A Service-Oriented Negotiation Model between Autonomous Agents

Carles Sierra*, Peyman Faratin, Nick R. Jennings

Dept. Electronic Engineering, Queen Mary and Westfield College,
University of London, London E1 4NS, UK.
{C.A.Sierra,P.Faratin,N.R.Jennings}@qmw.ac.uk

Abstract. We present a formal model of negotiation between autonomous agents. The purpose of the negotiation is to reach an agreement about the provision of a service by one agent for another. The model defines a range of strategies and tactics that agents can employ to generate initial offers, evaluate proposals and offer counter proposals. The model is based on computationally tractable assumptions and is demonstrated in the domain of business process management. Initial proofs about the convergence of negotiation are also presented.

1 Introduction

Autonomous agents are being increasingly used in a wide range of industrial and commercial domains [2]. These agents have a high degree of self determination – they decide for themselves what, when and under what conditions their actions should be performed. In most cases, such agents need to interact with other autonomous agents to achieve their objectives (either because they do not have sufficient capabilities or resources to complete their problem solving alone or because there are interdependencies between the agents). The objectives of these interactions are to make other agents undertake a particular course of action (e.g. perform a particular service), modify a planned course of action (e.g. delay or bring forward a particular action so that there is no longer a conflict), or come to an agreement on a common course of action. Since the agents have no direct control over one another, they must persuade their acquaintances to act in particular ways (they cannot simply instruct them). The paradigm case of persuasion is negotiation – *a process by which a joint decision is made by two or more parties. The parties first verbalise contradictory demands and then move towards agreement by a process of concession making or search for new alternatives*, (cf. [3]).

Given its pervasive nature, negotiation comes in many shapes and forms. However in this work we are interested in a particular class of negotiation –

* On sabbatical leave from Artificial Intelligence Research Institute –IIIA, Spanish Council for Scientific Research –CSIC. 08193 Bellaterra, Barcelona, Spain. sierra@iiia.csic.es. With the support of the Spanish Ministry of Education grant PR95-313.

namely *service-oriented negotiation*. In this context, one agent (the client) requires a service to be performed on its behalf by some other agent (the server)[2]. Negotiation involves determining a contract under certain terms and conditions. The negotiation may be iterative in that several rounds of offers and counter offers will occur before an agreement is reached or the negotiation is terminated.

When building an autonomous agent which is capable of flexible and sophisticated negotiation, three broad areas need to be considered [7] – what negotiation protocol will be used?, what are the issues over which negotiation takes place?, and what reasoning model will the agents employ? This paper concentrates predominantly on the final point although the protocol and negotiation object are briefly defined. A comprehensive reasoning model for service-oriented negotiation should determine: which potential servers should be contacted, whether negotiation should proceed in parallel with all servers or whether it should run sequentially, what initial offers should be sent out, what is the range of acceptable agreements, what counter offers should be generated, when negotiation should be abandoned, and when an agreement is reached.

This paper presents a formal account of a negotiating agent's reasoning component –in particular it concentrates on the processes of generating an initial offer, of evaluating incoming proposals, and of generating counter proposals. The model specifies the key structures and processes involved in this endeavour and defines their inter-relationships. The model was shaped by practical considerations and insights emanating from the development of a system of negotiating agents for business process management (see [5] and Section 2 for more details). The main contributions of this work are: (i) it allows rich and flexible negotiation schemes to be defined; (ii) it is based on assumptions which are realistic for autonomous computational agents (see Section 3.2 for the set of requirements and Section 7 for a discussion of related approaches); and (iii) it presents some initial results on the convergence of negotiation (see Section 6).

In this paper we concentrate on many-parties, many-issues, single-encounter negotiations with an environment of limited resources (time among them). Section 2 gives details of the type of applications and scenarios we are interested in. Sections 3 to 5 present the proposed model. Finally, some results on negotiation convergence and future avenues of work are outlined.

2 Service-Oriented Negotiation

This section characterises a context in which service oriented negotiation takes place. The scenario is motivated by work in the ADEPT project [5] which has developed negotiating agents for business process management applications. However, we believe that the characteristics emerging from this domain have a wide variety of application. To provide a detailed context for this work, a multi-agent

[2] A service is a problem solving activity which has clearly defined start and end points. Examples include diagnosing a fault, buying a group of shares in the stock market, or allocating bandwidth to transmit a video-conference.

system for managing a British Telecom (BT) business process is presented (section 2.1). This scenario is then analysed in terms of its key characteristics and assumptions as they relate to the process of negotiation (section 2.2).

2.1 BT's Provide Customer Quote Business Process

This scenario is based on BT's business process of providing a quotation for designing a network to provide particular services to a customer (figure 1)[3]. The overall process receives a customer service request as its input and generates as its output a quote specifying how much it would cost to build a network to realise that service. It involves up to six agent types: the sales department agent, the customer service division agent, the legal department agent, the design division agent, the surveyor department agent, and the various agents who provide the out-sourced service of vetting customers.

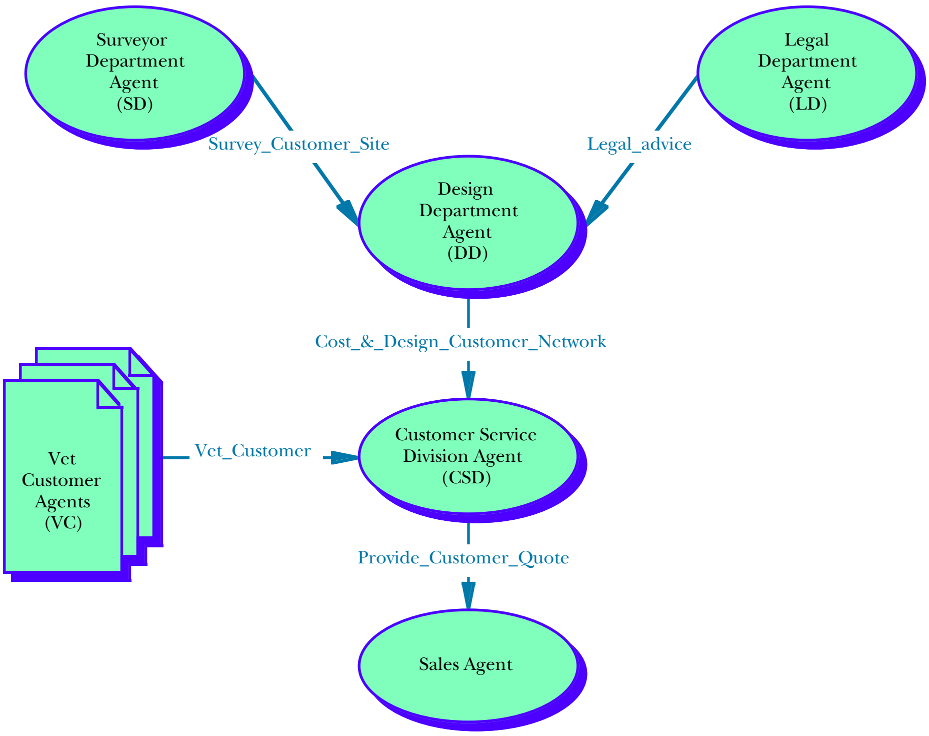

Fig. 1. Agent system for BT's provide customer quote business process.

[3] The negotiations between the agents are denoted by arrows (arrow head toward client) and the service involved in the negotiation is juxtaposed to the respective arrow.

The process is initiated by the sales agent which negotiates with the CSD agent (mainly over time, but also over the number of invocations and the form in which the final result should be delivered) for the service of providing a customer quote. The first stages of the *Provide_Customer_Quote* service involve the CSD agent capturing the customer's details and vetting the customer in terms of their credit worthiness. The latter sub-service is actually performed by one of the VC agents. Negotiation is used to determine which VC agent should be selected – the main attributes which are negotiated over are the price of the service, the penalty for contract violation, the desired quality of the service and the time by which the service should be performed. If the customer fails the vetting procedure, then the quote process terminates. Assuming the customer is satisfactory, the CSD agent maps their requirements against a service portfolio. If the requirements can be met by a standard off-the-shelf portfolio item then an immediate quote can be offered based on previous examples. In the case of bespoke services, however, the process is more complex. The CSD agent negotiates with the DD agent (over time and quality) for the service of designing and costing the desired network service. In order for the DD agent to provide this service it must negotiate with the LD agent (over time) and perhaps with the SD agent. The LD agent checks the design to ensure the legality of the proposed service (e.g. it is illegal to send unauthorised encrypted messages across France). If the desired service is illegal, then the entire quote process terminates and the customer is informed. If the requested service is legal then the design phase can start. To prepare a network design it is usually necessary to have a detailed plan of the existing equipment at the customer's premises. Sometimes such plans might not exist and sometimes they may be out of date. In either case, the DD agent determines whether the customer site(s) should be surveyed. If such a survey is warranted, the DD agent negotiates with the SD agent (over price and time) for the *Survey_Customer_Site* service. On completion of the network design and costing, the DD agent informs the CSD agent which informs the customer of the service quote. The business process then terminates.

The structure of the negotiation object is based almost directly on the legal contracts used to regulate agreements in the current manual approach to business process management. This structure is fairly rich and covers both service and meta-service attributes. In more detail, it contains: (i) the service name; (ii) a unique agreement identifier (covering the case where there are multiple agreements for the same service); (iii) the agents involved in the agreement (client and server); (iv) the type of agreement (one off agreement for a single service invocation versus on-going agreements for multiple invocations of the same service); (v) timing information (duration represents the maximum time the server can take to finish the service, and start time and end time represent the time during which the agreement is valid); (vi) the volume of invocations permissible between the start and end times (for on-going agreements only); (vii) the price paid per invocation; (viii) the penalty the server incurs for every violation of the agreement; (ix) the information the client must provide to the server on service invocation; and (x) the policy used for disseminating the service's intermediate and final results to the client.

2.2 Characteristics and Assumptions

The following negotiation characteristics can be noted from the ADEPT business process scenario. Moreover, it is believed that these characteristics are common to a wide range of service oriented negotiations between autonomous agents.

- A given service can be provided by more than one agent (e.g. multiple agents can provide the vet customer service to the CSD agent). The available services may be identical in their characteristics or they may vary along several dimensions (e.g. quality, price, availability, etc.).
- Individual agents can be both clients and servers for different services in different negotiation contexts.
- Negotiations can range over a number of quantitative (e.g. price, duration, and cost) and qualitative (e.g. type of reporting policy, and nature of the contract) issues. Each successful negotiation requires a range of such issues to be resolved to the satisfaction of both parties. Agents may be required to make trade-offs between issues (e.g. faster completion time for lower quality) in order to come to an agreement.
- The social context and inter-relationships of the participants influences the way agents negotiate. Some negotiations involve entities within the same organisation (e.g. between the CSD and DD agents) and hence are generally cooperative in nature. Other negotiations are inter-organisational and purely competitive – involving self interested, utility maximising agents (e.g. between the VC agents and the CSD agent). Some groups of agents often negotiate with one another for the same service (e.g. the CSD and DD agents), whereas other negotiations are more open in nature (for example, the set of VC agents changes frequently and hence the CSD agent often negotiates with unknown agents).
- As the agents are autonomous, the factors which influence their negotiation stance and behaviour are private and not available to their opponents (especially in inter-organisational settings). Thus agents do not know what utilities their opponents place on various outcomes, they do not know what reasoning models they employ, they do not know their opponent's constraints and they do not know whether an agreement is even possible at the outset (i.e. the participants may have non-intersecting ranges of acceptability).
- Since negotiation takes place within a highly intertwined web of activity (the business process) time is a critical factor. Timings are important on two distinct levels: (i) the time it takes to reach an agreement must be reasonable; and (ii) the time by which the negotiated service must be executed is important in most cases and crucial in others. The former means that the agents should not become involved in unnecessarily complex and time consuming negotiations – the time spent negotiating should be reasonable with respect to the value of the service agreement. The latter means that the agents sometimes have hard deadlines by which agreements must be in place

(this occurs mainly when multiple services need to be combined or closely coordinated).

3 The Negotiation Model

The negotiation model for autonomous agents proposed in this Section is based on a variation of the *two parties, many issues* value scoring system presented in [8]. That is, a model for bilateral negotiations about a set of quantitative variables. Our variation transforms that model into a *many parties, many issues* model (that is, multilateral negotiations about a set of variables). Multilateral negotiations are central to the application domains we are interested in. Our model of multilateral negotiations is based on a set of mutually influencing *two parties, many issues* negotiations. We will call the sequence of offers and counter-offers in a two-party negotiation a *negotiation thread*. Offers and counter offers are generated by lineal combinations of simple functions, called *tactics*. Tactics generate an offer, or counter offer, for a single component of the negotiation object using a single criteria (time, resources, etc.). Different weights in the lineal combination allow the varying importance of the criteria to be modelled. For example, when determining values of slots in the negotiation object it may initially be more important to take into account the other agent's behaviour than the remaining time. In which case, the tactics that emphasize the behaviour of other agents will be given greater preference than the tactics which base their value on the amount of time remaining.

To achieve flexibility in the negotiation, the agents may wish to change their ratings of the importance of the different criteria over time. For example, remaining time may become more relevant than the imitation of the other's behaviour as the time by which an agreement must be in place approaches. We use the term "*strategy*" to denote the way in which an agent changes the weights of the different tactics over time. Thus strategies combine tactics depending on the history of negotiations and the mental state of agents, and negotiation threads influence one another by means of strategies (see Section 6).

Before presenting our model, we introduce Raiffa's basic model for bilateral negotiation [8].

3.1 The bilateral negotiation model

Let i ($i \in \{a, b\}$) represent the negotiating agents and j ($j \in \{1, ..., n\}$) the issues under negotiation. Let $x_j \in [min_j, max_j]$ be a value for issue j. Here we consider issues for which negotiation amounts to determining a value between a delimited range. Each agent has a scoring function $V_j^i : [min_j, max_j] \rightarrow [0, 1]$ that gives the score agent i assigns to a value of issue j in the range of its acceptable values. For convenience, scores are kept in the interval $[0, 1]$.

The next element of the model is the relative importance that an agent assigns to each issue under negotiation. w_j^i is the importance of issue j for agent i. We assume the weights of both agents are normalized, i.e. $\sum_{1 \leq j \leq n} w_j^i = 1$, for all i in $\{a, b\}$. With these elements in place, it is now possible to define an agent's scoring function[4] for a *contract* – that is, for a value $x = (x_1, ..., x_n)$ in the multi-dimensional space defined by the issues' value ranges:

$$V^i(x) = \sum_{1 \leq j \leq n} w^i_j V^i_j(x_j)$$

If both negotiators use such an additive scoring function, it is possible to show how to compute the optimum value of x as an element on the efficient frontier of negotiation[5] (see [8], p. 164).

3.2 Service-oriented negotiation requirements

The above mentioned model for bilateral negotiation is valid for some service oriented settings. However, the model contains several implicit assumptions that, although they permit good optimisation results, are inappropriate for our scenarios:

1. *Privacy of information.* To find the optimum value, the scoring functions have to be disclosed. This is, in general, inappropriate for competitive negotiation.
2. *Privacy of models.* Both negotiators have to use the same additive scoring model. However, the models used to evaluate offers and generate counter offers are one of the things that negotiators try to hide from one another.
3. *Value restrictions.* There are pre-defined value regions for discussion (they are necessary to define the limits of the scoring function). However, it is not always possible to find these common regions and in many cases negotiation actually involves determining whether such regions even exist.
4. *Time restrictions.* There is no notion of timing issues in the negotiation. However, time is a major constraint on the agent's behaviour [6]. This is mainly true on the client side; agents often have strict deadlines by when the negotiation must be completed. For instance, a video link has to be provided at 16:00 because at that time a conference should start; negotiation about set up cannot continue after that instant in time.
5. *Resource restrictions.* There is no notion of resource issues in the negotiation. However, the quantity of a particular resource has a strong and direct influence on the behaviour of agents, and, moreover, the correct appreciation of the remaining resources is an essential characteristic of good negotiators. Resources from the client point of view relate directly with the number of servers engaging in negotiations; likewise from the server's point of view. Thus, the quantity of resource has a similar effect on the agents' behaviour as time.

[4] Non-linear approaches to modelling utility could be used if necessary without affecting the basic ideas of the model.

[5] Any contract not on this frontier is sub-optimal (i.e. not pareto-optimal) in that possible mutual gains are missed.

Taking the first consideration alone, it is clear that optimal solution cannot be found in our domains: it is not possible to optimize an unknown function. Hence, we shall propose a model for individual agent negotiation that looks for deals acceptable to its acquaintances but which, nevertheless, maximises the agent's own scoring function.

3.3 A service-oriented negotiation model

In service oriented negotiations, agents exhibit two possible behaviours that are, in principle, in conflict. Hence we shall distinguish (for notational convenience) two subsets of agents[6], $Agents = Clients \cup Servers$. We use roman letters to represent agents; $c, c_1, c_2, \ldots$ will stand for clients, $s, s_1, s_2, \ldots$ for servers and $a, a_1, b, d, e, \ldots$ for unspecific agents.

We adhere to an additive scoring system in which, for simplicity, the function V_j^a is either monotonically increasing or monotonically decreasing.

Clients and service providers may have mutual interest for particular variables. For example, Raiffa cites an example [8, pg. 133–147] in which the Police Officers Union and the City Hall realize, in the course of their negotiations, that they both want the police commissioner fired. Having recognised this mutual interest they quickly agree that this course of action should be selected. However, in general, clients and servers have opposing interests, e.g. a client wants a low price for a service, whereas his potential servers attempt to obtain the highest price. High quality is desired by clients but not by servers, and so on. Therefore, in the space of negotiation values, negotiators represent opposing forces in each one of the dimensions. In consequence, the scoring functions verify that given a client c and a server s negotiating values for issue j, then if $x_j, y_j \in [min_j, max_j]$ and $x_j \geq y_j$ then $(V_j^c(x_j) \geq V_j^c(y_j)$ iff $V_j^s(x_j) \leq V_j^s(y_j))$.

In contrast, where there is a mutual interest, a variable will be assigned one of its extreme values. Hence these variables can be taken out of the negotiation set. For instance, the act of firing the police commisioner can be removed from the set of issues under negotiation and assigned the extreme value "done".

Once the agents have determined the set of variables over which they will negotiate, the negotiation process between two agents $a, b \in Agents$ consists of an alternate succession of offers and counter offers of values for these variables. This continues until an offer or counter offer is accepted by the other side or one of the partners terminates negotiation (e.g. because the time deadline is reached without an agreement being in place). Negotiation can be initiated by clients or servers.

We represent by $x_{a \to b}^t$ the vector of values proposed by agent a to agent b at time t, and by $x_{a \to b}^t[j]$ the value for issue j proposed from a to b at time t. The range of values acceptable to agent a for issue j will be represented as the interval $[min_j^a, max_j^a]$. For convenience, we assume a common global time (the calendar time) represented by a linearly ordered set of instants, namely

[6] The subsets are not disjoint since an agent can participate as a client in one negotiation and as a service provider in another.

$Time$, and a reliable communication medium introducing no delays in message transmission (so we can assume that emission and reception times are identical). The common time assumption is not too strong for our application domains, because time granularity and offer and counter offers frequencies are not too high. Then,

Definition 1. A **Negotiation Thread** between agents $a, b \in Agents$, at time $t \in Time$, noted $x^t_{a \leftrightarrow b}$ or $x^t_{b \leftrightarrow a}$, is any finite sequence of the form $\{x^{t_1}_{d_1 \to e_1}, x^{t_2}_{d_2 \to e_2}, \ldots, x^{t_n}_{d_n \to e_n}\}$ where:

1. $e_i = d_{i+1}$, proposals are alternate between both agents,
2. $t_k \leq t_l$ if $k \leq l$, ordered over time,
3. $d_i, e_i \in \{a, b\}$, the thread contains only proposals between agents a and b,
4. $d_i \neq e_i$, the proposals are *between* agents, and
5. $x^{t_i}_{d_i \to e_i}[j] \in [min^{d_i}_j, max^{d_i}_j]$ or is one of the particles $\{accept, reject\}$.

Superindex t_n represents an instant in the set $Time$ such that $t_n \leq t$. We will say that a negotiation thread is **active**[7] if $x^{t_n}_{d_n \to e_n} \notin \{accept, reject\}$.

For simplicity in the notation, we assume in the sequel that t_1 corresponds to the initial time value, that is $t_1 = 0$. In other words, there is a local time for each negotiation thread, that starts with the utterance of the first offer. When agent a receives an offer from agent b at time t, that is $x^t_{b \to a}$, it has to rate the offer using its scoring function. If the value of $V^a(x^t_{b \to a})$ is greater than the value of the counter offer agent a is ready to send at the time t' when the evaluation is performed, that is, $x^{t'}_{a \to b}$ with $t' > t$, then agent a accepts. Otherwise, the counter offer is submitted. The interpretation function I^a expresses this concept more formally:

Definition 2. Given an agent a and its associated scoring function V^a, the **interpretation** by agent a at time t' of an offer $x^t_{b \to a}$ sent at time $t < t'$, is defined as:

$$I^a(t', x^t_{b \to a}) = \begin{cases} accept \text{ If } V^a(x^t_{b \to a}) \geq V^a(x^{t'}_{a \to b}) \\ x^{t'}_{a \to b} \quad otherwise \end{cases}$$

where $x^{t'}_{a \to b}$ is the contract that agent a would offer to b at the time of the interpretation.

The result of $I^a(t', x^t_{b \to a})$ is used to extend the current negotiation thread between the agents. This interpretation also models the fact that a contract unacceptable today can be accepted tomorrow merely by the fact that time has passed.

In order to prepare a counter offer, $x^{t'}_{a \to b}$, agent a uses a set of tactics that generate new values for each variable in the negotiation set. Based on the needs of

[7] We assume that any offer is valid (that is, the agent that uttered it is commited) until a counter offer is received. If the response time is relevant it can be included in the set of issues under negotiation.

our business process applications (Section 2), we developed the following families of tactics:

1. **Time-dependent**. If an agent has a time deadline by which an agreement must be in place, these tactics model the fact that the agent is likely to concede more rapidly as the deadline approaches. The shape of the curve of concession, a function depending on time, is what differentiates tactics in this set.
2. **Resource-dependent**. These tactics model the pressure in reaching an agreement that (i) the limited resources –e.g. remaining bandwidth to be allocated, money, or any other– and (2) the environment –number of clients, number of servers or economic parameters– impose upon the agent's behaviour. The functions in this set are similar to the time dependent functions except that the domain of the function is the quantity of resources available instead of the remaining time.
3. **Imitative**. In situations in which the agent is not under a great deal of pressure to reach an agreement, it may choose to use imitative tactics that protect it from being exploited by other agents. In this case, the counter offer depends on the behaviour of the negotiation opponent. The tactics in this family differ in which aspect of their opponent's behaviour they imitate, and to what degree the opponent's behaviour is imitated.

We do not claim that these family types are complete, nor that we have enumerated all possible instances of tactics within a given family. These are merely the types of tactics we found useful in our applications.

4 Negotiation tactics

Tactics are the set of functions that determine how to compute the value of a quantitative issue (price, volume, duration, ...), by considering a *single* criteria (time, resources, ...). The set of values for the negotiation issue are then the range of the function, and the single criteria is its domain. The criteria we have chosen for the application domain, as explained in the previous section, are time, resources and previous offers and counter offers.

Given that agents may want to consider more than one criterion to compute the value for a single issue, we model the generation of counter proposals as a weighted combination of different tactics covering the set of criteria. The values so computed for the different issues[8] will be the elements of the counter proposal. For instance, if an agent wants to counter propose taking into account two criteria: the remaining time and the previous behaviour of the opponent, it can select two tactics: *boulware* (sec. 4.1, based on the remaining time, and *Tit-For-Tat* (sec. 4.3) to imitate the behaviour of the oponent. Each of the so selected tactics will suggest a value to counter propose for the issue under negotiation.

[8] values for issues may be computed by different weighted combinations of tactics.

The actual value which is counter proposed will be the weighted combination of the two suggested values.

Given an issue j, for which a value is under negotiation, an agent a's initial offer corresponds to a value in the issue's acceptable region, that is, a value in $[min_j^a, max_j^a]$. For instance, a client with a range $[£0, £20]$ for the price p to pay for a good may start the negotiation process by offering the server $£10$ –what initial offer should be chosen is something the agent can learn by experience. The server, with range $[£17, £35]$ may then make an initial counter-offer of $£25$. With these two initial values, the strategy of the first agent may consist of using a time-dependent tactic giving $£12$ –e.g. if it has a short time to reach an agreement, it will start conceding. And then if the strategy of the latter agent is to use an imitative tactic, it could generate a counter-proposal of $£23$ (imitating the $£2$ shift of its opponent). And so on.

4.1 Time-dependent tactics

In these tactics, the predominant factor used to decide which value to offer next is time, t. Thus these tactics consist of varying the acceptance value for the issue depending on the remaining negotiation time (an important requirement in our domain –Section 2.2). This requires a constant t_{max}^a in agent a that represents an instant in the future by when the negotiation must be completed. We model the initial offer as being a point in the interval of values of the issue under negotiation. Hence, agents define a constant κ_j^a that multiplied by the size of the interval determines the value of issue j to be offered in the first proposal by agent a.

We model the value to be uttered by agent a to agent b for issue j as the offer at time t, with $0 \leq t \leq t_{max}^a$, by a function α_j^a depending on time as the following expression shows:

$$x_{a\to b}^t[j] = \begin{cases} min_j^a + \alpha_j^a(t)(max_j^a - min_j^a) & \text{If } V_j^a \text{ is decreasing} \\ min_j^a + (1 - \alpha_j^a(t))(max_j^a - min_j^a) & \text{If } V_j^a \text{ is increasing} \end{cases}$$

A wide range of time-dependent functions can be defined simply by varying the way in which $\alpha_j^a(t)$ is computed. Functions must ensure that $0 \leq \alpha_j^a(t) \leq 1$, $\alpha_j^a(0) = \kappa_j^a$ and $\alpha_j^a(t_{max}^a) = 1$. That is, the offer will always be between the value range, at the beginning it will give the initial constant and when the time deadline is reached the tactic will suggest to offer the reservation value[9]. We distinguish two families of functions with this intended behaviour, polynomial and exponential. Both families are parameterised by a value $\beta \in \Re^+$ that determines the convexity degree (see Figure 2) of the curve. We have chosen these two families of functions because of the very different way they model concession. For

[9] The reservation value for issue j of agent a represents the value that gives the smallest score for function V_j^a. The reservation value for agent a and issue j depends on the function V_j^a and the range $[min_j^a, max_j^a]$. If V_j^a is monotonically increasing, then the reservation value is min_j^a; if it is decreasing the reservation value is max_j^a.

the same big value of β the polynomial function concedes faster at the beginning than the exponential one, then they behave similarly. For a small value of β, the exponential function waits longer to start conceding than the polynomial one. Many other functions could eventually be defined.

- **Polynomial.** $\alpha_j^a(t) = \kappa_j^a + (1 - \kappa_j^a)(\frac{min(t,t_{max})}{t_{max}})^{\frac{1}{\beta}}$
- **Exponential.** $\alpha_j^a(t) = e^{(1-\frac{min(t,t_{max})}{t_{max}})^{\beta} \ln \kappa_j^a}$

These families of functions represent an infinite number of possible tactics, one for each value of β. However to better understand their behaviour we have classified them, depending on the value of β, into two extreme sets showing clearly different patterns of behaviour. Other sets in between these two could be defined:

1. **Boulware tactics [[8], pg. 48]**. Either exponential or polynomial functions with $\beta < 1$. This tactic maintains the offered value until the time is almost exhausted, whereupon it concedes up to the reservation value[10]. The behaviour of this family of tactics with respect to β is easily understood taking into account that $\lim_{\beta\to 0^+} e^{(1-\frac{min(t,t_{max})}{t_{max}})^{\beta} \ln \kappa_j^a} = \kappa_j^a$ or $\lim_{\beta\to 0^+} \kappa_j^a + (1 - \kappa_j^a)(\frac{min(t,t_{max})}{t_{max}})^{\frac{1}{\beta}} = \kappa_j^a$.
2. **Conceder [[3], pg. 20]**. Either exponential or polynomial functions with $\beta > 1$. The agent goes to its reservation value very quickly. For similar reasons as before, we have $\lim_{\beta\to +\infty} e^{(1-\frac{min(t,t_{max})}{t_{max}})^{\beta} \ln \kappa_j^a} = 1$ or $\lim_{\beta\to +\infty} \kappa_j^a + (1 - \kappa_j^a)(\frac{min(t,t_{max})}{t_{max}})^{\frac{1}{\beta}} = 1$.

4.2 Resource-dependent tactics

These tactics are similar to the time-dependent ones. Indeed time-dependent tactics can be seen as a type of resource-dependent tactic in which the sole resource considered is time. Whereas time vanishes constantly up to its end, other resources may have different patterns of usage. We model resource-dependent tactics in the same way as time-dependent ones, that is, by using the same functions, but by making the value t^a_{max} dynamic. Its value represents a heuristic about how many resources are in the environment. The scarcer the resource the more urgent the need for an agreement to be reached. In our application domains the most important resource to model is the number of agents negotiating with a given agent and how keen they are to reach agreements. On one hand, the greater the number of agents who are negotiating with agent a for a particular service s, the lower the pressure on agent a to reach an agreement with any

[10] Besides the pattern of concession that these functions model, Boulware negotiation tactics presume that the interval of values for negotiation is very narrow. Hence, when the deadline is reached and $\alpha(t_{max}) = 1$ the offer generated is not substantially different from the initial one.

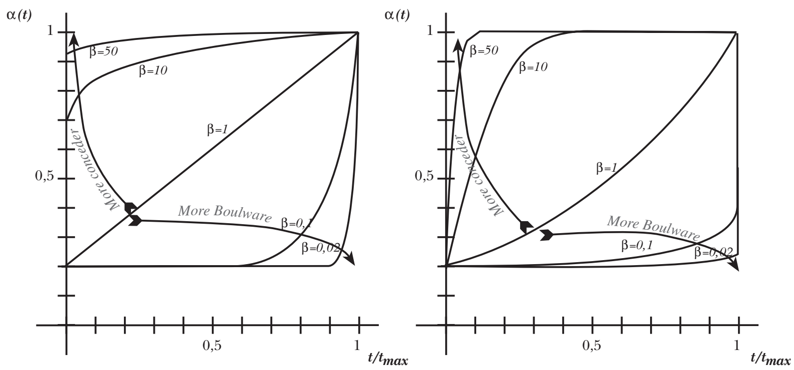

Fig. 2. Polynomial (left) and Exponential (right) functions for the computation of $\alpha(t)$. Time is presented as relative to t^a_{max}.

specific individual. While on the other hand, the longer the negotiation thread, the greater the pressure on a to come to an agreement. Hence, representing the set of agents negotiating with agent a at time t as: $N^a(t) = \{i | x^t_{i \leftrightarrow a} \text{is active}\}$, we define the dynamic time deadline for agent a as:

$$t^a_{max} = t_c + \mu^a \frac{|N^a(t_c)|^2}{\sum_i |x^{t_c}_{i \leftrightarrow a}|}$$

where μ^a represents the time agent a considers reasonable to negotiate with a single agent, t_c is the current time and $|x^{t_c}_{i \leftrightarrow a}|$ represents the length of the current thread between i and a. Notice that the number of agents is in the numerator – so quantity of time is directly proportional to it, and averaged length of negotiation thread is in the denominator – so quantity of time is inversely proportional to it.

4.3 Imitative tactics

This family of tactics compute the next offer based on the previous attitude of the negotiation opponent. These tactics have proved important in co-operative problem-solving negotiation settings [1], and so are useful in a subset of our contexts (see Section 2.2). The main difference between the tactics in this family is in the type of imitation they perform. One family imitates proportionally, another in absolute terms, and the last one computes the average of the proportions in a number of previous offers. Hence, given a negotiation thread $\{\ldots, x^{t_{n-2\delta}}_{b \to a}, x^{t_{n-2\delta+1}}_{a \to b}, x^{t_{n-2\delta+2}}_{b \to a}, \ldots, x^{t_{n-2}}_{b \to a}, x^{t_{n-1}}_{a \to b}, x^{t_n}_{b \to a}\}$, with $\delta \geq 1$, we distinguish the following families of tactics:

1. **Relative Tit-For-Tat**
   The agent reproduces, in percentage terms, the behaviour that its opponent performed $\delta \geq 1$ steps ago.

$$x_{a\to b}^{t_{n+1}}[j] = \begin{cases} min(max(\frac{x_{b\to a}^{t_{n-2\delta}}[j]}{x_{b\to a}^{t_{n-2\delta+2}}[j]} x_{a\to b}^{t_{n-1}}[j], min_j^a), max_j^a) & n > 2\delta \\ min_j^a + \kappa_j^a(max_j^a - min_j^a) & n \leq 2\delta, V_j^a \text{ decr.} \\ min_j^a + (1-\kappa_j^a)(max_j^a - min_j^a) & n \leq 2\delta, V_j^a \text{ incr.} \end{cases}$$

Depending on the value of the quotient between two consecutive counter offers, the agent exhibits a range of different behaviours: **mirror** if the quotient is greater than 1, **retaliatory** if it is lower than 1, and a type of time independent **boulware** if it is exactly 1.

2. **Random Absolute Tit-For-Tat**
The same as before but in absolute terms. It means that if the other agent decreases its offer by *£*2, then the next response should be increased by the same *£*2. Moreover, we add a component that modifies that behaviour by increasing or decreasing (depending on the value of parameter s) the value of the answer by a random amount. (This is introduced as it can enable the agents to escape from local minima.) M is the maximum amount by which an agent can change its imitative behaviour.

$$x_{a\to b}^{t_{n+1}}[j] = \begin{cases} min(max(x_{a\to b}^{t_{n-1}}[j] + (x_{b\to a}^{t_{n-2\delta}}[j] - x_{b\to a}^{t_{n-2\delta+2}}[j]) + \\ \qquad +(-1)^s R(M), min_j^a), max_j^a) & n > 2\delta \\ min_j^a + \kappa_j^a(max_j^a - min_j^a) & n \leq 2\delta, V_j^a \text{ decr.} \\ min_j^a + (1-\kappa_j^a)(max_j^a - min_j^a) & n \leq 2\delta, V_j^a \text{ incr.} \end{cases}$$

Where $s \in \{0, 1\}$ and $R(M)$ is a function that generates a random integer in the interval $[0, M]$.

3. **Averaged Tit-For-Tat** The agent computes the average of percentages of changes in a window of size $\gamma \geq 1$ of its opponents history when determining its new offer. When $\gamma = 1$ we have the relative Tit-For-Tat tactic with $\delta = 1$.

$$x_{a\to b}^{t_{n+1}}[j] = \begin{cases} min(max(\frac{x_{b\to a}^{t_{n-2}}[j]}{x_{b\to a}^{t_{n-2\gamma+2}}[j]} x_{a\to b}^{t_{n-1}}[j], min_j^a), max_j^a) & n > 2\gamma \\ min_j^a + \kappa_j^a(max_j^a - min_j^a) & n \leq 2\gamma, V_j^a \text{ decr.} \\ min_j^a + (1-\kappa_j^a)(max_j^a - min_j^a) & n \leq 2\gamma, V_j^a \text{ incr.} \end{cases}$$

5 Negotiation strategies

The aim of agent a's negotiation strategy is to determine the best course of action which will result in an agreement on a contract x that maximises its scoring function V^a. In practical terms, this equates to how to prepare a new counter offer.

In our model we consider that the agent has a representation of its mental state containing information about its beliefs, its knowledge of the environment (time, resources, etc.), and any other attitudes (desires, goals, obligations, intentions, etc.) the agent designer considers appropriate. The mental state of agent

a at time t is noted as MS_a^t. We denote the set of all possible mental states for agent a as MS_a.

When agent a receives an offer from agent b it becomes the last element in the current negotiation thread between both agents. If the offer is unsatisfactory, agent a generates a counter offer. As discussed earlier, different combinations of tactics can be used to generate counter offers for particular issues. An agent's strategy determines which combination of tactics should be used at any one instant. Hence, the following definition:

Definition 3. Given a negotiation thread between agents a and b at time t_n over domain $X = X_1 \times \ldots \times X_p$, with $x_{a\leftrightarrow b}^{t_n} = \{\ldots, x_{b\to a}^{t_n}\}$, and a finite set of m tactics[11] $T^a = \{\tau_i | \tau_i : MS_a \to X\}_{i\in[1,m]}$, a **weighted counter proposal** is any lineal combination of the tactics that generates the value at time t_{n+1} in the thread. That is, for issue j

$$x_{a\to b}^{t_{n+1}}[j] = \gamma_{j1}\tau_1(MS_a^{t_{n+1}})[j] + \gamma_{j2}\tau_2(MS_a^{t_{n+1}})[j] + \ldots + \gamma_{jm}\tau_m(MS_a^{t_{n+1}})[j]$$

such that for all issues j, $\sum_{i\in[1,m]} \gamma_{ji} = 1$ and $x_{a\leftrightarrow b}^{t_{n+1}} = \{\ldots, x_{b\to a}^{t_n}, x_{a\to b}^{t_{n+1}}\}$

Given a set of tactics, different types of negotiation behaviour can be obtained by weighting the tactics in a different way. That is, by changing the matrix Γ – particular to each negotiation thread:

$$\Gamma_{a\to b}^t = \begin{pmatrix} \gamma_{11} & \gamma_{12} & \cdots & \gamma_{1m} \\ \gamma_{21} & \gamma_{22} & \cdots & \gamma_{2m} \\ \vdots & \vdots & \vdots & \vdots \\ \gamma_{p1} & \gamma_{p2} & \cdots & \gamma_{pm} \end{pmatrix}$$

An example of when this weighted combination may be useful is when modelling a *smooth* transition from a behaviour based on a single tactic (e.g. Boulware, because the agent has plenty ot time to reach an agreement) to another one (e.g. Conceder, because the time is running out). Smoothness is obtained by changing the weight affecting the tactics progressively (e.g. from 1 to 0 and from 0 to 1 in the example).

We model many-parties negotiations by means of a set of interacting negotiation threads. The way this is done is by making a negotiation thread influence the selection of which matrix Γ is to be used in other negotiation threads. Thus,

Definition 4. Given $a, b \in Agents$, $t \in Time$, a's mental state MS_a^t, and $\Gamma_{a\to b}^t$, a **Negotiation Strategy**, is any function f of the following type:

$$\Gamma_{a\to b}^{t+1} = f(\Gamma_{a\to b}^t, MS_a^t)$$

[11] This definition uses the natural extension of tactics to the multi-dimensional space of issues' values.

A simplistic example of the application of our model would be to have a matrix Γ built up of 0s and 1s and having $\Gamma_{a \to b}^{t+1} = \Gamma_{a \to b}^{t}$ for all t. This would correspond to using a fixed single tactic for each issue at every instant in the negotiation.

6 Convergence results

Convergence in negotiation is achieved when the scoring value of the received offer is greater than the scoring value of the counter offer the agent intended to respond with. That is,

Definition 5. A Negotiation thread $x_{a \leftrightarrow b}^{t_n} = \{\ldots, x_{b \to a}^{t_n}\}$ *converges* at time t_{n+1} iff $V^a(x_{b \to a}^{t_n}) \geq V^a(x_{a \to b}^{t_{n+1}})$

With this definition in mind, we have obtained some preliminary results on convergence for a single variable, single tactic, bilateral negotiation. Wider convergence criteria will be forthcoming as future work. The second proposition (6.2) is particularly interesting because it allows an agent using a time-dependent tactic to know if the negotiation will converge with an agent playing relative Tit-For-Tat. Knowing if an opponent is playing Tit-For-Tat can easily be guessed by using a strategy that makes some initial random offers and then examines the responses.

Notice, however, that convergence cannot be guaranteed in general. For example, two agents using a Tit-For-Tat tactic might stay negotating forever if no limitation on time is established.

Proposition 6. *If two agents a and b negotiate values for an issue j over the value regions $[min_j^a, max_j^a]$, $[min_j^b, max_j^b]$, satisfying $[min_j^a, max_j^a] \cap [min_j^b, max_j^b] \neq \emptyset$, then the following properties hold:*

1. *If a and b follow a time-dependent tactic with V_j^a decreasing (resp. increasing), V_j^b increasing (resp. decreasing) and $t_{max}^a = t_{max}^b$ then the negotiation for issue j converges.*
2. *If a uses a time-dependent tactic with V_j^a increasing (resp. decreasing) and b uses a relative Tit-For-Tat tactic with V_j^b decreasing (resp. increasing), and a starts the negotiation thread, then the negotiation converges if $x_{a \to b}^{t_1}[j] x_{b \to a}^{t_2}[j] < (min_j^a)^2$ (resp. if $x_{a \to b}^{t_1}[j] x_{b \to a}^{t_2}[j] > (max_j^a)^2$).*

Proof. (1) We prove it for V_j^a decreasing, the other case is similar. We have $[min_j^a, max_j^a] \cap [min_j^b, max_j^b] \neq \emptyset$, then $max_j^a > min_j^b$. When time runs out the $\alpha(t)$ functions of both agents become 1, that is $\alpha_j^a(t_{max}^a) = \alpha_j^b(t_{max}^b) = 1$ and then their acceptance values will become $min_j^a + \alpha_j^a(t)(max_j^a - min_j^a) = max_j^a$ and $min_j^b + (1 - \alpha_j^b(t))(max_j^b - min_j^b) = min_j^b$. So, b will make the offer $x_{b \to a}^{t_{max}^b}[j] = min_j^b$ at the deadline. But given that a uses a monotonically decreasing function and $max_j^a > min_j^b$ we have $V_j^a(max_j^a) > V_j^a(min_j^b)$. In other words, by using the assumption $t^a_{max} = t^b_{max}$ we have $V^a_j(x^{t^a_{max}}_{b\to a}[j]) > V^a_j(x^{t^a_{max}}_{a\to b}[j])$. So the negotiation converges.

(2) Again we prove it for V^b_j increasing, the other case is similar. Without loss of generality assume that the thread is:

$$x^{t_{n-1}}_{a\leftrightarrow b} = \{x^{t_1}_{a\to b}, x^{t_2}_{b\to a}, \ldots, x^{t_{n-1}}_{a\to b}\}$$

and that $\delta = 1$. By using the definition of relative Tit-For-Tat, it can be seen that:

$$x^{t_n}_{b\to a}[j] = \frac{x^{t_{n-3}}_{a\to b}[j]}{x^{t_{n-1}}_{a\to b}[j]} x^{t_{n-2}}_{b\to a}[j] = \frac{x^{t_{n-3}}_{a\to b}[j]}{x^{t_{n-1}}_{a\to b}[j]} \frac{x^{t_{n-5}}_{a\to b}[j]}{x^{t_{n-3}}_{a\to b}[j]} x^{t_{n-4}}_{b\to a}[j] =$$

$$= \frac{x^{t_{n-3}}_{a\to b}[j]}{x^{t_{n-1}}_{a\to b}[j]} \frac{x^{t_{n-5}}_{a\to b}[j]}{x^{t_{n-3}}_{a\to b}[j]} \cdots \frac{x^{t_1}_{a\to b}[j]}{x^{t_3}_{a\to b}[j]} x^{t_2}_{b\to a}[j] = \frac{x^{t_1}_{a\to b}[j]}{x^{t_{n-1}}_{a\to b}[j]} x^{t_2}_{b\to a}[j]$$

The thread converges if $V^b_j(x^{t_{n-1}}_{a\to b}[j]) \geq V^b_j(x^{t_n}_{b\to a}[j])$, but given that V^b_j is monotonically decreasing, this happens only if $x^{t_n}_{b\to a}[j] < x^{t_{n-1}}_{a\to b}[j]$. Then, by substituting we get $\frac{x^{t_1}_{a\to b}[j]}{x^{t_{n-1}}_{a\to b}[j]} x^{t_2}_{b\to a}[j] < x^{t_{n-1}}_{a\to b}[j]$, that is $x^{t_1}_{a\to b}[j] x^{t_2}_{b\to a}[j] < (x^{t_{n-1}}_{a\to b}[j])^2$. But when time approaches t^a_{max} we have $\lim_{t\to t^a_{max}} x^t_{a\to b}[j] = min^a_j$ (by V^a_j being increasing). So, at the limit, if $x^{t_1}_{a\to b}[j] x^{t_2}_{b\to a}[j] < (min^a_j)^2$ the negotiation converges.

7 Related work

Research in negotiation models has been pursued in different fields of knowledge: game theory, social sciences and artificial intelligence. Each field has concentrated on different aspects of negotiation, making the assumptions that were pertinent for the goal of their study. In game theory, researchers have been interested in mechanism design: the definition of protocols that limit the possible tactics (or strategies) that can be used by players. For instance they are interested in defining protocols that give no benefit to agents that mis-represent or hide information [9]. In this work disclosure of information is acceptable, because by doing so it will benefit the agent in finding an optimal solution for itself. Contrary to our model, and as we discussed in Section 2, this is an inappropriate assumption from the point of view of real applications. As has been argued elsewhere [10], these and other assumptions limit the applicability of game theory models to solve real problems.

Our interests lie in invertigating the process of negotiation among agents and not only on the outcome. Hence, our study, and those forthcoming, are much more in the experimental line of [4]. Although we do not concentrate on learning, some similarities can be found with the formalism by Zeng and Sycara [10]. We have not concentrated however on the description of negotiation protocols that has been an important focus of attention for the community of distributed artificial intelligence (see [7] for extensive references).

8 Discussion and future work

The next stage in the development of our model is to undertake an experimental evaluation of the tactics and strategies described herein. We believe adequate strategies have to be developed in accordance with the desired properties and characteristics of the domain at hand. These strategies then need to be tested in repeated games over a range of typical scenarios to determine which are the most successful. Some initial modeling of concrete strategies has been made considering several variables in the mental state of an agent: (i) an approximation of the first and second derivatives of the other agent's behaviour, (ii) the relation between both negotiating agents (e.g. members of the same company, boss/employee, ...), and (iii) the time remaining to reach an agreement (in this case time is playing a role at both strategic and tactic levels). This model is being used in the real modeling of the domain presented in Section 2. Currently there are two versions of the model implemented in CLIPS and PROLOG.

The initial results on convergence, although simple, encourage us to make a more complete analysis of the types of negotiation situations that are likely to occur.

We have identified many research opportunities in extending the model. For instance, fuzzy control techniques could be used to relate a qualitative estimate of the first and second derivatives of the opponent and a qualitative value for the β to be used in a tactic; we imagine a rule like: *If the agent concedes* quite a lot *(first derivative) and the agent concession ratio (second derivative)* increases *then Beta is* Medium. Genetic algorithms could also be used to determine experimentally which weighted combinations of tactics survive better in the line of [4]. Moreover, genetic algorithms may help to determine which negotiating agents show the desired behaviour by using the strategies as the genetic code. Finally, case-based reasoning could be used to model strategies. The case memory could be used by the agent to determine which past combinations of tactics worked best in similar circumstances.

9 Acknowledgements

This project has received the support of the Spanish Research project SMASH (CICYT number, TIC96-1038-C04001) and the DTI/EPSRC Intelligent Systems Integration Programme (ISIP) project ADEPT.

References

1. R. Axelrod. *The Evolution of Cooperation.* Basic Books, Inc., Publishers, New York, USA., 1984.
2. B. Crabtree and N. (eds.). *The Practical Application of Intelligent Agents and Multi-Agent Technology.* London, UK., 1996.
3. D. G.Pruitt. *Negotiation Behavior.* Academic Press, 1981.

4. A. Ito and H. Yano. The emergence of cooperation in a society of autonomous agents - the prisoner's dilemma gamme under the disclosure of contract histories. In V. Lesser, editor, *Proceedings of the First International Conference on Multi-Agent Systems*, pages 201–208, San Francisco, USA, 1995. AAAI Press/The MIT Press.
5. N. R. Jennings, P. Faratin, M. J. Johnson, T. J. Norman, P. O'Brien, and M. E. Wiegand. Agent-based business process management. *Int Journal of Cooperative Information Systems*, 5(2–3):105–130, 1996.
6. S. Kraus, J. Wilkenfeld, and G. Zlotkin. Multiagent negotiation under time constraints. *Artificial Intelligence Journal*, 75(2):297–345, 1995.
7. H. Mueller. Negotiation principles. In G. M. P. O'Hare and N. R. Jennings, editors, *Foundations of Distributed Artificial Intelligence*, Sixth-Generation Computer Technology Series, pages 211–229, New York, 1996. John Wiley.
8. H. Raiffa. *The Art and Science of Negotiation.* Harvard University Press, Cambridge, USA, 1982.
9. J. S.Rosenschein and G. Zlotkin. *Rules of Encounter.* The MIT Press, Cambridge, USA, 1994.
10. D. Zeng and K. Sycara. How can an agent learn to negotiate. In J. Mueller, M. Wooldridge, and N. Jennings, editors, *Intelligent Agents III. Agent Theories, Architectures, and Languages*, number 1193 in LNAI, pages 233–244. Springer Verlag, 1997.