Supplier performance in a Digital Ecosystem

Angela Fabregues[1], Jordi Madrenas-Ciurana[1], Carles Sierra[1] and John Debenham[2]

[1]IIIA: Institut d'Investigacio en Intel.ligencia Artificial, CSIC: Spanish Scientific Research Council
UAB, 08193 Bellaterra, Catalonia, Spain, e-mail: (fabregues, jmadrenas, sierra)@iiia.csic.es
[2]Faculty of Information Technology, University of Technology
Sydney, NSW, Australia, e-mail: debenham@it.uts.edu.au

Abstract— **Autonomous entities in a digital ecosystem are expected to interact, negotiate and make agreements. The work introduced in this paper focuses on how these autonomous entities evaluate and rank the other participants in the digital ecosystem using past experiences. These past experiences allow agents to learn a probabilistic model of the behaviour of the others. We introduce a trust measure based on this probabilistic model of behaviour and a software tool that helps in the decision making process of how to choose the right partner for our next agreemeent.**

I. Introduction

One of the key issues in any model of self-organisation, as required by the Digital Ecosystem paradigm, is to have an adequate model of the members of our own species and of the other species. In the context of Digital Business Ecosystems, where the participants are autonomous entities (humans, companies, banks, brokers, ...) capable of making commitments to act, there is a need to form different forms of organisations in order to respond to changes in the environment. For instance, we might need to build coalitions of small companies producing the same products in order to participate in a large contract and compete with large producers. We might need to build teams that would organise companies and professionals with different profiles in order to solve a complex task for which no pre-existent solution has yet been found. All these processes require the adaptation of the agents incarnating the different principals in a digital ecosystem. Agents need to learn from experience who is reliable, who is capable of reacting quickly, who is expected to produce at the best quality. Experience and information exchanges are the basic building blocks of our model of the participants in a digital ecosystem.

In this context, we understand digital ecosystems as underpinned by the basic notion of *agreeement*. Agents (i.e. companies, humans or software agents) in a digital ecosystem are expected to interact, negotiate and sign agreements. Agreements to form coalitions, agreements to perform tasks in a team solving a complex problem, or agreements to structure supply chains that would efficiently produce certain goods. These agreements can then be honoured up to a certain extent. A member of a coalition may not contribute in the expected way, a team member may not show the committed capabilities in solving its part of the problem, or a node in a supply chain may deliver its committed goods late. The work introduced in this paper focuses on how to measure the execution of such agreements. In particular, we propose a model for trust based on a shared ontology and that takes into account time and conceptual similarity and that is based on a solid probabilistic modelling of agent behaviour.

Furthermore, we show how the model can be used in the particular case of eSourcing and give details of a software tool called SRM (*Supplier Relationship Management*) that gives support to a member of a digital ecosystem in the task of deciding which supplier to choose for a given supply order. It is now a prototype capable of working in a standalone mode but that will also be integrated in the iQuotes Suite, an enterprise resource planning system of the company iSOCO.

The paper is structured as follows. In Section II we describe the trust model. In Section III we show the use of the model in the context of eSourcing and in Section IV we discuss the contributions of the paper.

II. Trust Model

A lot of work can be found in the multiagent system literature that deals with trust, [1]. The models and techniques developed can be easily adapted to the context of digital business ecosystems. In this section we introduce such a trust model as an extension of [2] with a richer notion of similarity. This model allows an agent to evaluate the trust it has on another agent when the latter commits to a course of action. Trust values are calculated over a probabilistic representation of the expected performance of agents over signed agreements.

When an agreement is signed, agents involved get committed to a plan of action. How an agent performs a commitment directly affects the trust other agents will have on its behaviour in the future. Information about how an agreement is executed is called *experience*.

Definition 1: We define an experience as a tuple:

$$\mu = \langle \alpha, \beta, \varphi, \varphi', t \rangle$$

meaning that α signed an agreement with β at time t in which β committed to do φ but what α finally observed was φ' happening.

What is committed and observed is expressed on an ontology that agents engaged in a dialogue are assumed to share. The relation between commitment and observation is then represented as a conditional probability that is built along time by factoring in a number of experiences. To take maximum profit of experiences we assume that agents will have similar behaviours in similar semantic regions of the ontology. Thus, similarity measures are crucial to build the conditional probability distribution from a minimum set of experiences as we will see next.

A. Ontology

Agents involved in an agreement negotiation are required to have a shared ontology O. This ontology allows to specify what agents are arguing about and which is the content of the agreements being signed.

Definition 2: We define an ontology O as the tuple

$$O = \langle C, \sqsubset, : \rangle \tag{1}$$

where:

- C is a finite set of concepts.
- $: \subseteq C \times C$, is a *refinement* relationship usually referred to as *is-a*.
- $\sqsubset \subseteq C \times C$, is a *combination* relationship usually referred to as *part-of*.

Satisfying the following properties:

1) *transitivity:* if $c : c'$ and $c' : c''$ then $c : c''$
2) $\sqsubset$ *inherence:* if $c : c'$ and $c'' \sqsubset c'$ then $c'' \sqsubset c$
3) $:$ *inherence:* if $c' \sqsubset c$ and $c'' : c'$ then $c'' \sqsubset c$

The refinement relation defines increased specificity between concepts. It can also be seen as a relation between values and types. On the other hand, the combination relationship defines components of a concept. We represent the combination relation as a term where the predicate is the composed concept and the arguments the parts of it. For example: $wheels \sqsubset car$ and $engine \sqsubset car$ is represented as $car(wheels, engine)$. Even, we can represent it as $car()$ if we don't want to specify its components. The term representing a concept that has no components is the name of the concept itself, e.g. $nail$.

Definition 3: The set of terms associated to the concepts of an ontology $O = \langle C, \sqsubset, : \rangle$, noted $Term(c)$ for $c \in C$, is defined as:

1) if $c' : c$ then $c' \in Term(c)$
2) if $\exists c'.\ c' \sqsubset c$ then $c() \in Term(c)$ else $c \in Term(c)$
3) if $c_1 \sqsubset c, ..., c_n \sqsubset c$ then $c(c'_1, ..., c'_n) \in Term(c)$ where $c'_i \in Term(c_i)$

We abuse notation to define the terms of a set R as $Term(R) = \cup_{c \in R} Term(c)$. In particular, the terms of an ontology $O = \langle C, \sqsubset, : \rangle$ can be defined as $Term(C)$. In the rest of the paper we will refer $Term(C)$ as $Term(O)$.

B. Similarity measures

Similarity measures are crucial for the trust model. The idea is that a concrete experience about a commitment can be used to update the expectation of behaviour over semantically close commitments. To that end we have to be able to compare commitments that is concepts and terms of the shared ontology.

When comparing two terms whose associated concepts have components, we take into account the similarity between those components and also their relevance. For example, one may think that the $engine$ of a car is more important than its $colour$. So two cars with the same $engine$ would be more similar than two cars painted with the same $colour$. To measure the similarity between the components of two concepts we define the function at equation 2 where the components are compared and the result is aggregated weighted by the relevance each component has.

$$sim_{\sqsubset}(P_1(L_1), P_2(L_2)) = 1/2 \cdot \left(\frac{\sum_{t \in L_1} rel(h(t), P_1) \cdot \max_{t' \in L_2} sim(t, t')}{\sum_{t \in L_1} rel(h(t), P_1)} + \frac{\sum_{t' \in L_2} rel(h(t'), P_2) \cdot \max_{t \in L_1} sim(t', t)}{\sum_{t' \in L_2} rel(h(t'), P_2)} \right) \tag{2}$$

where:

- $P_1(L_1)$ and $P_2(L_2)$ are two terms which heads are P_1 and P_2 respectively and L_1 and L_2 are the lists of terms that represent their components.
- $sim(t, t')$ is an overall similarity function between terms that will be introduced later.
- $rel(c, c')$ is a function that represents how relevant is the component c' for the concept c when $c' \sqsubset c$.
- $h : T \to C$ gets the head of a term, eg. $h(P(L)) = P$

Focussing now on the *refinement* relation, it seams reasonable to let the designer define a specific similarity function between concepts c_1, c_2 that are direct specifications of another concept c. That is to say: a partial similarity function $sim_c : Term(O) \times Term(O) \mapsto [0, 1]$ that is defined for concepts c_1, c_2 when there exists c. $c_1 : c$ and $c_2 : c$ but there is no c' such that $c_1 : c' : c$ or $c_2 : c' : c$. For instance, if $high : level$, $medium : level$ and $low : level$ the designer can define a function sim_{level} that verifies $sim_{level}(high, medium) > sim_{level}(high, low)$. By default, when the designer does not define a specific similarity function, then we use the similarity described in equation 3, [3],

$$sim_{default}(c_1, c_2) = e^{-k_1 l} \cdot \frac{e^{k_2 h} - e^{-k_2 h}}{e^{k_2 h} + e^{-k_2 h}} \tag{3}$$

where:

- l is the length, number of hops, of the shortest path between the concepts following the *refinement* relation.
- h is the depth of the deepest concept subsuming both concepts following the *refinement* relation.
- k_1, k_2 balance the contribution of shortest path length and depth respectively[1].

Finally we put together all these similarity functions and create an overall measure for similarity between terms:

[1][3] argues that $k_1 \simeq 0.2$ and $k_2 \simeq 0.6$ represent a good model of human intuitions about similarity.

$$sim(t_1, t_2) =$$
$$= \begin{cases} 1 & \widehat{t_1} = \widehat{t_2} \\ sim(P,Q) \cdot sim_{\subset}(P(L_1), Q(L_2)) & \widehat{t_1} = P(L_1), \widehat{t_2} = Q(L_2) \\ sim_{:}(h(t_1), h(t_2)) & \exists c.\ h(t_1) : c,\ h(t_2) : c \\ 0 & otherwise \end{cases} \tag{4}$$

where:

- $sim_{:}(c_1, c_2)$ would be either a specific concept similarity $sim_c(c_1, c_2)$ or, otherwise, the default concept similarity $sim_{default}(c_1, c_2)$.
- $h : T \rightarrow C$ gets the head of a term, eg. $h(P(L)) = P$
- $\widehat{t}$ represents the expansion of the term t.

As introduced before, a concept with components can be expressed as a term that does not include all its components. It is a simplification on notation suitable when some of the components are the most general ones[2]. In that case, these general components can be omitted. The expansion operator completes the term description including all its components.

$$\widehat{t} = \begin{cases} t & \nexists c' \subset h(t) \\ & t = P(L). \\ P(\widehat{t'_1}, ..., \widehat{t'_j}, ..., \widehat{t'_n}) & t'_j = \begin{cases} t_i & \exists t_i \in L.\ t_i \in Term(c_j) \\ c_j & \text{otherwise} \end{cases} \\ & , \forall c_j \subset P.\ \nexists c_k \subset P.\ c_j : c_k \end{cases} \tag{5}$$

where L represents the list of arguments of the term which head is P.

C. Expected observations

The trust model being described models the expected behaviour of an agent involved in an agreement. The expected behaviour is represented as a conditional probability distribution function (PDF) over the possible observations given the possible agreements. That is $\mathbb{P}(\varphi'|\varphi)$ represents the probability of φ' happening when a commitment φ is made. This PDF is built using past relevant experiences; understanding as relevant all those experiences that have a commitment that is similar to φ over a threshold.

The PDF is initialized using background knowledge on the other agents in the digital ecosystem and updated for each new experience $\mu = \langle \alpha, \beta, \psi, \psi', t \rangle$ using minimum relative entropy inference[3] from the a priori PDF and a set of constraints Q. There are constraints for each term φ that is similar enough to those commitments appearing in previous experiences as described in equation 6.

$$Q(\varphi' \mid \varphi) = \mathbb{P}^t(\varphi'|\varphi) + 1/n_{ex} \cdot S^{inertia} \cdot (1 - \mathbb{P}^t(\varphi'|\varphi)) \tag{6}$$

where:

- n_{ex} limits the maximum influence of a single experience.
- $inertia$ amplifies the impact of an experience.
- S is a similarity measure between the past experience $\mu = \langle \alpha, \beta, \phi, \phi', t \rangle$ and the hipotetical future experience $\mu' = \langle \alpha, \beta, \varphi, \varphi', t' \rangle$ represented in equation 7:

$$S = (1 - \mid Sim(\phi', \phi) - Sim(\varphi', \varphi) \mid) \cdot Sim(\phi, \varphi) \tag{7}$$

D. Time

The model of the behaviour of agents in a digital ecosystem evolves along time. This is due to two factors: new experiences arrive and the information decays as times goes by leading to ignorance. These two factors are considered together in the next general equation that updates the PDFs.

$$\mathbb{P}^{t'} = (MRE(\mathbb{P}^t, Q) - \mathbb{D}) \cdot \nu^{\Delta t} + \mathbb{D} \tag{8}$$

where:

- $MRE(\mathbb{P}^t, Q)$ is the minimum relative entropy distribution from $\mathbb{P}^t$ satisfying the set of constraints Q.
- $\mathbb{D}$ is the decay limit distribution.
- $\nu \in [0, 1]$ sets the speed at which a probability distribution goes back to the decay distribution.
- $\Delta t = t' - t$ is the time increment from last update.

E. Trust

Finally, the trust that agent α has on β at time t' with respect to a potential commitment φ is evaluated using equation 9, where φ' is every possible observation.

$$T(\alpha, \beta, \varphi, t') = \sum_{\varphi'} \mathbb{P}(Prefer(\varphi', \varphi)) \cdot \mathbb{P}^{t'}(\varphi'|\varphi) \tag{9}$$

Trust is computed as the aggregation of the probability of each observation happening weighted by the probability of that observation being more preferable than the commitment.

As we don't assume a closed world we need to always represent the unknown possibilities. The set of unknown possibilities is represented with the symbol $\perp$. Because of that, for any commitment ϕ, there will always be at least two possibilities: ϕ and $\perp$. This permits the values of trust to respect some minimum commmon sense. For instance without information the probability of ϕ and $\perp$ would be the same, as the intuition indicates. As expereinces arrive with new possible observation: ψ, δ, ... the importance of $\perp$ decreases.

Trust is calculated on demand following algorithm 1 in this section. The main steps in the algorithm are:

1) *(lines 1 to 4)* build $Observations$ that is the set of all terms observed in the past plus $\perp$ and the current commitment.
2) *(lines 5 to 8)* build the decay limit distribution $\mathbb{D}$ and initializes the a priori PDF $\mathbb{P}^{t_{old}}$. We assume an equiprobable distribution for $\mathbb{D}$ and for the initial $\mathbb{P}^{t_{old}}$.
3) *(lines 9 to 20)* update the probability distribution $\mathbb{P}$ considering the elements in the experience history M_α that are similar enough (over a given threshold ω) to the current commitment φ. Note that the experiences are

[2]Remember that the refinement relation represents specificity between concepts.

[3]Given a probability distribution $\vec{q}$, the *minimum relative entropy distribution* $\vec{p} = (p_1, \ldots, p_I)$ subject to a set of J linear constraints $\vec{g} = \{g_j(\vec{p}) = \vec{a_j} \cdot \vec{p} - c_j = 0\}, j = 1, \ldots, J$ (that must include the constraint $\sum_i p_i - 1 = 0$) is: $\vec{p} = \arg\min_{\vec{r}} \sum_j r_j \log \frac{r_j}{q_j}$. This may be calculated by introducing Lagrange multipliers $\vec{\lambda}$: $L(\vec{p}, \vec{\lambda}) = \sum_j p_j \log \frac{p_j}{q_j} + \vec{\lambda} \cdot \vec{g}$. Minimising L, $\{\frac{\partial L}{\partial \lambda_j} = g_j(\vec{p}) = 0\}, j = 1, \ldots, J$ is the set of given constraints $\vec{g}$, and a solution to $\frac{\partial L}{\partial p_i} = 0, i = 1, \ldots, I$ leads eventually to $\vec{p}$. Entropy-based inference is a form of Bayesian inference that is convenient when the data is sparse [4] and encapsulates common-sense reasoning [5].

sorted by time. The equations implemented in these lines have already been described in this section. Concretely, *lines 11 and 12* correspond to the decay that affects to information as time goes by, and *lines 13 to 16* refer to the constraint computation necessary to update the expected observation PDF as stated in *line 17*.

4) *(lines 21 and 22)* apply decay over the expected observation PDF updated using the last experience. Time gets updated.
5) *(lines 23 to 26)* calculate the final trust value considering the probability of the terms in the set $Observations$ and the preference function.

Algorithm 1 function Trust(α, β, φ, t')

```
Require: $O$ {the shared ontology}
Require: $M_\alpha \subseteq M$ {$\alpha$'s history of experiences sorted by time}
Require: $Sim : Term(O) \times Term(O) \to [0,1]$ {term similarity}
Require: $Prefer : Term(O) \times Term(O) \to [0,1]$ {term preference}
Require: $\omega : [0,1]$ [Default 0.1] {minimum term similarity}
Require: $\nu : [0,1]$ [Default 0.95] {decay parameter}
Require: $n_{ex} : \mathbb{N}$ [Default 6] {influence limiter parameter}
Require: $inertia : [1,\infty)$ [Default 8] {influence amplifier parameter}
Ensure: $Trust(\alpha,\beta,\varphi,t') \in [0,1]$
 1: $Observations \leftarrow \{\varphi, \perp\}$
 2: for all $\mu = \langle \alpha,\beta,\phi,\phi',t\rangle$ do
 3:     $Observations \leftarrow Observations \cup \{\phi'\}$
 4: end for
 5: for all $\varphi'$ in $Observations$ do
 6:     $\mathbb{D}(\varphi' \mid \varphi) \leftarrow 1/size(Observations)$
 7: end for
 8: $t_{old} \leftarrow 0$; $\mathbb{P}^{t_{old}} = \mathbb{D}$
 9: for all $\mu = \langle \alpha,\beta,\phi,\phi',t\rangle$ in $M_\alpha$ do
10:     if $Sim(\phi,\varphi) \geq \omega$ and $t \leq t'$ then
11:         $\Delta t \leftarrow t - t_{old}$
12:         $\mathbb{P}^t \leftarrow (\mathbb{P}^{t_{old}} - \mathbb{D}) \cdot \nu^{\Delta t} + \mathbb{D}$ {Time goes by and $\mathbb{P}^t$ decays}
13:         for all $\varphi'$ in $Observations$ do
14:             $S \leftarrow (1 - \mid Sim(\phi',\phi) - Sim(\varphi',\varphi) \mid) \cdot Sim(\phi,\varphi)$
15:             $Q(\varphi' \mid \varphi) \leftarrow (\mathbb{P}^t(\varphi'|\varphi) + 1/n_{ex} \cdot S^{inertia} \cdot (1 - \mathbb{P}^t(\varphi'|\varphi)))$
                {Constraints to satisfy}
16:         end for
17:         $\mathbb{P}^t \leftarrow MRE(\mathbb{P}^t, Q)$ {Min. relative entropy from $\mathbb{P}^t$ satisfying $Q$}
18:     end if
19:     $t_{old} \leftarrow t$
20: end for
21: $\Delta t \leftarrow t - t_{old}$
22: $\mathbb{P}^{t'} \leftarrow (\mathbb{P}^{t_{old}} - \mathbb{D}) \cdot \nu^{\Delta t} + \mathbb{D}$ {Time goes by and $\mathbb{P}^{t_{old}}$ decays}
23: $T \leftarrow 0$ {Computing trust value}
24: for all $\varphi_i$ in $Observations$ do
25:     $T \leftarrow T + Prefer(\varphi_i,\varphi) \cdot \mathbb{P}^{t'}(\varphi_i \mid \varphi)$
26: end for
27: return $T$
```

III. SRM

As an application of the described trust model to strategic sourcing, we have developed the SRM toolbox. SRM, Supplier Relationship Management, establishes a technological framework to give support to the buying process in a supply chain. Strategic sourcing concerns: category management –sales analysis by product categories–, supplier relationships –operative buying process support– and supplier performance –advanced tools for supplier monitoring and selection–. The application SRM specially focus on supplier performance. It gives support for supplier selection evaluating experiences with past agreements and computing a trust value for each explored buying order.

Let us illustrate the scenario of supplier performance with an example.

Ana is the person in charge of buying the office supplies for her company. She has ordered thirty pens of a certain quality, and asked to receive them by tomorrow. She reaches an agreement with the supplier 'The Happier'. Unfortunately, she receives sixty pencils two days late. She feels disappointed as she wanted pens, not pencils, and she needed them immediately, not two days latter. The supplier 'The Happier' is not trustworthy.

Her level of satisfaction with the outcome of an agreement will depend on: (1) how important was for her each order dimension –in the example: product, quality, quantity and delivery day–, (2) how different is what she got –the observation– compared with what she asked for –the commitment– and, of course, (3) which are her preferences. All these points are contemplated in our trust model.

For supplier performance, the trust model is illustrated in the following using a simple ontology. The ontology describes an order as a concept with four components: product, delivery day, quality and quantity. We refer to them as order dimensions. As can be seen in figure 1, writing instruments are products. And these writing instruments can be either opaque (pen and pencil) or transparent (marker and highlighter). The ontology also represents that a delivery day can be +1 ... +7, the quality chosen for an order should be one of excellent, good, medium or bad. And finally, the quantity of products to be ordered is expressed by an integer value.

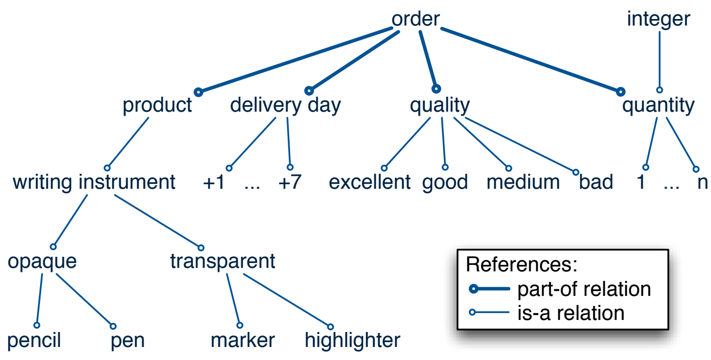


Fig. 1. SRM Ontology

Similarity between orders is computed as described in section II-B with three specific similarity functions for delivery day, quality and quantity. The relevance function is adjusted to give more importance to the product of an order and the quantity than to the rest of components.

The PDF is updated based on experiences and the decay of information is as described before. The preference function represents the knowledge of what is preferred with respect to the commitment and to what degree it is preferred, for instance, higher quality in the product is to be preferred in this illustration. More quantity or lower delivery day are not preferred because of the storage problems they can provoke. Any way, preference can be adjusted by the SRM user for every order component if it is required.

The experience reported in the previous example can thus be represented as:

$$\mu = \langle \text{Commit}(\textit{'The happier', 'Ana', order(pen, 30, good, +1)}), \text{Obs}(\textit{'Ana', 'The happier', order(pencil, 60, medium, +3)}), 12.5\rangle$$

An example of expected observation PDF is represented in figure 2. It belongs to the evaluation of the commitment of $order(pen, excellent, +1)$. In this order, quantity is not indicated. In fact, SRM can evaluate the trust in a supplier when only a subset of order components has been specified in the commitment. As it is shown in the PDF, it is expected to receive the requested product, pen, but it is known that sometimes there is a delay in the delivery of pens of such quality. If the user preferences specify that there is no problem with small $(\leq +5)$ delivery delays then the trust associated to this PDF will be high.

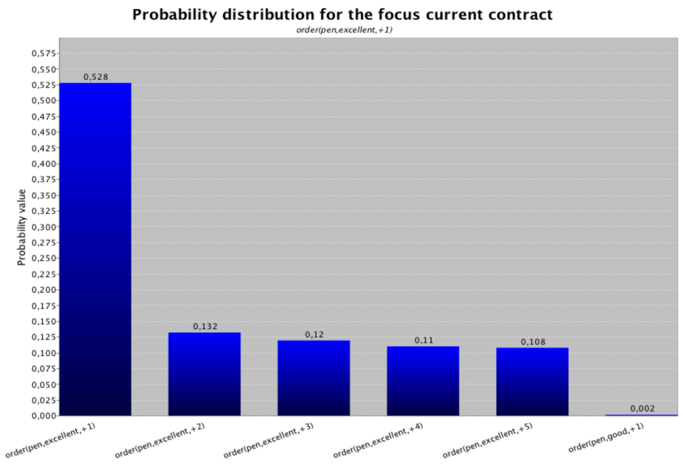


Fig. 2. Bar chart representing $\mathbb{P}(\varphi'|order(pen, excellent, +1))$. The height of the bars corresponds to the probability of a specific order being observed. Expected orders are represented at the bottom.

A. SRM functionalities

The SRM toolbox has four tools to analyse the performance of suppliers: Trust, Supplier, Critical Order and Minimal Cost. The features of these tools are described below.

1) Trust: The aim of the Trust tool is to show the trust on a supplier for a given commitment φ. Three different charts are visualized: a bar and a point similarity charts representing the expected observation PDF for the commitment φ, and a chart with the trust evolution over time.

Several parameters allow to adjust the trust measure as desired. One of them is the speed of memory loss. Changing this parameter, the relevance of old experiences to build the PDF changes.

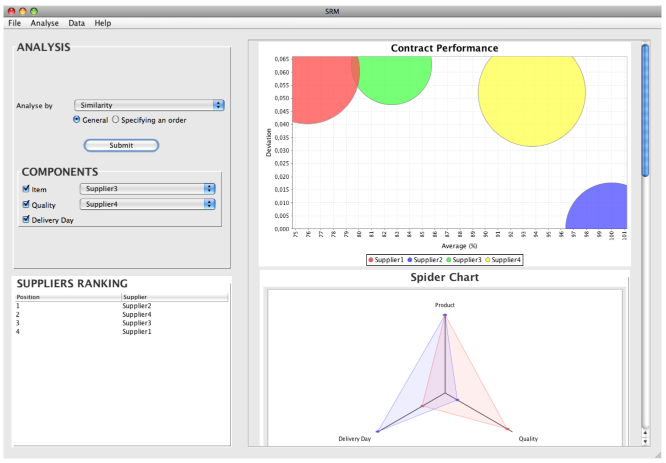
Trust can also be computed focusing on a subset of the order dimensions.

2) Supplier: The Supplier tool does not compute trust as described in section II. Experience history data is represented in several ways using mean and standard deviation measures of either similarity between commitment and observation or experience satisfaction. Satisfaction is evaluated using the preference function.

This tool provides a ranking of suppliers and a mean versus standard deviation chart representing each supplier as a circle whose diameter is proportional to the number of previous experiences with that particular supplier.

Besides, the tool includes a set of charts representing the comparison between two suppliers along each order dimension. These are spider charts –summarizing the behavior of the two selected suppliers along each order dimension– and a new chart for each dimension illustrating the behavior of the supplier for the corresponding order dimension over time.

Figure 3 is a screenshot of this tool where most of the described charts are visible. In addition, this tool allows both to ignore some of the order dimensions and to constrain the experiences to those which have a commitment quite similar to the one being analysed.



SUPPLIERS RANKING

Position	Supplier
1	Supplier2
2	Supplier4
3	Supplier3
4	Supplier1

Fig. 3. Screenshot of the Supplier Tool. At top left the user can choose between similarity or satisfaction and select which suppliers he wants to compare. At top right there is a mean versus standard deviation chart where all the suppliers are represented. The suppliers that behave better are in the bottom right part of the chart. Finally, at the bottom we can see a ranking on suppliers (left) and the spider chart of the two suppliers being compared. The blue one has problems in guaranteeing quality and the red doesn't succeed in delivering the product at the correct day.

3) Critical Order: This tool allows us to study how the suppliers are expected to perform when committing to a specific order. We can specify priorities between order dimensions –relevance of each dimension for the similarity calculations– and our preferences. Some dimensions can also be ignored.

The tool shows a ranking of suppliers based on their trust value for the desired order, priorities and preferences. It also shows a trust versus entropy chart where all suppliers are represented. Note that the most trustworthy suppliers are situated at the bottom right area.

4) Minimal Cost: Minimal Cost is perhaps the smartest tool included in SRM. It provides a split of orders along suppliers such that the split guarantees the levels of satisfaction that we want for each order dimension whilst minimizing the overall cost. To this end, an order and several percentages of satisfaction have to be indicated, one per dimension.

This tool aims to provide the cheapest solution that satisfies the user need. The solution indicates how many items should provide each supplier.

B. iQuotes integration of SRM

Several enterprise resource planning systems (ERP) are nowadays dealing with eSourcing. For example: iQuotes[4], SAP[5] and Ariba[6]. But no one is able to manage trust. In fact, most of the ERPs are more focused on data management than on decision support. Therefore, SRM is an innovation in the field of industrial strategic eSourcing that has its roots on an agent trust model.

In addition to its standalone use, SRM is being integrated into the iQuotes Suite of the company iSOCO[7]. In fact, SRM will be a new module of the suite from which it will get information about past experiences. This is, information about committed orders and feedback about their respective execution observation. For example, the user can inform that the execution was late, the quality different than committed, etc. With this information a new experience can be completed and thus be used to update the trust on the corresponding supplier.

IV. Discussion

In this paper we have described a trust model that takes into account (1) the passage of time, (2) past similar experiences, (3) number of previous interactions with a given agent, (4) preferences, as well as (5) the importance given to each of the dimensions that describe the new commitment. To that end, an ontology has to be defined together with a domain dependent similarity measure. The trust evaluation is computed on demand and bases its calculations on an expected observation PDF and a preference function. How this PDF is build is described in Section II-C.

In addition, we have applied this trust model to eSourcing, and concretely to strategic sourcing. A toolbox for supplier performance, SRM, is provided to give support to a company in selecting the supplier that better fulfills its needs when an agreement is negotiated. Indeed, the company's needs are expressed, as indicated by the trust model, by means of the terms of a shared ontology and a preference function. SRM is also being integrated in a larger ERP with the objective of being used as one of the components of the commercial sourcing application suite iQuotes.

Future work can be summarized in three points: aggregation of more tools to cover contract management as well as supplier performance, use of reputation to compute the trust value and support on negotiation. The design of more tools will come together with a users's demand of them. At the moment we will deploy the functionality described and will be in touch with the company iSOCO to report the need of extra functionality. Negotiation support is already being held in some way by iQuotes. We plan to improve the negotiation model by adding tools based on information theory to model information gain and ideas from cognitive science to help modelling the relationships between members of a supply chain. Finally, reputation can be applied to our trust model by the aggregation of external experiences and opinions. The use of experiences of other members of the digital ecosystem to update the expected observation PDF is studied in general terms in [6], but it doesn't include any software implementation or test results. The other way to incorporate reputation is by opinions. Users would be able to express their opinions about suppliers without making their experiences with them public. Some work is being done with this objective, [6], but how the trust model described in this paper can be modified to incorporate information about opinions is ongoing work.

[4]http://www.isoco.com/soluciones_es_iquotes_suite.htm
[5]http://www.sap.com
[6]http://www.ariba.com/
[7]http://www.isoco.com

Acknowledgment

Research supported by the PROFIT and the Consolider-Ingenio programs of the Spanish Ministry of Science and Innovation by means of the Strategic eSourcing and Agreement Technologies projects respectively. Partial support is given by the LiquidPub EU funded project.

References

[1] A. Jøsang, R. Ismail, and C. Boyd, "A survey of trust and reputation systems for online service provision," *Decis. Support Syst.*, vol. 43, no. 2, pp. 618–644, 2007.

[2] C. Sierra and J. Debenham, "An information-based model for trust," in *AAMAS '05: Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems*. New York, NY, USA: ACM, 2005, pp. 497–504.

[3] Y. Li, Z. A. Bandar, and D. McLean, "An approach for measuring semantic similarity between words using multiple information sources," *IEEE Transactions on Knowledge and Data Engineering*, vol. 15, no. 4, pp. 871 – 882, July / August 2003.

[4] P. Cheeseman and J. Stutz, *Bayesian Inference and Maximum Entropy Methods in Science and Engineering*. Melville, NY, USA: American Institute of Physics, 2004, ch. On The Relationship between Bayesian and Maximum Entropy Inference, pp. 445 – 461.

[5] J. Paris, "Common sense and maximum entropy," *Synthese*, vol. 117, no. 1, pp. 75 – 93, 1999.

[6] C. Sierra and J. Debenham, "Information-based reputation," in *ICORE'09: International Conference on Reputation*, 2009, p. [in press].