A New Model for Trust and Reputation Management with an Ontology Based Approach for Similarity Between Tasks*

Alberto Caballero, Juan A. Botia, and Antonio F. Gomez-Skarmeta

Universidad de Murcia. Campus Espinardo. Murcia, Espana
albe_cu@yahoo.com, {juanbot, skarmeta}@um.es

Abstract. This paper proposes a new trust and reputation model to assist decision making process into agents in P2P environments, taking WSMO as the base for definition of tasks to contract. This work shows the integration of trust and reputation model and WSMO in two ways: 1) how agents use WSMO as ontology to define their requirements, responses, domain-dependent features and metrics; and 2) how the Web services discovery process in WSMO may be improved using trust and reputation criteria given by the model from data stored by consumer agents in previous interactions.

1 Introduction

In P2P environments, the peers interact in a decentralized manner trying to obtain the solution for a given problem. For instance, peers can be providers or consumers of resources [6,13]. Each node or agent may expose very different behaviors, for this reason it is possible that consumers would want to contract only providers with the best behaviors. For that, it is necessary that each node manages his own updated model about the rest of nodes in the system. Trust and reputation based models can help to separate good and bad nodes.

The paper proposes a new model to manage trust and reputation values that an agent has about the rest of agents associated to the realization of a given task. These values are obtained from the agent experience and the information interchanged between agents. The experience of each agent must represent the satisfaction that this agent obtained from others. The way to measure the satisfaction may be very different according to the application domain, and representations for tasks and responses. Many times, the task, that agents negotiate, is a service request; and the response, the description of the service that satisfies it. This way, task and response representations using WSMO [14] may be very useful in order to define the domain-dependent features of the model. The model can be used in environments where agents take WSMO as ontology of reference.

Moreover, the proposed trust and reputation model carries out the Web service discovery process in a more flexible and intelligent way, using the previous

* This paper is supported by the Programme Alban scholarship No.E05D049799CU and also by the Spanish Ministry of Education and Science by the Research Project TIN-2005-08501-C03-02.

K. Fischer et al. (Eds.): MATES 2006, LNAI 4196, pp. 172–183, 2006.

knowledge of the system. Also, it identifies and uses some WSMO elements to obtain several measures such as satisfaction of the task given the response, and similarity between two tasks.

The rest of the paper is organized as follows: section 2 introduces the most important elements treated in previous trust and reputation works that are taken into account in our model; section 3 explains the role of WSMO as conceptual framework of the model. The proposed trust and reputation model is described in section 4. Section 5 shows, using an example, how WSMO elements may be used in order to determine the task satisfaction and similarity between tasks. In the final section, we draw conclusions and ongoing research work.

2 Antecedents on Trust and Reputation

A great amount of trust and reputation models considers trust as an emergent property of direct interactions between agents and assume that agents interact many times [8,14].Trust within an agent is calculated based on this performance in past interactions using expressions that use measurable quantities. REGRET system proposes a trust model based on direct experiences and reputations, providing measures of reliability for these concepts [10].

Reputation may be viewed as an aggregation of opinions of members of the community about one agent. Some authors propose to obtain some ratings from social networks and a procedure to aggregate them to obtain a unique reputation value. This way, it may be consider a community subset, through concepts like *groups* or *neighbors*, to take only closest agents for a specific link [11,10,17].

Our work only considers decision making based on interaction patterns, taking into account that a very simple trust model must be characterized by the following three features: (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; (3) it is possible to refine this value based on new acquired knowledge added to the experience of the agent.

Generally, trust and reputation values are obtained as global values only associated to a peer [3,5,7,13,16], but it is logical to suppose that these values must be associated also to the specification of tasks that agents need to delegate. This way, Griffiths proposes a model to manage trust between agents with respect to a particular task [4]. But, in some cases, it is possible that an agent does not have enough information to produce a trust value for a given task, but he knows instead the previous partner behavior performing similar tasks. It may obtain an approximate trust value for the specified task using available trust information about similar tasks.The way to estimate trust using the information about similar tasks is one contribution of this work (please see section 4.3).

3 Representing Contracting Tasks with WSMO

WSMO offers a conceptual framework for ontologies and Web services descriptions. WSMO consists on four main elements: ontologies (that define a common terminology, used by other elements, providing concepts and relationships between these concepts), Web services (that represent the computational entities providing access to services), goals (that represent the user desires) and mediators (that solve interoperability problems between the rest of elements) [14].

WSMO is a suitable framework to represent the knowledge structures needed by a trust model for P2P environments based on Web services. Service requests may be represented by tasks using the WSMO concept of "Goal". In other hand, the response (describing the Web service that satisfies the task) may be represented using the concept of "Web Service" given by WSMO.

We will use this conceptual framework because 1) it is a new standard proposal, enhancing existing standards in order to describe ontologies, that can be used to represent a broad range of situations where users need to find the most suitable resource in P2P environments based on Web services; 2) it offers great facilities to Web service representation and discovery from different variants according the application requirements; 3) Web services, goals and other elements have some non-functional properties that can be used to calculate trust and quality in Web service discovery process; and 4) the Web service discovery process in WSMO can be improved with the use of trust and reputation models taking into account the system previous experience and behavior exposed.

In line with reason 3, we identify some interesting non-functional properties of Web services and goals to manage trust, quality, costs, etc.:

- Accuracy - numbers of errors generated in a certain time interval.
- Network-Related QoS - network delay, delay variation and/or message loss.
- Performance - throughput, latency, execution time, and transaction time.
- Reliability - number of failures of the Web service in a certain time interval.
- Robustness - number of invalid inputs for which the Web service still function correctly.
- Scalability - number of solved requests in a certain time interval.
- Trust - the trust worthiness of a Web service.

The trust model, presented in the next section, uses service discovery based on simple semantic descriptions of services as a good and simple method to evaluate the quality of a given response. This needs that the agent stores its satisfaction degree for each initiated interaction. Stored information can be used to enhance the WSMO discovery process in later system interactions. This way, the proposed trust and reputation model allows to find the most suitable Web service, in an intelligent form, using knowledge about past experiences.

4 Trust/Reputation Model

The main goal of the proposed model is to offer mechanisms to support adaptive negotiations between agents. It enables mechanisms to decide which are the agents with which it is necessary to negotiate based on the calculation of a value of confidence, that is associated with the specification of the task that it is necessary to contract.

Given the decentralized nature of P2P environments, the model must follow a distributed approach. Each agent has its own bases of experiences to obtain trust values and to interact with its neighbors if it needs to calculate reputation.

When an interaction is finalized, the initiator agent (i.e. the agent that began the interaction) stores interaction data into a binnacle. If a_i is the initiator agent and a_j is the contracted agent to execute the task, the interaction data is represented as a tuple into the set (Initiator's Experience of Trust):

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

where $IET_i^{(t)}$ 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.

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

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

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.

In order to update the bases of experiences, at the end of each interaction t, the agent a_i evaluates the interaction, 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:

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

where w_j is the response given in this interaction by agent a_j; W is the set of all possible responses, and $C^{(t)}(a_i, s_k)$ is the set of the most reliable agents to give solutions to task s_k according to the experience of agent a_i.

Updating process combines the interaction results $I^{(t)}(a_i, s_k)$ and stored experiences in $IET_i^{(t)}$ and $IER_i^{(t)}$ using quality and satisfaction measurements.

4.1 Arranging Agents for Asking Them About Trust and Reputation

At the beginning of each interaction, initiator agent needs to identify the more reliable partners for the required task in order to interacts with them depending on its previous behaviour. For each task s_k, it can create two neighbors lists according to the partner trust degree to give a response for the required task:

- Set of the most reliable agents to give a response $(CT_{sup}^{(t)}(a_i, s_k))$
- Set of agents with a doubtful confidence to give a response $(CT_{dud}^{(t)}(a_i, s_k))$

In the same way, we may stand out the group $CR_{sup}^{(t)}(a_i, s_k)$ for the most reliable agents giving reputation values, according to confidence to give reputation values for the specific task s_k.

Agents from $CT_{sup}^{(t)}(a_i, s_k)$ should be asked from responses for task s_k, because they are the most reliable agents. However, it is possible that some agents from $CT_{dud}^{(t)}(a_i, s_k)$ could come up with valuable information. The system is dynamic and this allow the system to adapt. It would be desirable that these agents were also asked for a response about s_k. We define a set of agents with a doubtful trust to give response, but with a high reputation:

$$C_{prom}^{(t)}(a_i, s_k) = \{a_j |\ a_j \in CT_{dud}^{(t)}(a_i, s_k), R(a_i, a_j, s_k, CR_{sup}^{(t)}(a_i, s_k)) \geq \gamma_{sup}\}$$

where the function $R(a_i, a_j, s_k, CR_{sup}^{(t)}(a_i, s_k))$ represents the reputation value assigned by a_i to agent a_j for task s_k according to experience of the most reliable agent given reputation information, and γ_{sup} is a threshold value.

Hence, we define

$$C^{(t)}(a_i, s_k) = CT_{sup}^{(t)}(a_i, s_k) \cup C_{prom}^{(t)}(a_i, s_k)$$

as the set of requested agents that agent a_i will ask for task s_k. This list is populated by agents with a high trust value to give response and others with doubtful trust value to give response but with a high reputation value according to the most reliable agents given reputation information.

4.2 Obtaining Trust and Reputation

The concept of trust as used in this model not only takes into account the partner in a given negotiation, but the trust value associated to the given task specification. Also, it combines direct trust experience with opinions of high-trusted neighbors. The global value of trust $f_{i,j,k}^{(t)}$ is obtained from the bases of experiences like in REGRET [10], using this function:

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

where $DT(a_i, a_j, s_k, IET_i^{(t)})$ represents the direct trust value that agent a_i assigns to agent a_j for task s_k according to the experience in his own base $IET_i^{(t)}$; $R(a_i, a_j, s_k, CR_{sup}^{(t)}(a_i, s_k))$ 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.

Direct trust $DT(a_i, a_j, s_k, IET_i^{(t)})$ and its reliability $DTRL(a_i, a_j, s_k, IET_i^{(t)})$ are obtained using functions that query the base of experiences $IET_i^{(t)}$. Our model uses a discount approach taking into account that experiences lose relevance as they get older. The way to model the lose of relevance of experience is carried out different than REGRET. If $0 \leq \delta \leq 1$ is a time modulating parameter, that gives higher importance to experiences closer to t, trust can be calculated as follows:

$$DT(a_i, a_j, s_k, IET_i^{(t)}) = \sum_{l_p \in L} \delta^{|L|-p} et_{i,j,k,p}$$

where L is subset of different experiences that agent a_i has about the performance of agent a_j associated to task s_k ($L \subset IET_i^{(t)}$, $|L| \leq t$). Subindex p, in the new set L, indicates how old is the experience $et_{i,j,k,p}$: l_{p_2} is more recent that experience l_{p_1} only if $p_2 > p_1$. The $et_{i,j,k,0}$ represents the oldest experience that agent a_i has about the performance of agent a_j for task s_k.

To model how reliable the direct trust measure is, we follow the models given by SPORAS [18] and REGRET [10]. Reliability value is obtained from the amount and variability of experiences used to calculated the trust:

$$DTRL(a_i, a_j, s_k, IET_i^{(t)}) = N_o(a_i, a_j, s_k, IET_i^{(t)}) \cdot (1 - D_v(a_i, a_j, s_k, IET_i^{(t)}))$$

where

$$N_o(a_i, a_j, s_k, IET_i^{(t)}) = \begin{cases} sin(\frac{\pi \cdot |L|}{2 \cdot itm}) & : \quad |L| \leq itm \\ 1 & : \quad \text{otherwise} \end{cases}$$

and

$$D_v(a_i, a_j, s_k, IET_i^{(t)}) = \sum_{l_p \in L} \delta^{|L|-p}(|et_{i,j,k,p} - DT(a_i, a_j, s_k, IET_i^{(t)})|)$$

where itm is a domain-dependent parameter to control the maximum number of experiences taken into account to improve the reliability on the trust measurement. Values greater than itm do not improve the reliability of the metric.

The deviation of the experiences from direct trust ($D_v(a_i, a_j, s_k, IET_i^{(t)})$) is obtained following the same method to calculate direct trust. Differences between experience value and direct trust loses relevance thorugh time.

In other hand, $R(a_i, a_j, s_k, CR_{sup}^{(t)}(a_i, s_k))$ represents the reputation value that agent a_i assigns to agent a_j according to the experiences of the most reliable agents giving reputation information for task s_k.

Taking into account some trust and reputation models, given by Golbeck and Hedler [3,2], Zacharia [18] and Schillo [11], and considering that the reputation is a task-associated value, we propose a reputation function based on the propagation of reputation information from the most reliable agents:

$$R(a_i, a_j, s_k, CR_{sup}^{(t)}(a_i, s_k)) = \frac{\sum\limits_{a_q \in CR_{sup}^{(t)}(a_i, s_k)} DT(a_q, a_j, s_k, IET_q^{(t)}) \cdot er_{i,q,k}^{(t)}}{\sum\limits_{a_q \in CR_{sup}^{(t)}(a_i, s_k)} er_{i,q,k}^{(t)}}$$

where agent a_i, interested in obtaining reputation information, requests information to the reliable agents a_q to give reputation information for task s_k (grouped in $CR_{sup}^{(t)}(a_i, s_k)$) about trust on a_j.

The trust value to give reputation information $er_{i,q,k}^{(t)}$ is obtained directly from the base of experiences $IER_i^{(t)}$. The agent a_i stores, for each agent a_q, a unique trust value to give reputation information $er_{i,q,k}^{(t)}$, for each task s_k. This value is updated after each interaction is finalized (please, see section 4.4).

4.3 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, it needs 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 may obtain this approximation using a similarity degree between the most similar well-known task and the unknown one. In our model, the similarity between two tasks s_k and s_p is obtained from the comparison of the task attributes. This is a domain-dependent function.

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) or indirect reputation (IR) functions to approximate direct trust or reputation values, respectively:

$$IT(a_i, a_j, s_k, s_p, IET_i^{(t)})) = DT(a_i, a_j, s_p, IET_i^{(t)}) \cdot D(s_k, s_p),$$

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

According to Rodriguez and Egenhofer [9] the similarity can be calculated using elements from the set theory and Tversky's measure [12] as indicator of the semantic similarity between entities described using the same ontologies, in this case, between two tasks described using WSMO.

Tversky [12] defines a similarity measure in terms of a matching process. This measure produces a similarity value that is not only the result of the common, but also the result of the different characteristics between objects.

In our model, similarity between tasks ($D(s_q, s_p)$) is a domain - dependent concept that takes into account the Tversky's measure and set theory [9,12] over WSMO [14]. According to the Tversky's model, the similarity between two concepts a and b can be determined in the following way:

$$D(a, b) = \frac{|A \cap B|}{|A \cap B| + \alpha(a, b)|A \setminus B| + (1 - \alpha(a, b))|B \setminus A|}$$

where $0 < \alpha < 1$; A and B are the set of properties of concepts a and b, respectively.

$D(a, b)$ is not necessarily symmetrical, unless a and b are equal or $\alpha(a, b) = (1 - \alpha(a, b))$, that is to say, $\alpha(a, b) = 0.5$. Rodriguez and Egenhofer [9] define the function α taking into account the depth of compared concepts in the ontology hierarchy. Using the same expression to obtain α, and comparing the same concept of the same ontology (equal depth for each task) we take that $\alpha = 0.5$ (symmetrical similarity measure $D(a, b) = D(b, a)$).

4.4 Updating the Bases of Experiences *IET* and *IER*

At the end of each interaction, the model must update the two bases of experiences with the information generated in that time step.

To update the base of experiences $IET_i^{(t)}$, for each agent a_j, that gives the solution w_j to the task s_k, the model can generate the new experience:

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

where

$$(a_j, w_j) \in I^{(t)}(a_i, s_k)$$

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 4.5.

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 $IET_i^{(t)}$.

The base of experiences for reputation $IER_i^{(t)}$ has an unique value of reputation $er_{i,j,k}^{(t)}$ 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.

When agent a_j was requested by agent a_i about agents from $CT_{dud}^{(t)}(a_i, s_k)$, it may recommend some of them given their high reputation according to its experience. The recommended agents by a_j, to agent a_i for task s_k, can be grouped under the set:

$$M_j^{(t)}(a_i, s_k) = \{a_r | a_r \in CT_{dud}^{(t)}(a_i, s_k), f_{j,r,k}^{(t)} \geq \gamma_{sup}\},$$

where γ_{sup} is a thershold value.

This way, agent a_i must adjust $er_{i,j,k}^{(t)}$ reputation value on the requested agent a_j, taking into account the variation (produced during the interaction) on trust about recommended agents from $M_j^{(t)}(a_i, s_k)$.

For each requested agent $a_j \in CR_{sup}^{(t)}(a_i, s_k)$, for current task s_k, the model analyzes the cases of each agent $a_r \in M_j^{(t)}(a_i, s_k)$, taking into account the trust value that agent a_i had about a_r (denoted by $f_{i,r,k}^{(t)}$) at the beginning of the interaction and the new value (denoted by $f_{i,r,k}^{(t+1)}$) at the end.

The trust value to give reputation information $er_{i,j,k}^{(t)}$ is modified combining the mean of all differences between final and previous trust values for each agent a_r, about agent a_j. The value of the reputation $er_{i,j,k}^{(t+1)}$ will be better than $er_{i,j,k}^{(t)}$ when the trust on recommended agents from $M_j^{(t)}(a_i, s_k)$ is improved during the interaction:

$$er_{i,j,k}^{(t+1)} = sigmod(er_{i,j,k}^{(t)} + \frac{\sum\limits_{a_r \in M_j^{(t)}(a_i, s_k)} f_{i,r,k}^{(t+1)} - f_{i,r,k}^{(t)}}{|M_j^{(t)}(a_i, s_k)|})$$

where

$$sigmod(x) = \frac{1}{1 + e^{-\rho(x-\frac{1}{2})}}$$

The parameter ρ controls the squared-like shape of the function (the higher the ρ value, the faster the function gets to its maximum).

4.5 Satisfaction: Fulfillment and Quality

As we treated in the previous section, our model needs 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 $ec_{i,j,k}$. We understand this function as 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:

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

comparing the promised quality value ($ec_{i,j,k}$) 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 values.

The quality of the solution, $Q(w_j, s_k)$, 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, it is a domain-dependent function.

To obtain the value of satisfaction degree, our model proposes to consider the Web service discovery process in WSMO [15]. In this case, tasks are represented by goals and responses by Web services descriptions, discovery process 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). Section 5 shows an example of the definition of the quality function using WSMO discovery process based on simple semantic descriptions of services.

5 How to Compute Satisfaction and Similarity

It is possible to apply this model into a simple provider - consumer P2P scenario. This way, we try to ilustrate how we can use this model in an scenario where a consumer agent requests tasks and obtains solutions from providers.

Each task request s_k or response w_j (represented by Goal or WebService, respectively) is described by the set of non-functional properties listed in section 3 (i.e. accuracy, performance, reliability, etc.). Also, according to this application domain, we may add two properties: speed, representing the download speed, and quality, representing the quality of downloaded resource.

For each property of Goal or WebService, the model must define a normalization function to make independent the domain of the real world values from model-managed values. For that, the model uses values in the range [0,1] to represent the convenience of the property, independent of the original property domain: 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. For instance, when download speed is very fast, the value of the property "speed" is near to 1, but when the number of errors generated in a certain time interval is high, the value of the attribute "accuracy" is near to 0 (please, see section 3).

In WSMO, the Web Service discovery process using simple semantic descriptions of services is based on set theory and exploits ontologies as formal, machine-processable representation of domain knowledge [14,15].

The set of elements of Goals and WebServices can be analyzed in different ways, given a non-unique semantic interpretation. For instance, using the same set of elements to describe a Goal, we can specify that the user wants to satisfy all properties or only some of them. The same situation occurs with WebServices concept. For this reason, it is necessary to specify the intention (universal or existential) of the description of Goal or WebService, in order to determine the type of coincidence between Goal and WebService in the discovery process. For instance, if the user wants to satisfy all request attributes, the intention of the goal is universal; in other hand, if the purpose is to satisfy only some of them, the intention is existential.

Following the discovery approach based on the simple description of Web services [15], for each goal (s_k) or Web service (w_j), we need to group the good-value attributes in the sets R_g and R_w, respectively.

R_g and R_w 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 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 $w_j.b_i \geq \lambda_i$.

Considering universal intentions for goals and Web services $I_g = I_{s_k} = \forall$ and $I_w = I_{w_j} = \forall$ over the sets R_g and R_w (that contain good-values attributes of s_k and w_j, respectively), we may define the value of satisfaction degree:

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

According to this definition, 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 services w_j. Also, the satisfaction function considers intermediate cases.

To determine the similarity between two tasks s_q and s_p, using the definition of D defined in section 4.3, we consider the sets R_{gq} and R_{gp} of prominent non-functional and domain - dependent properties of tasks s_q and s_p, respectively:

$$D(s_q, s_p) = \frac{|R_{gq} \cap R_{gp}|}{|R_{gq} \cap R_{gp}| + 0.5|R_{gq} \setminus R_{gp}| + 0.5|R_{gp} \setminus R_{gq}|}$$

This way, we have a general method to obtain two needed domain-dependent measures in the proposed trust and reputation model: task satisfaction given a response and similarity between two tasks. It offers a very simple definition based on the set theory and WSMO elements.

When linking trust and reputation model to WSMO, the satisfaction and similarity measures use the concepts of Goals and WebServices in the definition of the tasks (the users' requirements) and the answers (services that satisfy the requirements), respectively. However, it can consider other domain-dependent characteristics like in this example: download speed and download quality. For this reason, the model can be adapted to different application domains where WSMO is the used ontology framework.

6 Conclusions and Future Work

This paper proposes a model to manage trust and reputation taking WSMO as conceptual framework in a P2P environment, where agents should be able to contract the Web service of best behavior. The combination of the trust model with the ontological representation offered by WSMO allows the service discovery process to take advantage of the previous knowledge of the system, taking into account the satisfaction degree of the previous tasks.

It is considered that the trust and the reputation in each agent 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 between this and a well known task.

For the description of the services and their requests, the model suggests the 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.

We intend to implement and prove the trust and reputation model based on WSMO, comparing different configurations, trying to affirm experimentally that the quality of discovery process in WSMO is improved when the trust and reputation model is used. We will identify the parameters that affect the system performance and their high-recommended values.

Now, the model is being adapted to ART [1] trying to prove its operation in front of other models of trust and reputation, evaluating and adjusting its capacities of reactivity and representation of the behavior of other agents. We identify some common characteristics and match some related concepts between this model and ART. However, there are several concepts difficult to match, that require ingenious and hard work. Also, we expect that the initial partition of neighbors, proposed by our model, enhances the agents' profits because it reduces unnecessary message interchanging, taking into account that in ART each opinion request has associated a given cost.

References

1. K. Fullam, T. Klos, G. Muller, J. Sabater, Z. Topol, K. S. Barber, J. S. Rosenschein, and L. Vercouter. The Agent Reputation and Trust (ART) Testbed Architecture. In *Proc. of Trust Workshop at AAMAS*, 2005.
2. 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.
3. 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.
4. N. Griffiths. Task delegation using experience-based multi-dimensional trust. In *Proc. of Trust Workshop at AAMAS*, 2005.
5. S. Marti. *Trust and Reputation in Peer-to-Peer Networks.* PhD thesis, Stanford University, 2005.
6. D. Milojicic, V. Kalogeraki, and R. L. R. Peer-to-peer computing. Tech report: Hpl-2002-57, Hewlett Packard, 2002.
7. M. Montaner, B. Lopez, and J. L. de la Rosa. Developing trust in recommender agents. In *Proc. of the First International Joint Conference on Autonomous Agents and Multi-Agent Systems.* C. Castelfranchi and L. Johnson (eds), 2002.
8. S. Ramchurn, D. Huynh, and N. Jennings. Trust in multi-agent systems. *Knowledge Engineering Review*, 1(19):1–25, 2004.
9. M. A. Rodriguez and M. J. Egenhofer. Determining semantic similarity among entity classes from different ontologies. *IEEE Transactions on knowledge and Data Engineering*, 15(2):442–456, 2003.
10. J. Sabater and C. Sierra. Regret: a reputation model for gregarious societies. In *Proc. of the First International Joint Conference on Autonomous Agents and Multi-Agent Systems*, pages 475–482. C. Castelfranchi and L. Johnson (eds), 2002.
11. M. Schillo, P. Funk, and M. Rovatsos. Using trust for detecting deceptive agents in artificial societies. *Applied Artificial Intelligence, Special Issue on Trust, Deception, and Fraud in Agent Societies*, 14:825–848, 2000.
12. A. Tversky. Features of similarity. *Psychological review*, 84(4):327–352, 1977.
13. Y. Wang and J. Vassileva. Trust and reputation model in peer-to-peer networks. In *Proc. of IEEE Conference on P2P Computing.* Sweden, September 2003.
14. WSMO Team. *Web Service Modeling Ontology (WSMO).* W3C, http://www.w3.org/Submission/WSMO/, 2005.
15. WSMO Team. *WSMO Web Service Discovery. WSML Working Draft.* W3C, http://www.wsmo.org/2004/d5/d5.1/v0.1/, 2005.
16. 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.
17. B. Yu and M. Singh. Distributed reputation management for electronic commerce. *Computational Intelligence*, 18(4):535–549, 2002.
18. G. Zacharia. *Collaborative reputation mechanisms for communities.* PhD thesis, Massachusetts Institute of Technology, 1999.