2

Constraint Based Automated Multi-attribute Negotiations

Miguel A. López-Carmona, Iván Marsá-Maestre and Juan R. Velasco
Universidad de Alcalá
Spain

1. Introduction

Multi-attribute (Multi-issue) negotiation protocols have been studied widely, and represent an important challenge in the multiagent systems community (Lai et al., 2004). Therefore, a lot of automated negotiation models and protocols have been developed, and manifold negotiation challenges have been already addressed. Most research in automated negotiation to date has focused on the competitive aspect (Vo et al., 2007). On the other hand, work by Dispute Resolution theorists in the social sciences has also focussed substantially on how to achieve negotiated agreements that are of a high value to all parties (Fischer & Ury, 1981). This approach is known as *Integrative* or *Interest-based negotiation*, and it has been recognised as the more successful approach to the negotiation problem. Example scenarios where such cases may arise are: business process management involving agents within the same organization, e-commerce negotiations where the seller is interested in having a satisfied buyer (e.g. long-term commercial relationships), or e-commerce scenarios where risk averse agents avoid the conflict in the negotiation processes. In the context of purchase negotiation scenarios, it is clear that every negotiation partner tries to maximize his preferences. However, when an agent aims at optimizing his own benefit with no regard for the others', it has been shown that negotiators more often than not reach inefficient compromises. Conflict theorists Lax and Sebenius (Lax & Sebenius, 1992) argue that negotiation necessarily includes both cooperative and competitive elements, and that these elements exist in tension. Therefore, he refers to the problem of deciding whether to pursue a cooperative or a competitive strategy at a particular time during a negotiation as the Negotiator's Dilemma. However, it is not always possible to separate the integrative bargaining process, i.e. when agents use cooperative strategies to search for joint gains, from the distributive bargaining process, i.e. when agents use competitive strategies in order to 'claim value'. The main problem is that distributive and integrative processes interplay with each other making information manipulation becomes part of the integrative bargaining process. *Integrative negotiation* contrasts with *distributive bargaining* in which the parties are trying to distribute a fixed resource, and where if an agent wins another agent looses. Distributive negotiation predicts that one party can only gain at the other party's expense. The key characteristics that distinguish integrative negotiations from distributive ones are: creation of value; focus on interests and not positions; openness and exchange of relevant

Source: Multiagent Systems, Book edited by: Salman Ahmed and Mohd Noh Karsiti,
ISBN 978-3-902613-51-6, pp. 426, February 2009, I-Tech, Vienna, Austria

information, and even learning; and problem restructuring. In order to achieve integrative approaches, literature of automated negotiation proposes a number of techniques such as *multi-attribute utility theory*, *distributed constraint satisfaction*, and *cojoint analysis*. A common aspect in all these techniques and in integrative negotiation approaches in general is that a multi-attribute negotiation scenario is required. Attributes are the characteristics of the negotiation item that are taken into account during the evaluation. The idea is that it may be beneficial for people to introduce multiple issues in a negotiation when they have different preferences over these issues because it may be possible to trade off one issue for another in order to reach agreements where both the negotiators are better off. So, in multi-attribute negotiations the parties involved need to settle more than one issue. For example, agents may need to come to agreements that are characterized by attributes such as price, quality, delivery time, and so on. If the impact of the issues under negotiation over the satisfaction function is different for each agent (that is, some issues are more important for a participant than for the others and vice versa), the issues may be traded-off against one another, increasing the social welfare of the deal.

Both in single and multi-issue negotiations the outcome depends on four key factors (Fatima et al., 2006): the *negotiation protocol*, the *participant's strategies*, the *players' preferences* over the possible outcomes, and the *information that the participants have about the others*. However, in multi-issue negotiations appears an additional factor: the *negotiation procedure*, which specifies how the issues will be settled. There are three ways of negotiating multiple issues: *Package deal* which links all the issues and discusses them together as bundle, *Simultaneous negotiation* which settles the issues simultaneously, but independently, and *Sequential negotiation* which negotiates the issues sequentially one after another. This chapter will focus on a package deal based procedure.

As we pointed out before, many automated negotiation models have been developed. They may be classified regarding many different criteria (Buttner, 2006). Regarding their theoretical approach, *game theoretic*, *heuristic* and *argumentation-based* approaches exist. The game-theoretic approach tries to find optimal strategies by the analysis of the equilibrium conditions (Nash, 1950). Game-theoretic models are deemed mathematically elegant, but are very restricted in use because of their assumptions of unlimited resources, perfect rationality and a perfect information situation. In heuristic approaches the mentioned assumptions are relaxed, and players try to find an approximate solution strategy according to principles of bounded rationality by utilizing heuristic search and evaluation techniques (Faratin et al., 1998; Ehtamo et al., 1999; Faratin et al., 2002; Klein et al., 2003; Gatti & Amigoni, 2005; Lai et al., 2006; Ito et al., 2008;). Both in game-theoretic and heuristic approaches, negotiation protocols are usually based on the communication of offers in the form of potential agreements. In contrast, in argumentation-based negotiations, the agents are able to reason their positions including a meta-level component that may use promises, rewards, threats, as well as issue various forms of appeal (Rahwan et al., 2003). In addition to the theoretical approach criterion, negotiation can be classified regarding its structure, regarding the negotiation process, and regarding the restrictions over time and information situations.

In addition to the problem of selecting the optimal strategies in the negotiation processes, the agent's decision making mechanisms in multi-attribute negotiations have to face the problem of characterize the preference on all attributes. The characterization of preferences has a critical influence on the negotiation protocols and decision-making mechanisms. To end up with this introduction we briefly review some of the most relevant approaches in

negotiation to model preferences, and pick up one of them to propose a multi-attribute negotiation protocol that will be presented in the following sections.

A typical way to model preferences is to use *utility functions*. In the case of multiple attributes, we talk about *multi-attribute utility theory (MAUT)*. Another approach to model preferences is to employ *multi-criteria decision making (MCDM)* (also called *multi-objective* or *multi-criteria optimization*) theory. In MCDM an agent has several objectives that are statements that delineate the desires of a decision maker. Thus, an agent wishes to maximise his objectives, which in some cases will conflict which each other in that the improved achievement with one objective can only be accomplished at the expense of another. Given an assignment of values to the corresponding attributes an agent measures how much the different objectives are fulfilled. Finally, a utility function is applied over the set of different levels of satisfaction of the agent's objectives. Research on those topics is conducted mostly in the field of decision theory. In the negotiation models described in the literature which use the utility based approaches to the modelling of preferences, the negotiation protocols are based on the communication of offers and counteroffers expressed as an assignment of values to the corresponding attributes. This approach to negotiation is known as *positional bargaining*, and is the predominant form of negotiation in the game-theoretic and heuristic approaches to negotiation. On the other hand, in argumentation-based negotiation the exchange of offers and counteroffers includes meta-information with the aim of reason the agents' positions. In the area of interest-based negotiation, another way to modelling preferences is to use *constraints* to restrict the attribute values that are preferred. *Constraints* in different formats, from *fuzzy* to *probabilistic* or *weighted constraints*, have been used in several models and approaches to multi-attribute negotiation (Luo et al., 2003; Lai & Lin, 2003; Ito et al., 2008). There are three main reasons that make very convenient the use of constraints as the core of a negotiation model. First, it is an efficient way of capturing requirements; second, constraints are capable of representing trade-offs between the different possible values for attributes; and third, using constraints to express offers in turns means that the solution space can be explored in a given exchange and so means that the search for an agreement is more efficient than in positional bargaining. The negotiation framework presented in this chapter falls within the heuristic approaches to non-mediated multi-attribute bilateral negotiations under incomplete information settings, and uses fuzzy constraints to model agent’s preferences. With incomplete information we mean that agents lack information about other's discounting factors, reservation prices, utility functions or deadlines, and with non-mediated we mean that agents negotiate without the intervention of a mediating agent. The negotiation model is based on the hypothesis that by means of an expressive approach to constraint based negotiation the negotiation processes may be more efficient than with other approaches where mainly positional bargaining is used. Behind this is the idea that with the cost of a bounded increase in the revelation of private information, the decision mechanisms are more accurate when searching the negotiation space.

The remainder of this chapter is organized as follows. The next Section recalls the most relevant concepts on modelling agent’s preferences and presents some preliminaries. Section 3 presents an example negotiation scenario where two different negotiation techniques are applied in order to show the possible advantages of expressive negotiation. Then the negotiation framework followed by an empirical evaluation is described. Finally, Section 6 presents the conclusions.

2. Modelling agent's preferences

A multi-attribute negotiation can be seen as a distributed *multi-objective optimization problem*. The participants in a negotiation have their own preferences over the negotiated attributes, and these preferences can be formulated in its most extensive form as a multi-objective or multi-criteria decision making problem. By definition, objectives are statements that delineate the desires of a decision maker. Thus, an agent wishes to maximise his objectives. However, it is quite likely that a decision maker's objectives will conflict with each other in that the improved achievement with one objective can only be accomplished at the expense of another. Therefore, a negotiator agent has to settle at a compromise solution. This is the topic of the *multi-criteria decision making theory*. Part of the solution to this problem is that the agent has to identify or approximate the *Pareto frontier* in the *consequence space* (i.e. in the space of the satisfaction levels of the different objectives). This task can be accomplished using different methods based on standard optimization techniques. Regarding the negotiation process it can be seen as a special case of multi-objective optimization problem. In this case, we have a set of distributed agent's objectives that should be satisfied. Each agent's objective depends on his individual objectives. The question now is if we can compute the Pareto frontier in a similar way. Assuming a set of agents which formalize their preferences as a multi-objective decision making problem, and that each agent computes his Pareto frontier, the only way to solve this problem in a similar way would be to share this information to formulate the global multi-objective optimization problem. In practice, this could be done by means of a trusted mediator, but it has a fundamental problem, agents and humans try to minimise the revelation of private information in negotiation to avoid strategic manipulation. Moreover, though Pareto optimality is a key concept in multi-objective optimization, we cannot forget that the aim of the negotiation is to reach an agreement, and so, it is necessary to pick up a fair solution from the Pareto frontier. However, fairness is not an easy concept to manage in negotiations.

2.1 Multi-attribute decision problems

As we stated before, negotiator agents are decision makers, and their decisions are based on preferences over the values of the different attributes. Formally, a *Multi-Attribute Decision Problem (MADP)* is defined as a set of attributes $X = \{x_1, ..., x_n\}$; a set of domain values $D = \{D_1, ..., D_n\}$ where each D_i is a set of possible values for attribute x_i; a set of constraints $C = \{C_1, ..., C_m\}$ where each C_j is a constraint function on a subset of attributes to restrict the values they can take; a set of available outcomes $O = \{o_1, ..., o_l\}$ where each o_j is an element of the possible outcome space D, and O is a subset of D; and a set of decision maker's preference statements $P = \{P_1, ..., P_m\}$. Agents negotiate over the same set of attributes and domain values, but each agent has a different set of constraints, available outcomes and preference statements. In a negotiation process, agents try to maximize their preferences, and in order to compute those values they have to solve the MADP. Among the different approaches to model agents' preferences from the MADPs perspective we survey two different categories of methods: the *constraint satisfaction problem (CSP) framework*, and the *multi-attribute utility theory (MAUT)*. For a detailed survey including more methods on MADPs see (Zhang & Pu, 2005).

A **CSP** is defined by a 3-tuple <*X*,*D*,*C*>, where *X* is a set of variables, *D* is a set of domains and *C* is a set of constraints. A solution to a CSP is a set of value assignment

$v = \{x_1 = v_1, ..., x_n = v_n\}$ where all constraints in *C* are satisfied. Therefore, the constraints are crisp (hard) since they are either respected or violated. A number of different approaches have been developed for solving this problem. One simple approach is to simply *generate-and-test*. However, when the CSP is complex the algorithm is not practical due to the computational complexity. A more efficient method is the *backtracking algorithm* that essentially performs a depth-first search of the space of potential CSP solutions. However, the complexity of backtracking for most nontrivial problems is still exponential. Other search algorithms for classical CSPs include: *forward checking*, *partial lookahead*, *full lookahead*, and *really full lookahead*.

We can see how a solution of a classical CSP needs to satisfy all the crisp constraints. Comparing the definition of classical CSP and MADP we can see that the main difference between them is that the MADP has a set of preferences, some of which can be violated when finding the optimal solution. Classical CSPs have been extended to *soft CSPs* in which not all the given constraints need to be satisfied. In the following, we recall several kinds of soft CSPs and a general framework which describes both classical and soft CSPs.

Fuzzy CSPs (FCSPs) extend the hard constraints by *fuzzy constraints*. A fuzzy constraint is a mapping from the direct product of the finite domain of the variables referred by the constraint to the [0,1] interval. The solution of a fuzzy CSP is the set of n-tuples of values that have the maximal value. The value associated with each n-tuple is obtained by minimizing the values of all its sub-tuples. An FCSP can be solved in a similar way as classical-CSP turning all fuzzy constraints into hard constraints.

Probabilistic CSPs (PCSPs) model those situations where each constraint *c* has a certain independent probability *p(c)* to be part of the given real problem. Let v be an n-tuple value set, considering all the constraints that the n-tuple violates, we can see that the probability of n-tuple being a solution is $\prod_{\text{all c that v violates}} (1 - p(c))$. The aim of solving PCSPs is to get the n-tuple with the maximal probability. The main difference between FCSPs lies in the fact that PCSPs contain crisp constraints with probability levels, while FCSPs contain non-crisp constraints. Moreover, the criteria for choosing the optimal solutions are different.

Weighted CSPs (WCSPs) allow to model optimization problems where the goal is to minimize the total cost of a solution. There is cost function for each constraint, and the total cost is defined by summing up the costs of each constraint. Usually WCSPs can be solved by the *Branch and Bound algorithm*.

A semiring-based CSP framework describes both classical and soft CSPs. In this framework, a semiring is a tuple *(A,+,x,0,1)* such that: *A* is a set and $0,1 \in A$; + is a close, commutative, and associative operation on A and 0 is its unit element; x is a closed, associative, multiplicative operation on A; and 1 is its unit element and 0 is its absorbing element. Moreover, x distributes over +. A c-semiring is a semiring such that + is idempotent, x is commutative, and 1 is the absorbing element of +.

Both the classical CSPs and the different type of soft CSPs can be seen as instances of the semiring CSP framework. The classical CSPs are Semiring-CSPs over the semiring $S_{CSP} = (\{false, true\}, \vee, \wedge, false, true)$ which means that there are just two preferences (*false* or *true*), that the preference of a solution is the logic *and* of the preferences of their subtuples in the constraints, and that *true* is better than *false*. FCSPs can be represented by $S_{FCSP} = ([0,1], \max, \min, 0, 1)$ which means that the preferences are over [0,1], and that we want to maximize the minimum preference over all the constraints. Similarly, the semiring

corresponding to a PCSP is $S_{PCSP} = ([0,1], \max, \times, 0, 1)$, and the WCSPs can be represented by the semiring $S_{WCSP} = (R^+, \min, +, +\infty, 0)$.

Utility theory and MAUT has been used in solving decision problems in economics especially for those involving uncertainty and risk. Given the utility function, the decision maker's preferences will be totally determined, and the optimal solution will be the outcome with the maximal utility. When using MAUT to solve a multi-attribute decision problem that only involves certainty, the main task is to assess the value function according to the decision maker's preferences.

Let $O = \{O_1, ..., O_n\}$ be a set of outcomes of the MADP, $\mathfrak{A}$ be the set of all lotteries on the set O where $\sum p_i o_i \in \mathfrak{A}$, $p_i \in [0,1]$, and $\sum p_i = 1$; and $\succsim$ be a binary relation on $\mathfrak{A}$. First we define 4 axioms: 1) $\succsim$ is complete, i.e. either $x \succsim y$ or $y \succsim x$; 2) $\succsim$ is transitive, i.e. if $x \succsim y$ and $y \succsim z$, then $x \succsim z$; 3) Continuity: given $x \succ y \succ z$, then there is an $\alpha, \beta \in (0,1)$ such that $\alpha x + (1-\alpha) z \succ y$ and $y \succ \beta x + (1-\beta) z$; 4) Independence: for all $x, y, z \in \mathfrak{A}$ and any $\alpha \in [0,1]$, $x \succsim y$ if and only if $\alpha x + (1-\alpha) z \succsim \alpha y + (1-\alpha) z$. Then the *von Neumann Morgenstern Theorem* proved the existence of utility function theoretically provided that the relation $\succsim$ satisfies the four axioms: Let $\mathfrak{A}$ be a convex subset of a linear space, and let $\succsim$ be a binary relation on $\mathfrak{A}$, then $\succsim$ satisfies the four axioms if and only if there is a real-valued function $u : \mathfrak{A} \to \Re$ such that:

a. $\forall x, y \in \mathfrak{A}, x \succsim y \Leftrightarrow u(x) \geq u(y)$;

b. $\forall x, y \in \mathfrak{A}$ and $\forall \alpha \in (0,1), u(\alpha x + (1-\alpha) y) = \alpha u(x) + (1-\alpha) u(y)$.

The function u is called the *utility function*.

Keeney and Raiffa (Keeney & Raiffa, 1976) extended the utility theory to the case of multi-attributes. Multi-attribute utility theory is concerned with the valuation of the consequences or outcomes of a decision maker's actions. For a decision problem where each action has a deterministic outcome, the decision maker needs only to express preferences among outcomes. The preference relation can be captured by an order-preserving, real-valued value function. Then, the optimal problem of the multi-attribute decision problem can be converted into the format of the standard optimization problem to maximize *u(x)*. When there is uncertainty involved in the decision problem, the outcomes are characterized by probabilities. It must be noted that a utility function is a value function, but a value function is not necessarily a utility function. In the case that only certainty is involved, the utility and value function are interchangeable.

3. A non-mediated bilateral negotiation model based on fuzzy constraints

Here we propose a non-mediated fuzzy constraint based negotiation framework for competitive e-marketplaces in which multiple buyer agents negotiate bilaterally with multiple seller agents to acquire products. In competitive markets, there is an inherent need to restrict the amount of private information the agent reveals. However, this restriction can have a detrimental effect on the search for a solution. As we stated above, especially in the case of multi-attribute negotiations, it is possible to reach a more satisfactory agreement by means of an adequate combination of attributes or constraints. However, most solutions put forward to tackle this problem are mediated, iterative and approach mechanisms, which are

applicable to preference models based on linear-additive or quasi-concave utility functions (Ehtamo et al., 1999; Faratin et al., 2002; Lai et al., 2006). Other approaches based on non-linear utility spaces include a mediator in the negotiation processes (Klein et al., 2003; Gatti & Amigoni, 2005; Ito et al., 2008). As an alternative to these solutions, we propose one based on the concept of communicative rationality rather than one which is merely strategic and retains as the fundamental criteria the minimization of private information revealed. Our solution is therefore based on a dialogue of offers in which preferences or satisfaction degrees are partially disclosed. The hypotheses on which the work is based is that of an interactive model which is sufficiently expressive to allow a discussion of proposals by means of a partial declaration of preferences which permits the agents to reach a more satisfactory agreement, being confined to the need to minimize the loss of privacy. The negotiation framework is defined by: a fuzzy constraint based model of preferences; the expressive behaviors and strategies of the agents; an interaction model that permits the automatic generation of expressive or non-expressive dialogues with different degrees of symmetry; and finally a set of decision mechanisms adapted to the interaction model and the preferences of the agents.

There are several works using fuzzy constraints to model preferences, however, most of them use single point offers (i.e. positional bargaining). The FeNAs (Fuzzy e-Negotiation Agent system) platform (Kowalczyk & Bui, 2000) uses fuzzy constraints and permits correlated multiple bilateral negotiations. It is one of the first works in which the problem of multi-attribute negotiation is clearly presented using a preference model based on FCSP. The main problem with FeNAs resides in its being a positional approach. Lai (Lai & Lin, 2004) presents a general framework for multi-attribute and multilateral negotiation based on fuzzy constraints. The negotiation model is based on FCSP, which when applied to a distributed domain of agents is organized as a network of distributed fuzzy constraints (DFCN). This work makes some very important contributions to the regularization of the mechanisms for calculating the satisfaction degree and to the available concession and compensation strategies. It introduces fuzzy logic techniques to the relaxation decision making area that allow concession strategies to be defined that are a function of the beliefs and desires of the agents. The model is also based on single-point offers and there is no argumentation, but decision-making is based on the behavior of the opponent and the type of offers received. In accordance with the mentioned above procedures, if there is no convergence in the first relaxation steps, the number of offers increases exponentially. If there are a large number of attributes, the number of possible proposals for a particular cut level becomes intractable. Although the similarity function can help with convergence, a certain amount of knowledge of the utility functions of the opponent is assumed. Finally, Luo (Luo et al., 2003) develops a fuzzy constraint based model for bilateral multi-issue negotiations in semi-competitive environments. It uses crisp constraints to express offers and includes the idea of rewards and restrictions. The most noticeable aspects are related to the acceptability function and with the operators used to apply the prioritization of the fuzzy constraints. Assuming the seller agents' dominant strategy is to offer the first product that satisfies the constraints, the model isn't efficient enough because it exhibits a large lack of symmetry. In this model a buyer agent has a great communication power (expressing offers by means of constraints) while the seller agent can only offer specific products or request a relaxation of the constraints. In this way, the opportunity to apply some form of solution compensation technique so that a win-win solution is obtained is lost.

3.1 Expressive vs inexpressive negotiation dialogues

In this subsection a bilateral negotiation scenario is presented, comparing two approaches, one expressive and the other non-expressive, in which all the advantages that our approach contributes to the problem will be discussed.

A buyer agent and a seller agent begin a negotiation dialogue about the sale of a vehicle. The buyer agent expresses a desire to buy in the following way: "*I want to acquire a car at a low price, of high quality and as new as possible*". From this statement, it can be taken there are three issues that are of interest to the buyer agent, the *price*, the *quality* and the *age* of the car. The requirements of the buyer agent are therefore defined by these three fuzzy constraints, so that a priori, no specific range is defined for each issue to determine whether a constraint has been satisfied. In the seller agent's case we could propose a formulation of preferences or sale needs in a similar way, however, in trading scenarios the seller agent may be more inclined towards the use of catalogues of products. In Figures 1 and 2, the buyer agent's preferences and a summary of the seller agent's catalogue are shown respectively. The labels above each step represent the range of the attributes value domain, in such a way that the states can appear as intervals, numeric groups or as linguistic terms. The higher steps represent greater satisfaction degrees. If we analyse the diagram we can see that, for

Fig. 1. Buyer agent's preferences

Product	price	quality	age	utility
p1	Very low	Very low	2006	Very low
p2	Very high	Very high	2006	Very high
p3	Low	High	2004	Medium
p4	Low	Medium	2005	Very low
...				
p_n				

Fig. 2. Seller's catalogue of products

example, the fuzzy constraint expressed as low price is divided in intervals in accordance with the different satisfaction degrees of the buyer agent. The catalogue of products is defined by a series of rows each one of which characterizes a product. For each product, the satisfied range of values of the buyer agent's attributes is shown. The last column represents the utility the seller agent obtains if the product is sold. This utility value does not have to have any direct correlation with the negotiable attributes, there may exist other private issues (non negotiated) that have a greater influence on the utility value.

To give an example of our working hypothesis we first present a possible negotiation dialogue between a buyer and seller agent (see Figure 3) that we will call *non-expressive*. In this type of dialogue the argumentation capability with respect to the offers is minimal. The buyer agent makes offers in the form of crisp constraints taken strategically from the fuzzy constraints that represent its overall requirements. On the other hand, the seller agent is only able to accept or reject an offer. So, we see in the example that the buyer agent successively relaxes its demands, as after each offer the seller agent responds with a refusal (as it does not have products that satisfy the constraints). Finally, in the last stage, the seller agent finds a product p4 that satisfies the buyer agent's requirements. However, this solution provides a very low profit for the seller agent. It is clear that the negotiating position of the buyer agent is much stronger, their requirements are described in detail in each offer, and at no time does the seller agent give any clue as to its preferences. The limitations of the language used mean that the only possible criteria that can be used to find solutions are local preferences. The question we must ask ourselves is whether there exists a solution that would have been more satisfying for the seller agent without worsening the

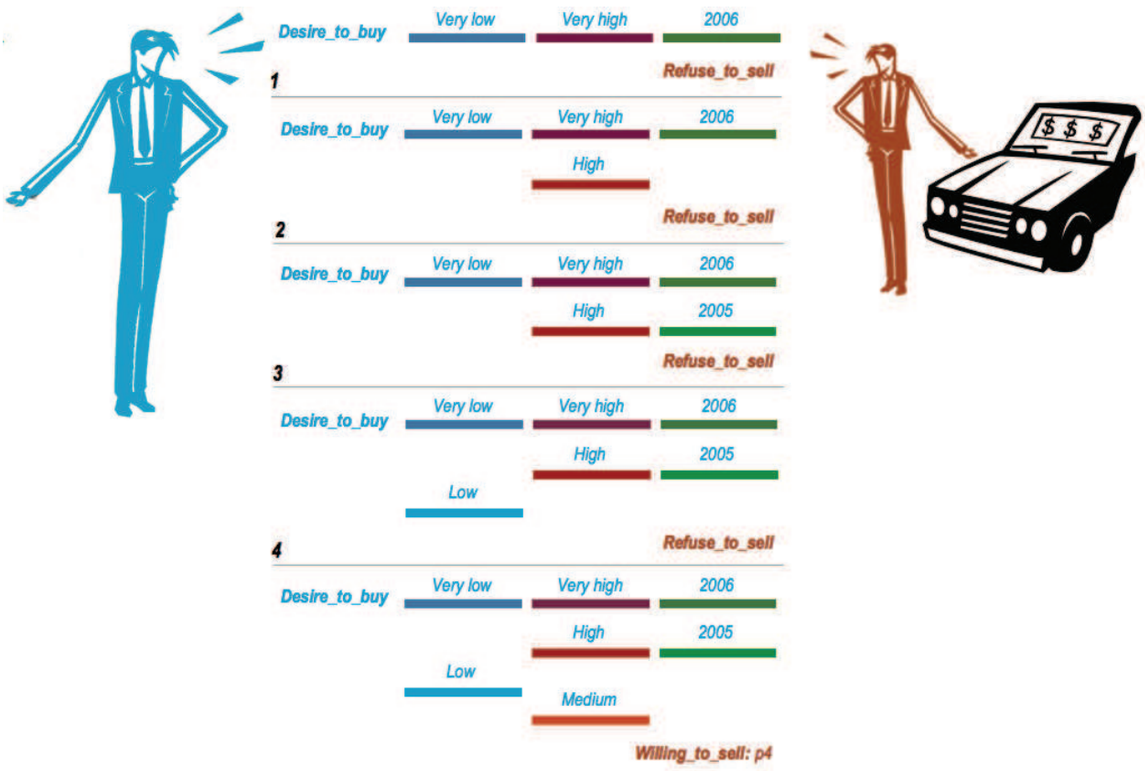

Fig. 3. Example of non-expressive dialogue

buyer agent satisfaction degree, and the answer rests in the solution p3, which would indeed have been more satisfactory for the seller agent without being less so for the buyer agent.□□As an alternative, we now present a new dialogue, which we term *expressive*, in which the concepts that form the basis of our hypothesis are applied.□□In Figure 4, the buyer agent and the seller agent negotiate the purchase of an automobile under the same preference conditions used in the previous dialogue. In this dialogue two important innovations appear: Firstly, the buyer agent is able to subjectively value its offers; and secondly, the seller agent is able to clarify its refusal to offer a product, by using expressions that allow it to state which constraints it wants the buyer agent to relax.

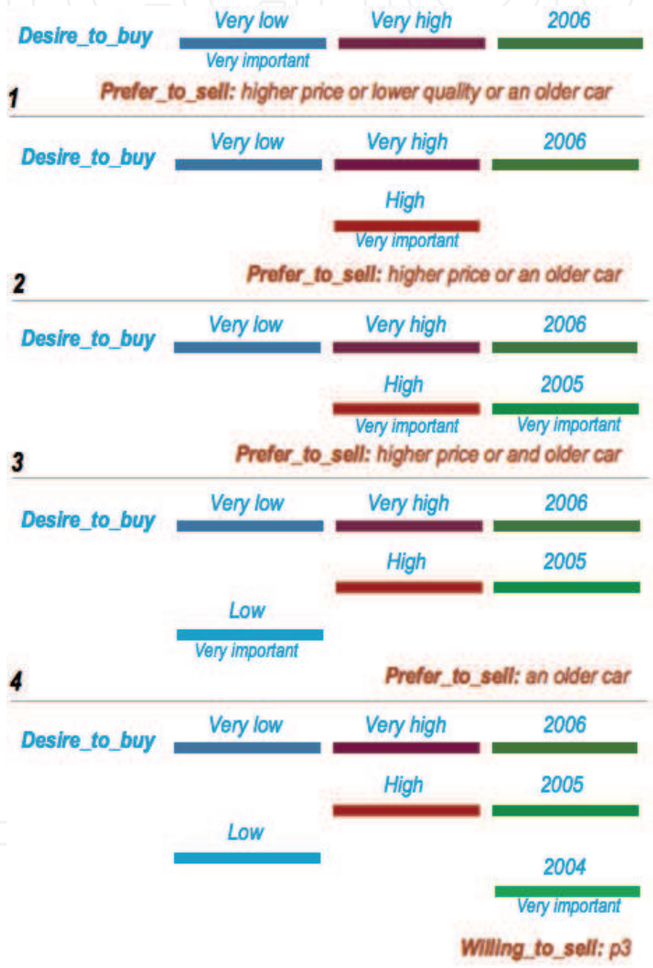

Fig. 4. Example of expressive dialogue

We will now analyse the course of the dialogue. □□

1. The first offer made by the buyer agent is the one that subjectively offers it the greatest satisfaction. Apart from the offer, defined as a set of crisp constraints, these constraints contain meta-information that grades them depending on the degree of importance each of them has. Thus, the constraint Very Low is considered as very important and it is expressed like this in the dialogue. The seller agent does not have a product that satisfies all the constraints, so it has no choice but to refuse the offer. However, it argues

its refusal with an attack based on preferences, suggesting that the buyer agent relax constraints with differing degrees of preference. From the seller agent point of view, any of the constraints in the initial dialogue can be relaxed.□

2. The buyer agent's second offer involves relaxing the quality constraint. As the seller agent had no preference for which constraint should be relaxed, the buyer agent relaxes at random one of the constraints (quality or age) that least affects its satisfaction degree. The quality constraint now becomes the buyer agent's choice, because to do so later would involve a greater loss of satisfaction than the relaxation of any other constraint. When the seller agent receives the offer, it is unable to find a product that satisfies all the constraints. However, it concludes that products p2 and p3 come close to the buyer agent's requirements. To be precise, the seller agent reasons in the following way: p2 will provide me with more profit, but on the other hand, although p3 will provide me with slightly less profit, it is closer to the buyer agent's requirements. After the seller agent has made the previous reasoning, it tries to persuade the seller agent by first asking it to relax the price and age constraints.□□
3. The third stage of the negotiation follows similar parameters to the previous one.
4. In the buyer agent's fourth offer, the price constraint is the most important. The seller agent analyses its catalogue and rejects p1 because of its low utility and estimated distance. With regards to p2, it decides that it satisfies the age and quality constraints, and that p3 satisfies the price and quality constraints, and finally, that p4 satisfies the price and age constraints. A priori, the three products are relatively close to the buyer agent requirements, but the description of the price constraint as very important affects the estimation of the closeness or distance of p2. The distance of products p3 and p4 is estimated to be similar, so the buyer agent discriminates depending on the utility of the solutions. The conclusion is that the seller agent decides that p3 is the best possible offer. He then puts all its effort into ensuring the sale of p3, although it does not satisfy the age constraint, which is why the request to relax concentrates on this constraint. □□
5. After receiving the request to relax, the buyer agent finds that a priori, it has no problem with relaxing either the quality or the age constraint. Under the assumption of negotiation based on interests or principles, the buyer agent accepts the request to relax the age constraint. The seller agent has a product, p3 that satisfies the present requirements. The overall satisfaction of the solution is greater than in the case with non-expressive negotiation. □□

The challenge of developing all the concepts in the example involves several aspects. Firstly, an agents' preference model formalization. Secondly, a definition of the negotiation profile for modelling the agent's behaviour towards their opponents. Creation of a communication model that, amongst other things, details the locutions needed to be able to deal with all the expressive nuances. Development of decision making mechanisms. Finally, a working language specification allowing the decision mechanisms to be linked to the expressions available to the agents.

4. Negotiation framework

The negotiation framework consists of a description of the *agent's domain knowledge*; a *dialogue model*; the *decision mechanisms*; and the *transition rules* that connect the locutions to the mechanisms.

4.1 Agent's domain knowledge

Buyer agent's requirements over the attributes of a product are described by means of a *fuzzy constraint satisfaction problem (FCSP)*, which is a 3-tuple (X, D, C^f) where $X = \{x_i \mid = 1,...,n\}$ is a finite set of variables, $D = \{d_i \mid = 1,...,n\}$ is the set of finite domains of the variables, and $C^f = \{R_j^f \mid j = 1,...,m\}$ is a set of fuzzy constraints over the variables. It is worth noting that a fuzzy constraint may restrict more that one variable or attribute. A fuzzy constraint corresponds to the membership function of a fuzzy set. The function that numerically indicates how well a given constraint is satisfied is the satisfaction degree function $\mu_{R_j^f} : X \to [0,1]$, where 1 indicates completely satisfied and 0 indicates not satisfied at all. Given the *cut level* $\sigma \in [0,1]$, the induced crisp constraint of a fuzzy constraint R^f is defined as R^c. It simply means that if R^c is satisfied, the satisfaction degree for the corresponding fuzzy constraint will be at least σ. Therefore, the *overall (global) satisfaction degree (osd)* of a given solution $x' = (x_1', ..., x_n')$ is:

$$\alpha(x') = \min\{\mu_{R^f}(x') \mid R^f \in C^f\} \tag{1}$$

On the other hand, a seller agent owns a private catalogue of products $S = \{s_k \mid s_k = (p_k, u_k)\}$, where p_k is the vector of attributes and u_k is the profit the seller agent obtains if the product is sold. We assume that the profit u_k may depend not only on the negotiated attributes but also on non-negotiated ones (stock period for instance).

Let A_b and A_s represent a buyer and a seller agent, a negotiation process is a finite sequence of alternate proposals from one agent to the other. During the negotiation stage, A_b utters *purchase requirements*,

$$\pi = \bigcap \left\{ R_j^{c(\sigma_j)} \mid j \in [1,m] \right\} \tag{2}$$

where $R_j^{c(\sigma_j)}$ is a crisp constraint induced from R_j^f at a cut level σ. Therefore, a purchase requirement is a purchase proposal that is formed by a set of crisp constraints extracted from the set of fuzzy constraints that describes the buyer's preferences regarding the attributes of the products. Each crisp constraint in the purchase requirement can be induced at a different cut level. Complementing the *osd* definition, the *potential or expected overall satisfaction degree (posd)* is the osd that a buyer agent may get if the corresponding purchase requirement is satisfied. It is defined as:

$$\alpha^\pi = \min\{\sigma_i \mid i = 1,...,m\} \tag{3}$$

A seller agent may respond to a buyer agent in three different ways: *rejecting the proposal*, *offering a product* that satisfies the purchase requirement, or *suggesting the relaxation* of the purchase requirement. A *relaxation requirement* is defined as a set:

$$\rho = \{r_j \mid r_j \in [0,1]\} \tag{4}$$

where ρ_j is the preference for constraint j to be relaxed. The negotiation process and the agreements achieved will mainly vary depending on the strategies followed by the agents when generating purchase requirements and when requesting its relaxation. We cover all

these aspects modeling the *agents' attitudes*. Agents' attitudes are related to the agents' strategic behavior in the negotiation process, where strategic behaviors are described in terms or expressiveness and receptiveness. A negotiation profile $\Pr ofile_{\text{seller}} = \{\psi, \beta\}$ describes the seller agent's attitude, where $\psi \in \{0,1\}$ controls whether it uses or not relaxation requests in order to express its preferences for a specific relaxation of the previous buyer's demands, and $\beta \in [0,1]$ modulates its attitude regarding a purchase requirement received from a buyer agent. Finally, a negotiation profile $\Pr ofile_{\text{buyer}} = \{\xi, \eta\}$ describes the buyer agent's attitude, where $\xi \in \{0,1\}$ controls whether it uses or not *purchase requirement valuations* defined as:

$$v = \{v_j \mid v_j \in [0,1]\} \tag{5}$$

where v_j is the degree of importance that the constraint j has for the buyer agent, and $\eta \in [0,1]$ modulates its attitude regarding a relaxation requirement received from a seller agent.

4.2 Negotiation dialogue

The framework of *formal dialogue games* is increasingly used as a base for structuring the interactions of agents communication protocols (McBurney et al., 2003), adopted from the theory of argumentation field. Formal dialogue games are those in which two or more players pronounce or transmit locutions in accordance with certain predetermined rules. In our negotiation model all dialogues are confined to two agents, one the buyer and the other the seller, so that the dialogues are exclusively bilateral. A dialogue is structured in accordance with the following stages:

1. *Opening* the dialogue.
2. *Negotiation*: this stage is defined by a sequence of iterations that are based on the domain knowledge mentioned earlier. These iterations are now itemised:
   - Buyer agent:

- Transmit purchase requirements.
- Transmit valuation of purchase requirements.
- Reject sale offers.

   - Seller agent:

- Transmit sale offers.
- Rejects purchase requirements.
- Propose the relaxation of purchase requirements.
- Reject purchase obligations.

3. *Confirmation*: the participants come to a compromise and reach an agreement.
4. *Close* of dialogue: the dialogue ends.

Our dialogue proposal is subject to the following rules:

a. The first stage in the dialogue is Opening of the dialogue.
b. The Opening and Closing stages of the dialogue can only occur once in the whole dialogue.
c. The only stages that must appear in all dialogues that end normally are Opening and Closing of the dialogue.
d. The Confirmation stage requires the negotiation stage to have occurred previously.
e. The last stage of all dialogues that end normally is Close of dialogue.

The participants can commute between the negotiation and confirmation stages, subject only to the rules and the constraints defined by the combination of locutions rules, which we describe later.

Our purchase negotiation dialogue is defined as sequence of four stages: **open dialogue** (L1-2), **negotiate** (L3-8), **confirm** (L9-10) and **close dialogue** (L11).

L1: open_dialogue (P_b, P_s, θ) P_b suggests the opening of a purchase dialogue to a seller participant P_s on product category θ. P_s wishing to participate must respond with *enter_dialogue*(.).

L2: enter_dialogue (P_s, P_b, θ) P_s indicates a willingness to join a purchase dialogue with participant P_b. Within the dialogue, a participant P_b must have uttered the locution *open_dialogue*(.).

L3: willing_to_sell (P_s, P_b, p_j) P_s indicates to the buyer P_b a willingness to sell a product. A buyer P_b must have uttered a *desire_to_buy*(.) or a *prefer_to_buy*(.) locution.

L4: desire_to_buy $(P_b, P_s, \pi_{\mathcal{B}_{req}})$ P_b, speaking to the seller P_s, requests to purchase a product that satisfies the purchase requirement $\pi_{\mathcal{B}_{req}}$.

L5: prefer_to_sell $(P_s, P_b, \pi_{\mathcal{B}_{req}}, \rho_{\mathcal{B}_{req}})$ P_s, speaking to the buyer, requests to relax the purchase requirement $\pi_{\mathcal{B}_{req}}$, and expresses which constraints are preferred to be relaxed, by means of the relax requirement $\rho_{\mathcal{B}_{req}}$.

L6: prefer_to_buy $(P_b, P_s, \pi^k_{\mathcal{B}_{req}}, \upsilon_{\mathcal{B}_{req}})$ P_b, speaking to the seller, requests to purchase a product which satisfies the purchase requirement $\pi^k_{\mathcal{B}_{req}}$, and expresses its preferences for the different constraints by means of the purchase requirement valuation $\upsilon_{\mathcal{B}_{req}}$.

L7: refuse_to_buy (P_b, P_s, p_j) Buyer agent expresses a refusal to purchase a product. This locution cannot be uttered following a valid utterance of *agree_to_buy*(.).

L8: refuse_to_sell $(P_s, P_b, p_j \mid \pi_{\mathcal{B}_{req}})$ Seller agent expresses a refusal to sell a product, or it expresses a refusal to sell products that satisfy the purchase requirement $\pi_{\mathcal{B}_{req}}$. This locution cannot be uttered following a valid utterance of *agree_to_sell*(.).

L9: agree_to_buy (P_b, P_s, p_j) Buyer agent P_b speaking to P_s commits to buy a product. A locution of the form *willing_to_sell*(.) must have been uttered.

L10: agree_to_sell (P_s, P_b, p_j) Seller agent speaking to buyer agent commits to sell a product. A locution of the form *agree_to_buy*(.) must have been uttered.

L11: withdraw_dialogue (P_x, P_y, θ) For P_x and P_y participants with different roles (i.e. sellers and buyers). P_x announces agent P_y the withdrawal from the dialogue.

Next step is to specify the mechanisms that will invoke particular locutions in the course of a dialogue.

4.3 Decision mechanisms

Syntactic rules are not enough to ensure that the dialogues are generated automatically. It is essential to equip each participant with mechanisms that allow it to invoke the correct locution at the right time, as a response to previous locutions or in anticipation of future ones. This type of mechanism we term *semantic decision mechanism*. The mechanisms are grouped together depending on the role of the participant: Buyer (B) or Seller (S). We will now describe each mechanism's general directive and then detail their specific features. In addition, we specify the output generated by the mechanisms, a key point for describing, in

the following Section, the working features or working semantics that connect the decision mechanisms and the locutions. We begin with the buyer agent's decision mechanisms.

B1: Recognize Need Allows a buyer agent to recognize the need to acquire a product. This recognition may be as a consequence of the explicit initiative of the user (e.g. through an interface the user gives an order to their personal agent of their intention to acquire a product), or it could be an automatic response based on thresholds that are triggered automatically (e.g. when a personal agent detects that it is within range of an electronic market that offers a particular type of product that falls within the preferences of the owner of the personal agent). When it detects the need and furthermore interprets that it is possible to begin a dialogue the mechanism's output is *have_need* (θ). Outputs: *wait*, *have_need* (θ), *have_no_need* (θ), where (θ) defines a *product category*.

B2: Generate Purchase Requirement This mechanism responds to a buyer agent's need to generate purchase requirements. Any purchase requirement must be compatible with the locution *desire_to_buy*(.) or *prefer_to_buy*(.). Two possible outputs are recognized, one that states that it is impossible to generate a requirement and another that specifies the requirement generated. Outputs: *empty_set* $\varnothing$, $\pi_{\mathcal{B}_{req}}$

The method for extracting crisp constraints directly affects the way a purchase requirement is accepted, and indirectly affects the potential overall satisfaction degree the buyer agent hopes to obtain. There are two possible strategies when extracting crisp constraints to satisfy a purchase requirement and generate a specific potential overall satisfaction degree:

(Concession strategy) Given a purchase requirement $\pi^{t}_{\mathcal{B}_{req}}$ sent at an instant $t \in \mathbb{N}$, a general concession strategy is defined as mechanism that generates a new purchase requirement $\pi^{t+1}_{\mathcal{B}_{req}}$ so that $\alpha^{\pi^{t+1}_{\mathcal{B}_{req}}} < \alpha^{\pi^{t}_{\mathcal{B}_{req}}}$ and $\alpha^{\pi^{t+1}_{\mathcal{B}_{req}}} \geq \alpha^{\pi^{t}_{\mathcal{B}_{req}}} - \varepsilon$, where $\varepsilon \in (0,1]$.

According to this definition, ε is an arbitrary value that fixes the maximum loss of potential overall satisfaction that the buyer agent is willing to accept when generating a new purchase requirement. It determines the agent's behaviour with respect to how rapidly it is willing to make concessions over its purchase requirements.

(Compensation strategy) Given a purchase requirement $\pi^{t}_{\mathcal{B}_{req}}$ sent in an instant $t \in \mathbb{N}$, a compensation strategy is a mechanism that generates a new purchase requirement $\pi^{t+1}_{\mathcal{B}_{req}}$ so that $\alpha^{\pi^{t+1}_{\mathcal{B}_{req}}} \geq \alpha^{\pi^{t}_{\mathcal{B}_{req}}}$.

We now move on to the specific mechanisms that put these strategies into practice. There are two ways to generate a new purchase requirement:

Adding a new fuzzy constraint. This way of generating purchase requirements is intended for two specific situations: the start of the negotiation, when the first purchase requirement should be prepared, and during the negotiation, after a sale offer that does not satisfy the constraints not included in the purchase requirement. In the first case, the agent selects a fuzzy constraint and applies the highest cut level to extract the corresponding crisp constraint and create the purchase requirement $\pi^{t}_{\mathcal{B}_{req}}$. By using this method, the agent is following the minimum revelation of information principle and the requirement obtained generates the greatest potential overall satisfaction degree. In the second case, a new constraint is selected from amongst those not satisfied by the sale offer received.

Modification of a previous purchase requirement. This way of creating a purchase requirement is intended for a specific situation: the locutions *prefer_to_sell*(.) or *refuse_to_sell*(.) sent by the seller agent during the negotiation with the intention of expressing its refusal to satisfy the buyer agent's requirements. Given a purchase requirement $\pi^t_{\mathcal{B}_{req}}$, and after receiving one of these locutions as a reply, the cut levels associated with the fuzzy constraints included must be changed and this change affects the potential overall satisfaction degree. Therefore, the generation of a new requirement $\pi^{t+1}_{\mathcal{B}_{req}}$ will be the aim of the application of one of the concession or compensation strategies and the problem is reduced to determining the plan for relaxing the previous purchase requirement $\pi^t_{\mathcal{B}_{req}}$. We propose the meta-strategy, which consists, when possible, of applying the compensation method and in its absence the concession method. The following algorithm implements the required function.

Algorithm 1. (*Modification of purchase requirements*)

1. Given a purchase requirement $\pi^t_{\mathcal{B}_{req}}$, a vector is obtained with the potential overall satisfaction degrees for all the possible purchase requirements resulting from the relaxation each time of only one of the constraints contained in $\pi^t_{\mathcal{B}_{req}}$:

$$[\alpha^{\pi^{(t+1)k_1}_{\mathcal{B}_{req}}} ... \alpha^{\pi^{(t+1)k_i}_{\mathcal{B}_{req}}}]$$

   where $\alpha^{\pi^{(t+1)k_x}_{\mathcal{B}_{req}}}$ represents the potential overall satisfaction degree obtained if the constraint $R^f_{k_x}$ is relaxed the minimum possible. The constraints that cannot be relaxed must be eliminated from the vector. If none of the constraints can be relaxed the function returns $\varnothing$.
2. The maximum of the previous vector is calculated:

$$\alpha^{t+1}_{max} = max([\alpha^{\pi^{(t+1)k_1}_{\mathcal{B}_{req}}} ... \alpha^{\pi^{(t+1)k_i}_{\mathcal{B}_{req}}}])$$

3. A new vector $\overrightarrow{\alpha^{t+1}_{max}}$ is generated in which only the elements that satisfy the following equality are included:

$$\alpha^{t+1}_{max} = \alpha^{\pi^{(t+1)k_x}_{\mathcal{B}_{req}}}$$

4. Finally, the following function is applied:

$$\arg\max_{\overrightarrow{\alpha^{t+1}_{max}}, \rho^t_{max}} \alpha^{\pi^{(t+1)k_x}_{\mathcal{B}_{req}}} + r_{k_x} * \eta$$

where ρ^t_{max} is a relax requirement from the seller agent, in which only those constraints included in the vector created in stage 3 are taken into account. If there are no relax requirements, r_{k_x} always takes the value 0. This function selects the constraint or constraints that maximize the total potential overall satisfaction that is induced if they are relaxed and of the relax requirement correspondingly weighted by the value η of the buyer agents receptive profile.

5. Once the constraint or constraints with the option of being relaxed are selected, one is chosen and a new purchase requirement is created $\pi_{\mathcal{B}_{req}}^{t+1}$, in which only the chosen constraint is modified.

The first three stages of the algorithm focus on the search for those constraints in $\pi_{\mathcal{B}_{req}}^{t}$ which if relaxed involve the smallest possible loss of potential overall satisfaction. Once these constraints have been detected, stage 4) takes into account only these constraints and, if there is a relax requirement, what the seller agent's preferences are in this respect. At one extreme if $\eta = 0$, the only criteria for relaxing is local, whereas if $\eta = 1$, the maximum importance possible is being given to the seller agent's recommendations. It is important to clarify that as we have defined in stage 2) the maximum value for filtering the potential satisfaction values, the function defined in 4) would only vary if r_{k_x} varies, being $\alpha^{\pi_{\mathcal{B}_{req}}^{(t+1)k_x}}$ a constant the same as α_{max}^{t+1}. However, we have decided to show the function in a more general form so we can easily extend the criteria of the maximum to other criteria.

B3: Generate Purchase Requirement Valuation This mechanism allows a valuation argument to be generated for a purchase requirement that has not yet been sent, i.e. a purchase requirement valuation $\upsilon_{\mathcal{B}_{req}}$. This can be communicated by the locution *prefer_to_buy*(.). The impossibility of obtaining a valuation generates the output an *empty_set*. Taking into account that the argumentation of a requirement is a reflection of the expressive character of the buyer agent, the mechanism will be controlled by its *expressive profile* ξ. If this has the value 1 the mechanism activates and tries to generate the valuation, if it has the value 0, the mechanism does not activate a valuation and returns an *empty_set*. When there are no valuations the buyer agent uses the locution *desire_to_buy*(.), whereas if there are valuations it uses the locution *prefer_to_buy*(.). Outputs: *empty_set* $\varnothing$, $\upsilon_{\mathcal{B}_{req}}$

A valuation of a purchase requirement is an expression of how important for the buyer agent the satisfaction of each of the purchase requirements constraints is. We propose the following algorithm.

Algorithm 2. (*Valuation of a purchase requirement*)

1. Given a purchase requirement, by sending $\pi_{\mathcal{B}_{req}}^{t+1}$, a vector is obtained with the potential overall satisfaction degrees for all the possible purchase requirements that result from relaxing only one of the constraints in $\pi_{\mathcal{B}_{req}}^{t+1}$ each time. The potential overall satisfaction degrees of those constraints that cannot be relaxed have the value 0:

$$[\alpha^{\pi_{\mathcal{B}_{req}}^{(t+2)k_1}} ... \alpha^{\pi_{\mathcal{B}_{req}}^{(t+2)k_i}}]$$

2. The elements of the previous vector are taken and a new standardized vector is defined that represents the valuation of the purchase requirement:

$$\upsilon_{\mathcal{B}_{req}} = [1-\alpha^{\pi_{\mathcal{B}_{req}}^{(t+2)k_1}} ... 1-\alpha^{\pi_{\mathcal{B}_{req}}^{(t+2)k_i}}] / sum([1-\alpha^{\pi_{\mathcal{B}_{req}}^{(t+2)k_1}} ... 1-\alpha^{\pi_{\mathcal{B}_{req}}^{(t+2)k_i}}])$$

The mechanism defines the valuation strategy of the purchase requirement as a strategy based on potential satisfaction degrees. To clarify which potential satisfaction degrees we are talking about we will describe a normal valuation process. When mechanism *B2:*

Generate Purchase Requirement is executed, a purchase requirement called $\pi_{\mathcal{B}_{req}}^{t+1}$ is generated from a requirement sent earlier, termed $\pi_{\mathcal{B}_{req}}^{t}$. To generate $\pi_{\mathcal{B}_{req}}^{t+1}$ the fictitious potential satisfaction degrees $\pi_{\mathcal{B}_{req}}^{t}$ have been used. However, in carrying out the valuation, the fictitious potential satisfaction degrees of the proposal $\pi_{\mathcal{B}_{req}}^{t+1}$, which has not yet been sent, are being generated. To sum up, what is being estimated is the potential satisfaction degree that could be obtained if the buyer agent requests another new requirement.
As we also take into account in the definition of the mechanism, the valuation of a constraint is contrary to the potential satisfaction degree that is obtained if it is relaxed. For this reason, we have chosen to carry out the operation $1-\alpha^{\pi_{\mathcal{B}_{req}}^{(t+2)k_x}}$ on each constraint. Finally, it is necessary to make clear that in the case of constraints that cannot be relaxed, by assigning a value of 0 to the potential overall satisfaction degrees in stage 1) a maximum valuation is obtained.
B4: Consider Offers This mechanism responds to the buyer agents need to decide at any given moment whether to: *accept or reject a sale offer* proposed by the seller agent, or *generate a new purchase requirement*. Sending a purchase requirement $\pi_{\mathcal{B}_{req}}^{t}$, a buyer agent accepts a sale offer p_j when the overall satisfaction degree $\alpha(p_j)$ is greater than or equal to the potential overall satisfaction degree of the purchase requirement $\pi_{\mathcal{B}_{req}}^{t+1}$. In this case the mechanism returns *accept_offer*(p_j). The acceptance of an offer opens the offer confirmation stage of the dialogue. If a sale offer cannot be accepted and the offer does not satisfy the constraints sent in $\pi_{\mathcal{B}_{req}}^{t}$, the mechanism returns *reject_offer*(p_j) indicating that a rejection expression must be generated. Finally, if the sale offer cannot be accepted, but satisfies the constraints sent in $\pi_{\mathcal{B}_{req}}^{t}$, the mechanism returns *generate_purchase_requirement*(p_j), indicating that a new purchase requirement must be generated.
Outputs: *accept_offer*(p_j), *reject_offer*(p_j), *generate_purchase_requirement*(p_j)
When a sale offer cannot be accepted, in one case the mechanism indicates that a new purchase requirement must be generated whilst in the other it indicates that a rejection expression must be generated. In both cases, the non-acceptance of the offer has its origin in the desired overall satisfaction degree not being reached. However, when the offer does not satisfy the constraints that have been sent we should interpret it as an anomaly. The seller agent may for example be sending offers indiscriminately. If however, a sale offer is not accepted, but satisfies the constraints that have been sent, this means that the sale offer does not adequately satisfy fuzzy constraints that have not been sent. In this case, we cannot assume that there is any anomaly in the seller agent's behaviour.
B5: Consider Withdrawal This mechanism responds to the buyer agent's need to decide at any given moment if it should terminate a dialogue with the seller agent. The mechanism can remain in wait mode and so returns *wait* or indicate whether the dialogue should be terminated in which case it returns *withdraw* (θ). Outputs: *wait*, *withdraw* (θ).
We now present the seller agent's decision mechanisms.
S1: Recognize Category Allows a seller agent to recognize the need to sell a product. The mechanism merely assures that the seller agent has available products of the *category* (θ) in

its catalogue. If the products are available the mechanism returns *wish_to_enter* (θ), and if not *wish_not_to_enter* (θ). The mechanism can remain in wait mode, and so returns wait. Outputs: *wait*, *wish_to_enter* (θ), *wish_not_to_enter* (θ)

S2: Assess Purchase_Requirement This mechanism responds to the seller agent's need to valuate purchase requirements. The main objective of a valuation is to detect the existence of products in the catalogue that satisfy the purchase requirements. If the products are not found, the mechanism decides whether to argue the rejection of the requirement received or not. This decision is function of the agent's expressive profile ψ, so that if it has value 1, the decision is to argue, and if it has value 0 not to argue. When there is a sale offer the mechanism returns *sale_offer*(p_j), if it decides to argue the purchase requirement it returns *purchase_requirement* ($\pi^t_{\mathcal{B}_{req}}$), and if it decides not to argue it returns an *empty_set*. Outputs: *empty_set* $\varnothing$, *purchase_requirement* ($\pi^t_{\mathcal{B}_{req}}$), *sale_offer*($p_j$)

Next, we propose an algorithm for implementing the mechanism.

Algorithm 3. (*Selection of sale offers*)

1. The products of maximum utility are selected.
2. Of the products selected in 1) those that have been sent the least number of times are chosen.
3. Of the products selected in 2) one is chosen at random. The function returns this product.
4. The seller agent's working memory is updated, increasing the number of times the product chosen in 3) is sent.

Choosing the product with the greatest utility is consistent with the agent's rationality with respect to utility maximization. Selecting the product that has been offered the least number of times is a consequence of logical reasoning, by estimating that if a product has not been accepted previously, it is unlikely to be accepted now, so better to try with a product with the same utility, but which has not been offered yet.

S3: Generate Potential Sale-Offers This mechanism for generating potential sale offers responds to the seller agent's need to use introspection to analyse which products in its catalogue would be considered as good offers at any given moment. We denominate these offers *potential sale offers* because their purpose is to serve as a reference in the rejection argumentation process of a purchase requirement. This mechanism returns a set of products from the catalogue S that it considers would be a good choice for a future offer to the buyer agent. Outputs: S_p, that details the set of products the seller agent considers would be a good choice for a future offer.

This mechanism constitutes the nucleus of the seller agent argumentation system. When a seller agent cannot satisfy a purchase requirement, it can motivate the buyer agent to change its preferences and consequently its proposals or purchase requirements. In a non-argumentative approach, the seller agent's only tool is rejection, but a mechanism based solely on rejection does not allow the negotiation to be taken to a more favourable position for the seller agent and even less arrive at a win-to-win solution. One way of promoting convergence in the negotiation from the seller agent's point of view is to express how a received purchase requirement should be relaxed. This expression is finally specified by the relax requirement $\rho_{\mathcal{B}_{req}}$. We understand a relax requirement as a two stage mechanism: the *generation of potential sale offers* (which are implemented in the mechanism we are describing), and *the generation of the relax requirement* $\rho_{\mathcal{B}_{req}}$ (which is implemented in the mechanism S4 described below).

The relax requirement is an expression of the seller agents preferences for the relaxation of specific constraints. The problem therefore is to discover which criteria the seller agent should apply to lean towards one constraint or another. We should recall that the final objective of the seller agent is to sell a product, and in particular decide which products it prefers to sell. The selection of preferred products is developed during the generation of potential sale offers stage. Once the products it prefers to sell have been selected, the relax requirement will try to lead the buyer agent towards those products, which is basically the same as trying to get it to relax those constraints that are not satisfied by the products that have been chosen as sale offers. However, choosing the candidate products is by no means an easy task. If we focus the search on strictly local criteria, we can lose the idea of what the buyer agent really wants, and on the other hand, if we concentrate solely on the needs of the buyer agent, the utility obtained by the sale can be diminished. Anyway, it seems obvious that there are two abstract ideas that should govern the selection of candidates, local and external criteria. We have identified two fundamental aspects in particular for carrying out the selection process:

Utility

Dependents on the u_j parameter defined for each product in the catalogue, would be a strictly local criterion focused on the utility of the sale.

Viability

Dependents on the degree of similarity between each product p_j and the purchase requirement $\pi_{\mathcal{B}_{req}}$, and the valuation the buyer agent is expected to make of the product on potential offer. This would be an external criteria connected to the buyer agents needs.

These two aspects are expressed by the function *prefer*(s_j)that assigns a preference value for each of the products in the catalogue. To modulate the importance and viability it turns to the agent's receptive profile β. So, we propose the following definition for the function *prefer*:

$$prefer(s_j) = \beta * u_j + (1-\beta) * \widehat{viability}(p_j, \pi^t_{\mathcal{B}_{req}}, \upsilon_{\mathcal{B}_{req}}) \tag{6}$$

At one extreme, if $\beta = 1$, the function remains as $\text{prefer}(s_j) = \beta * u_j$, whereas if $\beta = 0$ the function remains as $prefer(s_j) = \widehat{viability}(p_j, \pi^t_{\mathcal{B}_{req}}, \upsilon_{\mathcal{B}_{req}})$. Either the first case does not take into account the received purchase requirement or its possible valuation, which means the selection criteria, depends solely on the local utility. The second case does not take into account the utility, so the product selection criteria are based solely on the estimated viability of the sale. Intermediate β values consider both criteria at a specific rate.

Once the preference values for all the products in the catalogue have been calculated, by using a threshold filter for them, it is possible to generate a list of potential sale offers. The pre-set value threshold has a fundamental effect on the process of generating relax requirements. In general, a value below the threshold implies a high number of products in S_p, whereas a more selective or high threshold means that the number of products in S_p is smaller. If the seller agent tends to generate few products as candidates for potential sale offers, its behaviour can be interpreted as leaning towards leading the buyer agent towards a specific product. If, on the other hand, the seller agent is not very selective and the number of candidate products is always large, it can be assumed that it is not trying to lead the

negotiation towards a specific area. Therefore, this threshold is an indication of the impatience of the seller agent for sending constraint arguments.

Estimate of viability $\widehat{viability}$

The estimate of the viability of the sale of a product must be based, in accordance with the characteristics of the mechanism, on: the degree of similarity between the product and the purchase requirement $\pi_{\mathcal{B}_{req}}$; and an estimation of the valuation the buyer agent will make of the product in question if it is offered for sale. These aspects are condensed in the following definition of the estimation of viability:

$$\widehat{viability} = \widehat{sim}(p_j, \pi_{\mathcal{B}_{req}}) \Diamond \widehat{E_{bv}}(\upsilon_{\mathcal{B}_{req}}) \tag{7}$$

The first term represents an estimation of the similarity between the product and the purchase requirement $\pi_{\mathcal{B}_{req}}$. The second defines the estimation of the buyer agent valuation as an estimation function that depends on the product, and the valuation of the purchase requirement sent by the buyer agent (if the buyer agent's profile is expressive). The operator $\Diamond$ has the function of considering both terms (this operator is described later). We approach each term separately.

Similarity

We can define *similarity* as a function of distance in the following way:

$$\widehat{sim}(p_j, \pi_{\mathcal{B}_{req}}) = 1 - \widehat{dist}(p_j, \pi_{\mathcal{B}_{req}}) \tag{8}$$

where $\widehat{dist}(p_j, \pi_{\mathcal{B}_{req}})$ represents the estimation of the standardized distance between a product and a purchase requirement $\pi_{\mathcal{B}_{req}}$. To estimate the distance we propose using a measure based on Euclidean distance, making it clear that the selection is arbitrary and that it is possible to use other distance estimates. Our proposal is the following:

$$\widehat{dist}(p_j, \pi^t_{\mathcal{B}_{req}}) = sqrt(\sum_{i=1}^{n} \widehat{dist}(a_{ji}, \pi^{t,i}_{\mathcal{B}_{req}})^2 \ / \ n) \tag{9}$$

where $\widehat{dist}(a_{ji}, \pi^{t,i}_{\mathcal{B}_{req}})$ represents the distance of the attribute a_{ji} of the product to the set of constraints included in $\pi^t_{\mathcal{B}_{req}}$ which refer to said attribute. This means that in the first place the distances between the attribute and each one of these constraints must be calculated. This calculation is a measure of the distance of the closest estimated limit, which limits the corresponding crisp constraint. However, this measure of distance between an attribute and the limit of a crisp constraint is an absolute measure that has to be normalized. The following values need to be defined: the *estimated reservation value* a^{res}_i for each attribute; the *farthest limit of each attribute* a_{ji} , which is obtained by use of the boundary function $boundary(\pi^{t,i}_{\mathcal{B}_{req}})$; the *estimated relaxation speed* for each attribute a_{ji} , which we term $\tau^t_i \in (0, \infty)$; and finally, the *degree of certainty of the estimated distance* of an attribute a_{ji} which we term $\gamma^t_i \in [0,1]$. The reservation values, the relaxation velocity values and the

degree of certainty make up part of the seller agent's requirement model, in particular the set of beliefs $\Delta^t = \{(\delta_i^t, \gamma_i^t), i = 1,...,m\}$, where $\delta_i^t = (a_i^{res}, \tau_i^t)$. We now comment on each of these values.

The reservation value a_i^{res} expresses the seller agent's belief as to what value of relaxation limit the buyer agent would be willing to assume for a particular attribute. From this partial belief, it should not be induced that the buyer agent would be willing to relax all the constraints to these reserve values simultaneously. If this were the case, the seller agent could simply reject all the proposals until the buyer agent relaxed all its constraints. However, in a multiple seller agent system, using this type of strategy is conditioned by the seller agent's aversion to the risk of losing a sale. In our proposal, we do not specify how the seller agent establishes its belief about these reservation values. Beliefs could come from local factors, or it could be determined by external factors, for example, a record of negotiations with the same or other buyer agents.

The farthest limit requires in the first place the calculation of the limits that define the crisp constraints that limit an attribute a_{ji}. It is assumed that these limits are the closest to the attribute. Once these limits are obtained, the absolute distances to the attribute can be calculated. The farthest limit will be the limit generated by the largest of these distances. In other words, the absolute distance of the attribute a_{ji} to a purchase requirement will be the greatest distance from said attribute to the crisp constraints that limit it.

The estimated relaxation trend is an estimate of the buyer agent's predisposition to relax the constraints of a particular attribute. In this case, it is not the reserve value that is estimated, but the speed at which the reserve value will be reached. The idea is that the distance calculated for an attribute for which a rapid relaxation is estimated should be interpreted as shorter than for an equal distance calculated for an attribute for which a slow relaxation is estimated.

The degree of certainty γ_i^t is a measure of how sure the buyer agent is of the distance measure made for the attribute a_{ji}.

Finally, we now present the function $\widehat{dist}(a_{ji}, \pi_{\mathcal{B}_{req}}^{t,i})$ that binds all these concepts.

$$\widehat{dist}(a_{ji}, \pi_{\mathcal{B}_{req}}^{t,i}) = \begin{cases} (abs(\dfrac{a_{ji} - boundary(\pi_{\mathcal{B}_{req}}^{t,i})}{a_i^{res} - boundary(\pi_{\mathcal{B}_{req}}^{t,i})})^{1/\tau_i^t} - 1) * \gamma_i^t + 1 & a_{ji} \in [a_i^{res}, boundary(\pi_{\mathcal{B}_{req}}^{t,i})) \\ 0 & a_{ji} \in \pi_{\mathcal{B}_{req}}^{t,i} \\ 1 & rest \end{cases} \quad (10)$$

The distance estimate has the following properties:

1. The distance is standardized at one in all the cases.
2. The attribute values that exceed the estimated reservation values are fixed at distance 1.
3. If the attribute satisfies the constraint the distance is 0.
4. If the attribute is within the reservation value and the limit of the constraints, the distance is function of the remoteness of the constraints, and it is standardized by the distance between the reservation value and the limit of the constraints.

5. For an equal attribute value a_{ji}, a high value for τ_i^t implies a higher distance estimate than of a low value. Therefore, a high value for τ_i^t should be used when the seller agent does not believe that the constraints for the attribute will be relaxed largely.
6. For $\gamma_i = 0$ the degree of certainty about the estimation of the distance of the attribute a_{ji} is nil, so the distance has value 1.
7. For a predetermined value of $abs(.)^{1/\tau_i^t}$, a low value for γ_i^t penalizes the distance estimate in comparison with a high value.

Lastly, it is important to point out the dynamic character the parameters a_i^{res}, τ_i^t and γ_i^t can have. By this, we mean that these values can be updated even during the course of the negotiation because of the seller agent's beliefs, which may vary over time. For example, it is clear that if it is seen that the estimated reservation value is exceeded by the buyer agent's relaxation, the seller agent's belief with respect to the reservation value must be modified.

Estimate of the buyer agent's valuation

The estimate of the valuation the buyer agent will make of a sale offer is directly related to the explicit valuation the buyer agent may make of a purchase requirement, that's to say, of $v_{\mathcal{B}_{req}}$. This valuation follows a similar reasoning to that which causes us to define the relaxation velocity τ_i^t and the degree of certainty γ_i^t. It affects the distance estimate in the following way: if the constraints on an attribute are of high priority, the buyer agent will be much less predisposed to relax them, which means the distance measure should be increased, so, when the buyer agent sends purchase valuations we propose the operator implement the following viability function:

$$\widehat{viability} = 1 - sqrt(\sum_{i=1}^{n} (\widehat{dist}(a_{ji}, \pi_{\mathcal{B}_{req}}^{t,i}) * v_{boundary(\pi_{\mathcal{B}_{req}}^{t,i})})^2 / n) \quad (11)$$

where $v_{boundary(\pi_{\mathcal{B}_{req}}^{t,i})}$ represents the preference the buyer agent has that the constraint furthest from the attribute a_{ji} be satisfied.

To extend the use of this viability function to those cases in which the buyer agent is non-expressive, the seller agent considers $v_{boundary(\pi_{\mathcal{B}_{req}}^{t,i})} = 1$. The prefer function finally becomes:

$$prefer(s_j) = \beta * u_j + (1-\beta) * (1 - sqrt(\sum_{i=1}^{n} (\widehat{dist}(a_{ji}, \pi_{\mathcal{B}_{req}}^{t,i}) * v_{boundary(\pi_{\mathcal{B}_{req}}^{t,i})})^2 / n)) \quad (12)$$

S4: Generate Relax Requirement This mechanism responds to the seller agent's need to generate a relax requirement $\rho_{\mathcal{B}_{req}}$. It conforms to the second stage of the generation of the relax requirement process that we described in mechanism S3. Given a set of potential sale offers S_p, the mechanism generates a relax requirement with the aim of leading the buyer agent towards the products contained in S_p. The mechanism returns a relax requirement $\rho_{\mathcal{B}_{req}}$. Outputs: $\rho_{\mathcal{B}_{req}}$

The basic principle of the generation of the relax requirement is to get the buyer agent to relax those constraints of the purchase requirement that are not satisfied by the products contained in S_p. We apply the following algorithm:

Algorithm 4. (*Generate relax requirement*)

1. The relax requirement is generated as a vector $r_{k_1},...,r_{k_i}$, where $r_{k_x}=0$ indicates that constraint R_{k_x} is not satisfied by any product, and $r_{k_x}=1$ indicates that constraint R_{k_x} is satisfied by at least one product.

S5: Accept or Reject Offer This mechanism responds to the seller agent's need to decide at any given moment whether or not to accept an offer to purchase a product sent by the buyer agent using the locution *agree_to_buy*(.). There are only two possible return values, *accept* (p_j) if it accepts the offer, and *reject*(p_j) if it rejects the offer. This mechanism is easy to itemise if we assume a non-strategic behaviour, that is to say, if, as we have mentioned earlier, the agent does not keep back sale offers that satisfy the seller agents purchase requirements, while waiting for the buyer agent to relax its requirements further. If this supposition is valid, the mechanism returns *accept*(p_j) when p_j exists, and *reject*(p_j) in the opposite case. Outputs: *accept*(p_j), *reject*(p_j)

S6: Consider Withdrawal This mechanism responds to a seller agents need at any given moment to decide whether or not to terminate the dialogue with a buyer agent. The mechanism can remain in wait mode, and so returns wait, or indicate that it must withdraw from the dialogue, in which case it returns $withdraw(\theta)$. Outputs: *wait*, $withdraw(\theta)$

Equipped with the expressive mechanisms described through locutions, and the corresponding internal decision mechanisms, the next stage is to link these elements to finally shape the complete negotiation framework.

4.4 Operational semantics

Operational semantics in a dialogue game indicate how the state of the dialogue changes after locutions have been sent. It is assumed that the agents participating in the dialogue have the previously described decision mechanisms implemented and that the dialogue states are defined by the mechanisms inputs and outputs. The locutions sent throughout the course of the dialogue generate transitions between the different states, so that the locutions sent are inputs of one or more decision mechanisms, which in turn generate new outputs in the form of locutions. Therefore, the operational semantics is a formalization of the connection between the locutions available in the dialogue model and the defined decision mechanisms. To express the operational semantics we define the tuple $\langle P_x, \mathbf{K}, s\rangle$ that the decision mechanism K of agent P_x describes when it generates an output s. Operational semantics is defined by a series of transition rules between states. When the transitions are between the mechanisms of different agents, they are defined by the locutions that are sent and when they are between the mechanisms of the same agent, they are defined without locutions. In the first case, an arrow and the denomination of the pertinent locution indicate the transition. In the second case only an arrow appears. We now present the transition rules:

TR1 $\langle P_b, \mathbf{B1}, have\_need(\theta)\rangle \underline{\mathbf{L1}} \langle P_s, \mathbf{S1}, .\rangle$ This transition rule indicates that a buyer agent that wishes to acquire a product from category θ, is trying to start a purchase negotiation dialogue using the locution *L1: open_dialogue*(.). Said locution activates the mechanism *S1: Recognize Category* of the seller agent with which it wants to establish the dialogue.

TR2 $\langle P_b, \mathbf{B1}, have\_no\_need(\theta)\rangle \rightarrow \langle P_b, \mathbf{B1}, wait\rangle$ This transition rule indicates that a buyer agent that does not wish to acquire a product from category θ, will not start a purchase negotiation dialogue and will review the situation later on.

TR3 $\langle P_s, \mathbf{S1}, wish\_not\_to\_enter(\theta)\rangle \rightarrow \langle P_s, \mathbf{S1}, wait\rangle$ A seller agent that does not wish to start a trading dialogue with a buyer agent will review the situation later.

TR4 $\langle P_s, \mathbf{S1}, wish\_to\_enter(\theta)\rangle \underline{\mathbf{L2}} \langle P_b, \mathbf{B2}, .\rangle$ A seller agent that wishes to participate in a purchase negotiation dialogue will do so by sending the locution *L2: enter_dialogue*(.). This transmission induces the buyer agent to execute mechanism *B2: Generate Purchase Requirement* with the objective of generating the first purchase requirement.

TR5 $\langle P_b, \mathbf{B2}, \varnothing\rangle \rightarrow \langle P_b, \mathbf{B5}, .\rangle$ This transition rule affirms that when mechanism *B2: Generate Purchase Requirement* returns an *empty_set* in by a buyer agent it also activates mechanism *B5: Consider Withdrawal* in a buyer agent. This result is produced when a buyer agent cannot produce any more purchase requirements. Then it should consider withdrawing from the dialogue.

TR6 $\langle P_b, \mathbf{B5}, withdraw(\theta)\rangle \underline{\mathbf{L11}} \langle P_s, \mathbf{S6}, .\rangle$ When a buyer agent positively consider withdrawing from a dialogue, it sends the locution *L11: withdraw_dialogue*(). This activates in the seller agent mechanism *S6: Consider Withdrawal* so that it in turn considers withdrawing.

TR7 $\langle P_s, \mathbf{S6}, withdraw(\theta)\rangle \underline{\mathbf{L11}} \langle P_b, \mathbf{B5}, .\rangle$ When a seller considers withdrawing from the dialogue it sends locution *L11: withdraw_dialogue*(.), which in turn activates mechanism *B5: Consider Withdrawal* in the buyer agent.

TR8 $\left\langle P_b, \mathbf{B2}, \pi_{\mathcal{B}_{req}}\right\rangle \rightarrow \langle P_b, \mathbf{B3}, .\rangle$ This rule indicates that when a buyer agent generates a purchase requirement $\pi^t_{\mathcal{B}_{req}}$, it subsequently activates internally mechanism *B3: Generate Purchase Requirement Valuation* of generate purchase requirement valuation.

TR9 $\langle P_b, \mathbf{B3}, \varnothing\rangle \underline{\mathbf{L4}} \langle P_s, \mathbf{S2}, .\rangle$ Rule TR9 affirms that if mechanism *B3: Generate Purchase Requirement Valuation* induces an *empty_set* output, the buyer agent sends the locution *L4: desire_to_buy*(.). The locution in turn activates the seller agent's mechanism *S2: Assess Purchase Requirement* for the valuation of the purchase requirement.

TR10 $\left\langle P_b, \mathbf{B3}, \upsilon_{\mathcal{B}_{req}}\right\rangle \underline{\mathbf{L6}} \langle P_s, \mathbf{S2}, .\rangle$ This rule is identical to *TR9*, but the buyer agent sends the locution *L6: prefer_to_buy*(.) instead.

TR11 $\langle P_s, \mathbf{S2}, \varnothing\rangle \underline{\mathbf{L8}} \langle P_b, \mathbf{B2}, .\rangle$ This transition rule describes that when mechanism *S2: Assess Purchase Requirement* returns an *empty_set*, the seller agent sends the locution *L8: refuse_to_sell*(.). This locution activates mechanism *B2: Generate Purchase Requirement* in the buyer agent. This rule is the definitive one that generates a rejection locution without arguments to a previous purchase requirement.

TR12 $\left\langle P_s, \mathbf{S2}, sale\_offer(p_j)\right\rangle \underline{\mathbf{L3}} \langle P_b, \mathbf{B4}, .\rangle$ When the *S2:Assess Purchase Requirement* mechanism generates a sale offer that satisfies a purchase requirement, it is sent to the buyer agent by using the *L3: willing_to_sell*(.) locution, which in turn activates the buyer agents *B4:Consider Offers* mechanism for it to consider the offer.

TR13 $\left\langle P_s, \mathbf{S2}, purchase\_requirement(\pi^t_{\mathcal{B}_{req}})\right\rangle \rightarrow \langle P_s, \mathbf{S3}, .\rangle$ When the *S2: Assess Purchase Requirement* mechanism returns a purchase requirement, then the buyer agent itself activates the *S3: Generate Potential Sale-Offer* mechanism to explore the potential offers. In some way the *S2* mechanism is indicating that the purchase requirement be used as an input procedure to generate arguments.

TR14 $\langle P_s, \mathbf{S3}, S_P\rangle \rightarrow \langle P_s, \mathbf{S4}, .\rangle$ This transition rule states that the set S_p of potential sale offers generated by the *S3: Generate Potential Sale-Offer* mechanism, automatically activates the *S4:*

Generate Relax Requirement mechanism of the seller agent itself to generate a relax requirement.

TR15 $\langle P_s, \mathbf{S4}, \rho_{\mathcal{B}_{req}} \rangle \underline{\mathbf{L5}} \langle P_b, \mathbf{B2}, . \rangle$ This transition rule affirms that the relax requirement $\rho_{\mathcal{B}_{req}}$ which is obtained when the seller agents *S4: Generate Relax Requirement* is executed, is incorporated in the *L5: prefer_to_sell*(.) locution sent to the buyer agent. The locution activates the *B2: Generate Purchase Requirement* in the buyer agent with the aim of getting it to generate a new purchase requirement using the relax requirement that it has received as a basis.

TR16 $\langle P_b, \mathbf{B4}, generate\_purchase\_requirement(p_j) \rangle \rightarrow \langle P_b, \mathbf{B2}, . \rangle$

The $generate\_purchase\_requirement(p_j)$ output in the buyer agent's *B4: Consider Offers* mechanism activates *B2: Generate Purchase Requirement*. This transition appears when the buyer agent cannot accept a sale offer, so it needs to generate a new purchase requirement.

TR17 $\langle P_b, \mathbf{B4}, accept\_offer(p_j) \rangle \underline{\mathbf{L9}} \langle P_s, \mathbf{S5}, . \rangle$ This transition rule specifies that when a sale offer from the *B4: Consider Offers* mechanism is considered, if the offer is accepted, the mechanism sends the *L9: agree_to_buy*(.) locution, which in turn activates the *S5: Accept or Reject Offer* mechanism in the seller agent. This transition describes the beginning of the confirmation stage of the negotiation.

TR18 $\langle P_b, \mathbf{B4}, reject\_offer(p_j) \rangle \underline{\mathbf{L7}} \langle P_s, \mathbf{S2}, . \rangle$ If, after considering a sale offer from the *B4: Consider Offers* mechanism, the output of the mechanism is *reject_offer*(p_j), the buyer agent sends the *L7: refuse_to_buy*(.) locution, which activates the *S2: Assess Purchase Requirement* mechanism in the seller agent. This transition reflects the buyer agents rejection of a badly structured sale offer.

TR19 $\langle P_s, \mathbf{S5}, accept(p_j) \rangle \underline{\mathbf{L10}} \langle P_b, \mathbf{B5}, . \rangle$ When a seller agent considers, through the execution of the *S5: Accept or Reject Offer* mechanism, that an offer to purchase p_j is valid, it sends the *L10: agree_to_sell*(.) locution, which in turn activates the *B5: Consider Withdrawal* mechanism in the buyer agent. This transition describes the definitive confirmation of a purchase.

TR20 $\langle P_s, \mathbf{S5}, reject(p_j) \rangle \underline{\mathbf{L8}} \langle P_b, \mathbf{B2}, . \rangle$ If a seller agent considers that p_j is an invalid offer to purchase, when it executes the *S5: Accept or Reject Offer* mechanism, it sends the *L8: refuse_to_sell*(.) locution, that automatically activates the *B2: Generate Purchase Requirement* mechanism in the buyer agent. This result is produced when the confirmation stage cannot be completed due to the disappearance of the product p_j from the seller agent's catalogue.

TR21 $\langle P_x, \mathbf{K}, wait \rangle \rightarrow \langle P_x, \mathbf{K}, . \rangle$ When any mechanism *K* returns wait as an output, it indicates that the agent intention is to execute the same mechanism later.

One of the fundamental aims of our work is to develop an automated negotiation system. Therefore, the first thing we must demonstrate is that the dialogue model, the decision mechanism, and the operational semantics, that is to say our dialogue game framework for automated purchase negotiation is able to generate dialogue automatically. To demonstrate that the negotiation is automated we need to demonstrate: (a) that all the locutions can be activated by one or more of the decision mechanisms, and (b) that every time one of these mechanisms is executed it ultimately activates a locution. To support these propositions we first present for (a), a list of the locutions, together with the mechanisms that activate them, and the transition rule in which the activation is featured.

L1: Mechanism B1 (Rule TR1).
L2: Mechanism S1 (Rule TR4).
L3: Mechanism S2 (Rule TR12).
L4: Mechanism B3 (Rule TR9).
L5: Mechanism S4 (Rule TR15).
L6: Mechanism B3 (Rule TR10).
L7: Mechanism B4 (Rule TR18).
L8: Mechanism S2 (Rule TR11); Mechanism S5 (Rule TR20).
L9: Mechanism B4 (Rule TR17).
L10: Mechanism S5 (Rule TR19).
L11: Mechanism B5 (Rule TR6); Mechanism S6 (Rule TR7).

For (b), we show for each mechanism and their possible states: whether they activate a locution, or whether they indirectly activate a mechanism that in turn activates a locution. We also present the transition rules where these connections are established.

B1: Output *have_need* activates L1 (Rule TR1).
B1: Output *have_no_need* activates the mechanism B1 (Rule TR2).
B2: Output *empty_set* activates the mechanism B5 (Rule TR5).
B2: Output $\pi_{\mathcal{B}_{req}}$ activates the mechanism B3 (Rule TR8).

B3: Output *empty_set* activates the locution L4 (Rule TR9)
B3: Output $\upsilon_{\mathcal{B}_{req}}$ activates the locution L6 (Rule TR10).

B4: Output *generate_purchase_requirement* invokes the mechanism B2 (Rule TR16).
B4: Output *accept_offer* invokes the locution L9 (Rule TR17).
B4: Output *reject_offer* invokes L7 (Rule TR18).
B5: Output *withdraw_dialogue* invokes L11 (Rule TR6).
S1: Output *wish_not_to_enter* activates the mechanism S1 (Rule TR3).
S1: Output *wish_to_enter* activates the locution L2 (Rule TR4).
S2: Output *emtpy_set* invokes L8 (Rule TR11).
S2: Output *sale_offer* invokes L3 (Rule TR12).
S2: Output *purchase_requirement* invokes the mechanism S3 (Rule TR13).
S3: Output S_p activates the mechanism S4 (Rule TR14).

S4: Output $\rho_{\mathcal{B}_{req}}$ invokes the locution L5 (Rule TR15).

S5: Output *accept* invokes L10 (Rule TR19).
S5: Output *reject* invokes L8 (Rule TR20).
S6: Output *withdraw* invokes the locution L11 (Rule TR7).

We can easily prove that all the mechanisms generate a locution or activate a mechanism that then generates a locution, or activate a mechanism that then generates another mechanism that finally generates a locution.

5. Experimental analysis

Our negotiation framework allows us to test with different expressive and receptive strategies. An *expressive buyer* agent ($\xi = 1$) makes use of purchase valuations, whilst a *non-expressive* one ($\xi = 0$) does not. The *receptiveness* is determined by η, which has a

continuous domain that we limit to *non-receptive* ($\eta = 0$) and *receptive* ($\eta = 1$). On the other hand, an *expressive seller* agent ($\psi = 1$) makes use of relax requirements, whilst a *non-expressive* one ($\psi = 0$) does not. The *receptive profile* determined by β has a continuous domain that we limit to *0*, *0.5* and *1* (value 0 indicates no receptivity, 0.5 intermediate, and 1 maximum). However, not all the combinations of strategies are valid.

5.1 Analysis of validity of Individual strategies

We look first at agent level and we begin with the *buyer agent*:
(BAer) expressive and receptive, (BAner) non-expressive and receptive, and (BAnenr) non-expressive and non-receptive are valid strategies. However, **(BAenr) expressive and non-receptive** strategies make no sense as the purpose of the valuation is to redirect the negotiation in such a way that the seller agent sends useful relax requirements. If the agent does not analyse relax requirements, the valuation is of no use whatsoever. To sum up, the buyer agent can behave in three different ways: **BAer, BAner y BAnenr**.
We now analyse the *seller agent*:
(SAer1) expressive and receptive ($\beta = 0$), (SAer0.5) expressive and receptive ($\beta = 0.5$), (SAenr) expressive and non-receptive ($\beta = 1$), and **(SAnenr) non-expressive and non-receptive** are valid strategies. However, **(SAner1 or SAner0.5) non-expressive and any receptive** strategy makes no sense as a non-expressive seller agent does not send relax requirements, and the main purpose of a receptive strategy is to direct the relax requirements. To sum up, the seller agent can behave in accordance with four different strategies: **SAer1, SAer0.5, SAenr** and **SAnenr**.

5.2 Analysis of validity of combination of strategies

There are 12 possible combinations of strategies. However, some of these combinations are not coherent. In **BAer vs SAnenr** the buyer agent's valuations are not taken into account by the seller agent, which furthermore is not expressive. This aspect is detectable by the buyer agent, given that it does not receive relaxation requirements. A rational agent will not send valuations if it knows they are of no use. This combination is not stable and therefore equilibrium is not possible. The best strategy for a buyer agent under these circumstances is to change to a **non-expressive and non-receptive** strategy **BAnenr**. In **BAner vs SAnenr** neither of the agents is expressive, so for the buyer agent to be receptive makes no sense, and furthermore this fact it detectable by the buyer agent. A rational buyer agent would change to a **BAnenr** strategy. After this analysis, there are 10 pairs of balanced strategies : **BAer vs SAer1**, **BAer vs SAer0.5**, **BAer vs SAenr**, **BAner vs SAer1**, **BAner vs SAer0.5**, **BAner vs SAenr**, **BAnenr vs SAer1**, **BAnenr vs SAer0.5**, **BAnenr vs SAenr**, y **BAnenr vs SAnenr**. To simplify this repertoire we have made the following groupings:
BAer vs SAexrx This group has in common the fact that the buyer agent is simultaneously expressive and receptive, and the seller agent is expressive. Furthermore it seems obvious that the seller agents' different receptive profiles will affect the results of the negotiation, because the generation of relax requirements varies depending on this profile. Therefore, a priori we need to test with the three combinations that make up the group.
BAner vs SAexrx This group of strategies has in common the fact that the buyer agent is not expressive, but it is receptive and the seller agent is expressive. When the seller agent is receptive, intuitively we can affirm that the results of the negotiations are different to those

of the previous group. This is so because the buyer agent's valuations are not available to the seller agent. However, when the seller agent is not receptive the scenario is identical to the previous group. In other words, if the seller agent is not receptive it makes no difference whether the buyer agent sends valuations or not. In conclusion, the BAner vs SAenr pair is identical to BAer vs SAenr, as far as the results of the negotiation are concerned. To speed up the tests of this type, we have opted to define the test as BAner vs SAenr.

BAnenr vs SAxexrx This group of strategies is characterised by the non-expressivity and non-receptivity of the buyer agent. Therefore, it makes no difference whether the seller agent is expressive or receptive or not as the buyer agent will be unable to consider it. To speed up the execution of the tests in this group, we have opted to define as representative the BAnenr vs SAnenr pair.

Summarizing, there exist six pairs of representative strategies.

5.3 Buyer's preferences and seller's catalogue of products

The buyer agent's preferences are described as a 5 fuzzy constraint problem $R^f_{1...5}$ over 5 attributes $a_{1...5}$. Given a catalogue of products *S*, the set of products that may be a solution to a negotiation is the solution set $S_{sol} \subseteq S$. This set is comprised of the products that maximize the buyer agent's utility. This occurs when the seller agent is non-strategic with regards to the occultation of products and the buyer agent relaxes constraints minimizing the lost of *posd*. Finally, the noise set $S_{noise} \subseteq S$, is the set of complementary products to S_{sol}, so that $S_{sol} \cup S_{noise} = S$. In the experiments, the set S_{sol} is defined as a set of products where $\alpha(p_i) = 0.7$ for the buyer agent, while the utilities for the set S_{noise} are 0.1, 0.2 or 0.3. Once the products from the solution and noise sets have been generated, the next step is to assign utility values u_j to each of them. In the case of S_{noise} these are generated randomly using a uniform allocation between 0.9 and 1, while for S_{sol} a uniform allocation between 0 and 0.69 is used. To test the pareto-efficiency of the negotiations, also randomly, the utility of one of the products from the set S_{sol} is assigned 0.7. The aim is to see if this solution is reached after a negotiation. With this allocation of utilities, the seller agent's preferred sale offers are the noise set products. However, an intelligent seller agent would conclude that these products are not a valid sale offer and it would try to obtain the best solution from amongst those products that can really be a solution, in other words, from the solution set.

5.4 Test results

For each of the six pairs of strategies that we analyze and the different sizes of catalogues, 300 negotiations are carried out. We take as a reference the number of products from the solution set, so that the noise set is the same size as the solution set in every case. Taking into account that the buyer agent's overall satisfaction degree is known, the result we need to analyse is the utility the seller agent obtains from each negotiation. The results show the median and the success rate, where the success rate estimates the number of times that the pareto-optimal solution is obtained, that is to say, the solution in which $u_j = 0.7$. In every case, the calculated confidence interval is 95%. The summary of results is shown in Table 1.

It can be seen that with the *BAnenr vs SAnenr* strategies the success rate stabilizes around 10%, although for 4 and 8 products the rate is higher, which is logical, as the number of relax combinations is greater than the number of products. There is a dip in the median value

with 16 products. However, the median grows again as the number of products increases. This effect was foreseeable, as when the number of products is augmented, the probability that the seller agent has products with a high utility for the purchase requirement is greater. The results for the *BAer/BAner vs SAer1* strategies are similar to the *BAnenr vs SAnenr* strategies. This result was foreseeable, taking into account that the seller agent is not a utility maximizer. In the *BAer vs SAer0.5* strategies the valuation of the purchase requirements distracts the seller agent, so it unfavourably modifies the preferences of the different sale offers and ends up concentrating the search in the noise set, and the results are similar to the *BAnenr vs SAnenr* strategies. The *BAner vs SAenr* strategies are also similar to those obtained with the *BAnenr vs SAnenr* strategies. These results are very important, because they allow us to appreciate how, when a seller agent limits to looking out for its best interest and only thinks about the utility, the results are not good. Finally, with the *BAner vs SAer0.5* strategies we can check how the results are significantly better in every case. This test shows that the expressivity of the seller agent is key to obtaining satisfactory solutions.

BAnenr vs SAnenr, BAer/BAner vs SAer1,□BAner vs SAenr, BAer vs SAer0.5□			
Number of products	*Success rate*	*Confidence interv.*	*Median*
4	0.4400	0.3830 0.4982	0.6432
8	0.32	0.2676 0.3760	0.5502□
16	0.15	0.1116 0.1955	0.4867□
32	0.12	0.0855 0.1622	0.5362□
64	0.15	0.1116 0.1955	0.6326□
128	0.11	0.0769 0.1510	0.6419□
256	0.07	0.0439 0.1050	0.6686□
BAner vs SAer05□			
4	0.9500	0.9189 0.9717	0.7□
8	0.7600	0.7076 0.8072	0.7□
16	0.6900	0.6343 0.7419	0.7□
32	0.5200	0.4618 0.5778	0.7□
64	0.4500	0.3928 0.5082	0.6859□
128	0.3600	0.3056 0.4172	0.6647□
256	0.3300	0.2770 0.3864	0.6807□

Table 1. Summary of results

In Figure 5 the results obtained from the tests of the *BAnenr vs SAnenr*, and *BAner vs SAer05* strategies are summarised. In the top graphic the medians are shown, and in the bottom one the success rates. The success rates follow the same trend for all the catalogues, with a noticeable improvement in the case of the *BAner vs SAer05* strategies. As regards the medians, for catalogues with up to 64 products, the results are optimum. For catalogues with more than 256 products the strategies tend to converge, so expressivity is not a determinant factor. It should be recalled that a heavily populated catalogue means the seller agent will have high utility sale offers with a higher probability.

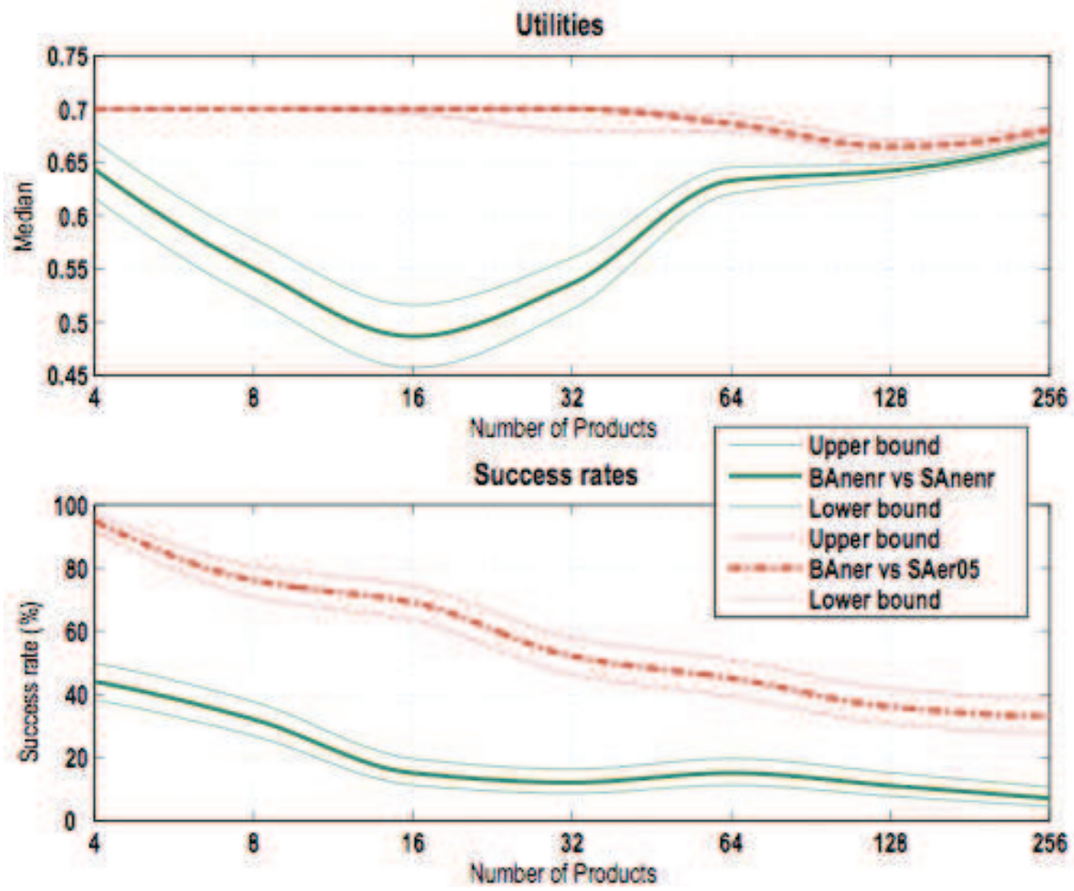

Fig. 5. Comparative of the *Banenr vs Sanenr* and the *Baner vs Saer05* strategies

Finally, in Figure 6 two graphs are presented that depict the percentage improvement in the success rates and the comparative percentage improvement in utility. The improvement in the success rates portrays the comparative between the percentage of success rates obtained with the *BAner vs SAer05* strategies and those obtained with the *BAnenr vs Sanenr* strategies. This graph presents a very important property, which is the exponential trend of comparative improvements in the success rates. For catalogues with a small solution set the improvement is of approximately 200%, with an increase of around 325% for medium sized catalogues of 16 and 32 products being observed. It should be taken into account that when there are very few products, the possibility of a good solution being found at random, is greater than when the catalogue is large, which is why the improvement is smaller for 4 and 8 products. Although for 64 products the success rate decreases to 250%, in general, as the size of the catalogues increases there is an exponential tendency for the rates to improve. As the catalogues become very large, the probability of obtaining an optimal solution without expressivity decreases exponentially down to zero, whereas with expressivity the optimal solution is explicitly searched.

The relative improvement in utility is a comparative measure that compares the improvements obtained with the *BAner vs SAer05* strategies with respect the maximum improvement obtained. It can be observed that the reference catalogue is the one with 16 products, which is the scenario with which the maximum utility is obtained. Therefore, the

graph shows a percentage of relative improvement of 100% for this catalogue. For large catalogues, the percentage of relative improvement decreases to below 10%. The minimum percentage improvement for smaller catalogues is 10% with an average value of around 60%.

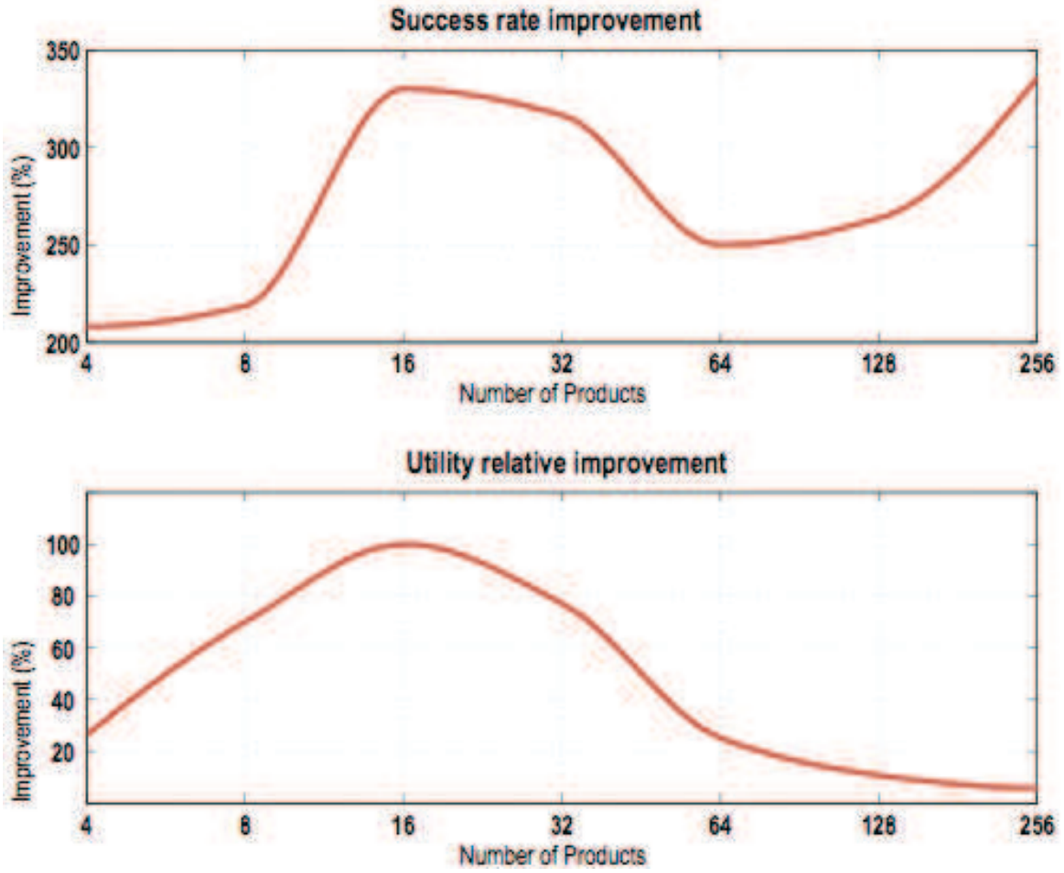

Fig. 6. Improvement in the success rate and relative improvement of utility

6. Conclusions and future work

This chapter presents a fuzzy constraint based model for automated purchase negotiations in competitive trading environments. The analysis of the model shows that the expressivity of the seller agent is essential to obtain an improvement in the negotiations. However, the expressivity of the buyer agent makes the results come close to those achieved with the reference non-expressive and non-receptive strategies. The viability of the potential sale offers is decisive in the improvement of the negotiations, and so, the *BAner vs SAenr* strategies, which focus on the seller agents' utility, are invalid. To sum up, we can affirm that the expressivity factor brings significant benefit to the negotiation process and the key element resides in the expressivity of the seller agent and the receptivity of the buyer agent. However, it must be pointed out that under our negotiation model, an 'inexpressive' buyer agent is more expressive than an 'inexpressive' seller agent because a buyer agent expresses offers as a set of constraints, while a seller agent expresses offers as concrete products or rejections to purchase requirements.

As future work, we propose the refinement of the mechanisms related to the purchase requirement valuation. As we have seen, the results using valuations of purchase requirements are not good because valuations distract the seller agent. We suggest to test different estimates for the viability parameter in the prefer function of the *generate potential sale offers seller's mechanism*. We believe that including valuations in the purchase requirements the negotiation processes may be improved.

7. Acknowledgements

This work has been supported by the Spanish Ministry of Education and Science grant TSI2005-07384-C03-03, and by the Comunidad de Madrid grant CCG07-UAH/TIC-1648.

8. References

Buttner, R. (2006). A Classification Structure for Automated Negotiations, *Proceedings of the 2006 IEEE/WIC/ACM Int. Conference on Web Intelligence and Intelligent Agent Technology*, pp. 1-8, ISBN 0-7695-2749-3, IEEE Computer Society

Faratin, P.; Sierra, C. & Jennings, N.R. (1998). Negotiation decision functions for autonomous agents. *Robotics and Autonomous Systems,* 24, 3-4, 159-182, ISSN 0921-8890

Faratin, P.; Sierra, C. & Jennings, N.R. (2002). Using similarity criteria to make issue trade-offs in automated negotiations. *Artificial Intelligence,* 142, 2, 205-237, ISSN 0004-3702

Ehtamo, H.; Hamalainen, P.; Heiskanen, P.; Teich, J.; Verkama, M. & Zionts, S. (2001). Generating Pareto Solutions in a Two-Party Setting: Constraint Proposal Methods. *Management Science,* 45, 12, 1697-1709, ISSN 1526-5501

Fatima, S. (2006). Multi-issue Negotiation with Deadlines. *Journal of Artificial Intelligence Research,* 27, 381-417, ISSN 11076 - 9757

Fisher, R. & Ury, W. (1981). *Getting to Yes: Negotiating an Agreement Without Giving In,* Houghton Mifflin Company, ISBN 0-395-63124-6, New York

Gatti, N. & Amigoni, F. (2005). An Approximate Pareto Optimal Cooperative Negotiation Model for Multiple Continuous Dependent Issues, *Proceedings of the 2005 IEEE/WIC/ACM Int. Conference on Web Intelligence and Intelligent Agent Technology*, pp. 1-8, ISBN 0-7695-2749-3, IEEE Computer Society

Ito, T.; Klein, M. & Hattori, H. (2008). A multi-issue negotiation protocol among agents with nonlinear utility functions. *Multiagent and Grid Systems,* 4, 67-83, ISSN 1574-1702

Keeney, R.L. & Raiffa, H. (1976). *Decisions with Multiples Objectives: Preferences and value Tradeoffs,* John Wiley and Sons, ISBN 0 521 43883 7, New York

Klein, M.; Faratin, P.; Sayama, H. & Bar-Yam, Y. (2003). Protocols for Negotiating Complex Contracts. *IEEE Intelligent Systems,* November-December 2003, 32-38, ISSN 1094-7167

Kowalczyk, R. & Bui, V. (2000). On fuzzy e-negotiation agents: Autonomous negotiation with incomplete and imprecise information, *Proceedings of the 11th Int. Workshop on Database and Expert Systems Applications*, pp. 1034-1038, London, ISBN 3-540-67978-2

Lai, G. ; Li, C. ; Sycara, K. & Giampapa, J. (2004). Literature Review on Multi-attribute Negotiations, *Technical Report CMU-RI-TR-04-66,* Carnegie Mellon University

Lai, G.; Li, C. & Sycara, K. (2006). Efficient Multi-Attribute Negotiation with Incomplete Information. *Group Decision and Negotiation,* 15, 511-528, ISSN 1572-9907

Lai, R. & Lin, M.W. (2004). Modelling agent negotiation via fuzzy constraints in e-business. *Computational Ingelligence,* 20, 4, 624-642, ISSN 1467-8640

Lax, D. & Sebenius, J. (1992). The manager as negotiator: The negotiators dilemma: Creating and claiming value, In: *Dispute Resolution,* F.S. Stephen Goldberg and N. Rogers, (2nd ed.), 49-62, Little Brown and Co., ISBN 978-0735528802, Boston

Luo, X.; Jennings, N.R.; Shadbolt, N.; Leung, H.F. & Lee, J.H. (2003). A fuzzy constraint based model for bilateral, multi-issue negotiations in semi-competitive environments. *Artificial Intelligence,* 148, 1-2, 53-102, ISSN 0004-3702

McBurney, P.; Van Eijk, R.M.; Parsons, S. & Amgoud, L. (2003). A Dialogue Game Protocol for Agent Purchase Negotiations. *Autonomous Agents and Multi-agents Systems,* 7, 235-273, ISSN 1573-7454

Nash, J. (2005). The Bargaining Problem. *Econometrica,* 18, 2, 155-162

Rahwan, I.; Ramchurn, S.; Jennings, N.; McBurney, P.; Parsons, S. & Sonenberg, L. (2003). Argumentation-based negotiation. *The Knowledge Engineering Review,*18, 4, 343-375, ISSN 1469-8005

Vo, Q.B.; Padgham, L. & Cavedon, L. (2007). Negotiating flexible agreements by combining distributive and integrative negotiation. *Intelligent Decision Technologies,* 1, 1-2, 33-47, ISSN 1872-4981

Zhang, J. & Pu, P. (2004). Survey on Solving Multi-Attribute Decision Problems, *Technical Report IC/2004/54,* EPFL

Multiagent Systems

Edited by Salman Ahmed and Mohd Noh Karsiti

ISBN 978-3-902613-51-6
Hard cover, 426 pages
Publisher I-Tech Education and Publishing
Published online 01, January, 2009
Published in print edition January, 2009

Multi agent systems involve a team of agents working together socially to accomplish a task. An agent can be social in many ways. One is when an agent helps others in solving complex problems. The field of multi agent systems investigates the process underlying distributed problem solving and designs some protocols and mechanisms involved in this process. This book presents an overview of some of the research issues in the field of multi agents. It is a presentation of a combination of different research issues which are pursued by researchers in the domain of multi agent systems as they are one of the best ways to understand and model human societies and behaviours. In fact, such systems are the systems of the future.

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

IntechOpen