Trust and Reputation Model based on WSMO

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

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

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

ABSTRACT

This paper shows a trust and reputation model for agents in P2P environments, where each agent can be either consumer or provider of resources. The model tries to help the consumer decision making process taking into account trust and reputation information in each partner. This information is obtained from data stored by consumer agents in previous interactions. This work details the structure and updating process for sets of experiences managed by each agent. Also, it proposes a manner to obtain trust and reputation measures from these. Only one task is negotiated in each interaction. Trust and reputation values are associated to the specific task of the interaction. If the initiator agent does not have any information about a given task, it uses stored information about similar tasks. The trust, that agent assign to other at the end of interaction, indicates the task satisfaction degree according to the response and the fulfillment of the agent own promise. The definition of these measures is based on the set theory, the Tversky's normalization measure, and WSMO as conceptual framework for the definition of domain-dependent elements.

Categories and Subject Descriptors

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

General Terms

Models, Reliability

Keywords

Trust, Reputation, Web Services, WSMO

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

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 interacting without any intermediate node [14, 25]. In this kind of environment, 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. As there is not a central entity to guide agent behaviors and give performance measures, it is necessary that each node manages his own updated model about the rest of nodes in the system. The model must bring out information to separate good nodes and bad ones in order to solve a given problem. Trust and reputation based models can help to separate good and bad nodes.

This paper considers a P2P environment where agents must be capable to contract tasks to agents to give the best solution. For that, it proposes a 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, for the interaction task and any similar.

The experience of each agent must represent the satisfaction that this agent obtained from others. Generally, the way to measure the satisfaction may be very different according to the application domain, and representations for tasks and responses. Many times, the task, that agents negotiate, is a service request; and the response, the description of the service that satisfies it. In this way, task and response representations using WSMO [5, 27] may be very useful in order to define the domain-dependent features of the model.

WSMO is a W3C standard proposal that gives a conceptual framework to describes ontologies and Web services. Also, it incorporates a language and Web services discovery processes according to its specified goals. In this way, the proposed model can be used in environments where a consumer agent takes WSMO as ontology of reference, representing its requirements in the form of a Web service request, and the solution is a Web service description that satisfies it.

In the other hand, the proposed trust and reputation model carries out the Web service discovery process in a more flexible and intelligent way, using the previous knowledge of the system.

The trust and reputation model takes as starting point the main ideas of others models [3, 15, 18, 29, 25, 16, 13, 10]. Besides, it identifies and uses some WSMO elements to obtain several measures such as satisfaction of the task (service request) given the response (service description), and similarity between two tasks (when the trust values for unknown tasks are obtained from trust values for similar tasks).

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

Trust in Multi-agent systems has been defined in different ways (please see [6]). For our purposes, we take into account that: "trust is a belief an agent has that the other part will do what it says it will for a specific task in order to decide who to interact with and how trust is the given response" [3]. In this way, Ramchurn *et al.* [19] see trust models into two levels of complementary approaches:

- Individual-level trust, whereby an agent has some beliefs about the honesty or reciprocative nature of its interaction partners. At this level, models can be divided into three groups: learning and evolution based, reputation based, and socio-cognitive based.

- System-level trust, whereby the actors in the system are forced to be trustworthy by the rules of encounter (i.e. protocols and mechanisms) that regulate the system.

Learning based and evolutionary models aim to endow agents with strategies that can cope with lying and non-reciprocative agents. These models consider trust as an emergent property of direct interactions between agents and assume that agents interact many times. In other words, trust is a social phenomenon based on multiple interactions between two parties [27, 5].

Generally, trust in an agent is calculated based on this performance in past interactions using equations that use measurable quantities [26]. In other models, performance is considered only as two possible values: true or false, good or bad, etc. [17, 23, 22]. Sabater and Sierra [21], through the REGRET system, propose a trust model based on direct experiences and reputations and provide a measure of reliability for trust and reputation. The calculation of these values is not only based on direct interactions, it also uses non-direct behavior.

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 (that shows the relationships between agents taking into account social concepts, roles and communication links) and a procedure to aggregate them to obtain a unique reputation value [22, 30, 31]. From social networks is possible to consider a community subset, through concepts like *groups* or *neighbors*, to take only closest agents for a specific link [21, 30].

A Socio-cognitive model adopts a rather higher level view of trust that takes the knowledge of motivations of other agents for granted and proposes ways to reason about these motivations [19].

Specifically, this paper deals with models based on trust and reputation from individual-level approach. We only consider decision making based on interaction patterns between agents.

Based on the characteristics of individual-level trust, presented above, we can affirm that a very simple trust model must be characterize 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, 15, 18, 29, 25, 16, 13, 10], but it is logical to suppose that these values must be associated also to the specification of tasks that agents need to delegate. In this way, Griffiths proposes a model to manage trust between agents with respect to a particular task [11]. But, in some cases, it is possible that an agent does not have enough information to produce a trust value for a given task. It needs to estimate trust. The way to estimate trust using the information about similar tasks is one contribution of this work.

In this paper we present a model to manage trust into Contract - Net interactions between agents [7]. This model supposes that the agents carry out their interactions according to partner confidence for a certain task, based on the experience about the behavior of the partner for this task. The model offers mechanisms to predict (estimate) the task satisfaction degree for each offered partner responses.

If, in the moment to decide, the agent does not have enough information about the previous partner behavior or it considers that obtained trust is a wrong measure, then it may use the trust that other agents have in his partner (i.e. reputation).

Moreover, it is possible that an initiator agent does not have any information about the previous partner behavior performing the specified 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 model proposes a way to manage similarity between tasks (please see section 4.9).

3. REPRESENTING CONTRACTING TASKS WITH WSMO

WSMO is a W3C standard proposal. It mainly offers a conceptual framework for ontologies and Web services descriptions. WSMO consists on four main elements: ontologies (that define an agreed 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) [27, 4].

How these elements are described consists on a reference to used ontology and mediators, a set of functional features (describing its behavior) and a set of non-functional properties, basically.

WSMO language: WSML

Also, WSMO has a formal language to describe all the defined elements: Web Services Modeling Language (WSML). The main aims of this language are: 1) developing a proper formalization language for semantic Web services; and 2)providing a rule-based language for the semantic Web [4].

WSML gives the needed syntaxis and semantics to describe all the elements in WSMO in order to facilitate the Web services discovery and invocation. WSML has different variants, each of them correspond with different levels of logical expressiveness. It takes Description Logics, First-Order Logic and Logic Programming as baselines of these variants [2]. All WSML variants are specified in terms of a human-readable syntax with keywords, similar to the elements of the WSMO conceptual model. It provides XML, RDF and OWL exchange syntaxes in order to guarantee the interoperability with XML, RDF and OWL-based applications [4]. This language is used in our model to specify contracted tasks and Web service responses.

Ontologies and Semantic Web services need formal languages for their specification in order to enable automated processing [4]. Automated Web service discovery process in WSMO uses WSML as formal language.

Web service discovery in WSMO

In WSMO, Web services are described by the following components:

- A set of non-functional properties,
- The domain ontology and possible mediators which allow reusing other ontologies previously defined,
- Capabilities: which define the Web service at the functional level (i.e. what the Web service is actually doing)
- Interfaces of the service: in order to allow for automatic choreography and orchestration [27, 5] of the Web service.

A central element in WSMO is the concept of goal. Goals represent the objective to reach with the Web service execution. They are describe similarly to Web services, representing these Web services which satisfy agents desires.

To discover the Web services, whose descriptions satisfy the goals, WSMO proposes different variants according to application requirements [28]:

- Keyword-based discovery,
- Discovery based on simple semantic descriptions of services and
- Discovery based on rich semantic descriptions of services.

WSMO is a suitable framework to represent the knowledge structures needed by a trust model for P2P environments based on Web services. Service requests can be represented by tasks using the WSMO concept of "Goal". In the other hand, the response (describing the Web service that satisfies the task) can be represented using the concept of "Web Service" given by 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 manage trust and quality in Web service discovery process; and 4) the Web service discovery process in WSMO can be improved by the use of trust and reputation models taking into account the system previous experience and behavior.

In line with reason 3, we identify some interesting non-functional properties of Web services and goals to manage trust, quality, costs, etc. in order to establish measures of similarity criteria between goals; and measures of goal satisfaction given Web services descriptions:

- 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 incomplete or 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 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. In 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 tries 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 in every moment with the specification of the task that it is necessary to contract.

This negotiation is deployed in dynamic and distributed environments, in the sense that in a concrete moment, an agent may be highly reliable for a little number of agents in the system and not reliable for the rest. This represents a situation in which there are different views of a same agent working with a single task. However, this situation appears in early moments of the execution of the system, since by using reputation values, confidence that agents have on a single agent working on a single task should converge. But notice that trust values on a single agent working on different tasks should not have to be equal since experience of the agent may be focused on a concrete kind of task.

The trust values to offer solutions and reputation information are obtained from information stored in bases of experiences.

Given the characteristics of P2P environments, the model must follow a distributed approach in order to manage these bases of experiences. It mean that 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 for to execute the task, the interaction data must be represented in a time-variable 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. The value $er_{i,j,k}$ must be updated at the end of each interaction.

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 some quality and satisfaction measurements.

The rest of this section details the trust model, explaining how each base of experiences is used to obtain trust and reputation values. Subsection 4.1 shows how the trust model identifies the most reliable agents to give response and reputation values about others. Subsection 4.2 introduces the manner to obtain the confidence on an agent for an specific task from combination of direct trust and reputation values offered by all requested agents. Subsections 4.3 and 4.4 are devoted to explain the way to obtain reputation and direct trust from bases of experiences IER and IET, respectively. The way to combine reputation values offered by all requested agents using direct trust in these agents is shown in subsection 4.5. Subsection 4.6 proposes to combine trust/reputation with similarity between tasks as an approximation of trust/reputation values for unknown tasks. The way to update the bases of experiences is treated in subsection 4.7.

Also, the model gives some functions to evaluate each interaction and similarity between tasks. The subsection 4.8 defines a function to evaluate the contract from a comparison between the agreement and real task satisfaction, given a response. Finally, subsection 4.9 proposes to use Tversky distance as similarity measure between tasks.

4.1 Arranging agents for asking them about trust and reputation

In a community of agents, each agent manages a list of its neighbors in order to interacts with them. The neighbors of agent a_i can be represented in the following way:

$$N_i = \{a_j | a_j \in A, neighbor(a_i, a_j) = true\},$$

where the boolean function *neighbor*() will have a domain-dependent definition.

An initiator agent interacts with a set of his neighbors to identify the more reliable partners for the required task. In this way, for each task s_k, and for different values of parameters γ_{sup} and γ_{inf} ($0 \le \gamma_{inf} < \gamma_{sup} \le 1$) we can create several 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:

  $$CT_{sup}^{(t)}(a_i, s_k) = \{a_j | a_j \in N_i, f_{i,j,k}^{(t)} \ge \gamma_{sup}\},$$

  that contains agents a_j whose confidence to give solutions for task s_k in the given moment t, denoted by $f_{i,j,k}^{(t)}$, overcomes the threshold γ_{sup}. The way to obtain $f_{i,j,k}^{(t)}$ from the base of experiences IET is described in section 4.2.

- Set of agents with a doubtful confidence to give a response:

  $$CT_{dud}^{(t)}(a_i, s_k) = \{a_j | a_j \in N_i, \gamma_{inf} < f_{i,j,k}^{(t)} < \gamma_{sup}\}$$

  that contains agents a_j whose confidence to give solutions for task s_k in the given moment t, denoted by $f_{i,j,k}^{(t)}$, is between γ_{inf} and γ_{sup}.

In the same way, we may stand out the group for the most reliable agents giving reputation values, according to confidence to give reputation values for the specific task:

$$CR_{sup}^{(t)}(a_i, s_k) = \{a_j | a_j \in N_i, er_{i,j,k}^{(t)} \ge \gamma_{sup}\},$$

This set contains agents a_j whose confidence to give reputation values for task s_k in the given moment t, denoted by $er_{i,j,k}^{(t)}$, overcomes the threshold γ_{sup}. The way to obtain $er_{i,j,k}^{(t)}$ from the base of experiences IER is described in section 4.3.

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. It would be desirable that these agents were also asked for a response about s_k.

For that, we define a set of agents with a doubtful trust value to give response, but with a high reputation:

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

where the function $R(a_i, a_j, s_k, CR^{(t)}_{sup}(a_i, s_k))$ 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, grouped in $CR^{(t)}_{sup}(a_i, s_k)$. Section 4.5 shows the manner to combine the reputation information given by agents from $CR^{(t)}_{sup}(a_i, s_k)$.

Also, we define $C^{(t)}(a_i, s_k)$ as the set of agents that agent a_i will ask for task s_k:

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

Summarising, the proposed model populates the list of requested agents, $C^{(t)}(a_i, s_k)$, by agents with a high trust value to give response (grouped in $CT^{(t)}_{sup}(a_i, s_k)$) and others with doubtful trust value to give response but with a high reputation value according to the most reliable agents given reputation information (grouped in $CT^{(t)}_{prom}(a_i, s_k)$).

4.2 Obtaining trust value to find values for $f^{(t)}_{i,j,k}$

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. The trust function, denoted by T, is defined in the range $[0, 1]$. Its algebraic definition is as follows:

$$T : A \times A \times S \to [0, 1].$$

The values $f^{(t)}_{i,j,k}$, used in the definition of sets $CT^{(t)}_{sup}(a_i, s_k)$ and $CT^{(t)}_{dud}(a_i, s_k)$ are referred as the global trust values obtained from the bases of experiences using this function. Hence

$$f^{(t)}_{i,j,k} \equiv T(a_i, a_j, s_k)$$

defined as

$$\begin{aligned} T(a_i, a_j, s_k) = \\ & DTRL(a_i, a_j, s_k, IET^{(t)}_i)\ DT(a_i, a_j, s_k, IET^{(t)}_i) + \\ & (1 - DTRL(a_i, a_j, s_k, IET^{(t)}_i))R(a_i, a_j, s_k, CR^{(t)}_{sup}(a_i, s_k)) \end{aligned}$$

where $DT(a_i, a_j, s_k, IET^{(t)}_i)$ 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 $IET^{(t)}_i$; $R(a_i, a_j, s_k, CR^{(t)}_{sup}(a_i, s_k))$ is the reputation value that agent a_i assigns to agent a_j for task s_k according to the experiences of the most reliable agents to give reputation; $0 \leq DTRL(a_i, a_j, s_k, IET^{(t)}_i) \leq 1$ indicates the relevance of the trust measure, as it is considered in REGRET [21].

Following REGRET, the value of $DTRL(a_i, a_j, s_k, IET^{(t)}_i)$ may be different in each interaction, and it is obtained from the reliability of direct trust measure $DT(a_i, a_j, s_k, IET^{(t)}_i)$. $DTRL(a_i, a_j, s_k, IET^{(t)}_i)$ is calculated from the deviation $D_v(a_i, a_j, s_k, IET^{(t)}_i)$ and multiplicity $N_o(a_i, a_j, s_k, IET^{(t)}_i)$ of the satisfaction values $et_{i,j,k,l}$, stored in the base of experiences $IET^{(t)}_i$. These values are obtained from an introspection in the base of experience $IET^{(t)}_i$, combining, in the convenient way, experiences associated to agents a_i and a_j for task s_k. Section 4.4 details the method used to obtain these measures.

$R(a_i, a_j, s_k, CR^{(t)}_{sup}(a_i, s_k))$ 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. Section 4.5 details the method to obtain this measure.

4.3 Obtaining trust values to produce reputation information $er^{(t)}_{i,j,k}$

The trust value to give reputation information $er^{(t)}_{i,j,k}$, used in the definition of set $CR^{(t)}_{sup}(a_i, s_k)$, is obtained directly from the base of experiences $IER^{(t)}_i$. Each agent a_i stores, for each agent a_j, a unique trust value to give reputation information $er^{(t)}_{i,j,k}$, for each task s_k. This value is updated after each interaction is finalized (please, see section 4.7).

Trust values to give reputation information are used to group agents according their confidence, to define the set of the most confidence neighbors (please, see section 4.1) and in the calculation of a single reputation value for a specific agent for a given task (please, see section 4.5).

4.4 Obtaining direct trust DT

Direct trust value ($DT(a_i, a_j, s_k, IET^{(t)}_i)$) and its reliability ($DTRL(a_i, a_j, s_k, IET^{(t)}_i)$) are obtained using functions that query the base of experiences $IET^{(t)}_i$. The model uses a discount approach taking into account that experiences lose relevance as they get older. 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:

$$\begin{aligned} & DT(a_i, a_j, s_k, IET^{(t)}_i) = \\ & \quad (1-\delta)^{|L|} et_{i,j,k,0} + \sum_{l_p \in L} \delta (1-\delta)^{|L|-p} et_{i,j,k,p} \end{aligned}$$

where L is subset of different experiences that agent a_i has about the performance of agent a_j associated to task s_k ($L \subset IET^{(t)}_i$, $|L| \leq t$). 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 know how reliable the direct trust measure is, we follow the models given by SPORAS [32] and REGRET [21]. Reliability value is obtained from the number of experiences used to calculated the trust and the variability of these rating experiences:

$$\begin{aligned} & DTRL(a_i, a_j, s_k, IET^{(t)}_i) = \\ & \quad N_o(a_i, a_j, s_k, IET^{(t)}_i) \cdot (1 - D_v(a_i, a_j, s_k, IET^{(t)}_i)) \end{aligned}$$

where

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

and

$$\begin{aligned} & D_v(a_i, a_j, s_k, IET^{(t)}_i) = \\ & \quad (1-\delta)^{|L|}(|et_{i,j,k,0} - DT(a_i, a_j, s_k, IET^{(t)}_i)|) + \\ & \quad + \sum_{l_p \in L} \delta(1-\delta)^{|L|-p}(|et_{i,j,k,p} - DT(a_i, a_j, s_k, IET^{(t)}_i)|) \end{aligned}$$

To analyze the multiplicity of the satisfaction values (denoted by $N_o(a_i, a_j, s_k, IET^{(t)}_i)$) the model follows the ideas given by REGRET. In this model, 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 other component of the reliability of the trust measure is $D_v(a_i, a_j, s_k, IET_i^{(t)})$, as the deviation of the experiences from the estimated direct trust. For that reason, we propose to obtain this value following the same method to calculate direct trust, taking into account that differences between experience value and direct trust loses relevance thorugh time.

4.5 Obtaining reputation R

Taking into account some trust and reputation models, given by Golbeck and Hedler [10, 9], Zacharia [32] and Schillo [22], 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.

The model defines the reputation function as follows:

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

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.

To obtain a global reputation value, agent a_i combines the information given by requested agents a_q with the trust to give reputation information that agents a_i assigns to requested a_q, for the task s_k ($er_{i,q,k}^{(t)}$). Our model supposes that each requested a_q informs its direct trust on the agent a_j.

4.6 Obtaining trust and reputation from similar tasks

Reputation, direct trust and its reliability measures, used to compute trust value $f_{i,j.k}^{(t)}$, are obtained from information stored in the bases of experiences $IET_i^{(t)}$ and $IER_i^{(t)}$ for task s_k involved in the interaction t (please, see previous sections).

However, it is possible that an agent does not have information from 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 similarity degree between the well-known task and the unknown one. For that, the model needs to incorporate a function to obtain similarity degree between two tasks, let be denoted with D and with algebraic definition as follows:

$$D : S \times S \to [0, 1]$$

The value of this function depends on the definition of the tasks and the distance metric used.

Using this function we can define s_p as the most similar well-known task to unknown s_k ($s_p \neq s_k$), such that:

$$\nexists s_r | D(s_k, s_r) \geq D(s_k, s_p),\ \forall s_r \in S, s_r \neq s_k, s_r \neq s_p.$$

In this way, we define indirect trust and reputation functions to approximate direct trust and reputation values when the base of experience of agent does not have information about task:

$$IT : A \times A \times S \times S \to [0, 1]$$

to approximate direct trust, and

$$IR : A \times A \times S \times S \to [0, 1]$$

to approximate reputation.

Analytical expressions for these functions may be given from the multiplication of trust/reputation in similar task by the similarity between tasks:

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

and

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

In other words, when an agent does not have trust information about a task (s_k) in its base of experiences, the model may combine the trust/reputation in a similar task (s_p) with the similarity degree between the two tasks $D(s_k, s_p)$ to approximate trust/reputation value.

4.7 Updating the bases of experiences IER and IET

At the end of each interaction, the model must update the two bases of experiences with the information generated in this time step. We give a proposal for each base below.

The base of experiences for reputation $IER_i^{(t)}$ has a unique value of reputation $er_{i,j,k}^{(t)}$ to indicate, according to the 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:

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

This way, agent a_i must to 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. For that we use the function:

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

with

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

where ρ is a modulating parameter to manage the movement of the reputation value between 0 an 1.

The value of the reputation $er^{(t+1)}_{i,j,k}$ will be better than $er^{(t)}_{i,j,k}$ when the trust on recommended agents from $M^{(t)}_j(a_i, s_k)$ is improved during the interaction.

In other hand, in the updating process of the base of experiences $IET^{(t)}_i$, for each agent a_j, that gives the solution w_j to the task s_k, the model can generate the 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, 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 4.8.

Here, the model uses the most general consideration taking into account that is possible to add the new experience $edt_{i,j,k}$, obtained from the current interaction, without having to analyze how many experiences are in the base $IET^{(t)}_i$:

$$IET^{(t+1)}_i(s_k) = IET^{(t)}_i(s_k) \cup \{edt_{i,j,k}\}$$

This alternative is feasible only when the base of experiences for trust can grow without limits.

4.8 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 ($ec_{i,j,k}$). Basically, the value of this function results 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:

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

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

The real quality of the solution, denoted by $Q(w_j, s_k)$, indicates how much the response w_j satisfies the requirements specified in the task s_k. Calculation of this value is based on the comparison of both concepts, 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 [28]. 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).

The model uses WSMO discovery based on simple semantic descriptions of services, taking into account the matching between the most important attributes of these concepts. Section 5 shows an example of the definition of the quality function using discovery process given by WSMO.

4.9 Similarity between tasks (D)

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.

According to Rodriguez and Egenhofer [20] the similarity can be calculated using the set theory and Tversky's measure [24] as indicator of the semantic similarity between entities described using the same ontologies, in this case, between two tasks described using WSMO.

Tversky [24] 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, which is in agreement to an information-theoretic definition of similarity [12].

The model proposes the definition of these two functions (quality and similarity) using a simple method based on the representation of domain concepts by means of WSMO. Tasks (represented by Goals) and responses (represented by WebServices) are described using properties defined in section 3. However, it may incorporate other properties according to the application domain, given the simplicity of these two domain-dependent functions.

The ways to obtain satisfaction and similarity measures, using recommended and domain-dependent properties, are shown in the example treated in next section.

5. AN EXAMPLE

To show how to obtain task satisfaction and similarity between tasks, we are considering a very simple P2P scenario, where each agent may be a resources provider and consumer.

We use WSMO to describe concepts. Resources consumer agents specify their tasks requests, s_k, using the concept of Goal given by WSMO. In the same way, the providers use WebService concept to represent the responses w_j.

Each task request s_k or response w_j is described by the set of non-functional properties listed in section 3 (i.e. accuracy, network-related QoS, performance, reliability, robustness, scalability, trust). 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.

Both non-functional properties defined by WSMO as the others incorporated in this example, are transformed using normalization functions. The model manages values in the range [0,1], where a value near to 0 indicates a non-desired value in the original property, and values near to 1 indicate high-desired values in the original properties. 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 (this is a WSMO property defined in section 3).

As general model (described in section 4) shows, and considering the particulars conditions of this scenario, the task satisfaction given a response (Q) is obtained using the WSMO Web Service discovery process [28].

In WSMO, the discovery using simple semantic descriptions of services is based on set theory and exploits ontologies as formal, machine-processable representation of domain knowledge [27].

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 [28], 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:

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

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

Similarity between tasks ($D(s_q, s_p)$) is other domain-dependent concept. As section 4.9 explains, the process to determine the similarity between two tasks takes into account the Tversky's measure and set theory [20, 24] over WSMO [27].

Considering that both tasks are described using the same concept of the same ontology, we only take into account the set of properties used to describe a Goal (non-functional and domain-dependent properties).

Generally speaking, the value of $D(s_q, s_p)$ represents the similarity degree of s_q with respect to s_p: our model uses the stored knowledge about s_p instead of s_q when it does not have information about s_q. To obtain this similarity degree, it uses a similarity measure based on the Tversky's normalization model and in functions of set theory. 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 $0 < \alpha < 1$, and 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 [20] 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 $D(a, b) = D(b, a)$).

In our model, to determine the similarity between two tasks s_q and s_p, we consider the sets R_{gq} and R_{gp} of prominent properties of tasks s_q and s_p, respectively:

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

In 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 service (or 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 degree of 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 [8, 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 (i.e. tasks may be considered as appraisal request; trust on agents to do a given task may be a reputation concept managed by agents in ART; etc.). However, there are several concepts difficult to match, that require ingenious and hard work (i.e. similarity between task is very difficult to implement in ART because eras, as a unique grouping criteria for paintings, has a very simple representation to write a similarity function). 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.

Furthermore, to test our model in ART, it would be necessary to incorporate mechanisms for the treatment of lies and the representation of the bad intentions of the partners.

7. REFERENCES

[1] ART Testbed Team. *Agent Reputation and Trust Testbed.* http://www.lips.utexas.edu/ kfullam/competition/, 2005.

[2] F. Baader, D. Calvanese, D. McGuinness, D. Nardi, and P. Patel-Schneider, editors. *Description Logic Handbook.* Cambridge University Press, 2002.

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

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

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

[6] R. Falcone, M. P. Singh, and Y. Tan. *Trust in Cyber-societies, Integrating the Human and Artificial Perspectives*, volume 2246 of *Lecture Notes in Computer Science.* Springer, 2001.

[7] Foundation for Intelligent Physical Agents. *FIPA Contract Net Interaction Protocol Specification SC00029.* http://www.fipa.org/, 2002.

[8] 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.

[9] 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.

[10] 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.

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

[12] D. Lin. An information-theoretic definition of similarity. In *Proc. of International Conference on Machine Learning ICML'98*, pages 296–304. Madison, WI, 1998.

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

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

[15] L. Molm, N. Takahashi, and G. Peterson. Risk and trust in social exchange: An experimental test of a classical proposition. *American Journal of Sociology*, 105(5):1396–1427, 2000.

[16] 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.

[17] L. Mui, M. Mohtashemi, and A. Halberstadt. A computational model of trust and reputation for e-business. In *Proc. of 35th Hawaii International Conference on System Science.* IEEE Computer society, 2002.

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

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

[20] M. A. Rodriguez and M. J. Egenhofer. Determining semantic similarity among entity classes from different ontologies. *IEEE Transactions on Knoledge and Data Engineering*, 15(2):442–456, 2003.

[21] 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.

[22] M. Schillo, P. Funk, and M. Rovatsos. Using trust for detecting deceitful agents in artificial societies. *Applied Artificial Intelligence, Special Issue on Trust, Deception, and Fraud in Agent Societies*, 14(8):825–848, 2000.

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

[24] A. Tversky. Features of similarity. *Psychological review*, 84(4):327–352, 1977.

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

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

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

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

[29] 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.

[30] B. Yu and M. Singh. Distributed reputation management for electronic commerce. *Computational Intelligence*, 18(4):535–549, 2002.

[31] B. Yu and M. Singh. Searching social networks. In *Proc. of the 2nd International Joint Conference on Autonomous Agents and Multi-Agent Systems*, pages 65–72. J. S. Rosenschein, T. Sandholm, M. Wooldridge, and M. Yokoo (eds.), 2003.

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