

# Adaptive Business Intelligence for an Open Negotiation Environment

S. Aciar<sup>1</sup>, P. Avesani<sup>2</sup>, J.L. De la Rosa<sup>1</sup>, N. Hormazabal<sup>1</sup>, A. Serra<sup>2</sup>

<sup>1</sup>University of Girona, Campus Montilivi, 17071 Girona, Spain, e-mail : (saciar, peplluis,nicolash)@udg.edu

<sup>2</sup>Fondazione Bruno Kessler, Via Sommarive 18, 38100 Trento, Italy, e-mail: (avesani, serra)@fbk.eu

**Abstract**—Engineering of negotiation model allows to develop effective heuristic for business intelligence. Digital Ecosystems demand open negotiation models. To define in advance effective heuristics is not compliant with the requirement of openness. The new challenge is to develop business intelligence in advance exploiting an adaptive approach. The idea is to learn business strategy once new negotiation model rise in the e-market arena. In this paper we present how recommendation technology may be deployed in an open negotiation environment where the interaction protocol models are not known in advance. The solution we propose is delivered as part of the ONE Platform, open source software that implements a fully distributed open environment for business negotiation.

**Index Terms**—Intelligent Digital Ecosystems and Technologies.

## I. INTRODUCTION

Once a negotiation model is known in advance it is straightforward to design an effective business intelligence that may support users to successfully take part to the corresponding marketplace. The auction model defined and deployed by eBay allows many people to conceive effective strategy and to deliver even business software agent that may attend the negotiation on behalf of the end users [1].

One of the fundamental shifts introduced by Digital Business Ecosystems (DBE) is the feature of openness. In an open market not only new users may join anytime the marketplace but also new negotiation model may arise to fulfil new emerging business habits. Known negotiation models as English auctions, Dutch auctions, and reversed auctions represent only a narrow portion of the business negotiations that may take place in DBE. Most of them refer to unstructured negotiation models [2] tailored to the custom needs of specific domains.

Aspire [3] represents one of the early attempts to design a system that covers the whole process of negotiation engineering: design, development and deployment of a negotiation model. More recently ONE [4] extended this work including the requirements of a digital business ecosystem. Introducing the notion of open environment new challenges arise: how to develop business intelligence for a broad range of new negotiation models that are not known in advance? How to develop a methodology that is model independent?

Our proposal is organized into two parts: (1) to define a meta-model of negotiation processes, (2) to exploit learning and recommendation technologies to develop the business

intelligence.

The former intuitive idea is to follow a model-driven approach. Despite the broad range of possible unstructured negotiations [5] we define an abstraction, i.e. the meta-model, that subsumes all the possible negotiation models. The design of model independent recommendation services may be achieved by referring the meta-model of negotiation rather than the specific negotiation model.

The latter intuition is to pursue the learning of business intelligence from data [6] [7] rather than encoding hard-wired heuristics [8]. Business intelligence may depend on negotiation models but also on the use that a population of users does of them. Learning from data allows to fit the business intelligence both with respect to the negotiation model and to a given population of users in a given period of time. Furthermore, learning-based approach allows to be much more flexible with respect the evolution of the behaviour of a community of users. We first designed the meta-model for negotiation, and then we defined the recommender functionalities to cover the whole process. Recommendation methodologies have been designed accordingly to the specific negotiation stages. Our contribution includes a deep empirical assessment through extensive simulation with different case studies.

This work was part of ONE [4], an European project devoted to develop methods and technologies for an open negotiation environment.

In the following Section we briefly introduce the notion and the platform of an open negotiation environment. In Section III we illustrate the negotiation meta-model and the recommendation services. Sections IV-VI are devoted to recommendation methodologies and their empirical evaluation.

## II. OPEN NEGOTIATION ENVIRONMENT

Open Negotiation Environment (ONE) is both an extension of Digital Business Ecosystem and at the same time a software platform [4]. The goal of ONE is to enable an open, decentralised negotiation environment and providing tools that will allow organisations to create contract agreements for supplying complex, integrated services as a virtual organisation/coalition. ONE is also an open source software platform (<http://one-project.eu>). The architecture includes many components that cover a wide range of services: a factory for modeling negotiations, a fully decentralized environment of negotiation execution, a distributed en-

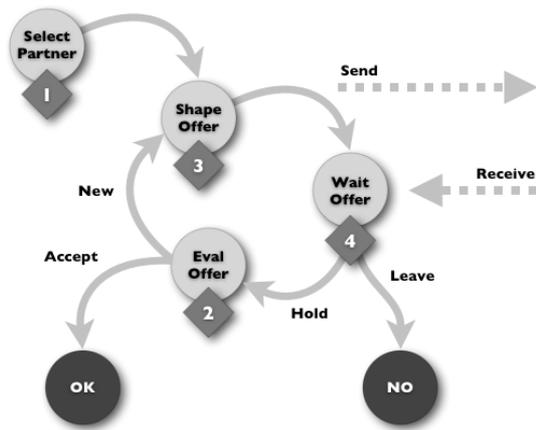


Fig.1 A simplified view of the negotiation meta-model

tity and authentication management, a trust and reputation mechanism, a self learning and recommendation strategies advisor, an ecosystem monitoring supervisor.

The most representative use case may be described by a SME (Small and Medium Enterprises) that has to buy some services in outsourcing. First it may define a negotiation model that fulfils the own business process. Afterwards it may proceed by deploying an instance of such a negotiation model on the internet. The execution of a negotiation includes many stages: the setup, the invitation, the admission, the offering, the bargaining, the consolidation, the agreement. The agreement is forwarded to arrange the contract to be signed while the log of negotiation is stored in the memory. Of course the model of negotiation may be reused for further businesses.

The software component devoted to recommendation and learning was designed to support the stage of negotiation execution.

### III. NEGOTIATION META MODEL

In an open negotiation environment the negotiation model is not hardcoded in advance. The users may design and deploy their own negotiation models that better fit their requirements. Nevertheless the design of negotiation models is constrained by a negotiation meta-model that subsumes all the negotiations that are supported by the negotiation engine. The negotiation meta-model is partitioned into three phases: (i) setup a negotiation, (ii) run a negotiation, (iii) close a negotiation.

Fig. 1 depicts a simplified view of the meta-model of negotiation restricted to phase of running a negotiation. The meta-model is encoded by super-states and the transitions among them. Super states includes (1) "Select Partner", (2) "Eval Offer", (3) "Shape Offer", (4) "Wait Offer". The super-state will become the state in the negotiation model once the meta-model will be instantiated.

The first super-state denotes the selection of partners. In a private negotiation it means to filter out from own contacts who to invite joining a negotiation while in a public negotiation it means to assess whether to accept an admission request. The transition from "Select Partner" brings to "Shape Offer" super-state. Shaping offer means to identify

what kind of issues to rise in the definition of an counter-offer. Next transition moves to "Wait Offer" super-state in the meanwhile a message is sent to the counter parts with the offer. While waiting for a counter-offer it is anytime possible to leave the negotiation. Once it is received a new counter-offer we have the transition to the next super-state "Eval Offer". The offer evaluation may bring to an agreement, i.e. a steady state "Ok", or to iterate the bargaining by moving to the "Shape Offer" super-state again.

The meta-model of negotiation allowed us to recognize what are the most crucial states in a negotiation process and then try to design the recommendation services with a good coverage of potential negotiation models that will be delivered by the negotiation factory.

We conceived a recommendation support for each of the four main super-state. First recommender supports the choice of partner when the negotiation is in the "Select Partner" super-state. The goal of recommender is twofold: on one hand to filter contacts with competences that fit as much as possible the matter of negotiations, on the other hand to filter contacts according to the estimate of their reputation. The trade off of these two elements should enable the recognition of partners that will allow to reach better agreements. The recommender output is a rank of contacts computed taking into account the properties of the partner such as demographical information or the product offer and reputation. A second recommender supports the formulation of an offer when the negotiation is in the "Shape Offer" super-state. An offer is a dynamic structure defined as a collection of issues and an hypothesis for their values. Shaping an offer means to select what kind of issue is critical for the agreement and to propose a value of assignment for such an issue. The recommender helps to detect what kind of issue is most critical for the counter-part. A new assignment of values will be restricted to such issues that prevent to achieve an agreement. The output of the recommender is a subsample of issues that should be elicited in the upcoming new offer. The third recommender supports the choice whether to leave the negotiation or to wait for a counter-offer when the process is in the "Wait Offer" super-state. To attend a negotiation is costly. When a negotiation ends without an agreement there is a loss of resources. If the expectation of receiving a satisfactory offer is low it might be worthwhile to leave the negotiation since the trade off between the effort and the quality of agreement would be too low. The recommender computes an estimate of the expectation to receive an offer better/worse of the current one in the subsequent iterations. Is in charge of the user to evaluate whether the current offer is enough good for an agreement or to leave the negotiation. The fourth recommender supports the assessment of the received offer when the negotiation is in "Eval Offer" super-state. The evaluation is concerned on what might be a value of final agreement. A reference value for an agreement may help the user to arrange a proper strategy for the subsequent bargaining. The output of recommender is an estimate of the values for open issues that should be part of a successful negotiation.

All four recommenders are required to not be dependent from a specific negotiation model. The recommendation methodologies have to be constrained only by the negotiation meta-model as described above.

#### IV. RECOMMENDATION METHODOLOGIES

The motivation of exploiting recommendation methodologies for an open negotiation environment is twofold. The former reason is that ONE, as presented above, is targeted to enable a market place among real users. Although negotiations take place in a virtual world the primary goal is not to provide an environment for bargaining among software agents that play on behalf of real users. The latter motivation is to overcome the approach based on heuristics [8] that might be really effective but suffers of a lack of adaptivity with respect to end users, negotiation models and business ecosystem.

In the following we will argue how recommendation methodologies allow to cover a wide range of different advisory supports. The recommendation services depicted above have been implemented using four different heterogeneous techniques.

##### A. Trust and Reputation Metrics

In an open negotiation environment users have to deal with the problem of selecting the appropriate partners to start a negotiation. Usually the main objective is to detect partners whose profiles better fulfil the quality of services that will be required in the negotiation process. Roughly speaking it is matter of finding a good match between the requirements of a tender and the profiles of tenders. Nevertheless such a match doesn't exhaust the selection of a suitable partner. A negotiation includes also many other factors. The reputation of a partner may provide helpful insights on negotiation style, like how much the partner is prone to leave the negotiation before the agreement or whether the partner tends to have longer bargaining interaction. The selection of partners is therefore concerned with many different levels of assessment that sometime interleave each other. The challenge is how to combine these factors in a comprehensive evaluation of a potential partner. The intuitive idea is to conceive the selection of partners as a process of two subsequent stages, the former devoted to user profile matching taking care of context information, the latter in charge of trustworthy user filtering [9]. This working hypothesis relies on the premise that the ONE software platform supports the elicitation and the management of trust statements with respect to users in the own contact list.

Given a set of partners  $\mathbf{P} = \{p_i\}$ , a preliminary filtering is computed by a similarity measure between two user profiles. A further rank is computed according to the notion of reputation (R) that might be conceived as the general evaluation of a partner in the user community of users. Reputation value for a partner  $p_i$  is derived from the trust ratings of other users (j) and it is defined as the average of the trust  $t_j(p_i)$  ratings from all partners in the system ( $n$ ).

$$R(p_i) = \frac{\sum_{j=1}^n t_j(p_i)}{n} \quad (1)$$

The challenge is to prove that combining sequentially the filtering based on context information and the ranking according to reputation estimate the quality of the final agreement would increase. In the next Section VI-A we will provide an implementation of the notion of trust that enables an effective exploitation of reputation as defined above.

##### B. Case-Based Reasoning

A case-based approach [10] has been adopted to support the stage of offer evaluation. Case-based reasoning relies on the assumption that similar problems share similar solutions. In our setting the problem was defined as the decision to accept the current offer or to proceed by negotiating a new counter-offer. We designed the notion of case encoding the main information that describe a stage of negotiation: (1) the context of negotiation – e.g. the partners, (2) the negotiation protocol – e.g. the specific model of interaction, (3) the negotiation process – e.g. the current open issues, (4) the final agreement, if any.

The recommender engine works computing two subsequent operations of filtering and adaptation. Given a description of the current stage of a running negotiation according to the definition of case above, a subsample of similar situations are filtered out from the collection of past negotiations. The filtering takes place as computation of a similarity metric that combine the assessment of information concerned with the context, the protocol and the process. The step of adaptation is in charge to compute an estimate of the final value of possible agreement and the expected number of interactions required to conclude the negotiation.

The benefit for the user is twofold. On one hand the evaluation of an offer may take advantage of a reference hypothetical agreement that allows to estimate how far it is the conclusion of the negotiation. On the other hand, the recommender provides a pointer to specific past negotiations that may represent the source of additional insight for a deeper assessment of the current offer evaluation.

The lazy learning technique implemented by case-based reasoning fits better the requirements of an open negotiation environment. The low bias of such a soft computing method is compliant with the potential high variance of negotiation models that a recommender will have to deal with.

##### C. Bayesian Experimental Design

The task of shaping an offer is supported by a recommender based on Bayesian experimental design [11]. The problem setting is based on the assumption that an offer may be rejected because it doesn't satisfy the counter-part preference model or because the offer doesn't include the information that the counter-part needs to take a decision. The possible answers are three,  $A = \{0, \infty, 1\}$ : reject, unknown, accept respectively. From the point of view of the negotiation the first two answers prevent both to achieve an

agreement. In this case the main purpose of recommender is to reduce the number ‘unknown’ feedbacks including in the offer all the issues that allow the counter-part to take a decision. The challenge is to design a policy that actively selects issues to probe the counter-part. The goal is to learn those issues that affect the evaluation of the offer. The learning effort is restricted to the issues that trigger the decision model and to acquire a deeper preference model of the counter-part is out of the scope. Differently from other works we are not interested to learn the values of issues that will enable a positive answer. The intuitive idea is that if we reduce the number of ‘unknown’ answers, part of them might be converted in positive answers. Once detected the relevant issues, the negotiation will be matter of finding a compromise for their values.

The recommender engine may be conceived as a policy to select at each stage of negotiation what kind of issues to include in the offer. The idea is to compute for each issue an estimate of its expected benefit whether it would be used to shape the offer. Once such a benefit measure is computed for each issue it is straightforward to select the one with the highest value.

Let see how we may introduce a measure of benefit. We define a matrix  $\mathbf{F}_m = \{f_{ij}\}$  that records after ‘m’ steps all the past offers shaped in the ongoing negotiation, where  $f_{ij}$  denotes the conditional probability that for the i-nth the issue j-nth might be relevant for the counter-part. We represent the history of answers as a vector  $\mathbf{A}_m = \{a_i\}$ , where  $a_i$  denotes the feedback of the counter-part to the offer i-nth and takes values in  $\{0, \infty\}$ . Let  $g$  be a relevance function that computes the mutual information for a given issue for the matrix  $\mathbf{F}_m$  with respect to the vector  $\mathbf{A}_m$ . High relevance of an issue means that it might play a key role in the assessment of an offer.

Given these premises we can define a benefit function as:

$$B(\text{issue}_j) = \sum_{a \in \{0, \infty\}} [g_j(F_m, \text{issue}_j | a) - g_j(F_m | a)]^2 \cdot \Pr(a | F_m) \quad (2)$$

The benefit is computed as the expected squared increment of the relevance for a given issue weighted by the probability to receive one of the possible answers. See [12] for a detailed definition of the computational model.

#### D. Trust-aware Look ahead

If we consider that agents are aware of their own preferences, and they know how far are willing to go on the negotiation process (in a simple bargain, how low they can go with their offer in order to achieve an agreement).

Let’s consider that there’s available a knowledge base based on the past negotiations ran on the environment which contains a set of tuples with the offers made and the result of each one of them; this is if that offer has been successful accepted or not. If the agent is about to make an offer, it can look for similar offers in the past (similarity can be calculated by several values such as price, items included, etc) and estimate a success rate for it (the number of success offers divided by the total number). This is a situational trust value  $-ST_a(y, z)$ -, as it is based on the trust an agent ( $a$ ) evaluates towards another ( $y$ ) in a specific situa-

tion ( $z$ ) taking information from past similar ones [13]. If the agent is aware of the following offers it is going to make in case the current one is not accepted (a set  $S$  of possible offers), it can also estimate the maximum value of the situational trust of the set of future possible offers.

Having a trust value for the current offer and another one for the possible future offers, we can get an acceptance rate  $AR$ , which can suggest if it is better to continue on the negotiation or ending it as the expectations will not be likely fulfilled with the future offers.

$$AR = \frac{ST_a(y, z)}{E_a(y, S)} \quad (3)$$

If  $AR$  has a value greater than 1, it means that the expected future offer results will not be better than the current one. This will be called from now on, Trust Aware Negotiation Dissolution (TAND from now on), more details on it in [14] and [15].

## V. EXPERIMENTAL SIMULATION

Since we are interested to investigate the performance of the recommendation methodologies on a scale of digital business ecosystem, we designed the empirical evaluation by defining an experimental simulation rather than a trail on the field. Our main purpose is to collect statistical evidence of the effectiveness of an approach based on recommendation. In the following we focus our discussion only to results achieved by simulations.

The empirical assessment is performed using two different datasets, the former a kind of benchmark for the recommendation community, the latter concerned with a real world example of business negotiations. The original dataset of Movielens has been used to model the profiles of users involved in bilateral negotiations to find a common agreement on what kind of movie to watch together. The Italian archive of tenders for service provisioning was sampled to extract a collection of negotiation logs that occurred in the last year. The mining of such a logs provided the information to replay a simplified version of past negotiations.

We tested the partner invitation recommender mainly with the dataset concerned with real business negotiation because the logs allowed to derive the implicit information on reputation. The recommenders for offer shaping and negotiation dissolution were tested with the dataset derived from Movielens. The evaluation of the case-based recommender was performed arranging a trial with end users because it is not straightforward to model different strategies for bargaining.

Since ONE is concerned with unstructured negotiations whose models are not known in advance, it is difficult to adopt the usual measure of BVPM (Best Value Per Money). As reference criteria for the evaluation we computed two kinds of measures: (1) the average number of successful negotiations i.e. closed with an agreement, (2) the average number of steps required to achieve an agreement.

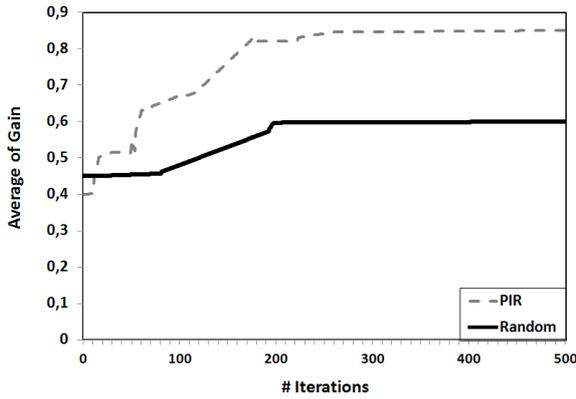


Fig.2 Empirical results for PIR

VI. EMPIRICAL RESULTS

The results presented in this Section refer to simulations with the following setting: ~100 users, ~1000 negotiations, ~10000 interactions. The discussion addresses the comparison of recommendation methodologies with respect to a random strategy, that might be considered an heuristic to average the performance when it is not available any knowledge in advance of the specific negotiation model.

A. Partner Invitation Recommender

As mentioned in Section IV-A the recommender for partner invitation refers to the notion of reputation as a derived measure from trust. In our experiments we implemented the trust of each partner based on the record of successful or unsuccessful negotiations. A trust value is computed for each one of the partners. Given the information about the successful negotiations the measure of trust defined by Patel et al. [16] is applied. They define the value of trust in the interval between [0,1], where 0 means an unreliable partner and 1 a reliable partner. The trust of a partner *p* is computed as the expected value of a variable *B<sub>s</sub>* given the parameters *v* and *d*. *B<sub>s</sub>* is the expected value that *p* could have to perform the task. This value is obtained using the next equation.

$$T_j(p_i) = E[B_s / v, d] \tag{4}$$

E is computed as follows:

$$E[B_s / v, d] = \frac{v}{v + d} \tag{5}$$

Where the parameters *v* and *d* denote the number of successful experienced negotiations and the number of unsuccessful ones respectively. For the experiments, we designed simulated negotiations where the recommender has to provide as result a partner start a negotiation about a given service request. The negotiation takes place as subsequent steps of generation of an offer and counter offer. Once the process has finalized the gain or loss is calculated for both user and partner as follow:

$$Loss(u) = GainTota_u - GainAgreement_u \tag{6}$$

The *GainTota<sub>u</sub>* is the expected gain that the user/partner expects to obtain if the final agreement is composed by all the issues that receive higher gain; the *GainAgreement<sub>u</sub>* is

Table 1 - Empirical Results for AOS

AOS	RND	%	μ	σ
Unsuccessful	Unsuccessful	0.05	85.9	8.1
Successful	Successful	0.17	257.2	13.1
Unsuccessful	Unsuccessful	0.01	20.4	4.9
Faster	Faster	0.26	400.0	13.06
Slower	Slower	0.46	696.0	20.7
	Same speed	0.02	41.3	6.2

the gain obtained only from the services in the final agreement.

We performed the simulations applying two competing strategies: the former using our approach to select the partner, the latter one selecting the partner randomly. The empirical analysis addressed a case study with data extracted from the Italian archive of tenders for service provisioning. Fig. 2 illustrates the evolution of the average gain of the users with respect to an increasing number of simulated negotiations. The results show how the overall utility of a ONE user improves over time when the recommendation of partner invitation is taken into account. The benefit in terms of gain increment may overcome the 30% of the average expected gain using a uniform strategy for partner selection. Learning the estimate of partner reputation requires only few tens of negotiations.

B. Offer Shaping Recommender

From the simulations we obtained that the rate of success for random strategy is 0.81, whereas for the AOS strategy is 0.93. There is an increment of 12% (180 negotiations) in the number of successful negotiations when using the AOS recommender. It is interesting to analyze in detail the disaggregated results, as reported in Table 1. The first row reports the mean number of counts (and the corresponding standard deviation) when adopting the AOS recommender the negotiation fails, whereas, using a random strategy the negotiation succeeds. The second row depicts exactly the opposite, when using the AOS recommender is beneficial and random strategy does not. The difference between these two counts is about 170 which explain the increase in the rate of success when using the AOS strategy. The fourth row depicts the situation where both the strategies produce a successful negotiation,

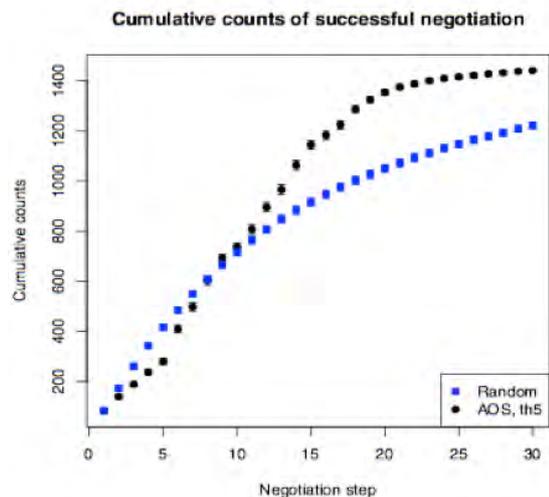


Fig.3 Empirical results for AOS

Table 2 – Empirical results of TAND

	Case 1	Case 2
AD	0,1548	0,1125
AS	5,6993	4,6683

but using AOS a faster agreement is reached. The fifth line is the opposite. Here we can notice that, typically, there are about 700 negotiations where the random strategy produces faster successful negotiations. Another interesting result is the comparison of the speed in obtaining a successful negotiation. In Fig. 3 we plot the cumulative counts of successful negotiations (and error bars associated), where we can clearly see that in the first part of the negotiation, the random strategy is better, but after the 20th offer the AOS recommender provides additional successes. This indicates that at certain point in the negotiation, the uniform sampling of the random strategy stops to be fruitful, whereas the AOS recommender keeps being effective.

### C. Negotiation Dissolution Recommender

For testing purposes, we implemented a negotiation environment where two agents negotiate to reach an agreement from a limited number of options; agents consecutively offer their next best option at each step until the offer is no better than the received one. The scenario consists of different agents that each represent a person who wants to go to a movie with a partner, so they negotiate between them from different available movie genres to choose which movie to go to together.

The idea is to test the TAND suggested in Section IV-D, in a scenario where there will be a fixed number of available movie genres (for example, drama, comedy, horror, etc.) during the whole simulation. Each agent will have a randomly generated personal preference value (from a uniform distribution) for each genre between 0 and 1, where 0 is a genre it does not like at all, and 1 is its preferred movie genre. One of these genres, randomly chosen for each agent, will have a preference value of 1, so each agent will have always a favourite genre. Each interaction will be saved in a knowledge base, so future negotiations will have information of past negotiations to calculate a trust value for each possible offer. Comparing two cases, where Case 1 is a simple bargain where agents offer and counteroffer starting from the option they like the most until the received offer is no better than the next one they are going to make. Case 2 is using the TAND. We will use as a reference value the distance from the perfect agreement. We define the perfect agreement as the highest value for the product of each agent's preference among the different possible agreements. This is then, the best possible agreement for both agents. Being the perfect agreement  $P$ , and the product of the final agreement  $A$ , the distance from the perfect agreement is  $AD=P-A$ , the lower the value the better. The other value used for measuring the performance will be the average steps needed to achieve an agreement ( $AS$ ). As we can see in Table 2, the results show an improvement on the distance from the perfect agreement of near a 35%, and the steps needed for reaching an agreement are a 20% lower. So we

have faster and better agreements.

## VII. CONCLUSIONS

The summary of the main contributions of this work is: (1) we faced with the openness requirement of DBE, (2) we argued the use of learning and recommendation technologies for ex-post developing of business intelligence, (3) we covered the whole negotiation process by means of a broad scope of heterogeneous recommendation methodologies, (4) we provided empirical evidence of the benefits.

## VIII. ACKNOWLEDGEMENT

This work is partially funded under the IST program of the EU Commission by the STREP-project "ONE" (INFSO-IST-034744).

## IX. REFERENCES

- [1] M. Lenar and J. Sobacki. "Using recommendation to improve negotiations in agent-based systems", *Journal of Universal Computer Science*, 13(2):267–286, 2007.
- [2] M. Bichler, G. Kersten, and S. Strocker. "Towards a structured design of electronic negotiations", *Group Decision and Negotiation*, vol.12, n.4, 2003.
- [3] Kersten, G.E. and Lo, G., "Aspire: an integrated negotiation support system and software agents for e-business negotiation", in *International Journal of Internet and Enterprise Management*, 2003, 1(3).
- [4] L.Telesca, J.Finnegan, P.Ferronato, P.Malone, F.Ricci, K.Stanoevska-Slabeva, "Open Negotiation Environment: An Open Source Self-Learning Decentralised Negotiation Framework for Digital Ecosystems", In *Proceedings of the Inaugural IEEE International Conference on Digital Ecosystems and Technologies, (IEEE-DEST)*, February 2007, Cairns, Australia.
- [5] A.R. Lomuscio, A.R., M. Wooldridge and N. R. Jennings, "A Classification Scheme for Negotiation in Electronic Commerce", in: *Agent-Mediated Electronic Commerce: A European AgentLink Perspective*, Editors F. Digham and C. Sierra, Springer Verlag., Heidelberg: 2001.
- [6] D.Zeng, K.Sycara, "Benefits of learning in negotiation", in *Proceedings of the 14th National Conference on Artificial Intelligence, 1997*.
- [7] D. Zeng and K. Sycara, "Bayesian learning in negotiation", *International Journal of Human-Computer Studies*, vol.48, 1998.
- [8] H.J. Mueller. "Negotiation principles", pages 221–229. *Foundations of Distributed Artificial Intelligence*, Sixth-Generation Computer Technology. John Wiley and Sons, Inc., New York, 1996.
- [9] P.Massa, P.Avesani, "Trust Metrics in Recommender Systems", in *Computing with Social Trust*, J.Golbeck (Ed.), Springer, 2009.
- [10] D.Bridge, M.Goecker, L.McGinty, B. Smith, "Case-based Recommender Systems", *The Knowledge Engineering Review*, vol.20, n.3, 2005.
- [11] E.Olivetti, S.Veeramachaneni, P.Avesani, "Active Learning of Feature Relevance", Book Chapter in *Computational Methods of Feature Selection*, edited by H.Liu and H. Motoda. Chapman and Hall, 2008.
- [12] A.Malossini, A.Serra P.Avesani, "Active Offer Shaping", in *Proceedings of Group Decision and Negotiation (GDN-08)*, Coimbra, Portugal, June 2008.
- [13] S. Marsh, "Formalising Trust as a Computational Concept", *Ph.D. dissertation, Department of Mathematics and Computer Science*, University of Stirling, 1994.
- [14] N.Hormazabal, J.L. de la Rosa, S. Aciar, "Trust Aware Negotiation Dissolution", in *Proceedings of the 18th European Conference on Artificial Intelligence. (ECAI2008)*, Patras, Greece, 2008.
- [15] N. Hormazabal, S. Aciar, J. L. de la Rosa, "Agent Negotiation Dissolution", *Proceedings of the 11th International Conference of the Catalan association for Artificial Intelligence (CCIA)*, 2009.
- [16] Patel J., Teacy W. T. L., N. R. Jennings, and M. Luck. "A probabilistic trust model for handling inaccurate reputation sources", in *Proceedings of Third International Conference on Trust Management*, 23-26 May, 2005, Rocquencourt, France, 2005.