs.uregina.ca}; \text{sadaouis}@\text{uregina.ca}\)

Abstract—Trust management is becoming crucial in open systems because they may contain malicious and untrustworthy service providers. Trust management in multi-agent systems (used to model open systems) has gained a huge amount of attention from researchers in recent years. In our previous work, we proposed a generic agent trust management framework, called ScubAA, which is based on the theory of Human Plausible Reasoning. ScubAA first recommends to the trustee (e.g. a user) a personalized ranked list of the most trusted trustees (e.g. service providers), within the context of the trustee's request, and then forwards the request to those trusted trustees only. In this article, we are particularly interested in comparing, from a theoretical perspective, ScubAA with four other trust management systems that we selected from the vast literature on trust management. This comparison highlights significant factors that agent trust management systems utilize in their trust evaluation process. It also shows that ScubAA is able to consider more trust evidences towards a more accurate value of trust. Indeed, ScubAA introduces a single unified framework that considers various important aspects of trust management, such as the trustee's feedback, history of trustee's interactions, context of the trustee's request, third-party references from trustees as well as from trustees, and the structure of the society of trustees.

Keywords—Trust Management; Multi Agent Systems; Human Plausible Reasoning; Trust-based Service Selection; Trust Factors

I. INTRODUCTION

Managing the trust in an open system is becoming crucial because of the growth of the number of autonomous service providers. Diverseservice providers can join and leave the open system at any time, and in addition there is no supervision on their actions [1, 2]. Due to the autonomous nature of the providers, they may release inaccurate information about the services they offer. Open systems are usually developed as Multi Agent Systems (MASs) [1, 2]. A MAS consists of numerous agents as well as an environment in which the agents are situated; each agent acts autonomously in a goal-oriented manner to achieve the functionality of the MAS. Because of the availability of a vast variety of service providers, the consumer needs to be presented with rigorous tools to assist him finding the most trustworthy service agents. The process of selecting a set of the most trusted agents is called Agent Trust Management (ATM) [2-4]. It is a common case when there is more than one agent capable of satisfying a particular sub-goal. As a result, the consumer must decide which best agent(s) the task is to be assigned to. To help the consumer in his decision, ATM methods rank the agents according to a number of measures that are usually domain specific [2, 3, 5, 6].

With the explosion of information on the Internet, users face the problem of overloaded information. Search engines have been proposed to tackle this issue. Nevertheless, they suffer from the fact that users who do not have the same preferences and interests, will obtain same results.Recommender systems are the alternative option to search engines to address the personal preference problem [7, 8]. Trust management approaches, on the other hand, can provide important information to recommender systems regarding the behaviour model of the information providers [9, 10]. In fact, trust-based recommender systems utilize this model to recommend to the user the set of providers that satisfy his information needs in the best way. As discussed in [11], the trustworthiness of agents has to be taken into consideration in recommendation systems. Authors of [12] clearly state that “The goal of a trust-based recommendation system is to generate personalized recommendations by aggregating the opinions of other users in the trust network.” Most of the recommender systems take into account the trust network build in the underlying MAS. However, as shown in the related works section, many do not utilize the structure of the MAS as an evidence towards recommendation. Furthermore, one of the applications of trust-based recommender systems is trust-based service selection in open systems. In other words, trust-based service selection methods try to suggest a set of the most trusted service providers that have been evaluated to be the best in response to the user’s request. The trust evaluation is based on different factors such as functional QoS properties, structural properties, supported APIs, and past behaviour of the trustees[5, 13-15].

In our previous work[16, 17], we propose a generic ATM framework, called ScubAA, that recommends to the user a ranked list of the most trustworthy service providers in an open system, such as the Web. ScubAA introduces a MAS, named Service Agent System, to handle the autonomy of the agents residing in the open system. Moreover, ScubAA divides the entities involved in a trust management problem into two groups: trustees (such as users) and trustees (such as service providers). A trustee evaluates the trust of another set of entities, and a trustee is the entity for which the trust is to be calculated. ScubAA, based on the theory of Human Plausible Reasoning (HPR) [18], assesses a trustee's trust in terms of a
single personalized trust value derived from different sources of evidences, such as trustor’s feedback, history of trustee’s interactions, context of the submitted request, third party references from trustors as well as from trustees, and structure of the society of entities. Third-party references are only accepted from those entities that are associated to the same context. To the best of our knowledge, this is the first time HPR has been applied to the field of trust management. As shown in the related works section, an ATM system that takes into account all these evidences at once has not been proposed. In fact, these evidences have been addressed separately and differently in the literature.

In this paper, we are particularly interested in comparing ScubAA with four other ATM systems that we chose from the very vast literature in trust management, including FIRE [2], AFRAS [19], POYRAZ[20], and the approach introduced in [21] by Nusrat and Vassileva (that we name Nusrat in this document). FIRE (which has been cited 463 times) is a good candidate for this comparison since it is one of the closest systems to ScubAA in terms of the kind of references it employs to assess the trustworthiness of agents. POYRAZ and Nusrat (more recent articles) are both context-based approaches to trust management, and hence may provide a good comparison to ScubAA. Finally, AFRAS (cited 113 times) has been selected because it employs a different point of view, based on fuzzy logic, for addressing the trust management problem. We compare in detail these five trust management systems in terms of their inputs, outputs, aggregation methods and underlying theories. Thanks to this comparison, significant factors that trust management systems utilize in their trust evaluation process are highlighted. We show the benefits of each trust factor and discuss whether or not these systems consider each of them. Nevertheless, since the other four ATM systems (FIRE, AFRAS, POYRAZ, and Nusrat) are not available for testing, consequently, we only perform a theoretical comparison. This comparison shows that ScubAA is able to consider more trust evidences towards a more accurate value of trust by introducing a single framework that takes into account a number of important aspects of trust management.

II. RELATED WORKS

A. Trust Management

Trust management has gained a huge amount of attention from researchers in recent years. To assess the trust for each trustee, trust management methods consider different factors that are related to the behaviour of both trustees and trustors. These methods can be roughly divided into two categories: the “cognitive” and “mathematical” approaches [22]. The cognitive approaches firstly try to build a belief with regard to the other party, and secondly use this belief for evaluating the trust. For instance, [23] introduces such a method to trust management by incorporating the Bayesian network theory. For each agent, this approach creates a belief of how trustworthy the agent is in different situations, and then by applying a Bayesian network model, it combines these believes in order to acquire the final value of trust. [24] proposes a belief theory approach that stores the belief of the two possible outcomes that show whether an agent is trustworthy or not. Next, by subtracting these two beliefs, this method can obtain the reputation score of a given agent.

On the other hand, mathematical methods do not build a belief, yet they utilize a number of measures based on the characteristics of the underlying MAS. After gathering quantitative values for these measures, they calculate the degree of trust. For example, [25] presents a directed agent trust graph in which agents are the nodes; an edge exists between two nodes if there is a trust relation between them. By combining the trust sub-graphs, and then trying to find a trust path in the graph, the trusted agents for a given node can be found. In [26], authors expose an approach for measuring reputation in three phases: checking the feedback acquired from the user, adjusting the feedback by incorporating the feedback similarity, and detecting malicious feedback by using a cumulative sum method. [27] reports a trust management method that employs the Markov model and time series. This method clusters the direct interactions and referenced trust values using local learning techniques based on time series in different time slots. Subsequently, it predicts the future trust values by the help of a Markov matrix. [28] introduces a reputation management scheme for P2P networks where there is no centralized authority. It uses the past behaviour of the peer to predict its future behaviour. The malicious peers are then put away from the network which, in turn, results in the significant reduction of harmful activities in the network. [29] reviews the individuals who create and maintain their trust in others in complex societies. The review is based on comparative biology, behavioural economics and social cognitive neuroscience. It suggests that trust management in the nature has its roots in evolution and is happening every day. [4] examines a number of trust management approaches as well as some mechanisms to exchange trust values. It concludes that although trust is a subjective concept, it may be useful to take into account the trust evaluated by other agents as recommendations. [30] studies trust models from three different points of view: socio-cognitive, numerical and reputational. It also considers the Service Oriented Computing (SOC) as a MAS. Next, it focuses on trust in P2P networks and Grid computing, and examines a number of methods to employ trust and reputation management in SOC. [31] defines a probability density function for the positive experiences. Authors in [31] state that the key intuition of this approach is the assumption that the agent’s trust corresponds to the increasing deviation from the uniform distribution.

B. Trust-Based Recommender Systems

Trust-based recommendation systems have been investigated by many researchers. [10] introduces a recommendation system in the social networks according to the trust between different agents in the network of trust. The trust knowledge is used in this paper to filter the information agents can find in the social network. It also examines if the dynamic nature of trust between agents biases outcome of the system. [32] presents TREPES that employs the trust information of the service
providers in a peer production system to recommend the best information to users. The method is able to evaluate the trustworthiness as well as accuracy of services provided by each peer using fuzzy inference system and fuzzy multi-criteria decision model. [33] also provides a model for using trust in recommender systems (in collaborative filtering specifically). The model approximates the rating for an unvisited product by defining two heuristics that are based on trust. Finally, the model proposes different techniques to constantly providing trust information for the products. [9] supposes that there exists a network of trust between the parties in a recommender system. Each agent provides its feedback in terms of a fuzzy value. By using a temporal ontology, the proposed model records all the changes different parties may make on the ontology. This ontology, in contrast to the traditional database approach, is then used to generate the recommendations. [34] tries to solve the problem of quality assessments of the database in a recommender system. For this purpose, the authors propose a trust-aware recommender system which, in addition to the traditional way of accepting the rating matrix, takes into account the trust network in terms of a matrix depicting the trust relations between agents. Using the trust matrix, the system is able to find similar users, i.e. those that the current user trusts.

C. Trust-Based Service Selection

Several methods of trust-based service selection have been reported in the literature. [35] introduces an agent-based framework to facilitate the process of dynamically selecting Web services by means of an ontology that defines the QoS parameters from both objective (e.g. response time) and subjective (e.g. by concentrating on the user’s interaction) perspectives. [36] exposes a new approach for agent trust management based on the theory of Rough Sets. By collecting the values of the trust attributes from different sources, this approach generates a set of trust rules that is used to extract the most trusted agents, and forwards the user’s request to those agents only. The trust attributes are derived from the properties of the user, the user’s request and the open system. [37] identifies the challenges of calculating the trust and proposes SCOUT as a middleware to collect the evidences, and a client interface that, once implemented, makes it possible for the consumer to submit the evidences to the middleware layer. SCOUT helps a software system to easily compute the degree of trust for its entities. The evaluated degrees, then, are used to select the best service. [38] produces the trust that a consumer has in a service based on both direct and indirect rating for that service. The direct rating is the assessment of the service after the consumer finishes the interaction, and indirect one is provided by third parties. The paper employs the Dempster-Shafer theory of evidence to generate such a trust value. As a result, the model is able to overcome the problem of transitive trust evaluation as well as to remove fake evidences from the assessment process. [39] proposes an intuitionistic fuzzy sets-based approach. Using an improved version of fuzzy indexing method, the authors are able to combine both “concord” and “discord” satisfaction rates in order to assess the satisfaction degree of the service provider. Moreover, a weighting scheme is employed to prioritize different QoS metrics. [21] introduces a method for evaluating trust and reputation of a service provider in a decentralized user modelling system. The proposed method takes into account two metrics: a number of measures with regard to the QoS of service provider, and the degree of trust over each of these measures in the case where they are provided by other agents that show how much the user trusts the referrers values in that context. Finally, by taking into account the defined weights as well as adjusting the referred values according to degree of trust to the provider, the consumer agent is able to generate his degree of trust in a service provider. [40] provides a classification of trust metrics for services (functionalities used by consumers) and service providers (organizations which own services). It considers the direct trust between the trustors and trustees, and do not consider the third party references. In fact, it does not incorporate the trustee’s opinion about the services/service providers by stating that the first-parties should provide the trust information. [41] proposes a new reputation model for Web service selection in a Web service collaboration network. It proposes two reputation metrics to calculate the trust. The first one considers the recommendation selected from the structure of the community, and the second one uses a modified version of the page rank algorithm to assess the reputation of the invoked Web service. [14] presents a method for selecting and ranking services according to their reputation. For this purpose, it utilizes the certainty measure of social networks, in terms of how close two nodes are in the social graph, as the reputation of each service by counting the number of adjacent services to the current service.

III. AN OVERVIEW ON SCUBAA

We consider the case where there are a lot of services available to the user to choose from. It looks impossible for the user to be able to select “the best” services amongst them. ScubAA [16, 17] returns to the user a personalized ranked list of the most trusted service agents within the context of the user's request, and submit the request to those service agents only. We may note that the name ScubAA is composed of two terms: “Scub” standing for “Service Suggestion System” i.e. “S-Cubed”, and “AA” for “Autonomous Agents.

The generic architecture of ScubAA, depicted in Fig. 1, allows users to apply it to any kind of trust management problems. We employ the theory of Human Plausible Reasoning (HPR) [18, 42] to manage the trust for MASs. As mentioned before, ScubAA divides the entities involved in a trust management process into two groups, trustors and trustees. Using HPR, we are capable of not only considering the references from other trustees (service agents), but also from other trustors (users). In other words, if a relation between two trustors ti and tj exists, it would be possible to reflect the trustees whom tj already trusts in the process of evaluating the trusted trustors for ti.
Moreover, by associating each element of the theory with a context, HPR equips ScubAA with the capability of considering the context in all its calculations. The context is beneficial in this process in the sense that it eliminates the third parties that do not belong to the same identified context. As a result, it ensures that the third-party references are only accepted from entities that have some commonality with the trustor or the trustee. In ScubAA, the evaluated degree of trust for each trustee is the aggregation of several factors: the trustee’s current degree of trust, the direct interaction rating value provided from the trustor, any third-party references from entities with whom the trustor or trustee has a relation, the context of the trustor's request, and the structure of the society of entities. ScubAA infers the relations between entities by applying the three HPR transformation functions (generalization, similarity and specialization) to its Knowledge Base (KB) for a given context. The KB contains several hierarchies, which model the global knowledge of the MAS, along with the certainty parameter (i.e. the degree of trust) and the context for each entry in those hierarchies. For every submitted request, ScubAA adjusts and improves the KB by inferring new trust relations between trustors and trustees, and by finding new similarity relations between trustors according to their history of interactions, and between trustees based on the results they return to the trustor.

With the intention of finding the most trusted trustees for a trustor, U, the main contributions of ScubAA are as follows:

- Reasoning to find the related trustees of the trustees that U already trusts, adjusting their corresponding degrees of trust, and adding them to the list of trusted trustees of U. The reasoning is performed by applying the three HPR transformation functions to the KB for a given context.
- Reasoning to find the trusted trustees of the related trustors to U, and adding them to the list of trusted trustees of U along with their adjusted degrees of trust.
- Improving the trust information in the KB by updating the existing degrees of trust.
- Inferring to generate new information that relate the nodes in the hierarchy of trustors to denote similar trustors, and also in the hierarchy of trustees to show similar trustees, and, finally, between the nodes in these two hierarchies to indicate the new trusted agents of trustors.
- Identifying the context according to the properties of both the trustor and the submitted request, and accepting references only from those entities that are associated to that context.
- At the time of reasoning, taking into account the certainty of how well a context has been identified for a given request.
- Providing generic aggregation functions in order to enable the developers customize ScubAA for any ATM applications. For instance in[17], we have employed the mathematical average as well as Dempster-Shafer theory [43] as two individual aggregation functions in our specific implementation of ScubAA.

In [17], we also demonstrate the feasibility of ScubAA by applying it to a real-world experiment of Web search. Indeed, ScubAA assists user by submitting his query to the most trusted search engines. From the Information Retrieval (IR) perspective, this specific application is viewed as a high-precision IR system in which a search engine is trusted more if it returns results with higher precision. ScubAA firstly identifies the context of the submitted query, and then by employing HPR finds the most trusted search engines in that specific context. These engines are sorted according to their degree of trust. The user’s query is forwarded to the most trusted search engines in the next step. After getting the user’s feedback in terms of the
precision of retrieval for each search engine, the values of trust are updated by merging three metrics: the current value of trust, the precision of the retrieved results as the direct interaction rating, and third-party references from related users as well as from related search engines within the context of the query. The related search engines are inferred using the concepts of HPR. In [17], we also employ the statistical analysis tool ANOVA [44] to compare the degrees of trust produced by ScubAA to their real values. The result of this analysis clearly shows that there are no statistically significant differences between those two sets of trust values.

IV. SELECTED ATM SYSTEMS FOR COMPARISON

To perform the comparison, we selected the following ATM systems. FIRE [2] is a good candidate for this comparison since it is one of the closest systems to ScubAA in terms of the kind of references it employs to assess the trustworthiness of agents. POYRAZ [20] and Nusrat [21] are both context-based approaches to trust management, and hence can provide a good comparison to ScubAA. Finally, AFRAS [19] has been chosen because it employs a different point of view, based on fuzzy logic, to address the trust management problem. Below, we summarize these four ATM systems.

A. FIRE

FIRE [2] is an agent trust management model that takes into account four sources of trust evidences in order to obtain a more accurate degree of trust. FIRE employs weighted mean to aggregate all the evidences collected by each agent. It also benefits from the fact that it can handle situations in which some of those sources are missing, consider the cold-start problem [45] for instance. Moreover, there is no centralized trust authority, i.e. each agent individually evaluates the degree of trust in another agent. FIRE incorporates the following types of trust evidences [2]:

- “Interaction Trust”: the value of this evidence is calculated based on the history of direct interactions between two agents.
- “Role-based Trust”: this component makes it possible for FIRE to consider any role-based relation between agents, if such relation exists. For instance, an agent must be configured in such a way that it always trusts the agents it owns.
- “Witness Reputation”: through this module, FIRE is able to consume the references from other third parties with respect to the reputation of the current agent.
- “Certified Reputation”: a trustee can also provide a list of agents that are willing to reveal their reputation.

B. AFRAS

AFRAS [19], a fuzzy reputation agent system, applies fuzzy sets in order to manage trust in a MAS. It mostly emphasizes on the reputation each agent, in an e-commerce scenario, gains by interacting with other agents. AFRAS proposes a conceptual model for agents based on the following three abstract layers:

- “World Model”: it takes care of tasks dealing with the environment and other agents in the society, e.g. answering a question, or purchasing a particular product from another agent. This layer will be activated when a new event has been percept from the environment.
- “Social Model”: this layer deals with the arrangement of those actions that are required to take place so that the World Model can perform a specific task. World Model invokes this layer at the time of responding to an event in the environment.
- “Mental Model”: it keeps track of the inner state of an agent. The other two layers will make use of the mental state of the agent in their decisions. This layer is called from the Social Model when a new change in the mental state of an agent is required, e.g. when the result of an interaction rating is submitted.

As stated in our work, reputation and trust are meaningful within a specific context. However, AFRAS uses the overall reputation of each agent in all contexts as the general witness of its behaviour. Furthermore, AFRAS uses a weighted average scheme in order to aggregate the current value of reputation with the interaction rating provided by the consumer agent. By normalizing this weight, AFRAS achieves a single weight parameter \( W \in [0,1] \) such that the higher value of \( W \) gives more importance to the current value of reputation, and a lower onedotes the new reputation in favour of the most recent interactions.

C. POYRAZ

POYRAZ, a context-based service selection system under deception, is introduced in [20]. In this model, agents store their experiences objectively using an ontology which can be easily interpreted by other agents. POYRAZ also presents an information module that helps to identify deceptive experiences. [20] states that the weak points of using a rating-based methods for trust management is two folds. Firstly, there are no semantics associated with the ratings, and secondly, ratings only depict the criteria and perception of the rater. Consequently, it is possible that a consumer gets misrepresented if he has a
completely different taste than of the rater.

In order to capture the context, POYRAZ defines two separate ontologies. The base-level ontology handles the domain independent concepts, and the second-level one captures the domain specific properties. These two ontologies share the same “description” class as their base class. The description class of the domain-level ontology is used to illustrate service requirements, delivered services, tasks and accomplishments of both consumer and provider agents. POYRAZ also considers third-party references from other consumers. To this end, each consumer collects the related experiences from other consumers in the context of his request. Then, the consumer rates each service agent by assigning a Boolean value to indicate whether or not the interaction was satisfactory based on his own criteria. Finally, POYRAZ employs a machine learning classification technique to calculate the trust in each agent.

D. Nusrat

Nusrat and Vassileva propose a multi-faceted model of trust [21] where agents assess the trustworthiness of other agents based on two factors: trust in the quality of the service, and trust in the references from third parties. The recommendations are supplied by other users, and the degree of trust in those referee agents are aggregated by using a weighting scheme. In this work, user models are employed to keep record of the user preferences and objectives. Nusrat enable the users to define their own preference model by assigning different weights to different properties, including the context. Moreover, agents are able to aggregate the recommendations from other agents in a decentralized fashion.

In the example scenario presented in paper [21], an agent, a, can use recommendation from another agent, b, about the agent of interest, c, by getting the values of judgment of b with regard to different aspects of c. Nevertheless, an ontology would be beneficial to unify these aspects. For instance, two different agents may consider ‘approachability’ and ‘friendliness’ of a doctor as the same aspect. Therefore, the ontology is required here to connect these two concepts.

<table>
<thead>
<tr>
<th>TABLE I  COMPARING SCUBAA, FIRE, AFRAS, POYRAZ AND NUSRAT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Factor</td>
</tr>
<tr>
<td>-----------------</td>
</tr>
<tr>
<td>Context of Trustor's Request</td>
</tr>
<tr>
<td>References</td>
</tr>
<tr>
<td>History of Interactions</td>
</tr>
<tr>
<td>References</td>
</tr>
<tr>
<td>Current Value of Trust</td>
</tr>
<tr>
<td>Interaction Rating Value from Trustees</td>
</tr>
<tr>
<td>Structure of MAS</td>
</tr>
<tr>
<td>Ranked List of Trusted Trustees</td>
</tr>
<tr>
<td>Aggregation Method</td>
</tr>
<tr>
<td>Underlying Theory</td>
</tr>
</tbody>
</table>

V. COMPARING ATM SYSTEMS WITH RESPECT TO TRUST FACTORS

Table I compares five trust management approaches, ScubAA [17], FIRE [2], AFRAS [19], POYRAZ [20] and Nusrat [21]. This comparison highlights some of the most important factors that trust management systems take into account. We explain each factor separately and discuss whether or not these five systems utilize each of them.

A. Context of Trustor's Request

The context plays an important role in trust management problems in the sense that a trustor evaluates a trustee within a specific context. In real-life situations, trust is strictly bound to a context. For instance, you may trust your doctor in the context of medical diagnostics, but you may not trust him in the context of repairing a watch.

However, not many ATM models, as studied in this paper and related works section, consider the context when calculating the values of trust. According to the comparison of Table I, ScubAA, POYRAZ and Nusrat employ this factor. ScubAA uses the context of the request by associating it to all the statements in its knowledge base, and also when applying the HPR transformation functions on those statements. In this way, third parties are only accepted if they belong to the same context. POYRAZ defines a common description class for the domain specific and domain independent ontologies, and uses it as the context. Nusrat not only takes into account the context, but also a faceted trust model in which the agents can define their trust according to different aspects of trust. We may note that FIRE and AFRAS do not take into consideration the context in their trust evaluation methods.
B. Third-Party References from Trustors

If a trustor $t_1$ is related to another trustor $t_2$, and $t_2$ trusts a trustee $t_3$, one can conclude that $t_1$ also trusts $t_3$ with some degree. The definition of the term “related” in the above sentence can be customized in different ATM applications (as done by ScubAA). Third-party references are important as they provide more trust evidences for trustees that are already trusted by a trustor. In addition, they deliver valuable information regarding the trustees for which the trustor does not have any trust information, and as a result, they can extend the trust network of trustors.

ScubAA, FIRE, POYRAZ and Nusrat consume third party references from other trustors. By applying the transformation functions of the HPR theory, ScubAA is able to find related trustors to the trustor of interest. Furthermore, ScubAA determines the related entities (with generalization, specialization and similarity) only if the corresponding relations exist in the knowledge base within the same context that has been identified for the request. FIRE utilizes this kind of references through its “witness reputation” module in which the reputation in a trustee is provided by other agents. POYRAZ, by collecting related experiences from other consumers in the context of the request, and Nusrat, by aggregating recommendations from other agents in a decentralized fashion, are able to make use of the trustors’ third party references. The system AFRAS do not consider references from other trustors.

C. History of Trustor’s Interactions

The history of interactions can provide useful information about the behaviour of entities, trustors specifically, which, in turn, can be used to generate a model of them. It can also be employed to compare the behaviours of two entities to find out how similar they are.

ScubAA and Nusrat are the only ones that utilize the history of trustors’ requests and interactions in order to find the degree of similarity between them in a specific context. ScubAA uses this factor in order to update the certainty parameter of the similarity transformation functions between trustors. Agents in Nusrat model collect and aggregate different aspects of their corresponding users from different sources in a distributed fashion in order to build a model for those users. The user model is then employed to create a preference model to find more personalized service providers. Nevertheless, this factor is not used by the other three ATM approaches.

D. Third-Party References from Trustees

As mentioned before, third-party references are important as they provide more trust evidences for trustees that are already trusted by a trustor, and as a result they can extend the trust network.

ScubAA and FIRE are the only two models that take into consideration the references from other trustees. ScubAA uses the HPR transformation functions (generalization, specification, and similarity) to find the related trustees. More precisely, ScubAA finds the related trustees only in the context of the request. FIRE also introduces the “certified reputation” module which provides such a reference but the context is not considered. AFRAS, POYRAZ and Nusrat do not utilize this kind of references in their models.

E. Current Value of Trust of Trustee

The current value of trust plays an important role in trust management problems because it shows the overall degree of trustworthiness for trustees from the trustors’ point of view. The majority of trust management models utilize this factor in their calculations in order to have an iterative process for updating the degrees of trust. To this end, the ATM methods aggregate this value, as an indicator of past behaviours of the trustees, with several other factors that are derived from the current state of the MAS.

All the systems in this comparison take into account the current value of trust.

F. Interaction Rating Value from Trustors

Interaction rating, which is provided by the trustors, is the common factor that all these ATM systems utilize in their calculations of trust. This completely complies with the fact that the trust is subjective, and, therefore, the opinion of the trustor of interest plays an essential role in the evaluation of trust.

For example, in [17], when we apply ScubAA to the field of Web search, we utilize the precision of retrieval as the interaction rating values. Indeed, for every document returned from the service agents, the trustor judges whether or not it is relevant to the submitted query. We may note that the precision is defined as the ratio of relevant documents to the total number of returned documents.

G. Structure of the MAS

The organization of trustees and trustors in the society of agents can play a significant role in managing trust. For instance, if an agent, a, owns another agent, b, then a may always trust b with some degree. Moreover, there may be some protocols defined for different groups of agents that affect the trustworthiness of those agents.
In our comparison, only ScubAA and FIRE make use of such structure in their calculations. Regarding ScubAA, all the agents (trustees and trustors) along with their relations (through generalization, specialization and similarity), structures and roles in the society are stored in the knowledge base. The theory of HPR makes it possible to define any relation between two entities, between an entity and a group of entities, and between different groups of entities as needed. FIRE introduces the “role-based trust” module which provides the structural protocols between agents. AFRAS, POYRAZ and Nusrat do not consider this factor.

H. Ranked List of Trusted Trustees

In all these systems, each agent in the set of trusted trustees is associated with a degree of trust. Since degree of trust is calculated based on the values of several factors, trust management models usually employ some sort of aggregation functions in order to convert those evidences into a single quantitative value. As a result, trusted trustees are associated with a number between zero and one. Therefore, the list of trusted agents can be sorted based on the degree of trust for each agent.

VI. CONCLUSIONS

In this article, we conducted a theoretical comparison of a number of agent trust management systems, including ScubAA, FIRE, POYRAZ, AFRAS and Nusrat. We compared these five ATM systems in terms of their inputs, outputs, aggregation methods and underlying theories. We examined in total seven important trust factors and showed their benefits for addressing trust management problems. ScubAA is a generic agent trust management framework that takes into account all these factors, such as the context of the submitted request, trustor’s judgement, trustor’s history of interactions, references from both trustees and trustors, and structure of the society of agents.

This research can be improved in a number of ways. Including more ATM systems from the literature in the comparison would be the first step towards a future extension of this paper. Also, due to the lack of access to the ATM research prototypes, an empirical study on a sample application (such as e-commerce or Web search) would improve this research. Trust management field suffers from lacking a standard dataset that may be used to evaluate and compare different ATM approaches. Building a standard trust dataset, and empirically comparing various ATM systems can broaden this work.

REFERENCES

Dr. Sadra Abedinzdeh obtained his Ph.D. in Computer Science from University of Regina, Canada in 2013. He also received his B.Sc. in computer science and his M.Sc. in software engineering from University of Tehran, Iran in 2004, and 2007 respectively.

His main research area is trust management in multi agent systems. His other research interests include human plausible reasoning, theory of rough sets, information retrieval, software architecture, and database systems.

Dr. Samira Sadaoui obtained her MSc and PhD in Computer Science from the University of Nancy I in France. She is currently Professor of Computer Science at the University of Regina in Canada.

Her research interests are in the areas of Software Engineering and Artificial Intelligence, including: Formal Methods; Multi-Agent Systems; Multi-Attribute Auctions; Trust Management; Constraint Programming and User’s Preferences. Her research is supported by the Natural Sciences and Engineering Research Council of Canada (NSERC) federal grant.